Theoretical Guarantees and Complexity Reduction in Information Planning

by

Georgios Papachristoudis

Eng.Dipl., Electrical and Computer Engineering, Aristotle University of Thessaloniki, 2007
S.M., Electrical Engineering and Computer Science, MIT, 2010

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

June 2015

Signature of Author: _____

Department of Electrical Engineering and Computer Science
May 20, 2015

Certified by: _____

John W. Fisher III
Senior Research Scientist of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by: _____

Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Committee for Graduate Students

Theoretical Guarantees and Complexity Reduction in Information Planning

by Georgios Papachristoudis

Submitted to the Department of Electrical Engineering
and Computer Science on May 20, 2015
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Information planning addresses the problem of determining the optimal set of measurements that would reduce the uncertainty over latent variables of interest under a set of constraints. A commonly used reward for quantifying the expected reduction in uncertainty is mutual information (MI). One application of information planning can be found in object tracking where a network of sensors generate noisy measurements of a moving object's position and we are interested in selecting the set of sensors that would maximally reduce the uncertainty of predicting the object's track. Optimal measurement selection can be combinatorially complex and intractable for large-scale problems; interestingly, it has been shown that simple greedy algorithms that choose the best measurement at each time step given past selections, provide nearly optimal solutions for submodular monotone rewards.

In this thesis, we examine several challenges that arise when performing real-world information planning. Our contributions are three-fold: (i) we provide theoretical guarantees for the greedy algorithm used in the submodular monotone case when it is applied to different problem settings: (1) non-monotone rewards, (2) budget constraints, (3) settings where only a set of latent variables is of relevance; (ii) we demonstrate how to substantially reduce the complexity of evaluating MI in Gaussian models by taking advantage of sparsity in the measurement process; and (iii) we propose a variant of belief propagation that is suitable for adaptive inference settings.

In the first part, we present conditions under which open-loop is equivalent to closed-loop information planning. Furthermore, we provide bounds on the greedy performance for submodular non-monotone functions. We also consider the case when measurements are valued based on their relative information content and explore the conditions under which the known bounds for the submodular monotone case apply to this modified setting. Lastly, we provide bounds for budgeted and focused planning. In the former, each measurement induces a cost and there is a limited budget constraint, while in the latter only a set of latent variables is of interest.

In the second part of the thesis, we propose a method for reducing the complexity of function evaluations that take place during information planning. Previous works have assumed oracle-value models, where the informational value of any set of measurements is provided in constant time, an assumption which is often unrealistic. We focus on

Gaussian models and MI and show that we can substantially reduce complexity by exploiting sparsity in the measurement process.

In the third part, we propose a variant of belief propagation (BP) that is well-suited to adaptive inference settings and is exact on trees. During information planning we need to update the marginal distribution of only one latent variable at each step of the greedy algorithm. This task can be achieved naïvely, where inference is performed from scratch at every step, or by taking advantage of repeating calculations. Adaptive inference is concerned with the latter approach. We suggest a minimal messaging schedule, where only the necessary messages are sent at every step to guarantee the correct marginal distributions at the nodes of interest. We also provide extensions to Gaussian loopy graphs and to the problem of finding the most likely sequence of hidden variables.

Thesis Supervisor: John W. Fisher III
Title: Senior Research Scientist of Electrical Engineering and Computer Science

Acknowledgments

I would like to extend my deepest thanks to my thesis committee, John Fisher, Jon How and Costis Daskalakis for their valuable comments and suggestions during my thesis writing. This thesis would not have been possible without the guidance and support by John. John trusted me by welcoming me to his group during a critical transitional period for me (from Masters to PhD) when I was looking for an advisor who would not have only been able to guide me with his technical expertise, but would also be fun to work with. John's mentoring and great insights helped me to not only advance my knowledge in the area of information theory and probabilistic inference, but also led me through the entire process of my PhD. It has been remarkable to listen to John's intuition about different topics in group meetings and then go back and see on paper that these claims were true.[1] Another great characteristic of John is the relative freedom he gives to his students to pursue their passion and develop abstract ideas to promising research directions. It is not an exaggeration to say had there not been for John and Sensing, Learning, and Inference (SLI) group, in general, the PhD experience would have been a much poorer experience.

Different aspects of this thesis were supported by the Office of Naval Research and the Army Research Office Multidisciplinary Research Initiative programs as well as Shell through the MIT Energy Initiative. I would like also to thank the Alexander Onassis Foundation for their financial support during the first years of my PhD. Special mention goes to to Mrs. Roula and Mr. Leuteris Kanellakis, whom I owe my deepest gratitude. Mrs. and Mr. Kanellakis created the Paris Kanellakis Fellowship in memory of their son that supported me during the first year of my studies. Both these kind people have embraced me like their own child and filled me with advice of wisdom before I embark on my adventure to MIT. I thank them for giving me the opportunity to be one of the Kanellakis fellows and hope will prove to be an exceptional ambassador of their son's legacy.

The everyday routine would be much more different and uneventful had there not been for my wonderful labmates; Randi Cabezas, Sue Zheng, Jason Chang, Zoran Dzunic, Julian Straub, Chris Dean, Vadim Smolyakov, David Hayden, Dahua Lin and our postdocs Guy Rosman, Oren Freifeld and Hossein Mobahi. Especially, some of

[1] With a few exceptions of course...

them had not only been great colleagues, but amazing friends as well. Sue's placid personality, Chris' amazing sense of humor and Randi's nagging would be just a few of the things I would remember dearly from this group. A special mention goes to Randi, whom I got to know a little more. Randi has been one of the most caring people I happened to meet in CSAIL. It takes some time to get used to him, but he is really a guy who would help everyone even at his own expense. To support this claim, he read a full draft of my thesis on his own initiative and provided me with really useful comments. Besides that, he has been a great -yet tough- gym buddy and friend!

Life in the office would be boring without the engaging conversations with constant hints of sarcasm by Adrian Dalca, the cryptic one-sentence phrases by George Chen, the sound of Zoran eating nuts and Ramesh's high volume laughters and abundant energy. Jokes aside, it would not be an exaggeration to say that I have spent maybe most of my PhD life around these guys and I am thankful to them for being such a great company and source of information. Special thanks go to Christian Wachinger for his ever resourceful teases and Kayhan Batmanghelich for his dear company.[2]

Switching gears to people outside my group, I would like to sincerely thank from the bottom of my heart Nikos Trichakis (the Spaceman) and Eleni Malliou. Nikos and Eleni have served as my older siblings, who gave me great advice throughout this process and helped me in every possible way. Especially, Nikos was the first guy I talked to before making the decision coming to MIT, and Nikos and Eleni were the first people who helped me settle in when I moved to Boston. Nikos' advice has always been and will be of special weight to me.

I'd like to thank all my friends who have filled my life with great moments all these years; Dimitris Chatzigeorgiou, Kimon Drakopoulos, Gerasimos Skaltsas, Yannis Bertsatos, Nikos Legbelos,[3] Dimitris Bisias, Manolis Kasseris, Spiros Lekkakos, Vangelis Sfakianakis, George Christou, Giorgos Angelopoulos, Penelope Pani, Iro Palaska, Aristeidis, Tim, Yola, Michos, Eva, Theodora and so many others that I am possibly forgetting right now. A special mention goes to Dimitris Tzeranis and Angelos Tsoukalas. Despite the heated conversations we might have had at times, they were one of the most genuine and generous persons I've met in Boston. I'd always be grateful for their company and the really great moments we had.

This acknowledgments section would be incomplete without commemorating my basketball team and all the wonderful and bitter (mostly wonderful) moments we have had. The following people will always bring the sweetest memories to my mind; Giorgos Kanavakis, Andrej Košmrlj, Gabe Juarez, Peter Scrivanos, Kostas Nanopoulos, Gabriel Zaccak, Alex Mentzelopoulos, Nacho Arnaldo, Nick Vandewiele, Maria-Alkistis, as well as ex-members of the team. Last but not least, I'd like to thank my Basque brother, Enrique Lizarraga for leading this team to three championships, but most importantly for being my dearest friend.

[2] I have to confess I bought once a sound horn to scare Christian to revenge him for a prank he played on me, but I never really used it...

[3] Even though he is a fan of Aris.

Continuing the everlasting thank you notes, I would like to sincerely thank Stella Kounelaki and Nina Panagiotidou. They both have been unique and true friends. No matter how rarely we saw each other, I knew they would always be there for me. Closing the circle of my friends, I'd wish to pay a special tribute to my three roommates. Antonis has been the first roommate I've had. I can still remember all the numerous cooking experimentations, the countless parties, the impromptu live guitar concerts, and all the great events we participated in during the three years of being roommates. They say it takes time, common experiences and friction to build a true friendship and that certainly has been the case. Spyros Zoumpoulis, my second roommate, has been one of the most fun and easy-going people I met in Boston. The effortless communication and the great discussions of any matter we dove into were remarkable. Lastly, Giorgos Dimitrakopoulos has been one of the most considerate guys I've met. It might have been only for a short while, but it was enough to realize the great qualities he carries.

During the ups and downs of my Phd, there are generally two groups of people I mostly grateful to, because they helped me stay afloat and have been a source of inspiration for me. Youngjoo and my family. Youngjoo has been one of the most loving, truthful, inspiring, generous and ambitious persons I've met. Her continuous support has always been a great motivating force for me that kept me going. It would not be an exaggeration to say that despite the distance, her continuous companionship usually through our endless Skype sessions, has made her the closest person to me here in the US. There are not enough words to express my deep gratitude to her. She has been part of my life in one of the most adventurous and challenging periods for me, and helped me broaden my horizons and become a better person. Regarding the other group, had it not been for my constant presence of my family, this journey would not have been completed. My parents, Nikos and Anna have always been there for me, and their constant reminders of what is really important in life have helped me so many times to put things in perspective. If I were to isolate just two key elements from them, these would be my father's minimalist approach towards life (that simple things in life are the ones which matter) and my mother's loud laughters stemming from her spontaneous and extremely optimistic personality. My sister Kiki holds a very special place in my heart and is one of the dearest people in my life. She has been my best pal since we were kids. I would always be grateful to her for being constantly by my side when I needed her and for being so loving and caring. I'm also extremely thankful to the two newest members of my family, Nikos and Anna. Nikos, Kiki's husband, has been dear to all of us, while our little angel, Anna, my two-year old niece, has been the cutest little being I've met who could bring infinite amounts of joy to me with literally no effort. She has no idea how many times has brightened my days just by watching her funny videos or by listening to her silky voice on the phone calling my name! Lastly, I'd wish to commemorate my beloved grandfather Giorgos, whom I am named after. I hope his exemplary humble lifestyle and purity of soul, would always lead me and constantly remind me where I am coming from.

Ever tried. Ever failed. No matter. Try Again. Fail again. Fail better. – Samuel Beckett

Contents

List of Figures

List of Algorithms

List of Tables

Notational Conventions

General Notation

Symbol	Definition
X	Random variable
$\mathsf{X}_{1:T}$	Sequence of random variables $\mathsf{X}_1, \ldots, \mathsf{X}_T$
x	Value of random variable
$\mathcal{X}$	Alphabet of X (set of values X takes when it is discrete, $x \in \mathcal{X}$)
$\mathbb{1}(x = j)$	One if $x = j$, zero otherwise
$P(\mathsf{X} = x)$	Probability that random variable X taking value x
$p_{\mathsf{X}\mid\mathsf{Y}}(\cdot \mid \cdot)$	Probability distribution of X given Y
$p_{\mathsf{X}}(\cdot)$	Probability distribution of random variable X (subscript will be omitted when there is no ambiguity)
$p_i(x_i)$	Marginal probability distribution of random variable X_i
$\mathbb{E}[\cdot]$	Expected value
$\mathbb{E}_{p_{\mathsf{X}}}[\cdot]$	Expected value over the distribution p_{X} (when there is no ambiguity as to which distribution is the expectation over, we will denote it simpler by $\mathbb{E}_{\mathsf{X}}[\cdot]$)
$\mathrm{cov}(\cdot)$	Covariance

Matrix/Vector Notation

Symbol	Definition
$(\cdot)^T$	Matrix or vector transpose
$(\cdot)^{-1}$	Matrix inverse
$\lvert \cdot \rvert$	Matrix determinant
$\mathbb{I}$	Identity matrix

Matroid Theory/Greedy Heuristics

Symbol	Definition
$f(\cdot)$	Set function
$\mathcal{V}$	Ground set of measurements
$\mathscr{I}$	Collection of (independent) sets
$\mathscr{M}$	Matroid
$\mathcal{O}$	Optimal set
$\mathcal{G}$	Greedy set
$\mathcal{A}, \mathcal{B}, \mathcal{S}, \mathcal{I}, \mathcal{J}$	Subsets of $\mathcal{V}$
u, e, j	Single measurements (items) from $\mathcal{V}$
$f(\mathcal{A} \mid \mathcal{B})$	Incremental value/gain by adding set $\mathcal{A}$ given $\mathcal{B}$: $f(\mathcal{A} \mid \mathcal{B}) = f(\mathcal{A} \cup \mathcal{B}) - f(\mathcal{B})$
$\boldsymbol{w}$	Walk of greedy selection process
N	Size of observation set $\mathcal{V}$, $\lvert\mathcal{V}\rvert = N$
k	Number of measurements to select from observation set $\mathcal{V}$
N_t	Size of observation set $\mathcal{V}_t$, $\lvert\mathcal{V}_t\rvert = N_t$
k_t	Number of measurements to select from observation set $\mathcal{V}_t$
$c(u)$	Non-negative cost of measurement u
b	Budget (selected measurements' cost from $\mathcal{V}$ must be below b)
b_t	Budget (selected measurements' cost from $\mathcal{V}_t$ must be below b_t)

Graph Theory

Symbol	Definition
G	Undirected graph
V	Vertex set
$\mathcal{E}$	Edge set
$\lvert V\rvert$	Cardinality of set V (number of nodes in graph G)
$V \setminus i$	All vertices except i (abbreviation for $V \setminus \{i\}$)
(i, j)	Undirected edge
$N(i)$	Set of neighbors of node i
$\text{lca}(u, v)$	Lowest common ancestor of nodes u, v
$\text{diam}(G)$	Diameter: maximum distance between any pair of nodes
$\boldsymbol{w}$	Walk (sequence of connecting nodes in graph G)

Graphical Models

Symbol	Definition
φ_i	Node potential
ψ_{ij}	Edge (pairwise) potential
Z	Partition function (normalization constant)
$m_{i \to j}(x_j)$	Message from node i to node j (in BP)
$h_{i \to j}, J_{i \to j}$	Messages in Gaussian BP (GaBP)
μ	Mean for a Gaussian distribution
Σ	Covariance for a Gaussian distribution
h	Potential vector for a Gaussian distribution
J	Information (precision) matrix of a Gaussian distribution
$\mathcal{N}(x; \mu, \Sigma)$	Gaussian distribution with mean μ and covariance Σ (moment form)
$\mathcal{N}^{-1}(x; h, J)$	Gaussian distribution with potential vector h and information matrix J (information form)

Gaussian HMMs

Symbol	Definition
d	Dimension of hidden variable
m	Dimension of observed variable
t	Point in time
X_t	Hidden variable ($\mathsf{X}_t \in \mathbb{R}^d$)
Y_t	Observed variable ($\mathsf{Y}_t \in \mathbb{R}^m$)
A_t	State transition matrix ($A_t \in \mathbb{R}^{d \times d}$)
V_t	Process noise ($\mathsf{V}_t \sim \mathcal{N}(v_t; 0, Q_t)$)
C_t	Observation matrix ($C_t \in \mathbb{R}^{m \times d}$)
W_t	Observation noise ($\mathsf{W}_t \sim \mathcal{N}(w_t; 0, R_t)$)

Chapter 1

Introduction

MANY estimation problems, both sequential and non-sequential, can be formulated as inference in graphical models. Here we propose to investigate such problems in the context of distributed information gathering subject to limited resources by representing the stochastic part of the problem as a graphical model. A graphical model is a probabilistic model that represents the conditional independencies between random variables. We will assume that the graph is comprised of a set of hidden (latent) variables X and a set of observed variables Y. Estimation theory is concerned with the usage of an appropriate set of observations to increase our confidence about the hidden states. Applications are seen in a wide range of areas such as object tracking, flight control, medical diagnosis, water pollution monitoring, temperature control monitoring and influence in social networks. For example, in object tracking we are often interested in choosing the measurements under limited resources that would be the most crucial in improving the posterior belief of a quantity of interest, e.g., location of a moving object. In medical diagnosis, we often need to select the tests that are most "orthogonal" to each other to better predict a patient's condition. In water pollution monitoring, we need to select the sensors that would give us better detection accuracy of polluted areas. In social networks, we are interested in finding the users that are most influential to other users.

Providing an optimal solution for the above problems is intractable as the number of variables grow. Therefore, there is need for approximate techniques that provide nearly optimal solutions in an efficient way, which usually translates to polynomial complexity in the number of hidden variables and the size of observation sets. Many previous works have approached the above problems with so-called *myopic* approaches, where the reward of the next step is being optimized based on what has been seen in the past. Most of the non-myopic extensions of these methods are either appropriate to very specific problems (e.g., observation models, dynamics models) or are limited to two or three time intervals (longer planning horizons are typically computationally prohibitive) [101]. Sensor management problems involving multiple time steps (in which decisions at a particular stage may utilize information received in all prior stages) can be formulated and conceptually solved using dynamic programming. However, the optimal solution of these problems often requires computation and storage of continuous functions with no finite parameterization, hence even problems involving small numbers of objects,

sensors, control choices and time steps may be intractable. In general, finding the optimal set that maximizes the reward may result in time and space complexity that makes it intractable even in problems with small number of hidden states, sensors, control modes and time steps. Finding greedy approaches that achieve nearly optimal solutions is an alternative that has been extensively used in the past [54, 55, 57, 102].

The aforementioned works focus on problems with selection constraints, in other words, in problems where we can select up to a fixed number of measurements for optimizing a predefined reward function. Determining a solution becomes even more challenging when the constraint set is more complicated. For example, in more realistic scenarios, different measurements have different costs and there is a certain budget. In addition to that, costs of measurements might change over time depending on the relative information content of measurements as we proceed in the planning process. Lastly, a commonly used metric in information planning is mutual information which measures the average reduction in the uncertainty of an underlying process of interest given measurements. When the mild condition of independence of measurements given the underlying process holds, a simple polynomial-time greedy algorithm gives nearly optimal solutions for the problem of maximizing mutual information under selection constraints. Another equally interesting problem though is what happens when conditional independence among measurements breaks.

The above formulations usually view the complexity of information planning in terms of the total number of feasible plans that satisfy a given set of constraints and neglect the complexity of evaluating the reward of each set of measurements. In other words, the evaluation of a set's information content, which is used for building the measurement plan, is assumed to be given in constant time. In many realistic settings though, the computational load of evaluating rewards cannot be ignored, since it depends on the size of latent graph, the number of measurements and size of constraint sets. In fact, as these parameters grow larger, the complexity of evaluating rewards becomes a really daunting task, which can make the application of even simple greedy techniques prohibitive.

■ 1.1 Motivation

Many active sensing problems are formulated as Bayesian inference. The problem of determining a set of measurements from an available set that maximizes an objective with respect to an underlying quantity of interest subject to constraints is known as *experimental design* [56]. In the context of distributed sensor networks, the idea of using information measures as an objective for Bayesian experimental design is a well-studied problem. Previous works [30, 40, 58, 103, 107] have considered the use of information rewards in sensor selection problems. Zhao et al. [107] suggests an one-step look-ahead approach in which the most informative sensor is chosen at each step and where information is quantified by the well-known mutual information measure. Kreucher et al. [58] suggest Rényi entropy [85] as a criterion for tasking sensors for multi-target

tracking. Both methods are characterized as myopic and greedy in the sense that they look one time-step ahead and select the single best measurement available. Resource constraints are implicit in these formulations.

Williams et al. [103] formulated active sensing as an approximate dynamic program also using mutual information, resulting in a tractable information-driven approach that allowed for multiple measurements and looking ahead for multiple time steps. The problem for multiple time steps is that the expected mutual information depends on previous selections. They overcome this by linearizing the measurement model about the state and approximating the uncertainty as Gaussian. This way they exploit the well known property that the information provided by jointly Gaussian measurement models is independent on the *value* of the measurement. This was shown to work well in practice when the spread of the state distribution is narrow with respect to the smoothness of the measurement model. As a consequence, sensor selection does not require simulation. It remains an open question if only Gaussian models satisfy this property.

Furthermore, in realistic settings, constraints on resources, such as energy and communication, preclude the use of all measurement actions and as such result in a combinatorial subset selection problem. These constraints are linked with poor link qualities between sensors, presence of obstacles, battery consumption, communication costs and in a more abstract level with the quality of the conveyed information. The result is a stochastic optimization problem whose complexity grows exponentially with the number of sensing actions. In the past, simpler greedy approaches have been proposed that attain near-optimal solutions. Some of the previous work determine bounds compared to the optimal solution in unit-cost settings where all measurements are available [53, 54], in sequential settings where measurement sets are assigned to different hidden states [102] and in budgeted settings where there is a cost associated with each measurement and a total budget b [52, 57]. All of the cited works rely on the assumption of non-decreasing reward functions. Also, greedy heuristics had to be modified *ad hoc* to take into account cost constraints. In addition, bounds in budgeted problems seemed to be much lower than those in the unit-cost case [52]. We are not aware of any extensive treatment of theoretical guarantees for the budgeted sequential setting. Existing literature has considered fixed cost settings [38, 52, 53, 54, 57]. An interesting alternative scenario would be when costs change based on the state of the world of each requesting party. Previous research that touches upon these issues has been conducted by Kempe et al. In [86], they consider problems where items can be allocated to many bidders, and the valuation of each individual bidder decreases as the items get allocated to additional bidders and derive sufficient conditions for truthful allocations. In essence, a player would be willing to pay less for a piece of information when it is shared with others. Kempe et al. [48] study the impact of budgets. They consider a scenario where not only the bidders' willingness to buy information is taken into account, but also their ability to pay.

Previous works have often made the assumption of *oracle value* models [51]. That is,

the reward corresponding to a set of measurements can be provided in constant time. In many realistic settings though such an assumption does not hold. For instance, we know that when mutual information is chosen as the reward function in Gaussian models, the complexity of information planning depends on the hidden dimension, on the size of a set of measurements under consideration and the available measurements. As such, Guestrin et al. [38] propose approximating truncation methods for Gaussian models. They also note the prohibitive costs of evaluating conditional entropies in [54], while Kempe et al. [47] acknowledge the complexity in evaluating the underlying influence function that guides the selection of the most influential nodes in a social network problem. The simply one-step look-ahead technique that has been proposed in previous works can be summarized as finding the best measurement at each step in some defined sense given past selections. To judge the contribution of a measurement, we compare the reward that a set of measurements conveys before and after the incorporation of that measurement. The sequential incorporation of new measurements into the model during the greedy selection process can be considered as a special case of adaptive inference. Adaptive inference refers to the problem of handling changes to the model more efficiently than performing inference from scratch. Adaptive inference is encountered not only in information planning problems, but in several other areas such as temperature monitoring, computational biology, when there is sequential changes in the model, either in the form of updates in the parameters or in the form of structural changes. These settings require repeated inference on variations of essentially the same model. It is therefore desirable to re-use as much information as we can from the previous computation while repeatedly solving the inference tasks.

■ 1.2 Contributions and thesis outline

This thesis makes contributions in two main areas. It studies the performance of greedy algorithms on a different variety of problems in information planning spanning from problems where the reward is non-monotone, to ones with finite budget allocation, with changing costs, or when a small part of the underlying process is of interest. Our goal is to not only design efficient algorithms, but also provide worst-case guarantees with respect to the (intractable) optimal solution that would justify their usage. Secondly, we will analyze the complexity of information planning in Gaussian models when mutual information is chosen as the reward function. We will show that sparsity can reduce substantially the computational load. The last task can be seen as a special case of adaptive inference, where the sufficient statistics of the underlying (hidden) process need to be updated at every step of the greedy algorithm. We will present a variation of belief propagation that is more suited to adaptive settings than standard belief propagation and can also be applied for information planning. We will provide below an overview of the thesis and summarize the problems and specific contributions.

1.2.1 Chapter 2: Background

In Chap. 2, we include a discussion of background material. We begin the chapter by describing Hidden Markov Models (HMMs), which are a simple type of graphical models with applications to several different areas. We continue the chapter by describing Gaussian HMMs, which have been extensively used in information planning and sensor selection problems. We present two standard inference techniques used in this setting, the Kalman filter and smoother. We expand the discussion to information measures that can be used as rewards in information planning problems. We continue with presenting essential elements of matroid theory and submodularity, two key properties for the existence of theoretical bounds when using greedy algorithms. The chapter continues by outlining the greedy algorithms that are typically used in such settings and presents worst-case bounds with respect to the optimal solution. Lastly, we discuss graphical models and how they can be represented, we present the exponential family of distributions, which holds a few very interesting properties and forms a broad group under which common distributions such as gaussian, uniform, poisson among others fall into. An extension to Gaussian graphical models is provided as Gaussian models are the subject of Chap. 4 and the chapter concludes by presenting typical algorithms that are suitable for inference in graphical models.

1.2.2 Chapter 3: Theoretical Guarantees of Greedy Algorithms

Chap. 3 begins by discussing *value independent* models and continues by providing necessary and sufficient conditions for existence of such models. Value independent models are models where knowledge of the value of measurements does not change the measurement plan. We show that since Gaussian models satisfy the sufficiency conditions, they can be considered as value independent models. We continue the chapter by presenting worst-case guarantees of greedy algorithms for several different settings. We first prove lower bounds for the case where we have multiple observation sets and the reward function is not monotone. The results of the above analysis can be used in information settings with constraints, where a penalized form of mutual information is more appropriate for incorporating both the information content and cost of a measurement. We also consider the case of varying costs of measurements depending on their relative information content. We show that under certain conditions, the bounds derived for the monotone unit-cost case can be extended to the varying-cost case. We continue the chapter by presenting upper bounds for the optimal solution in the Submodular Knapsack Maximization (SKM) problem. Submodular knapsack maximization is encountered when measurements have different costs and there is a limited budget. We additionally explore the case of focused inference, where only a small part of the latent graph is of interest. In this case, conditional independence between measurements given the latent graph breaks, which is a necessary condition for the existing bounds to hold. We show that by providing an extended latent superset (of the set of interest), which guarantees conditional independence between measurements, we can still apply the fa-

miliar myopic one-step look-ahead approaches and obtain worst-case bounds under mild conditions. We further show a method to obtain approximately minimal extended sets. We conclude the chapter by presenting an example, where measurements induce different costs and show that with the selection of penalized mutual information as a reward function, the generated greedy solution can have higher cumulative informational value than the solution generated when mutual information as selected as the objective.

1.2.3 Chapter 4: Complexity Reduction of Information Planning in Gaussian Models

The chapter commences by outlining the hardness of information planning problems. We continue the chapter by highlighting the complexity of evaluating mutual information rewards, a fact that has been underestimated or even ignored in previous works. We present related work, which mostly has assumed the existence of *oracle value* models. That is, models where answers to queries about rewards of sets of measurements are provided in constant time. We cite previous works that hint upon the inappropriateness of such value models in certain settings. We propose a different approach to perform greedy planning by taking sparsity between measurements and latent variables into account. Under the assumption that observation set sizes are large and each measurement depends only on a few latent variables, this analysis provides speedups several orders of magnitude larger than the standard approach. Another computational bottleneck during the greedy procedure is the propagation of uncertainty after the incorporation of a measurement at the end of each iteration. The standard approach is to use Kalman filtering and smoothing techniques that propagate the uncertainty to the node that is of interest at each iteration. We propose a variant of belief propagation that sends only the absolutely necessary messages to update the covariance at the end of interest at each iteration. Our analysis focuses on Gaussian HMMs for simplicity of exposition. We show later extensions to trees, loopy graphs as well as to non-linear models. We additionally suggest further reductions in complexity by ignoring measurements that are not valuable to planning. We do this by using the notion of submodularity. Lastly, we show synthetic experiments that show the tremendous speedups that we obtain through sparsity and by using our proposed variant of belief propagation instead of standard Kalman filtering approaches. The significant computational reductions in evaluating information rewards allow for consideration of more visitation orders of observation sets that satisfy the constraints. As such, we can find better measurement plans suggested by the greedy algorithm and better upper bounds for the optimal solution. As a last result, we show an example where the value of measurement sets is decoupled from the complexity they incur during the greedy process. This latter property can help us guide the selection process and characterize the number of different visitation orders that needs to be explored before we arrive at a satisfying greedy solution.

1.2.4 Chapter 5: Adaptive Belief Propagation

Chap. 5 focuses on adaptive inference settings, that is, in settings where there are sequential changes in the parameters of the graphical model. In such settings, sufficient statistics need to be computed without performing inference from scratch. We will additionally concentrate in focused inference settings, where only a few marginals are of interest. We present a variant of BP, termed *adaptive BP*, that is well-suited for such settings. We show that the algorithm is exact for trees and it applies both to discrete and Gaussian variables. Interestingly, we demonstrate that this algorithm is exact for Gaussian loopy graphs as well, when combined with the method by Liu et al. [64]. We provide a thorough complexity analysis and show that adaptive BP is always faster than standard BP in adaptive inference settings. We include extensions when multiple nodes are of interest or multiple nodes arrive at a time and extend the method to the MAP sequence problem, where the inference problem concerns finding the most likely sequence of the full latent graph. Furthermore, we consider the reverse problem, when we only have constraints on the number of measurements we can obtain from each set and our goal is to minimize the complexity of performing inference at every step. Lastly, we present experiments on both synthetic and real data. We show that our method is orders of magnitude faster than standard BP in the average case and provide conditions under which it outperforms state-of-the-art method by Sümer et al. [89]. We demonstrate the applicability of our method in two real datasets, on computational biology and temperature monitoring data, where we observe similar findings.

1.2.5 Chapter 6: Conclusion

In Chap. 6, we summarize the work and contributions of this thesis. We conclude by providing future directions.

1.2.6 Appendix

Derivations that would detract from the natural flow of the narrative have been included in the appendix.

Chapter 2

Background

In this chapter, we review relevant background material that we utilize to demonstrate our results. The primary problem of interest is that of determining a nearly optimal measurement plan. As a related problem, we are interested in sequential inference, where we make inference over latent parts of the graph as new measurements arrive. Both problems require the understanding of several related topics. One should be familiar with the most common approaches of modeling underlying physical phenomena as well as constructing estimators for conducting inference. For purposes of information planning, one should also select the objective under which valuable measurements would be chosen and therefore a discussion of different information measures is necessary. In addition, knowledge of the approximating methods that are used for planning is also crucial in understanding the theoretical guarantees and complexities that come with them. Lastly, a discussion of the most common inference algorithms is useful to become familiar with some of the different approaches used that provide solutions to different inference problems.

In Sec. 2.1, we discuss Hidden Markov Models (HMMs), a commonly used model for describing object tracking problems as well as other problems that can be seen as applications of information planning. We provide a special treatment of HMMs for the Gaussian case in Sec. 2.2, where we present the Kalman filter and smoother used for estimation of sufficient statistics at the latent nodes. In Sec. 2.3, we provide the most common information measures that are used as objectives in information planning. We present the entropy and mutual information, discuss their properties and outline generalizations of them such as the Kullback-Leibler divergence and f-divergences. In Sec. 2.4, we present the matroid theory which applies to the structures that we consider. In Sec. 2.5, we discuss submodularity, a necessary property that the chosen objectives should have such that the approximating methods have desirable theoretical guarantees. In Sec. 2.6, we present (approximating) greedy heuristics for different settings that come with nearly optimal lower bounds with respect to the optimal solutions. We are interested in two settings; a *batch* one where all measurements are available at all times and a *sequential* one, where all measurements might not be available at every step during the greedy selection process but may arrive at different points in time. Also, we deal with two types of constraints, one where there is constraint on the number of measurements and one where there is a budget constraint assuming that measurements have different

costs. In Sec. 2.7, we present graphical models and introduce the necessary language for describing stochastic phenomena. In Sec. 2.8, we give a quick overview of exponential families, which capture a large spectrum of known distributions, where inference is done in closed form, since posterior parameters can be derived in closed form. In Sec. 2.9, we focus on Gaussian graphical models, since they present special graph properties and derive the entropy and mutual information forms in the Gaussian case. We conclude with Sec. 2.10, where we present one of the most common algorithms for performing inference in graphical models; belief propagation. We outline the algorithm both for discrete and Gaussian models and discuss a new algorithm by Liu et al. [64], which provides exact results for Gaussian loopy graphs.

■ 2.1 Hidden Markov Models

Hidden Markov models (HMMs) are a generalization of a Markov chain. A *Markov chain* is a memoryless random process. In a first-order Markov chain, the next state depends only on the previous state (see Fig. 2.1a). The "memorylessness" property is known as *Markov property* [8]. Mathematically, the Markov property is expressed as

$$P(X_{t+1} = j \mid X_t = i, X_{t-1} = i_{t-1}, \ldots, X_1 = i_1) = P(X_{t+1} = j \mid X_t = i).$$

A Markov chain is *homogeneous/stationary*, when the conditional distribution is not time dependent. That is,

$$P(X_{t+1} = j \mid X_t = i) = P(X_{t+s+1} = j \mid X_{t+s} = i) = p_{ij}, \forall s.$$

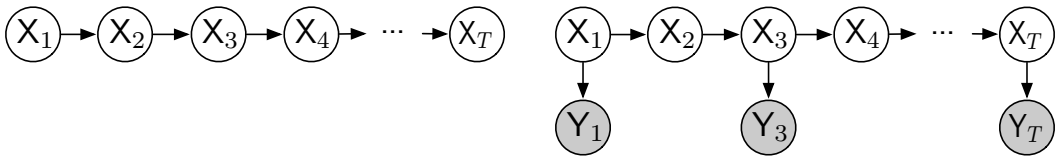
The joint distribution of a Markov chain of length T is given by

$$p(x_1, \ldots, x_T) = p(x_1) \prod_{t=2}^{T} p(x_t \mid x_{t-1}).$$

In a hidden Markov model, the state is not directly visible but rather indirectly observed through a (usually) noisy measurement. Intuitively, the variables denoted by X represent a process evolving over time that we do not directly observe. This process forms a Markov chain. We refer to the variables of this process as hidden variables. The variables denoted by Y (depicted as gray nodes in Fig. 2.1b) depend on the hidden variable of the same time step and are usually referred to as observed nodes. HMMs have numerous applications in several fields such as speech recognition [43], handwriting [75], gesture recognition [104], cryptanalysis [45] and bioinformatics [25]. The joint distribution in an HMM is given by

$$p(x_1, \ldots, x_T, y_1, \ldots, y_T) = p(x_1) \prod_{t=2}^{T} p(x_t \mid x_{t-1}) \prod_{t=1}^{T} p(y_t \mid x_t).$$

A common task under consideration is the posterior probability of a hidden state given



(a) Markov chain (b) HMM

Figure 2.1: Markov chain and HMM. (a) A Markov chain is a memoryless random process, where future values do not depend on past history given present. Here, we show an example of a first order Markov chain, where next state depends only on the current state and not on the sequence of events that preceded it. (b) A Hidden Markov model (HMM) is a model comprising of a latent stochastic process modeled by a Markov chain, and a series of observations that are generated by the hidden states.

all the observations, $P(\mathsf{X}_t = x_t \mid \mathsf{Y}_{1:T} = y_{1:T})$ There are certain algorithms that make inference rather easy for this type of problems. One of them is the forward-backward algorithm [84], which is a dynamic programming algorithm that uses two types of messages to determine the posterior marginal at every node. The forward (or alpha) message proceeds forward in time and represents the probability $\alpha_t(x_t) = p(x_t, y_{1:t})$, while the backward (beta) message goes backwards and represents probability $\beta_t(x_t) = p(y_{t+1:T} \mid x_t)$. The posterior marginal is proportional to

$$p(x_t \mid y_{1:T}) \propto \alpha_t(x_t)\beta_t(x_t).$$

The forward–backward algorithm constitutes a special case of a more general algorithm known as sum-product or belief propagation. We will delay the discussion of the latter method to Sec. 2.10.1.

■ 2.2 Gaussian Hidden Markov Models

In the previous section, we considered the case when the hidden variables are discrete and the observed ones are either discrete or continuous. A Gaussian HMM is a state space model, when both the hidden and observed variables follow a multivariate Gaussian distribution and are given by the following dynamics:

$$\mathsf{X}_{t+1} = A_t\mathsf{X}_t + \mathsf{V}_t \tag{2.1}$$

$$\mathsf{Y}_t = C_t\mathsf{X}_t + \mathsf{W}_t, \tag{2.2}$$

where $A_t \in \mathbb{R}^{d\times d}$, $C_t \in \mathbb{R}^{m\times d}$, $\mathsf{V}_t \sim \mathcal{N}(v_t; 0, Q_t)$, $\mathsf{W}_t \sim \mathcal{N}(w_t; 0, R_t)$, $\mathsf{X}_1 \sim \mathcal{N}(x_1; 0, \Sigma_1)$. Eq. (2.1) represents the dynamics of the underlying process, while Eq. (2.2) the measurement generating process. Variables $\mathsf{V}_t, \mathsf{W}_t$ represent the white Gaussian noises added in the dynamics and measurement processes. Here, we consider them to be zero-mean, but they can easily be generalized to non-zero Gaussians. When A_t, C_t, Q_t, R_t do not change w.r.t. time, the model is time-invariant.

2.2.1 Kalman Filter

The Kalman filter was introduced in 1960 by R. Kalman [44]. It is the minimum mean squared estimator (MMSE) for the quadratic loss objective

$$\hat{x}(y_{1:T}) = \arg\min_{\hat{x}} \mathbb{E}[(\hat{x} - X)^T(\hat{x} - X) \mid y_{1:T}],$$

where X is the true (unknown) underlying stochastic process. It turns out that MMSE is $\hat{x}(y_{1:T}) = \mathbb{E}[X \mid y_{1:T}]$, which is exactly the output of the Kalman filter. Kalman filter is also the optimal estimator to other criteria such as the mean absolute error and uniform cost, when the noise processes are Gaussian [101].

When the underlying system is a Gaussian HMM, the Kalman filter updates return the posterior mean and covariance of the hidden process $X_1, \ldots, X_T$ given the measurements. It is comprised of two steps, a *propagation* step

$$\hat{x}_{t|t-1} = \mathbb{E}[X_t \mid y_{1:t-1}] = A_{t-1}\hat{x}_{t-1|t-1} \tag{2.3}$$

$$\Sigma_{t|t-1} = \mathrm{cov}(X_t \mid y_{1:t-1}) = A_{t-1}\Sigma_{t-1|t-1}A_{t-1}^T + Q_{t-1}, \tag{2.4}$$

where the current state is predicted given the past history of measurements, and an *update* step

$$\hat{x}_{t|t} = \mathbb{E}[X_t \mid y_{1:t}] = \hat{x}_{t|t-1} + G_t(y_t - C_t\hat{x}_{t|t-1}) \tag{2.5}$$

$$\Sigma_{t|t} = \mathrm{cov}(X_t \mid y_{1:t}) = \Sigma_{t|t-1} - G_tC_t\Sigma_{t|t-1} \tag{2.6}$$

$$G_t = \Sigma_{t|t-1}C_t^T(C_t\Sigma_{t|t-1}C_t^T + R_t)^{-1}, \tag{2.7}$$

where the belief of the current hidden state is updated after the incorporation of the measurement at the same time point.

Since variables $X_1, \ldots, X_T, Y_1, \ldots, Y_T$ are jointly Gaussian, the expectations generated by the Kalman filter correspond to the MAP estimate of hidden process given the observations. It is worth noting that the covariance estimates do not depend at all on the measurement values. This is an observation which will prove critical later for determining the measurement schedule since commonly used information measures (entropy, mutual information) depend only on the covariance in Gaussian models. This allows for planning in advance that is no different from online planning that is taking place as new measurements are incorporated into the model.

2.2.2 Extended Kalman Filter

Kalman filter applies to linear Gaussian models. Extended Kalman filter (EKF) is the non-linear extension which linearizes about the current estimate value [28]. EKF linearizes about a working point by using multivariate Taylor expansions.

The non-linear model is given by:

$$X_{t+1} = f_t(X_t) + V_t \tag{2.8}$$

$$Y_t = h_t(X_t) + W_t, \tag{2.9}$$

where f_t, h_t are differentiable functions. In this case too, V_t, W_t are uncorrelated white noise processes.

The EKF propagation and update equations take the following form:

Propagation

$$\hat{x}_{t|t-1} = f_{t-1}(\hat{x}_{t-1|t-1}) \tag{2.10}$$

$$\Sigma_{t|t-1} = A_{t-1}\Sigma_{t-1|t-1}A_{t-1}^T + Q_{t-1} \tag{2.11}$$

Update

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + G_t(y_t - h_t(\hat{x}_{t|t-1})) \tag{2.12}$$

$$\Sigma_{t|t} = \Sigma_{t|t-1} - G_t C_t \Sigma_{t|t-1} \tag{2.13}$$

$$G_t = \Sigma_{t|t-1}C_t^T(C_t\Sigma_{t|t-1}C_t^T + R_t)^{-1}, \tag{2.14}$$

where $A_{t-1} = \nabla f_{t-1}(\hat{x}_{t-1|t-1})$, $C_t = \nabla h_t(\hat{x}_{t|t-1})$. Higher order EKFs retain more terms of the Taylor expansion and tend to provide benefits when the measurement noise is small. The estimated covariance tends to underestimate the true one and, in general, the estimates of an EKF can be far from the true value if the underlying model is highly non-linear.

2.2.3 Kalman Smoother

The Kalman filter provides the estimate of the hidden variable at the current time point given the past history of measurements. A more interesting quantity of interest is the belief of X_t given all the observations, $y_{1:T}$. This is accomplished with the so-called *Kalman smoothing*. The Rauch-Tung-Striebel (RTS) smoother is an efficient two-pass algorithm that outputs the posterior belief of each hidden variable given all the acquired measurements [26]. It starts with a forward pass, where the Kalman filter estimates are obtained and proceeds with a backward pass using the following recursive equations:

$$\hat{x}_{t|T} = \hat{x}_{t|t} + F_t(\hat{x}_{t+1|T} - \hat{x}_{t+1|t}) \tag{2.15}$$

$$\Sigma_{t|T} = \Sigma_{t|t} + F_t(\Sigma_{t+1|T} - \Sigma_{t+1|t})F_t^T \tag{2.16}$$

$$F_t = \Sigma_{t|t}A_t^T\Sigma_{t+1|t}^{-1}. \tag{2.17}$$

2.3 Information Measures

An essential part in a planning problem is to define a reward/cost function that determines the order of actions as well as the particular types of actions taken for a particular problem. In some settings, the choice of a reward function is tied to the task at hand. For example, in a supply-chain problem, the cost might be the distance between locations. The objective would be to choose the optimal order to visit a fixed set of locations that minimizes the distance traveled. In a medical decision problem, the reward might be the probability of a patient's survival and the objective to maximize this probability by choosing a number of medical tests to run (as there are limited resources). In finance, the cost might be the risk of an investor's portfolio as expressed by the covariance of

the financial assets constituting this portfolio and the objective to choose the particular assets (and their relative weighting) that would minimize this risk. It is evident that to determine a choice of actions, it is necessary to define a reward (or cost) linked to them. In a more abstract formulation, not tied to a particular problem, rewards can be defined in terms of the bits of information gained by choosing a particular set of items (measurements) to refine our belief over an underlying hidden process.

In information theory, objectives that are commonly used for this purpose are entropy, mutual information (MI), Kullback-Leibler (KL) divergence as well as the generalization of the previous measures known as f–divergences (or Ali–Silvey distances). We will continue by presenting definitions and basic properties of these quantities.

2.3.1 Entropy

Entropy is a measure of uncertainty in a random variable [20]. The term usually refers to Shannon entropy, which quantifies the expected value of information contained in a message. In other words, it is the amount of information required on average to fully describe the random variable.

It is defined as

$$H(X) = \mathbb{E}[-\log P(X)]. \tag{2.18}$$

For a discrete variable, Eq. (2.18) expands to

$$H(X) = -\sum_{x \in \mathcal{X}} p_X(x) \log p_X(x). \tag{2.19}$$

When X is continuous, we simply replace the sum with an integral

$$H(X) = -\int_x p_X(x) \log p_X(x) \, \mathrm{d}x. \tag{2.20}$$

The *joint entropy* of two variables X, Y is defined as

$$H(X, Y) = \mathbb{E}[-\log P(X, Y)]. \tag{2.21}$$

In the discrete case, the joint entropy of X, Y takes the form

$$H(X, Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p_{X,Y}(x, y) \log p_{X,Y}(x, y). \tag{2.22}$$

It is defined similarly for the continuous case. Lastly, the *conditional entropy* is defined as

$$H(X \mid Y) = \mathbb{E}[-\log P(X \mid Y)], \tag{2.23}$$

which boils down to

$$H(X \mid Y) = -\sum_{y \in \mathcal{Y}} p_Y(y) \sum_{x \in \mathcal{X}} p_{X|Y}(x \mid y) \log p_{X|Y}(x \mid y) \tag{2.24}$$

in the discrete case.

We define the *pointwise conditional entropy*

$$H(X \mid Y = y) = -\sum_{x \in \mathcal{X}} p_{X|Y}(x \mid y) \log p_{X|Y}(x \mid y). \tag{2.25}$$

It is connected to the conditional entropy with the following formula

$$H(X \mid Y) = \mathbb{E}_Y[H(X \mid Y = y)]. \tag{2.26}$$

We continue by presenting two important properties of entropy, the non-negativity for discrete variables and the chain rule.

Lemma 2.3.1 (Non-negativity of entropy of a discrete r.v.)**.** *For a discrete variable X, we have that*

$$H(X) \geq 0. \tag{2.27}$$

Proof. We have

$$H(X) = -\sum_{x \in \mathcal{X}} p_X(x) \log p_X(x) = \sum_{x \in \mathcal{X}} p_X(x) \log \frac{1}{p_X(x)} \geq 0,$$

since $p_X(x) \geq 0$, we have $\log \frac{1}{p_X(x)} \geq 0, \forall x$. ■

The same holds true for pointwise conditional entropy and hence for conditional entropy as well.

Theorem 2.3.1 (Chain rule of entropy)**.** *Joint entropy, entropy and conditional entropy are related as*

$$H(X, Y) = H(X \mid Y) + H(Y). \tag{2.28}$$

Proof. We have

$$\begin{aligned} H(X, Y) &= -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p_{X,Y}(x, y) \log p_{X,Y}(x, y) \\ &= -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p_{X,Y}(x, y) \log p_{X|Y}(x \mid y) - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p_{X,Y}(x, y) \log p_Y(y) \\ &= H(X \mid Y) + H(Y). \end{aligned}$$

■

It trivially holds that

$$H(X, Y) = H(Y \mid X) + H(X).$$

Corollary 2.3.1 (Chain rule of conditional entropy)**.** *Let X, Y, Z be random variables. It holds that*

$$H(X, Y \mid Z) = H(X \mid Y, Z) + H(Y \mid Z). \tag{2.29}$$

Corollary 2.3.2 (Chain rule of entropy for multiple variables)**.** *Let $X_1, \ldots, X_N$ be drawn according to $p_{X_1,\ldots,X_N}(x_1, \ldots, x_N)$. Then, the join entropy can be decomposed as*

$$H(X_1, \ldots, X_N) = \sum_{i=2}^{N} H(X_i \mid X_1, \ldots, X_{i-1}) + H(X_1). \tag{2.30}$$

Theorem 2.3.2 ("Information never hurts" principle for entropy)**.** *Let X, Y be random variables. Conditioning on Y, the uncertainty over X reduces on average.*

$$H(X \mid Y) \leq H(X). \tag{2.31}$$

Proof. We will defer the proof for this theorem until the introduction of mutual information in Sec. 2.3.2. ■

It is important to note that this inequality is true on average. In other words, it might not hold for a particular instance of pointwise entropy, but it holds on average.

Example 2.3.1. *Let X, Y be binary random variables with the following joint distribution.*

X \ Y	0	1
0	0	ϵ
1	$1-2\epsilon$	ϵ

We have that

$$\begin{aligned} H(X) &= -\epsilon \log_2(\epsilon) - (1-\epsilon)\log_2(1-\epsilon) \\ H(X \mid Y=0) &= -(1-2\epsilon)\log_2(1-2\epsilon) \\ H(X \mid Y=1) &= -2\epsilon \log_2(\epsilon) \\ H(X \mid Y) &= -4\epsilon^2 \log_2(\epsilon) - (1-2\epsilon)^2 \log_2(1-2\epsilon). \end{aligned}$$

If we set $\epsilon = 1/4$, we have $H(X) = 2 - \frac{3}{4}\log_2(3) \approx 0.8113$, $H(X \mid Y=0) = 1/2$, $H(X \mid Y=1) = 1$, $H(X \mid Y) = 3/4$. Even though we have $H(X) < H(X \mid Y=1)$, it holds on average that $H(X) \geq H(X \mid Y)$.

■ 2.3.2 Mutual Information

Mutual information (MI), $I(X;Y)$, is a measure of distance between the joint distribution $p_{X,Y}(\cdot,\cdot)$ and the product distribution $p_X(\cdot)p_Y(\cdot)$.

$$I(X;Y) = \mathbb{E}\left[\log \frac{P(X,Y)}{P(X)P(Y)}\right]. \tag{2.32}$$

For a discrete variable, it expands to

$$I(X;Y) = \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{X,Y}(x,y) \log \frac{p_{X,Y}(x,y)}{p_X(x)p_Y(y)}. \tag{2.33}$$

We also define the *pointwise mutual information* as

$$I(X; Y = y) = \sum_{x \in \mathcal{X}} p_{X|Y}(x \mid y) \log \frac{p_{X|Y}(x \mid y)}{p_X(x)}. \tag{2.34}$$

MI is the mean of pointwise mutual information over distribution $p_Y(\cdot)$:

$$I(X; Y) = \mathbb{E}_Y[I(X; Y = y)]. \tag{2.35}$$

From Eq. (2.35), we can see that MI is the gain of information on average (reduction in bits of uncertainty), that is achieved by incorporating observation Y to update our belief over X's distribution.

The *conditional mutual information* of variables X, Y given Z is defined as

$$I(X; Y \mid Z) = \mathbb{E}\left[\log \frac{P(X, Y \mid Z)}{P(X \mid Z)P(Y \mid Z)}\right]. \tag{2.36}$$

We will present now a few key properties of MI that will prove useful for later.

Theorem 2.3.3 (MI and entropy)**.** *MI is linked to entropy with the following equivalent relations*

$$I(X; Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X) \tag{2.37}$$
$$= H(X) + H(Y) - H(X, Y). \tag{2.38}$$

Theorem 2.3.4 (Symmetry of MI)**.** *Let X, Y be random variables. The MI between X, Y is symmetric.*

$$I(X; Y) = I(Y; X). \tag{2.39}$$

In other words, the reduction in uncertainty that Y brings to X's posterior belief on average is the same with the reduction that X brings to Y's posterior belief on average.

Corollary 2.3.3 (Symmetry of conditional MI)**.** *Let X, Y, Z be random variables. The MI between X, Y given Z is symmetric.*

$$I(X; Y \mid Z) = I(Y; X \mid Z). \tag{2.40}$$

Theorem 2.3.5 (Non-negativity of MI)**.** *MI is non-negative, in other words, it never hurts to obtain an observation on average.*

$$I(X; Y) \geq 0. \tag{2.41}$$

Proof. We have

$$-I(X; Y) = \mathbb{E}_{X,Y}\left[\log \frac{P(X)P(Y)}{P(X, Y)}\right].$$

Let $Z = \frac{P(X)P(Y)}{P(X,Y)}$, then

$$-I(X;Y) = \mathbb{E}_Z[\log Z] \overset{(a)}{\leq} \log\left(\mathbb{E}_Z[Z]\right) \overset{(b)}{=} \log 1 = 0,$$

where (a) is due to Jensen's inequality. The second equality (b) holds, because $\mathbb{E}_Z[Z] = \sum_z p_Z(z)z = \sum_{x,y} p_{X,Y}(x,y)\frac{p_X(x)p_Y(y)}{p_{X,Y}(x,y)} = \sum_{x,y} p_X(x)p_Y(y) = 1$. Therefore, $I(X;Y) \geq 0$. If X, Y are continuous, sums are replaced with integrals. ■

Corollary 2.3.4 (Non-negativity of conditional MI)**.** *It holds that*

$$I(X;Y \mid Z) \geq 0, \tag{2.42}$$

with equality if and only if X and Y are conditionally independent given Z.

Thm. 2.3.2 holds trivially, since $I(X;Y) \geq 0$ and $I(X;Y) = H(X) - H(X \mid Y)$.

The equivalent of "information never hurts" principle for MI is the following

Theorem 2.3.6 ("Information never hurts" principle for MI)**.** *Let X, Y, Z be random variables. Then, the following inequality holds*

$$I(X;Y) \leq I(X;Y,Z). \tag{2.43}$$

That is, the gain is always bigger the more observations we add to update the posterior belief of a variable of interest (here, X).

Theorem 2.3.7 (Chain rule of MI)**.** *Let $X_1, \ldots, X_N$ be drawn according to $p_{X_1,\ldots,X_N}(x_1,\ldots,x_N)$. Then, it holds that*

$$I(X_1,\ldots,X_N;Y) = I(Y;X_1,\ldots,X_N) = \sum_{i=2}^{N} I(X_i;Y \mid X_1,\ldots,X_{i-1}) + I(X_1;Y). \tag{2.44}$$

The chain rule holds for conditional MI as well. That is, the MI of $X_1,\ldots,X_N, Y$ given Z can be decomposed as

$$I(X_1,\ldots,X_N;Y \mid Z) = \sum_{i=2}^{N} I(X_i;Y \mid X_1,\ldots,X_{i-1},Z) + I(X_1;Y \mid Z). \tag{2.45}$$

Theorem 2.3.8 (Data-processing inequality)**.** *Let X, Y, Z form the following Markov chain, $X \to Y \to Z$. That is, Z directly depends on Y and Y depends on X. Then, the following inequality holds*

$$I(X;Y) \geq I(X;Z). \tag{2.46}$$

Proof. By the chain rule, we have

$$I(X;Y,Z) = I(X;Y \mid Z) + I(X;Z) = I(X;Z \mid Y) + I(X;Y).$$

Since $X \perp\!\!\!\perp Z \mid Y$, $I(X;Z \mid Y) = 0$. Therefore,

$$I(X;Y) = I(X;Y \mid Z) + I(X;Z)$$

and since $I(X;Y \mid Z) \geq 0$, we have that

$$I(X;Y) \geq I(X;Z).$$

■

Corollary 2.3.5 (Data-processing inequality for conditional MI)**.** *Let X, Y, Z form the following Markov chain, $X \to Y \to Z$. Then, the following inequality holds*

$$I(X;Y) \geq I(X;Y \mid Z). \tag{2.47}$$

Proof. The proof is exactly the same with that of Thm. 2.3.8. ■

Obviously, if $Z = g(Y)$, that is, if Z is a function of Y, the Markovianity assumption of Thm. 2.3.8 is satisfied and the inequality $I(X;Y) \geq I(X;g(Y))$ holds. It is worth noting that the dependencies can be reversed and the data-processing inequalities would still hold. In other words, we could have $X \leftarrow Y \leftarrow Z$, while inequalities (2.46), (2.47) would still be in place. Lastly, as in the entropy case, these inequalities hold on average. In other words, for a particular pointwise quantity, the inequality might be violated, but it is satisfied on average.

Example 2.3.2. *Let X, Y, Z be binary random variables with the following joint distribution.*

$X = 0$

Y \ Z	0	1
0	$\frac{1}{3} - 2\epsilon$	$\frac{2}{3}\epsilon$
1	$\frac{2}{3} - 4\epsilon$	$\frac{4}{3}\epsilon$

$X = 1$

Y \ Z	0	1
0	ϵ	ϵ
1	ϵ	ϵ

We have

$$p_X(x) = \begin{cases} 1-4\epsilon, & x = 0 \\ 4\epsilon, & x = 1 \end{cases}, \quad p_Y(y) = \begin{cases} \frac{1}{3} + \frac{2}{3}\epsilon, & y = 0 \\ \frac{2}{3} - \frac{2}{3}\epsilon, & y = 1 \end{cases}, \quad p_Z(z) = \begin{cases} 1-4\epsilon, & z = 0 \\ 4\epsilon, & z = 1 \end{cases}$$

$$p_{X|Y}(x \mid y = 0) = \begin{cases} \frac{1-4\epsilon}{1+2\epsilon}, & x = 0 \\ \frac{6\epsilon}{1+2\epsilon}, & x = 1 \end{cases}, \quad p_{X|Y}(x \mid y = 1) = \begin{cases} \frac{1-4\epsilon}{1-\epsilon}, & x = 0 \\ \frac{3\epsilon}{1-\epsilon}, & x = 1 \end{cases}$$

$$p_{X|Z}(x \mid z=0) = \begin{cases} \frac{1-6\epsilon}{1-4\epsilon}, & x=0 \\ \frac{2\epsilon}{1-4\epsilon}, & x=1 \end{cases}, \quad p_{X|Z}(x \mid z=1) = \begin{cases} \frac{1}{2}, & x=0 \\ \frac{1}{2}, & x=1 \end{cases}$$

$$p_{Z|Y}(z \mid y=0) = \begin{cases} \frac{1-3\epsilon}{1+2\epsilon}, & z=0 \\ \frac{5\epsilon}{1+2\epsilon}, & z=1 \end{cases}, \quad p_{Z|Y}(z \mid y=1) = \begin{cases} \frac{2-9\epsilon}{2-2\epsilon}, & z=0 \\ \frac{7\epsilon}{2-2\epsilon}, & z=1 \end{cases}$$

$$p_{X|Y,Z}(x \mid y=0, z=0) = \begin{cases} \frac{1-6\epsilon}{1-3\epsilon}, & x=0 \\ \frac{3\epsilon}{1-3\epsilon}, & x=1 \end{cases}, \quad p_{X|Y,Z}(x \mid y=0, z=1) = \begin{cases} \frac{2}{5}, & x=0 \\ \frac{3}{5}, & x=1 \end{cases}$$

$$p_{X|Y,Z}(x \mid y=1, z=0) = \begin{cases} \frac{2-12\epsilon}{2-9\epsilon}, & x=0 \\ \frac{3\epsilon}{2-9\epsilon}, & x=1 \end{cases}, \quad p_{X|Y,Z}(x \mid y=1, z=1) = \begin{cases} \frac{4}{7}, & x=0 \\ \frac{3}{7}, & x=1 \end{cases}$$

We can easily show that $Y \perp\!\!\!\perp Z \mid X$*, since*

$$p_{Z|X}(z \mid x=0) = p_{Z|X,Y}(z \mid x=0, y=0) = p_{Z|X,Y}(z \mid x=0, y=1) = \begin{cases} \frac{1-6\epsilon}{1-4\epsilon}, & z=0 \\ \frac{2\epsilon}{1-4\epsilon}, & z=1 \end{cases}$$

$$p_{Z|X}(z \mid x=1) = p_{Z|X,Y}(z \mid x=1, y=0) = p_{Z|X,Y}(z \mid x=1, y=1) = \begin{cases} \frac{1}{2}, & z=0 \\ \frac{1}{2}, & z=1 \end{cases}$$

Therefore, X, Y, Z *form the Markov chain,* $Z \leftrightarrow X \leftrightarrow Y$ *and Cor. 2.3.5 should hold. In order to show that Cor. 2.3.5 holds only on average, we need to evaluate the quantities* $I(X;Y), I(X;Z \mid Y=0), I(X;Z \mid Y=1), I(X;Z \mid Y)$*:*

$$I(X;Y) = \sum_y p_Y(y) \sum_x p_{X|Y}(x \mid y) \log \frac{p_{X|Y}(x \mid y)}{p_X(x)}$$

$$I(X;Z \mid Y=0) = \sum_z p_{Z|Y}(z \mid y=0) \sum_x p_{X|Y,Z}(x \mid y=0, z) \log \frac{p_{X|Y,Z}(x \mid y=0, z)}{p_{X|Y}(x \mid y=0)}$$

$$I(X;Z \mid Y=1) = \sum_z p_{Z|Y}(z \mid y=1) \sum_x p_{X|Y,Z}(x \mid y=1, z) \log \frac{p_{X|Y,Z}(x \mid y=1, z)}{p_{X|Y}(x \mid y=1)}$$

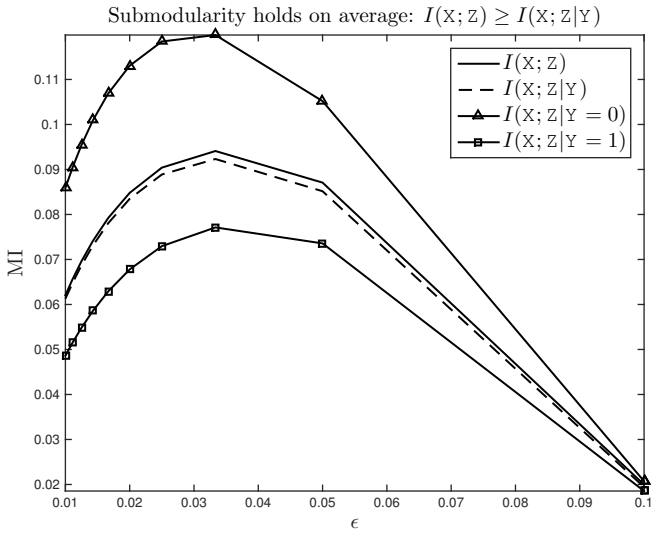
$$I(X;Z \mid Y) = p_Y(y=0) I(X;Z \mid Y=0) + p_Y(y=1) I(X;Z \mid Y=1),$$

which are depicted in the following figure for $\epsilon \in [0.01, 0.1]$*. We observe that* $I(X;Z \mid Y=0) \geq I(X;Z)$*, but on average* $I(X;Z) \geq I(X;Z \mid Y)$*.*

■ 2.3.3 Kullback-Leibler Divergence

The *Kullback-Leibler (KL) divergence* or *relative entropy* is a measure of distance between two distributions. It is defined as

$$D(p \| q) = \mathbb{E}_p \left[\log \frac{p(X)}{q(X)} \right]. \tag{2.48}$$



For example, if X is a discrete random variable, the above relation takes the form

$$D(p\| q) = \sum_{x\in\mathcal{X}} p(x) \log \frac{p(x)}{q(x)}.$$

KL-divergence is not symmetric, in other words, $D(p\| q) \neq D(q\| p)$ in general.

Theorem 2.3.9 (Non-negativity of KL-divergence)**.** *KL is non-negative,*

$$D(p\| q) \geq 0, \tag{2.49}$$

with equality if and only if $p(x) = q(x), \forall x$.

Proof. The proof follows the same logic to that of Thm. 2.3.5. ■

■ 2.3.4 f–divergences

f–divergence (or else known as Ali-Silvey distance) is a generalization of KL-divergence. It measures the difference between two probability distributions p, q weighted by a function f [2]. It is defined as

$$D_f(p\| q) = \mathbb{E}_q \left[f\left(\frac{p(X)}{q(X)} \right) \right], \tag{2.50}$$

where f is a convex function. For example, if X is a discrete random variable, the above relation transforms to

$$D_f(p\| q) = \sum_{x\in\mathcal{X}} q(x) f\left(\frac{p(x)}{q(x)} \right).$$

It degenerates to KL-divergence, if we set $f(t) = t\log(t)$:

$$D_{t\log(t)}(p\| q) = \sum_{x\in\mathcal{X}} q(x) \frac{p(x)}{q(x)} \log \frac{p(x)}{q(x)} = \sum_{x\in\mathcal{X}} p(x) \log \frac{p(x)}{q(x)} = D(p\| q).$$

■ 2.4 Matroid Theory

Matroids are an abstraction of the notion of linear independence in vector spaces [36]. They are very important because they correspond to a rich class of combinatorial optimization problems that can be solved rather efficiently. Before we proceed in defining what a matroid is, we will first present the notions of set system and an independence system.

Definition 2.4.1 (Set system)**.** *A (finite) ground set $\mathcal{V}$ and a set of subsets of $\mathcal{V}$, $\mathscr{I} \subseteq 2^{\mathcal{V}}$ is called a set system, denoted by $(\mathcal{V}, \mathscr{I})$.*

Set $\mathcal{V}$ can represent the full set of measurements, a subset of which needs to be selected under a particular optimization objective to update the belief over an underlying process. More generally, $\mathcal{V}$ can represent a set of items.

Definition 2.4.2 (Independence system)**.** *A set system $(\mathcal{V}, \mathscr{I})$ is an independence system, if $\emptyset \in \mathscr{I}$ (emptyset containing) and $\forall \mathcal{I} \in \mathscr{I}, \mathcal{J} \subset \mathcal{I}$, it holds that $\mathcal{J} \in \mathscr{I}$ (down-closed or subclusive).*

Example 2.4.1. *Let $\mathcal{V} = \{1, 2, 3, 4\}$ be the ground set of measurements and $\mathscr{I} = \{\emptyset, \{1\}, \{1, 2\}, \{1, 2, 4\}\}$. $(\mathcal{V}, \mathscr{I})$ is not an independence system, since even though $\{1, 2, 4\} \in \mathscr{I}$ and $\{1, 4\} \subset \{1, 2, 4\}$, $\{1, 4\} \notin \mathscr{I}$. However, if $\mathscr{I} = \{\emptyset, \{1\}, \{2\}, \{4\}, \{1, 2\}, \{1, 4\}, \{2, 4\}, \{1, 2, 4\}\}$, then $(\mathcal{V}, \mathscr{I})$ is an independence system.*

Essentially, for a set system to be independent, every subset of a set belonging to the independence system should belong to the system as well.

Definition 2.4.3 (Maximal set)**.** *A maximal set $\mathcal{A} \in \mathscr{I}$ is a set such that adding any other element from ground set $\mathcal{V}$ makes $\mathcal{A}$ no longer a member of the independence system. More formalistically, $\mathcal{A} \in \mathscr{I}$ is maximal if $\mathcal{A} \cup \{e\} \notin \mathscr{I}, \forall e \in \mathcal{V} \setminus \mathcal{A}$.*

Example 2.4.2. *Suppose we have the independence system $(\mathcal{V}, \mathscr{I}) = (\{1, 2, 3, 4\}, \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 4\}, \{2, 4\}, \{1, 2, 4\}\})$. Singleton set $\{1\}$ is not maximal since we can add elements $\{2\}$ or $\{4\}$ to it that would belong to the independence system as well. However, $\{3\}$ is a maximal set since by adding any other element, the concatenation will not belong to $\mathscr{I}$ anymore. The same holds for set $\{1, 2, 4\}$.*

Definition 2.4.4 (Maximal independent subsets of $\mathcal{A}$)**.** *Assume we have the independence system $(\mathcal{V}, \mathscr{I})$. The maximal independent subsets or bases of set $\mathcal{A}$ is a collection of subsets of $\mathcal{A}$ that are maximal sets of the independence system $(\mathcal{V}, \mathscr{I})$. In other words,*

$$\text{maxInd}(\mathcal{A}) = \{\mathcal{I} \mid \mathcal{I} \subseteq \mathcal{A}, \mathcal{I} \in \mathscr{I} \text{ and } \forall e \in \mathcal{A} \setminus \mathcal{I}, \mathcal{I} \cup \{e\} \notin \mathscr{I}\}.$$

Matroids are a particular type of an independence system for which efficient optimization algorithms exist for certain problems. They represent a generalization of algebraic linear independence. Matroids are independence systems with the additional property that $\forall \mathcal{I}, \mathcal{J} \in \mathscr{I}$ with $|\mathcal{I}| = |\mathcal{J}| + 1$, there exists an element $e \in \mathcal{I} \setminus \mathcal{J}$ such that $\mathcal{J} \cup \{e\} \in \mathscr{I}$. In more detail, a matroid is defined as follows:

Definition 2.4.5 (Matroid)**.** *A set system $\mathscr{M} = (\mathcal{V}, \mathscr{I})$ is a matroid, if*

(a) $\emptyset \in \mathscr{I}$ *(emptyset containing)*

(b) $\forall \mathcal{I} \in \mathscr{I}, \mathcal{J} \subset \mathcal{I}$, $\mathcal{J} \in \mathscr{I}$ *(down-closed or subclusive).*

(c) $\forall \mathcal{I}, \mathcal{J} \in \mathscr{I}$ *with* $|\mathcal{I}| = |\mathcal{J}| + 1$*, there exists* $e \in \mathcal{I} \setminus \mathcal{J}$ *such that* $\mathcal{J} \cup \{e\} \in \mathscr{I}$*.*

Equivalent conditions for (c) are:

(c′) $\forall \mathcal{I}, \mathcal{J} \in \mathscr{I}$ with $|\mathcal{I}| > |\mathcal{J}|$, there exists $e \in \mathcal{I} \setminus \mathcal{J}$ such that $\mathcal{J} \cup \{e\} \in \mathscr{I}$.

(c″) Let $\mathcal{A} \subseteq \mathcal{V}$. For all $\mathcal{I}_1, \mathcal{I}_2 \in \text{maxInd}(\mathcal{A})$, $|\mathcal{I}_1| = |\mathcal{I}_2|$. In other words, all maximal subsets (bases) of $\mathcal{A}$ have the same size.

Definition 2.4.6 (Rank of matroid)**.** *The rank function $r_{\mathscr{M}} : 2^{\mathcal{V}} \mapsto \mathbb{N}$ of a matroid $\mathscr{M} = (\mathcal{V}, \mathscr{I})$ for a set $\mathcal{I} \subseteq \mathcal{V}$ is defined as the cardinality of the maximal subset of $\mathcal{I}$ that is still a member of the independence system $\mathscr{I}$:*

$$r_{\mathscr{M}}(\mathcal{I}) = \max\{|\mathcal{J}| \mid \mathcal{J} \subseteq \mathcal{I}, \mathcal{J} \in \mathscr{I}\}.$$

It obviously holds that $0 \leq r_{\mathscr{M}}(\mathcal{I}) \leq |\mathcal{I}|$.

■ 2.4.1 Examples

There are numerous mathematical structures which satisfy the properties of a matroid. Below we outline a few of them.

Linear Matroid

Let A be an $M \times N$ matrix and $\mathcal{V} = \{1, \ldots, N\}$. Also, let $\mathscr{I}$ be the collection of sets such that if $\mathcal{I} = \{i_1, \ldots, i_k\} \in \mathscr{I}$, then the column vectors $A_{i_1}, \ldots, A_{i_k}$ are linearly independent. The system $\mathscr{M} = (\mathcal{V}, \mathscr{I})$ forms a linear matroid.

Graphic Matroid

Let $G = (V, \mathcal{E})$ be a graph, with V, $\mathcal{E}$ representing the node and edge sets, respectively. Define $\mathcal{E}$ as the ground set and $\mathscr{I}$ the collection of subsets of $\mathcal{E}$ that do not form a cycle. That is, if $\mathcal{E}_s \in \mathscr{I}$ and covers the set of nodes V_s, then graph $G_s = (V_s, \mathcal{E}_s)$ is either a forest or a tree. In other words, a graphic matroid contains all the forests and trees for a given graph G. The system $\mathscr{M} = (\mathcal{E}, \mathscr{I})$ is a graphic matroid.

Uniform Matroid

Let $\mathcal{V} = \{1, \ldots, N\}$. For any non-negative integer $k \leq N$, we define $\mathscr{I} = \{\mathcal{I} \mid \mathcal{I} \subseteq \mathcal{V}, |\mathcal{I}| \leq k\}$. Then, $\mathscr{M} = (\mathcal{V}, \mathscr{I})$ is a uniform matroid of rank k. It is usually denoted by $\mathscr{M}_{k,N}$.

Partition Matroid

Let $\mathcal{V} = \mathcal{V}_1 \cup \cdots \cup \mathcal{V}_T$ be a partition of $\mathcal{V}$ into T disjoint sets. We define $\mathscr{I} = \{\mathcal{I} \mid \mathcal{I} \subseteq \mathcal{V}, |\mathcal{I} \cap \mathcal{V}_t| \leq k_t, \forall t = 1, \ldots, T\}$, where $k_1, \ldots, k_T$ are fixed parameters. Then, $\mathscr{M} = (\mathcal{V}, \mathscr{I})$ is a partition matroid. Partition matroid is a generalization of a uniform matroid.
The last two structures would be of most interest to us in this thesis. The rank of the uniform matroid is $r_{\mathscr{M}}(\mathcal{I}) = \min\{|\mathcal{V} \cap \mathcal{I}|, k\}$, while it is $r_{\mathscr{M}}(\mathcal{I}) = \sum_{t=1}^{T} \min\{|\mathcal{V}_t \cap \mathcal{I}|, k_t\}$ for a partition matroid.

■ 2.5 Submodularity

Submodular functions is a special case of set functions that when applied on matroidal structures in a greedy fashion can give nice theoretical guarantees with respect to the optimal value. They have been important in several fields, such as combinatorics [92], social networks [72], computer vision [6, 12], finance [7] and economics [27]. Below, we define formally the notions of a set function, normalized function, non-negative function, monotone function and submodular function. We present a few key properties of the latter.

A set function $f : 2^{\mathcal{V}} \mapsto \mathbb{R}$ is a real-valued function that takes a set as an input and returns a real number. It can be thought of as utility measure that quantifies the value/cost of a set of items (measurements) over some defined metric.

Definition 2.5.1 (Normalized function)**.** *A set function f is normalized if $f(\emptyset) = 0$.*

Definition 2.5.2 (Non-negative function)**.** *A set function f is non-negative if $f(\mathcal{A}) \geq 0, \forall \mathcal{A} \subseteq \mathcal{V}$.*

Definition 2.5.3 (Non-decreasing (monotone) function)**.** *A set function f is non-decreasing (monotone) if $f(\mathcal{A}) \leq f(\mathcal{B}), \forall \mathcal{A} \subseteq \mathcal{B}$.*

Corollary 2.5.1 (Non-negativity for monotone functions)**.** *If for a monotone function holds that $f(\emptyset) \geq 0$, then $f(\mathcal{A}) \geq 0, \forall \mathcal{A}$.*

Definition 2.5.4 (Submodular function)**.** *A set function f is submodular if*

$$f(\mathcal{A} \cup \mathcal{B}) + f(\mathcal{A} \cap \mathcal{B}) \leq f(\mathcal{A}) + f(\mathcal{B}), \forall \mathcal{A}, \mathcal{B} \subseteq \mathcal{V}. \tag{2.51}$$

If the inequality sign in Eq. (2.51) is reversed, that is, $f(\mathcal{A} \cup \mathcal{B}) + f(\mathcal{A} \cap \mathcal{B}) \geq f(\mathcal{A}) + f(\mathcal{B}), \forall \mathcal{A}, \mathcal{B} \subseteq \mathcal{V}$, the function is called *supermodular*. When Eq. (2.51) is satisfied with equality for every pair $\mathcal{A}, \mathcal{B}$, the function is called *modular*. Obviously, a modular function is both submodular and supermodular. Usually, certain function properties are more easily identifiable through an alternative definition, that of, *increment function*.

Definition 2.5.5 (Increment function)**.** *Increment function $f(\mathcal{A} \mid \mathcal{B})$ is defined as $f(\mathcal{A} \mid \mathcal{B}) \triangleq f(\mathcal{A} \cup \mathcal{B}) - f(\mathcal{B})$.*

If $\mathcal{A} = \{e\}$, in other words, it is a single element, it represents the marginal gain we have by adding element e to set $\mathcal{B}$. We can usually check more easily for monotonicity and submodularity of a function via its incremental definition.

Definition 2.5.6 (Non-decreasing (monotone) function - alternative)**.** *A set function f is non-decreasing (monotone) if $f(j \mid \mathcal{A}) \geq 0, \forall j, \mathcal{A}$, where j a single element of ground set $\mathcal{V}$.*

Definition 2.5.7 (Submodular function - alternative)**.** *A set function f is submodular if*

$$f(j \mid \mathcal{A}) \geq f(j \mid \mathcal{B}), \forall \mathcal{A} \subseteq \mathcal{B}, j \notin \mathcal{B}. \tag{2.52}$$

Eq. (2.52) embodies the notion of "diminishing returns". In other words, an item (observation) is worth more, when we have obtained fewer rather than more items. Below are some equivalent definitions of submodularity that might as well prove useful depending on the problem setting.

$$\begin{aligned}
&(i) \quad f(\mathcal{C} \mid \mathcal{A}) \geq f(\mathcal{C} \mid \mathcal{B}), \forall \mathcal{A} \subseteq \mathcal{B}, \mathcal{C} \subseteq \mathcal{V} \setminus \mathcal{B}. \\
&(ii) \quad f(j \mid \mathcal{A}) \geq f(j \mid \mathcal{A} \cup \{e\}), \forall \mathcal{A} \subseteq \mathcal{V}, j \in \mathcal{V} \setminus (\mathcal{A} \cup \{e\}). \\
&(iii) \quad f(\mathcal{B}) \leq f(\mathcal{A}) + \sum_{j \in \mathcal{B} \setminus \mathcal{A}} f(j \mid \mathcal{A}), \forall \mathcal{A} \subseteq \mathcal{B}. \\
&(iv) \quad f(\mathcal{A}) \leq f(\mathcal{B}) - \sum_{j \in \mathcal{B} \setminus \mathcal{A}} f(j \mid \mathcal{B} \setminus \{j\}), \forall \mathcal{A} \subseteq \mathcal{B}.
\end{aligned}$$

A non-negative, normalized submodular function is subadditive:

$$f(\mathcal{A} \cup \mathcal{B}) \leq f(\mathcal{A}) + f(\mathcal{B}). \tag{2.53}$$

Lastly, submodularity is closed under the operations of non-negative addition, restriction and conditioning [10].

Lemma 2.5.1 (Submodularity closed under conic combination)**.** *Given submodular functions $f_1, f_2, \ldots, f_n$, their conic combination (non-negative addition) is submodular.*

$$f(\mathcal{A}) = \sum_{i=1}^{n} \alpha_i f_i(\mathcal{A}), \tag{2.54}$$

where $\alpha_i \geq 0$.

Proof. Since every function f_i is submodular, and $\alpha_i \geq 0, \forall i$ we have

$$f_i(j \mid \mathcal{A}) \geq f_i(j \mid \mathcal{B}) \Rightarrow \sum_{i=1}^{n} \alpha_i f_i(j \mid \mathcal{A}) \geq \sum_{i=1}^{n} \alpha_i f_i(j \mid \mathcal{B}) \Rightarrow f(j \mid \mathcal{A}) \geq f(j \mid \mathcal{B}), \forall \mathcal{A} \subseteq \mathcal{B}.$$

■

Lemma 2.5.2 (Submodularity closed under restriction/marginalization)**.** *If function g is submodular, then*

$$f(\mathcal{A}) = g(\mathcal{A} \cap \mathcal{C}). \tag{2.55}$$

is submodular for every fixed set $\mathcal{C}$ as well.

Proof. For every pair $\mathcal{A}, \mathcal{B}$ such that $\mathcal{A} \subseteq \mathcal{B}$ and for a fixed set $\mathcal{C}$ we consider the following two cases: $j \in \mathcal{V} \setminus (\mathcal{B} \cup \mathcal{C})$, $j \in \mathcal{C} \setminus \mathcal{B}$.

For $j \in \mathcal{V} \setminus (\mathcal{B} \cup \mathcal{C})$, we have that

$$\begin{aligned} f(j \mid \mathcal{A}) &= f(j \cup \mathcal{A}) - f(\mathcal{A}) = g((j \cup \mathcal{A}) \cap \mathcal{C}) - g(\mathcal{A} \cap \mathcal{C}) = g((\mathcal{A} \cap \mathcal{C}) \cup (j \cap \mathcal{C})) - g(\mathcal{A} \cap \mathcal{C}) \\ &= g((\mathcal{A} \cap \mathcal{C}) \cup \emptyset) - g(\mathcal{A} \cap \mathcal{C}) = g((\mathcal{A} \cap \mathcal{C})) - g(\mathcal{A} \cap \mathcal{C}) = 0. \end{aligned}$$

Similarly, $f(j \mid \mathcal{B}) = 0$. Therefore, $f(j \mid \mathcal{A}) \geq f(j \mid \mathcal{B}), \forall \mathcal{A} \subseteq \mathcal{B}, j \in \mathcal{V} \setminus (\mathcal{B} \cup \mathcal{C})$.

For $j \in \mathcal{C} \setminus \mathcal{B}$, we have

$$\begin{aligned} f(j \mid \mathcal{A}) &= f(j \cup \mathcal{A}) - f(\mathcal{A}) = g((j \cup \mathcal{A}) \cap \mathcal{C}) - g(\mathcal{A} \cap \mathcal{C}) = g((\mathcal{A} \cap \mathcal{C}) \cup (j \cap \mathcal{C})) - g(\mathcal{A} \cap \mathcal{C}) \\ &= g((\mathcal{A} \cap \mathcal{C}) \cup j) - g(\mathcal{A} \cap \mathcal{C}) = g(j \cup (\mathcal{A} \cap \mathcal{C})) - g(\mathcal{A} \cap \mathcal{C}) = g(j \mid \mathcal{A} \cap \mathcal{C}). \end{aligned}$$

It holds that $f(j \mid \mathcal{A}) = g(j \mid \mathcal{A} \cap \mathcal{C}) \geq g(j \mid \mathcal{B} \cap \mathcal{C}) = f(j \mid \mathcal{B})$, due to g's submodularity and because $\mathcal{A} \subseteq \mathcal{B}$ and consequently $\mathcal{A} \cap \mathcal{C} \subseteq \mathcal{B} \cap \mathcal{C}$ as well. ■

Lemma 2.5.3 (Submodularity closed under contraction/conditioning)**.** *If function g is submodular, then*

$$f(\mathcal{A}) = g(\mathcal{A} \cup \mathcal{C}) - g(\mathcal{C}) \tag{2.56}$$

is submodular for every fixed set $\mathcal{C}$ as well.

Proof. For every $\mathcal{A} \subseteq \mathcal{B}, j \in \mathcal{V} \setminus \mathcal{B}$ and for a fixed set $\mathcal{C}$, we have

$$f(j \mid \mathcal{A}) = f(j \cup \mathcal{A}) - f(\mathcal{A}) = g(j \cup \mathcal{A} \cup \mathcal{C}) - g(\mathcal{C}) - g(\mathcal{A} \cup \mathcal{C}) + g(\mathcal{C}) = g(j \mid \mathcal{A} \cup \mathcal{C}).$$

Similarly, $f(j \mid \mathcal{B}) = g(j \mid \mathcal{B} \cup \mathcal{C})$. Therefore, $f(j \mid \mathcal{A}) = g(j \mid \mathcal{A} \cup \mathcal{C}) \geq g(j \mid \mathcal{B} \cup \mathcal{C}) = f(j \mid \mathcal{B})$, since g is submodular and for $\mathcal{A} \subseteq \mathcal{B}$, it holds that $\mathcal{A} \cup \mathcal{C} \subseteq \mathcal{B} \cup \mathcal{C}$ as well. ■

Lemma 2.5.4 (Submodularity closed under composition)**.** *Given functions $f : \mathbb{R} \mapsto \mathbb{R}$, $g : 2^{\mathcal{V}} \mapsto \mathbb{R}$, the composition $h = f \circ g$ is non-decreasing submodular, if f is non-decreasing concave and g is non-decreasing submodular.*

Proof. It is clearly non-decreasing, because for $\mathcal{A} \subseteq \mathcal{B}$, we have that $g(\mathcal{A}) \leq g(\mathcal{B})$, since g is non-decreasing. In addition, $f(g(\mathcal{A})) \leq f(g(\mathcal{B}))$ since f is non-decreasing as well. To prove submodularity, we initially observe that due to g's monotonicity, it holds that $g(\mathcal{A} \cap \mathcal{B}) \leq g(\mathcal{B}) \leq g(\mathcal{A} \cup \mathcal{B})$. Then, there exists some number $\lambda \in [0, 1]$, such that

$$g(\mathcal{B}) = \lambda g(\mathcal{A} \cap \mathcal{B}) + (1 - \lambda) g(\mathcal{A} \cup \mathcal{B}). \tag{2.57}$$

Due to g's submodularity it holds that $g(\mathcal{A}) + g(\mathcal{B}) \geq g(\mathcal{A} \cup \mathcal{B}) + g(\mathcal{A} \cap \mathcal{B})$. Therefore,

$$\begin{aligned}
g(\mathcal{A}) &\geq g(\mathcal{A} \cup \mathcal{B}) + g(\mathcal{A} \cap \mathcal{B}) - g(\mathcal{B}) \overset{f\text{'s monotonicity}}{\Rightarrow} \\
f(g(\mathcal{A})) &\geq f(g(\mathcal{A} \cup \mathcal{B}) + g(\mathcal{A} \cap \mathcal{B}) - g(\mathcal{B})) \overset{\text{Eq.(2.57)}}{\Rightarrow} \\
&\geq f(\lambda g(\mathcal{A} \cup \mathcal{B}) + (1-\lambda) g(\mathcal{A} \cap \mathcal{B})) \overset{f\text{'s concavity}}{\Rightarrow} \\
&\geq \lambda f(g(\mathcal{A} \cup \mathcal{B})) + (1-\lambda) f(g(\mathcal{A} \cap \mathcal{B})) \\
&= f(g(\mathcal{A} \cup \mathcal{B})) + f(g(\mathcal{A} \cap \mathcal{B})) - (\lambda f(g(\mathcal{A} \cap \mathcal{B})) + (1-\lambda) f(g(\mathcal{A} \cup \mathcal{B}))) \overset{f\text{'s concavity}}{\Rightarrow} \\
&\geq f(g(\mathcal{A} \cup \mathcal{B})) + f(g(\mathcal{A} \cap \mathcal{B})) - f(\lambda g(\mathcal{A} \cap \mathcal{B}) + (1-\lambda) g(\mathcal{A} \cup \mathcal{B})) \overset{\text{Eq.(2.57)}}{\Rightarrow} \\
&= f(g(\mathcal{A} \cup \mathcal{B})) + f(g(\mathcal{A} \cap \mathcal{B})) - f(g(\mathcal{B})) \Rightarrow \\
f(g(\mathcal{A})) + f(g(\mathcal{B})) &\geq f(g(\mathcal{A} \cup \mathcal{B})) + f(g(\mathcal{A} \cap \mathcal{B})).
\end{aligned}$$

■

■ 2.5.1 Entropy

Entropy can be seen as a set function. In other words, if we denote by $\mathcal{A}$ a set of random variable indices, then entropy can be defined as

$$f(\mathcal{A}) \triangleq H(X_{\mathcal{A}}). \tag{2.58}$$

Lemma 2.5.5 (Monotonicity of entropy for discrete random variables)**.** *In the case of discrete random variables, entropy is a non-decreasing function.*

Proof. For all $\mathcal{A} \subseteq \mathcal{B}$, $H(X_{\mathcal{B}}) = H(X_{(\mathcal{B} \setminus \mathcal{A}) \cup \mathcal{A}}) = H(X_{\mathcal{B} \setminus \mathcal{A}} \mid X_{\mathcal{A}}) + H(X_{\mathcal{A}}) \geq H(X_{\mathcal{A}})$, since $H(X_{\mathcal{B} \setminus \mathcal{A}} \mid X_{\mathcal{A}}) \geq 0$, for discrete variables. ■

Lemma 2.5.6 (Submodularity of entropy [54])**.** *Entropy is a submodular function.*

Proof. For all $\mathcal{A}, \mathcal{B} \subseteq \mathcal{V}$,

$$\begin{aligned}
I(X_{\mathcal{A} \setminus \mathcal{B}}; X_{\mathcal{B} \setminus \mathcal{A}} \mid X_{\mathcal{A} \cap \mathcal{B}}) &\overset{(a)}{\geq} 0 \\
H(X_{\mathcal{A} \setminus \mathcal{B}} \mid X_{\mathcal{A} \cap \mathcal{B}}) + H(X_{\mathcal{B} \setminus \mathcal{A}} \mid X_{\mathcal{A} \cap \mathcal{B}}) - H(X_{(\mathcal{A} \setminus \mathcal{B}) \cup (\mathcal{B} \setminus \mathcal{A})} \mid X_{\mathcal{A} \cap \mathcal{B}}) &\geq 0,
\end{aligned} \tag{2.59}$$

where (a) is due to MI's non-negativity. We additionally have that

$$\begin{aligned}
H(X_{\mathcal{A} \setminus \mathcal{B}} \mid X_{\mathcal{A} \cap \mathcal{B}}) &= H(X_{(\mathcal{A} \setminus \mathcal{B}) \cup (\mathcal{A} \cap \mathcal{B})}) - H(X_{\mathcal{A} \cap \mathcal{B}}) = H(X_{\mathcal{A}}) - H(X_{\mathcal{A} \cap \mathcal{B}}) \\
H(X_{\mathcal{B} \setminus \mathcal{A}} \mid X_{\mathcal{A} \cap \mathcal{B}}) &= H(X_{(\mathcal{B} \setminus \mathcal{A}) \cup (\mathcal{A} \cap \mathcal{B})}) - H(X_{\mathcal{A} \cap \mathcal{B}}) = H(X_{\mathcal{B}}) - H(X_{\mathcal{A} \cap \mathcal{B}}) \\
H(X_{(\mathcal{A} \setminus \mathcal{B}) \cup (\mathcal{B} \setminus \mathcal{A})} \mid X_{\mathcal{A} \cap \mathcal{B}}) &= H(X_{(\mathcal{A} \setminus \mathcal{B}) \cup (\mathcal{B} \setminus \mathcal{A}) \cup (\mathcal{A} \cap \mathcal{B})}) - H(X_{\mathcal{A} \cap \mathcal{B}}) = H(X_{\mathcal{A} \cup \mathcal{B}}) - H(X_{\mathcal{A} \cap \mathcal{B}})
\end{aligned}$$

Therefore, Eq. (2.59) becomes

$$H(X_{\mathcal{A}}) + H(X_{\mathcal{B}}) \geq H(X_{\mathcal{A} \cup \mathcal{B}}) + H(X_{\mathcal{A} \cap \mathcal{B}}).$$

■

2.5.2 Mutual Information

MI can be expressed as a set function as follows

$$f(\mathcal{A}) \triangleq I(X; Y_{\mathcal{A}}). \tag{2.60}$$

Here, we make the distinction between latent variables (X) and observed variables (Y). For any pair $\mathcal{I}, \mathcal{J} \subseteq \mathcal{V}$ such that $\mathcal{I} \cap \mathcal{J} = \emptyset$, we assume that $Y_{\mathcal{I}} \perp\!\!\!\perp Y_{\mathcal{J}} \mid X$. In that case, MI is both non-decreasing and submodular [54].

Lemma 2.5.7 (Monotonicity of MI)**.** *MI is a non-decreasing function.*

Proof. For all $j, \mathcal{A}$, $f(j \mid \mathcal{A}) = I(X; Y_{j \cup \mathcal{A}}) - I(X; Y_{\mathcal{A}}) = I(X; Y_j \mid Y_{\mathcal{A}}) \geq 0$, due to the monotonicity of MI. ■

Lemma 2.5.8 (Submodularity of MI)**.** *MI is a submodular function if observed variables are conditionally independent given the latent state.*

Proof. For all $\mathcal{A} \subseteq \mathcal{B}, j \in \mathcal{V} \setminus \mathcal{B}$, we have

$$\begin{aligned}
I(Y_j; Y_{\mathcal{B} \setminus \mathcal{A}} \mid Y_{\mathcal{A}}) &\overset{(a)}{\geq} 0 \\
H(Y_j \mid Y_{\mathcal{A}}) - H(Y_j \mid Y_{(\mathcal{B} \setminus \mathcal{A}) \cup \mathcal{A}}) &\geq 0 \\
H(Y_j \mid Y_{\mathcal{A}}) &\geq H(Y_j \mid Y_{\mathcal{B}}) \\
H(Y_j \mid Y_{\mathcal{A}}) - H(Y_j \mid X) &\geq H(Y_j \mid Y_{\mathcal{B}}) - H(Y_j \mid X) \\
H(Y_j \mid Y_{\mathcal{A}}) - H(Y_j \mid X, Y_{\mathcal{A}}) &\overset{(b)}{\geq} H(Y_j \mid Y_{\mathcal{B}}) - H(Y_j \mid X, Y_{\mathcal{B}}) \\
I(Y_j; X \mid Y_{\mathcal{A}}) &\geq I(Y_j; X \mid Y_{\mathcal{B}}) \\
I(X; Y_j \mid Y_{\mathcal{A}}) &\geq I(X; Y_j \mid Y_{\mathcal{B}}),
\end{aligned}$$

where (a) is due to MI's monotonicity and (b) due to the conditional independency assumption of observations given the latent state X. ■

2.6 Greedy Heuristics

In this thesis, we focus on open-loop control structures. In other words, on structures where the value of future actions is averaged over all values of current observations. In this framework, planning is done completely in advance and does not take into account realizations of observations. In addition, we focus on two types of selection problems, one where all measurements are available and another one where different measurements might be available at different times. In the latter setting, measurements might belong to different sets of observations, where different constraints are applied to them. We will call the first setting, *batch* setting and the second one, *sequential* setting. The goal is to select the set of measurements that maximizes some predefined reward f under certain constraints. For the batch setting, we will examine both the selection constraint

problem, where there is a constraint on the number of measurements we can obtain and the budget constraint problem where measurements have different costs and there is a finite budget b. For the selection constraint version of batch setting, we have a set $\mathcal{V}$ of N observations/items and we are interested in selecting up to k of them that maximize some predefined reward, while for the budget constraint problem each measurement u has some fixed cost $c(u)$ and there is a certain budget b that we cannot exceed.

Similarly, for the sequential setting, we will consider the selection constraint and the budget constraint problem. In the sequential setting, the observation set is split into T disjoint observation sets, $\mathcal{V} = \cup_{t=1}^{T} \mathcal{V}_t$, where $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset, \forall i \neq j$ and $|\mathcal{V}_t| = N_t, \forall t$. Each of these observation sets is generated from a different part of the hidden state $\boldsymbol{X}$. In the selection constraint version, the task is to select up to k_t observations from each set $\mathcal{V}_t$. In the budgeted version, the task is to select measurements with varying costs that would maximize some specified reward of interest under finite budget b_t for each observation set $\mathcal{V}_t$. The budgeted problem is also known as *submodular knapsack maximization (SKM)* problem.

For the *batch* setting, the selection problem can be cast as a combinatorial optimization problem

$$\mathcal{O} \in \underset{\{\mathcal{S} \subseteq \mathcal{V} |\ |\mathcal{S}| \leq k\}}{\arg\max} f(\mathcal{S}), \tag{2.61}$$

while the budgeted problem can be cast as

$$\mathcal{O} \in \underset{\{\mathcal{S} \subseteq \mathcal{V} |\ c(\mathcal{S}) \leq b\}}{\arg\max} f(\mathcal{S}), \tag{2.62}$$

where $c(\mathcal{S}) = \sum_{u \in \mathcal{S}} c(u)$ is the cost of set $\mathcal{S}$, which is additive and b is the budget. To get an indication of the hardness of problems (2.61), (2.62), we need to take into account all $\binom{N}{k}$ combinations of sets of size k to find the optimal solution for problem (2.61), while we need to consider all 2^N sets from size 1 to N to find the optimal solution of problem (2.62).

The selection version of the *sequential* setting is similar to the batch one

$$\mathcal{O} \in \underset{\{\mathcal{S} \subseteq \mathcal{V} |\ |\mathcal{S} \cap \mathcal{V}_t| \leq k_t\}}{\arg\max} f(\mathcal{S}), \tag{2.63}$$

while the budgeted one

$$\mathcal{O} \in \underset{\{\mathcal{S} \subseteq \mathcal{V} |\ c(\mathcal{S} \cap \mathcal{V}_t) \leq b_t\}}{\arg\max} f(\mathcal{S}). \tag{2.64}$$

Assuming $N_t = N, k_t = k, \forall t$, finding the optimal solutions for problems (2.63), (2.64) would require the consideration of $\binom{N}{k}^T$ and 2^{NT} sets of measurements for the first and second problem, respectively.

It becomes evident that solving the above problems optimally becomes intractable as the number of observation sets T and observation set sizes N grow. However, we often resort to greedy approximate techniques that choose the current action based on one-step horizon policies and run in polynomial time. Usually, these methods come with

no guarantees and can be far from the optimal value. Fortunately, when the objective f is non-decreasing and submodular, there are lower-bound guarantees compared to the optimal solution. We will additionally assume that f is normalized, $f(\emptyset) = 0$.

■ 2.6.1 Batch setting - Selection problem

Nemhauser et al. [77] proposed the following greedy algorithm for the batch setting

Algorithm 2.1 Batch Selection Greedy Heuristic

Initialization
Set $\mathcal{G}_0 = \emptyset$.

Iteration
for j in $1 : k$ **do**
  Select g_j s.t. $g_j \in \arg\max_{u \in \mathcal{V}} f(u \mid \mathcal{G}_{j-1})$.
  Set $\mathcal{G}_j = \mathcal{G}_{j-1} \cup \{g_j\}$.
  Set $\mathcal{V} = \mathcal{V} \setminus \{g_j\}$.
end for
$\mathcal{G} = \mathcal{G}_k$ is the greedy solution.

That is, at every stage we select the observation g_j that maximizes the incremental reward based on the observations that have been selected so far, $\mathcal{G}_{j-1}$. The collection of greedy sets $\{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3, \ldots\}$ at every stage form a uniform matroid. The value of greedy solution is at least 63.2% close to the optimal in addition to reward f being non-decreasing and submodular. The complexity of this approach is $\mathcal{O}(Nk)$ as at every step we need to explore at most N measurements and there is a total of k steps. Since $k \leq N$, we can also argue that the running time is $\mathcal{O}(N^2)$.

Theorem 2.6.1 (Performance bounds in the batch setting [77])**.** *If the greedy method described in Alg. 2.1 is applied to problem* $\max_{|\mathcal{S}| \leq k} f(\mathcal{S})$ *under a non-decreasing submodular reward* f*, it achieves a value no worse than* $1 - 1/e \approx 0.632$ *of the optimal solution.*

Proof. We start by observing that since f is non-decreasing the selection constraint would be satisfied with equality as it does not hurt to acquire more measurements. Therefore, the cardinality of the greedy $\mathcal{G}$ and optimal set $\mathcal{O}$ would be k, $|\mathcal{G}| = |\mathcal{O}| = k$. In addition, let us denote the greedy set at stage j by $\mathcal{G}_j$. It obviously holds that $\mathcal{G}_1 \subseteq \mathcal{G}_2 \subseteq \cdots \subseteq \mathcal{G}_k \equiv \mathcal{G}$. We have that

$$f(\mathcal{O}) \leq f(\mathcal{O} \cup \mathcal{G}_{j-1}) \leq f(\mathcal{G}_{j-1}) + \sum_{u \in \mathcal{O} \setminus \mathcal{G}_{j-1}} f(u \mid \mathcal{G}_{j-1}). \tag{2.65}$$

The first and second inequalities hold due to monotonicity and submodularity of f, respectively.[1]

[1] We have made use of definition (*iii*) of submodularity (see Sec. 2.5).

Eq. (2.65) becomes

$$f(\mathcal{O}) \leq f(\mathcal{G}_{j-1}) + \sum_{u \in \mathcal{O} \setminus \mathcal{G}_{j-1}} f(u \mid \mathcal{G}_{j-1}) \overset{(a)}{\leq} f(\mathcal{G}_{j-1}) + \sum_{u \in \mathcal{O} \setminus \mathcal{G}_{j-1}} f(g_j \mid \mathcal{G}_{j-1})$$
$$\leq f(\mathcal{G}_{j-1}) + |\mathcal{O} \setminus \mathcal{G}_{j-1}| f(g_j \mid \mathcal{G}_{j-1}) \overset{(b)}{\leq} f(\mathcal{G}_{j-1}) + k f(g_j \mid \mathcal{G}_{j-1}), \tag{2.66}$$

where inequality (a) comes from the definition of greedy method presented in Alg. 2.1 and (b) because $|\mathcal{O} \setminus \mathcal{G}_{j-1}| \leq k$.

We additionally have

$$f(\mathcal{G}_{j-1}) = \underbrace{f(\mathcal{G}_{j-1}) - f(\mathcal{G}_{j-2})}_{f(g_{j-1} \mid \mathcal{G}_{j-2})} + \underbrace{f(\mathcal{G}_{j-2}) - f(\mathcal{G}_{j-3})}_{f(g_{j-2} \mid \mathcal{G}_{j-3})} + \cdots + \underbrace{f(\mathcal{G}_2) - f(\mathcal{G}_1)}_{f(g_2 \mid \mathcal{G}_1)} + \underbrace{f(\mathcal{G}_1)}_{f(g_1 \mid \emptyset)}$$
$$= \sum_{i=1}^{j-1} f(g_i \mid \mathcal{G}_{i-1}),$$

where $\mathcal{G}_1 = \{g_1\}, \mathcal{G}_0 = \emptyset, f(g_1 \mid \emptyset) = f(g_1)$ since $f(\emptyset) = 0$.

Therefore, Eq. (2.66) becomes

$$f(\mathcal{O}) \leq \sum_{i=1}^{j-1} f(g_i \mid \mathcal{G}_{i-1}) + k f(g_j \mid \mathcal{G}_{j-1}). \tag{2.67}$$

Inequality (2.67) holds for every $j = 1, 2, \ldots, k$. One natural question to ask is how close is the worst-case greedy solution to the optimal one. In other words, what is

$$\min_{f(g_1), f(g_2 \mid \mathcal{G}_1), \ldots, f(g_k \mid \mathcal{G}_{k-1})} f(\mathcal{G}) = \sum_{j=1}^{k} f(g_j \mid \mathcal{G}_{j-1})$$
$$\text{s.t. } f(\mathcal{O}) \leq \sum_{i=1}^{j-1} f(g_i \mid \mathcal{G}_{i-1}) + k f(g_j \mid \mathcal{G}_{j-1}), \forall j = 1, 2, \ldots, k,$$

since any greedy solution as described in Alg. 2.1 needs to satisfy the above constraints. Here, $\mathcal{G} = \{g_1, g_2, \ldots, g_k\}$.

The above problem can be easily cast as a linear optimization problem by setting $y_j \triangleq f(g_j \mid \mathcal{G}_{j-1})$:

$$\min_{y_1, \ldots, y_k} \sum_{j=1}^{k} y_j$$
$$\text{s.t. } f(\mathcal{O}) \leq \sum_{i=1}^{j-1} y_i + k y_j, \forall j = 1, 2, \ldots, k.$$

By taking the dual of the above problem, we have

$$\max_{x_1,\dots,x_k} \; f(\mathcal{O}) \sum_{j=1}^{k} x_j$$
$$\text{s.t. } kx_j + \sum_{i=j+1}^{k} x_i = 1, \forall j = 1, 2, \dots, k,$$

which leads to the unique (maximizing) solution

$$x_j^* = \frac{1}{k}\left(1 - \frac{1}{k}\right)^{k-j}, \forall j = 1, 2, \dots, k.$$

It can be easily seen that $\sum_{j=1}^{k} x_j^*$ is a geometric series with constant ratio $1 - 1/k$ and therefore $\sum_{j=1}^{k} x_j^* = 1 - \left(1 - \frac{1}{k}\right)^k$. Since the optimization problem is linear, there is no duality gap and hence the maximizing value of the dual problem is the minimizing value of the original one

$$\sum_{j=1}^{k} y_j^* = f(\mathcal{O})\left(1 - \left(1 - \frac{1}{k}\right)^k\right).$$

As a small reminder, $\sum_{j=1}^{k} y_j^*$ represents the worst (minimum) value that the greedy solution can take. Therefore, any greedy solution obtained from Alg. 2.1 is greater equal to it

$$f(\mathcal{G}) \geq \sum_{j=1}^{k} y_j^* = \left(1 - \left(1 - \frac{1}{k}\right)^k\right) f(\mathcal{O}). \tag{2.68}$$

Finally, note that

$$1 - \left(1 - \frac{1}{k}\right)^k \geq \lim_{k\to\infty}\left(1 - \left(1 - \frac{1}{k}\right)^k\right) = 1 - 1/e.$$

Therefore, inequality (2.68) becomes

$$f(\mathcal{G}) \geq (1 - 1/e)\, f(\mathcal{O}).$$

■

■ 2.6.2 Batch setting - Budgeted problem

In the budgeted case, we assume that each measurement u has a fixed cost $c(u) \geq 0$ and there is a finite budget b. The problem we wish to solve is finding the set of measurements that maximizes some specified reward f under the budget constraint b.

$$\mathcal{O} \in \arg\max_{c(\mathcal{S}) \leq b} f(\mathcal{S}).$$

Algorithm 2.2 Batch Budgeted Greedy Heuristic - Sviridenko

Determining best set of cardinality one or two
Select $\mathcal{G}^{[1,2]}$ s.t. $\mathcal{G}^{[1,2]} \in \arg\max_{\{\mathcal{S}\subseteq\mathcal{V}||\mathcal{S}|\leq 2, c(\mathcal{S})\leq b\}} f(\mathcal{S})$.
Determining best set of cardinality three or more
Set $\mathcal{G}^{[3]} = \emptyset$ and $f(\mathcal{G}^{[3]}) = -\infty$.
for each $\mathcal{S}$ s.t. $|\mathcal{S}| = 3$ **do**
  Set $\mathcal{G}_0^s = \mathcal{S}, \mathcal{I}_0^s = \mathcal{V}, i = 1$.
  while $\mathcal{I}_{i-1}^s \setminus \mathcal{G}_{i-1}^s \neq \emptyset$ **do**
    Select g_i s.t. $g_i \in \arg\max_{u\in\mathcal{I}_{i-1}^s\setminus\mathcal{G}_{i-1}^s} f(u \mid \mathcal{G}_{i-1}^s)/c(u)$.
    if $c(\mathcal{G}_{i-1}^s) + c(u) \leq b$ **then**
      Set $\mathcal{G}_i^s = \mathcal{G}_{i-1}^s \cup \{g_i\}$ and $\mathcal{I}_i^s = \mathcal{I}_{i-1}^s$.
    else
      Set $\mathcal{G}_i^s = \mathcal{G}_{i-1}^s$ and $\mathcal{I}_i^s = \mathcal{I}_{i-1}^s \setminus \{g_i\}$.
    end if
  end while
  if $f(\mathcal{G}_{N-3}^s) > f(\mathcal{G}^{[3]})$ **then**
    Set $\mathcal{G}^{[3]} = \mathcal{G}_{N-3}^s$.
  end if
end for
$\mathcal{G} = \mathcal{G}^{[1,2]}$, if $f(\mathcal{G}^{[1,2]}) \geq \mathcal{G}^{[3]}$ or $\mathcal{G} = \mathcal{G}^{[3]}$, otherwise.

Sviridenko [90] proposed Alg. 2.2. The algorithm works in two phases. In the first phase, we determine the feasible set $\mathcal{G}^{[1,2]}$ of cardinality one or two that has the largest value of the objective function f. In the second phase, the algorithm considers all feasible sets of cardinality three. For each such set $\mathcal{G}_0^s = \mathcal{S}$ such that $|\mathcal{S}| = 3$, the method adds new elements that correspond to the maximum ratio of marginal value $f(u \mid \mathcal{G}_{i-1}^s)$ to cost $c(u)$ as long as they correspond to a feasible solution: $c(\mathcal{G}_{i-1}^s) + c(u) \leq b$. Otherwise, this measurement is exempted from further consideration. At the end, we keep the set $\mathcal{G}^{[3]}$ with the largest value f. The greedy solution would be $\mathcal{G} = \mathcal{G}^{[1,2]}$, if $f(\mathcal{G}^{[1,2]}) \geq \mathcal{G}^{[3]}$ or $\mathcal{G} = \mathcal{G}^{[3]}$, otherwise. The complexity of this approach is $\mathcal{O}(N^5)$ as for each set of cardinality three, we need to explore at most $\mathcal{O}(N^2)$ measurements and there are $\binom{N}{3}$ possible combinations of sets of size 3. Feige [31] showed that no such algorithm exists with better guarantees unless P = NP.

Krause and Guestrin [52] proposed a technique that gives a lower approximating ratio $(1 - 1/\sqrt{e}) \approx 0.394$, but runs only in $\mathcal{O}(N^2)$ time. The algorithm works in two phases just like Sviridenko's method. In the first phase, it determines the single feasible element with the highest reward and stores in set $\mathcal{G}^{[1]}$. In the second phase, it starts by setting the greedy set $\mathcal{G}^{[2]} = \emptyset$ and the exploration set $\mathcal{I} = \mathcal{V}$. Then, as long as there are still elements in the exploration set $\mathcal{I}$, it determines the one with the highest incremental reward $f(g \mid \mathcal{G}^{[2]})$ to cost $c(g)$ ratio. If the solution with the addition of element g remains feasible, g is added to the existing greedy set $\mathcal{G}^{[2]}$. At the end of

the current iteration, element g is removed from the exploration set $\mathcal{I}$. At the end, we select greedy set $\mathcal{G}^{[1]}$ if $f(\mathcal{G}^{[1]}) \geq \mathcal{G}^{[2]}$ or $\mathcal{G}^{[2]}$, otherwise. A summary of this method is given in Alg. 2.3.

Algorithm 2.3 BATCH BUDGETED GREEDY HEURISTIC - KRAUSE ET AL.

Determining best set of cardinality one
Select $\mathcal{G}^{[1]}$ s.t. $\mathcal{G}^{[1]} \in \arg\max_{\{u \in \mathcal{V} | c(u) \leq b\}} f(u)$.
Determining best set of cardinality two or more
Set $\mathcal{G}^{[2]} = \emptyset, \mathcal{I} = \mathcal{V}$.
while $\mathcal{I} \neq \emptyset$ **do**
  Select g s.t. $g \in \arg\max_{u \in \mathcal{I}} f(u \mid \mathcal{G}^{[2]})/c(u)$.
  if $c(\mathcal{G}^{[2]}) + c(u) \leq b$ **then**
    Set $\mathcal{G}^{[2]} = \mathcal{G}^{[2]} \cup \{g\}$.
  end if
  Set $\mathcal{I} = \mathcal{I} \setminus \{g\}$.
end while
$\mathcal{G} = \mathcal{G}^{[1]}$, if $f(\mathcal{G}^{[1]}) \geq \mathcal{G}^{[2]}$ or $\mathcal{G} = \mathcal{G}^{[2]}$, otherwise.

The selection problem can be thought of a special case of the budgeted problem, as someone can assign unit costs to all measurements and set the budget to be $b = k$. Lee et al. [61] provide also a 1/5 bound for the case when the reward function f is not monotone.

■ 2.6.3 Sequential setting - Selection problem

As we showed in Sec. 2.6.1, in the batch setting we have a universal set of measurements and a maximum number of measurements that can be selected. This structure complies with the uniform matroidal structure that Nemhauser et al. [77] studied earlier. In a more general setting, there is no need to have a single, universal set of measurements neither all measurements need to be available at all times. One example is an HMM that represents temporal progress of a stochastic process. In this scenario, each group of measurements is linked to a latent (multidimensional) variable from the stochastic process and is available only at the time that the latent variable is generated. We assume we have T groups of measurements (observation sets) $\mathcal{V} = \{\mathcal{V}_1, \ldots, \mathcal{V}_T\}$, one for each latent variable X_t, and they are disjoint. The selection constraint in this case is to select up to k_t measurements from each set $\mathcal{V}_t$. This structure corresponds to a partition matroid. Fisher et al. [34] studied the following problem

$$\begin{aligned} \max \quad & f(\mathcal{S}) \\ \text{s.t.} \quad & \mathcal{S} \cap \mathcal{V}_t \in \mathscr{I}_t, \forall t \in \{1, \ldots, T\}, \end{aligned}$$

where $\mathcal{V} = \cup_{t=1}^{T} \mathcal{V}_t$ and $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset, \forall i \neq j$. Each matroid $\mathscr{M} = (\mathcal{V}_t, \mathscr{I}_t)$ can be thought of as a uniform matroid $\mathscr{M}_{k_t, N_t}$. They proposed a greedy approach described in Alg. 2.4, which comes with the following guarantees.

Algorithm 2.4 Sequential Selection Greedy Heuristic - Fisher et al.

Initialization
Set $\mathcal{G}_0 = \emptyset$.
Label the T observation sets arbitrarily as $\mathcal{V}_1, \ldots, \mathcal{V}_T$.

Iteration
for t in $1:T$ **do**
  Set $\mathcal{A}_t = \emptyset$.
  while $\mathcal{V}_t \neq \emptyset$ **do**
    Select g s.t. $g \in \arg\max_{u \in \mathcal{V}_t} f(u \mid \mathcal{G}_{t-1} \cup \mathcal{A}_t)$.
    if $\{g\} \cup \mathcal{G}_{t-1} \cup \mathcal{A}_t \in \mathscr{I}_t$ **then**
      Set $\mathcal{A}_t = \mathcal{A}_t \cup \{g\}$.
    end if
    Set $\mathcal{V}_t = \mathcal{V}_t \setminus \{g\}$.
  end while
  Set $\mathcal{G}_t = \mathcal{G}_{t-1} \cup \mathcal{A}_t$.
end for
$\mathcal{G} = \mathcal{G}_T$ is the greedy solution.

Theorem 2.6.2 (Performance bounds in the sequential setting [34])**.** *If the greedy method described in Alg. 2.4 is applied to problem* $\max_{\{\mathcal{S} \cap \mathcal{V}_t \in \mathscr{I}_t | t=1,\ldots,T\}} f(\mathcal{S})$ *under a non-decreasing submodular reward* f*, it achieves a value no worse than half of the optimal solution.*

Proof. The interested reader can refer to [34] for a proof of this theorem. ■

Williams et al. [102] showed that the same guarantee applies for a more general greedy algorithm. For this, they introduced the notion of a *walk*, which is the particular order that observation sets are visited to greedily select a measurement at each iteration. This order has to correspond to a feasible solution, in other words, it has to satisfy the constraints of every observation set. In more detail, they studied the problem

$$\begin{aligned} \max \quad & f(\mathcal{S}) \\ \text{s.t.} \quad & \mathcal{S} \cap \mathcal{V}_t \leq k_t, \forall t \in \{1, \ldots, T\}, \end{aligned}$$

where f is a non-decreasing submodular function. Since f is non-decreasing, all the selection constraints are met with equality, since it never hurts to obtain more measurements/items. A walk $\boldsymbol{w}$ is any order comprising of k_1 items from observation set $\mathcal{V}_1$, k_2 items from observation set $\mathcal{V}_2$, ..., k_T items from observation set $\mathcal{V}_T$. In other words, $\boldsymbol{w} = \{w_1, \ldots, w_M\}$, where $M = \sum_{t=1}^{T} k_t$ and $\sum_{j=1}^{M} \mathbb{1}(w_j = t) = k_t, \forall t = 1, \ldots, T$. Williams et al. [102] proposed the following greedy algorithm which comes with the following guarantees.

Algorithm 2.5 Sequential Selection Greedy Heuristic - Williams et al.

Initialization

Set $\mathcal{G}_0 = \emptyset$.

Define a visit walk $\boldsymbol{w} = \{w_1, \ldots, w_M\}$ such that $\sum_{j=1}^{M} \mathbb{1}(w_j = t) = k_t, \forall t$.

Iteration

for j in $1 : M$ **do**

Select g_j s.t. $g_j \in \arg\max_{u \in \mathcal{V}_{w_j}} f(u \mid \mathcal{G}_{j-1})$.

Set $\mathcal{G}_j = \mathcal{G}_{j-1} \cup \{g_j\}$.

Set $\mathcal{V}_{w_j} = \mathcal{V}_{w_j} \setminus \{g_j\}$.

end for

$\mathcal{G} = \mathcal{G}_M$ is the greedy solution.

Theorem 2.6.3 (Performance bounds in the sequential selection setting [102])**.** *If the greedy method described in Alg. 2.5 is applied to problem* $\max_{\{|\mathcal{S} \cap \mathcal{V}_t| \leq k_t, t=1,\ldots,T\}} f(\mathcal{S})$ *under a non-decreasing submodular reward* f*, it achieves a value no worse than half of the optimal solution.*

Proof. Let us denote the optimal and greedy solutions by $\mathcal{O} = \{o_1, \ldots, o_M\}, \mathcal{G} = \{g_1, \ldots, g_M\}$, respectively. The optimal and greedy solutions have the same length, that is, $M = \sum_{t=1}^{T} k_t$ and the same number of elements from each observation set, since f is a non-decreasing function. We choose a shuffling of the optimal elements $\{s_1, \ldots, s_M\}$ such that $o_{s_j} \in \mathcal{V}_{w_j}$. This is possible since there are k_t elements in the optimal set from observation set $\mathcal{V}_t, \forall t$. Obviously, $f(o_1, \ldots, o_M) = f(o_{s_1}, \ldots, o_{s_M}) = f(\mathcal{O})$. Therefore, we have

$$f(\mathcal{O}) \overset{(a)}{\leq} f(\mathcal{O} \cup \mathcal{G}) \overset{(b)}{\leq} f(\mathcal{G}) + \sum_{u \in \mathcal{O} \setminus \mathcal{G}} f(u \mid \mathcal{G}) \leq f(\mathcal{G}) + \sum_{u \in \mathcal{O}} f(u \mid \mathcal{G}) \overset{(c)}{=} f(\mathcal{G}) + \sum_{j=1}^{M} f(o_{s_j} \mid \mathcal{G})$$

$$\overset{(d)}{\leq} f(\mathcal{G}) + \sum_{j=1}^{M} f(o_{s_j} \mid \mathcal{G}_{j-1}) \overset{(e)}{\leq} f(\mathcal{G}) + \sum_{j=1}^{M} f(g_j \mid \mathcal{G}_{j-1}) = f(\mathcal{G}) + f(\mathcal{G}) = 2f(\mathcal{G}),$$

where (a) is due to monotonicity of f, (b) due to submodularity, (c) since elements $o_{s_1}, \ldots, o_{s_M}$ constitute the optimal set $\mathcal{O}$, (d) due to submodularity, and (e) due to the definition of the greedy heuristic. ■

The complexity of the above two algorithms is $\mathcal{O}(kTN)$ assuming $N_t = N, k_t = k, \forall t$, while the approximation ratio is $1/2$. Recently, there have been approaches which achieve the familiar $1 - 1/e$ optimal ratio that we saw in the batch setting [14, 32, 94]. However, they all result in complexities involving high order terms of the observation set size N and the number of observation sets T and hence make these methods impractical for more complex problems. For example, the algorithm of Filmus and Ward [32] runs in $\mathcal{O}(r^7 T^2 N^2)$, where r is the rank of the matroid. For a partition matroid that conforms

to the sequential setting, we have that $r = kT$, as discussed in Sec. 2.4. Therefore, the complexity boils down to $\mathcal{O}(k^7T^9N^2)$. Similarly, the method by Buchbinder et al. [14] gives a $(1-1/e-\epsilon)$ approximating ratio in $\mathcal{O}\left(r\sqrt{\frac{NT}{\epsilon^5}}\log\left(\frac{NT}{\epsilon}\right)+\frac{NT}{\epsilon^5}\log^2\left(NT\right)\right)$ time. For $r = kT$ and $\epsilon \sim \frac{1}{NT}$, the complexity simplifies to $\mathcal{O}\left(T^6N^6\log^2\left(NT\right)\right)$.

■ 2.6.4 Sequential setting - Budgeted problem

The T-knapsack problem is defined as follows. Given a T-dimensional budget vector $b = \begin{bmatrix} b_1 & \cdots & b_T \end{bmatrix}^T$ and a ground set of measurements $\mathcal{V}$, the subset $\mathcal{S}$ is sought that maximizes a specified reward f whose cost satisfies all budget constraints. In this setting, each measurement u has T different costs $c(t,u), \forall t \in \{1,\ldots,T\}$. So, the problem is collectively represented as

$$\mathcal{O} \in \underset{\{\mathcal{S}\subseteq\mathcal{V}|c(t,\mathcal{S})\leq b_t,\forall t\}}{\arg\max} f(\mathcal{S}).$$

We can easily convert the sequential budgeted problem to a T-knapsack one by choosing a partition of measurements in T observation sets $\mathcal{V}_1,\ldots,\mathcal{V}_T$. Then, if a measurement u belongs to observation set $\mathcal{V}_t$, we can just set the costs of this measurement for the remaining observation sets to zero, $c(s,u) = 0, \forall s \neq t$, since in the sequential setting a measurement has effect only on the budget of the observation set it belongs to. With this approach the sequential budgeted problem has been converted to a T-knapsack one.

Kulik et al. [59, 60] proposed the following algorithm which consists of two main phases and gives a $1-1/e$ approximation ratio with respect to the optimal solution. The first phase known as *profit enumeration* phase guesses (by enumeration) a constant number of elements of highest value in some optimal solution. Then, the algorithm proceeds to the randomized procedure taking the value residual problem with respect to the guessed subset. The *randomized procedure* uses randomized rounding in order to attain an integral solution from a fractional solution returned by the continuous greedy algorithm as described in [93]. However, simple randomized rounding may not guarantee a feasible solution, as some of the knapsack constraints may be violated. Lastly, the algorithm enumerates on the elements with high costs, something that enables the bounding of the variance of the cost in each dimension, and hence the event of discarding an infeasible solution occurs with small probability. Second, a fixing procedure is applied, in which a nearly feasible solution is converted to a feasible solution, with small harm to the objective function. The problem with this approach is that it is very difficult to implement and involves high order terms with respect to the number of budgets T and the total number of measurements NT, which makes it inappropriate for large problems.

■ 2.6.5 Remarks

We should note that in the batch setting all measurements need to be available at all times. On the contrary, in the sequential setting, only the observation set that the

greedy method dictates at every iteration needs to be available. In the selection problem, the method by Fisher et al. [34] considers measurements from a new observation set after all the required measurements have been greedily obtained from another observation set. Williams et al. [102] relax this requirement by allowing the consideration of measurements from different observation sets at every iteration as long as selection constraints are met at all times.

■ 2.6.6 Unconstrained Submodular Maximization (USM)

Lastly, for reasons that would become apparent in Sec. 3.4, we are interested in the problem of unconstrained maximization of a submodular non-monotone function f. Buchbinder et al. [13] proposed a really elegant algorithm that runs in just $\mathcal{O}(N)$ time, where $N = |\mathcal{V}|$ and provides a $1/2$ bound to the optimal solution on average. The problem they are solving is known as *Unconstrained Submodular Maximization (USM)* and is formulated as

$$\mathcal{O} \in \arg\max_{\mathcal{S}\subseteq\mathcal{V}} f(\mathcal{S}),$$

where f is a submodular but non-monotone function.

They use a doubly-greedy technique whose main idea is to accept a new measurement depending on how beneficial it would be if added to the existing greedy set $\mathcal{G}$ or discard a measurement if it is harmful to the existing exploration set $\mathcal{I}$. The algorithm chooses an arbitrary order of measurements $\mathcal{V} = \{u_1, \ldots, u_N\}$ and initializes the greedy and exploration set to $\mathcal{G}_0 = \emptyset$ and $\mathcal{I}_0 = \mathcal{V}$, respectively. Each measurement u_i is valued on how beneficial it is if added in the existing greedy set $\mathcal{G}_{i-1}$ or how harmful it is if retained in the existing exploration set $\mathcal{I}_{i-1}$ by the following metrics $f(u_i \mid \mathcal{G}_{i-1})$ and $-f(u_i \mid \mathcal{I}_{i-1} \setminus u_i)$, respectively. Since f is non-monotone, these two quantities can be negative, so we truncate them to zero if they become negative. We set $a_i = \max\{f(u_i \mid \mathcal{G}_{i-1}), 0\}$ and $b_i = \max\{-f(u_i \mid \mathcal{I}_{i-1} \setminus u_i), 0\}$ and decide whether to add measurement u_i in the existing greedy set $\mathcal{G}_{i-1}$ with probability $a_i/(a_i+b_i)$. Otherwise, measurement u_i is discarded from the exploration set $\mathcal{I}_{i-1}$ and as a consequence is guaranteed to not be part of the final greedy solution. A summarized description of their methodology is provided in Alg. 2.6. When a new measurement is accepted with probability $a_i/(a_i+b_i)$, greedy set $\mathcal{G}_i$ increases by one element, since it incorporates measurement u_i, while exploration set $\mathcal{I}_i$ stays the same. When measurement u_i is discarded with probability $b_i/(a_i + b_i)$, greedy set $\mathcal{G}_i$ stays the same, while exploration set $\mathcal{I}_i$ decreases by one element, since measurement u_i is removed. This protocol guarantees that a new greedy set $\mathcal{G}_i$ would be a superset of the previous one $\mathcal{G}_i \supseteq \mathcal{G}_{i-1}$, a new exploration set $\mathcal{I}_i$ would be a subset of the previous one $\mathcal{I}_i \subseteq \mathcal{I}_{i-1}$ and that a greedy set would always be a subset of an exploration set. Summarizing the above information, it holds that $\emptyset = \mathcal{G}_0 \subseteq \mathcal{G}_1 \subseteq \mathcal{G}_2 \subseteq \cdots \subseteq \mathcal{G}_N = \mathcal{I}_N \subseteq \cdots \subseteq \mathcal{I}_2 \subseteq \mathcal{I}_1 \subseteq \mathcal{I}_0 = \mathcal{V}$. It is noteworthy that in the final iteration N, greedy and exploration sets completely coincide $\mathcal{G}_N = \mathcal{I}_N$.

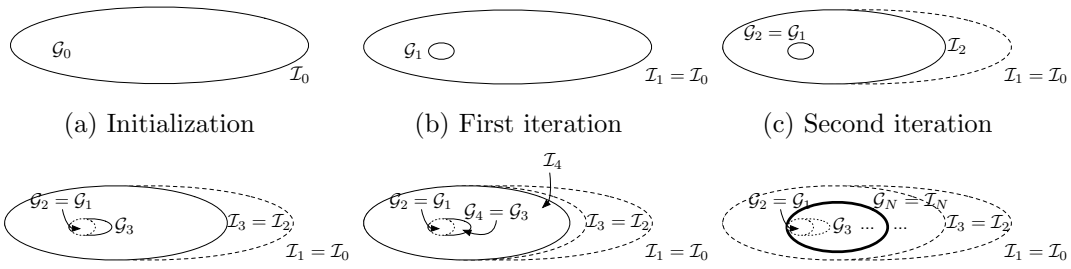
Theorem 2.6.4 (Performance bounds for USM)**.** *If the greedy method described in Alg. 2.6 is applied to problem* $\max_{\mathcal{S}\subseteq\mathcal{V}} f(\mathcal{S})$ *under a submodular reward* f*, it generates*

Algorithm 2.6 Randomized USM Greedy Heuristic – Buchbinder et al.

$\mathcal{G}_0 = \emptyset, \mathcal{I}_0 = \mathcal{V}$
for $i = 1, \ldots, N$ **do**
$a_i = \max\{f(u_i \mid \mathcal{G}_{i-1}), 0\}$, $b_i = \max\{-f(u_i \mid \mathcal{I}_{i-1} \setminus \{u_i\}), 0\}$.

$$(\mathcal{G}_i, \mathcal{I}_i) = \begin{cases} (\mathcal{G}_{i-1} \cup \{u_i\}, \mathcal{I}_{i-1}) & \text{, with probability } a_i/(a_i + b_i) \\ (\mathcal{G}_{i-1} \quad , \mathcal{I}_{i-1} \setminus \{u_i\}) & \text{, otherwise.} \end{cases}$$

end for



(a) Initialization (b) First iteration (c) Second iteration

(d) Third iteration (e) Fourth iteration (f) Last iteration

Figure 2.2: Randomized USM flow. In this particular example for exposition purposes, we assume that measurements u_1, u_3 are added, while measurements u_2, u_4 are discarded. (a) During initialization, greedy and exploration sets are initialized to $\mathcal{G}_0 = \emptyset$ and $\mathcal{I}_0 = \mathcal{V}$, respectively. (b) Measurement u_1 is accepted with probability $a_1/(a_1 + b_1)$ and so greedy set becomes $\mathcal{G}_1 = \{u_1\}$, while exploration set stays the same $\mathcal{I}_1 = \mathcal{V}$. (c) In the second iteration, measurement u_2 is discarded with probability $b_2/(a_2 + b_2)$ and so greedy set stays the same, while exploration set decreases by one element, $\mathcal{G}_2 = \mathcal{G}_1 = \{u_1\}$, $\mathcal{I}_2 = \mathcal{V} \setminus \{u_2\}$. (d) At iteration three, measurement u_3 is accepted and therefore $\mathcal{G}_3 = \{u_1, u_3\}, \mathcal{I}_3 = \mathcal{I}_2$. (e) In the fourth iteration, measurement u_4 is discarded and so $\mathcal{G}_4 = \mathcal{G}_3, \mathcal{I}_4 = \mathcal{V} \setminus \{u_2, u_4\}$. (f) After the end of the last iteration, greedy and exploration sets completely coincide, $\mathcal{G}_N = \mathcal{I}_N$.

a solution that on average is no worse than half of the optimal solution.

Proof. The interested reader can refer to [13] for a proof of this theorem. ■

■ 2.7 Graphical models

Probabilistic graphical models provide a framework for describing the statistical dependencies among random variables. The richness of representation allowed them to be extensively used in various disciplines such as statistics, applied mathematics, machine learning, information theory, statistical physics, computational biology, signal processing, and computer vision [95]. There two types of graphical models; directed and undirected. The two most common types of undirected models are Markov Random Fields (MRFs) and factor graphs. We will focus on MRFs as it can be shown that one type can be converted to another easily [11, 106]. Before proceeding to describing MRFs, we will give a brief overview of graph theory.

■ 2.7.1 Graph Theory

A graph $G = (V, \mathcal{E})$ is specified as a collection of vertices (or nodes), V, with a collection of edges, $\mathcal{E} \subset V \times V$. Unless stated otherwise, all edges are assumed to be undirected. We will refer to an (undirected) edge connecting vertex i to vertex j as (i, j). The *neighborhood* of vertex i is the set $N(i) = \{j \in V | (i, j) \in \mathcal{E}\}$. In other words, it is the set of all vertices that vertex i is connected directly with an edge. The *degree* of vertex i is the number of its neighbors $|N(i)|$. A subgraph of G is a graph $G_s = (V_s, \mathcal{E}_s)$, where $V_s \subset V$, and $\mathcal{E}_s \subset V_s \times V_s$. In other words, G is a supergraph of G_s and G_s is embedded in G. A *clique* C of graph G is a fully connected subgraph of G, i.e., $C = (V_s, \mathcal{E}_s)$ with every pair of vertices connected: $i, j \in V_s \Rightarrow (i, j) \in \mathcal{E}_s$. A *maximal clique* is a clique that is not contained within another clique. In other words, no other vertex can be added while still retaining full connectivity. A *walk* $\boldsymbol{w} = (w_0, w_1, \ldots, w_\ell)$ of length ℓ is a sequence of vertices $w_0, \ldots, w_\ell \in V$, where each pair of consequent vertices is connected with an edge. A *path* is a walk with distinct vertices and edges. A graph is *connected* if there is a path between any two vertices. The *diameter* of a graph, $\text{diam}(G)$, is the maximum distance between any pair of vertices, where distance is defined as the length of the shortest path between the pair of vertices. A *cycle* is a connected graph, where each vertex has exactly two neighbors. A *tree* is a connected graph with no cycles. A graph is called *chordal* if every cycle of the graph of length 4 or more contains a chord (an edge between two non-adjacent vertices of the cycle). The *treewidth* of a graph G is the smallest size of the largest clique minus one over all chordal graphs containing G. The treewidth of a graph serves as a lower bound on the complexity of exact inference [68].

■ 2.7.2 Markov Random Fields

In Markov Random Fields, the joint distribution can be expressed as a product of factors (or *potentials*), where each factor corresponds to a clique in the graph. As a reminder, cliques are subsets of nodes that are fully connected. In more detail, assume that we have N random variables $\mathsf{X}_1, \mathsf{X}_2, \ldots, \mathsf{X}_N$ corresponding to N nodes in graph G and $\mathscr{C}$ is a collection of all cliques in G. Also, each random variable X_i is either discrete or continuous, $\mathsf{X}_i \in \mathcal{X}$. Then, the joint distribution can be expressed as

$$p(x) = \frac{1}{Z} \prod_{c \in \mathscr{C}} \psi_c(x_c), \tag{2.69}$$

where Z is a normalizing constant (or partition function), c is a clique comprised of variables X_c and $\psi_c(\cdot)$ is the potential for clique c. The partition function is defined as $Z = \sum_x \prod_{c \in \mathscr{C}} \psi_c(x_c)$.

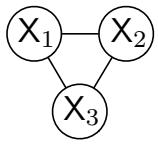
Each individual node forms a clique with itself by default, called *singleton clique*, while every pair of nodes connected with an edge forms a *pairwise clique*. It is customary to assign a potential to each node and edge of the graph. In this case, expression (2.69) simplifies to

$$p(x) = \frac{1}{Z} \prod_{i=1}^{N} \varphi_i(x_i) \prod_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j), \tag{2.70}$$

where $\varphi_i(\cdot)$ is the node potential and $\psi_{ij}(\cdot, \cdot)$ the edge potential.

Even though representation through MRFs might be ambiguous because potentials might not correspond to maximal cliques, it is very convenient in providing the conditional independence properties of the model and hence enabling the usage of efficient algorithms for inference purposes. To illustrate the above, we cite the following example:

Example 2.7.1. *Suppose we have the model represented by the following graph*



This model can receive multiple equivalent representations. One factorization might include only maximal cliques

$$p(x) \propto \psi_{123}(x_1, x_2, x_3),$$

or it can be expressed in terms of node (φ) and edge (ψ) potentials as well

$$p(x) \propto \varphi_1(x_1)\varphi_2(x_2)\varphi_3(x_3)\psi_{12}(x_1, x_2)\psi_{23}(x_2, x_3)\psi_{13}(x_1, x_3).$$

MRFs that involve only node and edge potentials are referred to as pairwise MRFs. In this thesis, we will focus on pairwise MRFs as it can be shown that any MRF can be converted to such by state space augmentation [97, 106]. The mapping from the conditional independence properties to the structure of the graph comes from the concept of graph separation. Let the set of nodes V be partitioned in the disjoint sets (of nodes) A, B, C: $V = A \cup B \cup C$. We say that C separates A from B, if any path from any vertex in A to any vertex in B goes through some vertex in C. Set C is referred to as *vertex cutset*. A distribution $p(x)$ is called Markov with respect to undirected graph G, if for any such partition, $\mathsf{X}_A \perp\!\!\!\perp \mathsf{X}_B \mid \mathsf{X}_C$.

Theorem 2.7.1 (Hammersley-Clifford [95])**.** *If the joint distribution p of $\mathsf{X}_1, \ldots, \mathsf{X}_N$ factorizes over graph G, then $\mathsf{X}_1, \ldots, \mathsf{X}_N$ are Markov with respect to G, meaning that the graph separation implies conditional independence properties. Conversely, if $\mathsf{X}_1, \ldots, \mathsf{X}_N$ are Markov with respect to G and $p(x) > 0, \forall x$, then p factorizes over the graph.*

Proof. An elegant proof of this theorem can be found in p. 11 of [95]. ■

Tree-structured MRFs

When MRF is a tree, there are methods such as the belief propagation algorithm that can provide very efficiently exact solutions to inference problems. Tree-structured MRFs can either be used directly to model a problem or as an approximation (embedded) structure for a more general type of graph [68]. In tree-structured MRFs, potentials can be constructed in such a way that are directly linked to node and edge marginals:

$$p(x) = \prod_{i=1}^{N} p_i(x_i) \prod_{(i,j)\in\mathcal{E}} \frac{p_{ij}(x_i, x_j)}{p_i(x_i)p_j(x_j)}, \tag{2.71}$$

where $\varphi_i(x_i) = p_i(x_i)$ and $\psi_{ij}(x_i, x_j) = p_{ij}(x_i, x_j)/(p_i(x_i)p_j(x_j))$.

Another equivalent representation is by picking an arbitrary node as the root and continue by considering the conditional probabilities of parent-child pairs as edge potentials. That is, if we arbitrarily let node 1 to be the root, then we can express the joint distribution as

$$p(x) = p_1(x_1) \prod_{(i,j)\in\mathcal{E}, i>j} p(x_i \mid x_j), \tag{2.72}$$

where $i > j$ indicates that i is a child of j. Here, $\varphi_1(x_1) = p_1(x_1), \varphi_i(x_i) = 1, \forall i \neq 1$, $\psi_{ij}(x_i, x_j) = p(x_i \mid x_j)$.

■ 2.8 Exponential Families

Exponential families are families of distributions where inference can be derived analytically. In more detail, exponential family is the only family with finite-sized sufficient statistics, meaning that data can be summarized in a statistic without any loss of information. Exponential families are also closed under conjugacy. That is, if a prior is

used in an appropriate form, the posterior distribution will also belong in the exponential family. Lastly, it is very convenient for inference as the posterior parameters after the incorporation of measurements can be obtained in closed form. Many known probability distributions can be represented as members of an exponential family. We give below the definition of an exponential family.

Definition 2.8.1 (Exponential Family [73])**.** *A parameterized family of distributions $p(\cdot;\theta)$ is an exponential family with natural parameter $\lambda(\theta)$, natural statistic $\phi(x)$ and base distribution $q(x)$ if each member of the family is represented as*

$$p(x;\theta) = q(x)\exp(\langle\lambda(\theta),\phi(x)\rangle - A(\theta)) = q(x)\exp(\lambda(\theta)^T\phi(x) - A(\theta)), \tag{2.73}$$

where $\langle\cdot,\cdot\rangle$ is the dot product of two vectors and $A(\theta)$ is called the log-partition function.

The log-partition function is defined as

$$A(\theta) = \log\int q(x)\exp(\lambda(\theta)^T\phi(x))\,\mathrm{d}x. \tag{2.74}$$

Usually, the base distribution $q(x)$ is a scaling constant. Several distributions (such as the Bernoulli, Exponential, Laplace, Gaussian, Gamma, Dirichlet) that can be expressed in the exponential family form, have a constant base distribution. A family where $\lambda(\theta) = \theta$ is called a *canonical family*. If, furthermore, $\phi(x) = x$, the family is called a *natural family*. Lastly, an exponential family is *regular* if the support of the variable whose distribution belongs in the exponential family does not depend on the parameter.

We will focus on canonical families, $\lambda(\theta) = \theta$, as we can always convert to a canonical family with a transformation of parameters. We will give below of a few examples of exponential families for further clarification.

Example 2.8.1 (Bernoulli)**.** *The Bernoulli distribution* $\mathrm{Ber}(\cdot;p)$ *takes the form:*

$$\mathrm{Ber}(x;p) = p^x(1-p)^{1-x}.$$

It can be written in exponential form as

$$\mathrm{Ber}(x;p) = \exp(\log(p^x(1-p)^{1-x})) = \exp(x\log p + (1-x)\log(1-p)) = \exp(x\log\frac{p}{1-p}).$$

Therefore, $\theta = \log\frac{p}{1-p}$ and $\phi(x) = x$. The log-partition function is evaluated as

$$A(\theta) = \log\sum_x \exp(\theta x) = \log(1+\exp(\theta)).$$

Example 2.8.2 (Multivariate Gaussian [73])**.** *The multivariate Gaussian is expressed in information form as*

$$\begin{aligned} p(x;h,J) &\propto \exp(h^Tx - \frac{1}{2}x^TJx) = \exp(h^Tx)\exp(x^T(-\frac{1}{2}J)x) = \exp(h^Tx)\exp(\mathrm{trace}(-\frac{1}{2}Jxx^T) \\ &\propto \exp(\langle h,x\rangle + \langle\!\langle -\frac{1}{2}J, xx^T\rangle\!\rangle), \end{aligned} \tag{2.75}$$

where the dot product of two matrices $\langle\!\langle \cdot, \cdot \rangle\!\rangle$ *is defined as the trace of the product of these two matrices. From Eq. (2.75),* $\theta = \begin{bmatrix} \theta_1 \\ \Theta_2 \end{bmatrix} = \begin{bmatrix} h \\ -\frac{1}{2}J \end{bmatrix}$ *and* $\phi(x) = \begin{bmatrix} x \\ xx^T \end{bmatrix}$*. With a little bit of algebra, we can show that the log-partition function takes the form*

$$A(\theta) = \log \int_x \exp(\langle h, x \rangle + \langle\!\langle -\frac{1}{2}J, xx^T \rangle\!\rangle) \, \mathrm{d}x = -\frac{1}{4}\theta_1^T \Theta_2^{-1} \theta_1 - \frac{1}{2}\log|-2\Theta_2| = \frac{1}{2}h^T J^{-1} h - \frac{1}{2}\log|J|.$$

Example 2.8.3 (Ising model [96])**.** *The Ising model is a widely used graphical model, which represents a grid graph, where each hidden variable takes two values* $\{-1, +1\}$*. Each node is connected to four nodes (one on the left, one on the right, one above and one below) unless the node is on the boundaries of the graph. In that case, it connects to either 2 or 3 nodes. The joint distribution of variables is represented as*

$$p(x; \theta) = \exp\left(\sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in \mathcal{E}} \theta_{st} x_s x_t - A(\theta) \right),$$

where θ_s *is a potential for node* s *and* θ_{st} *represents the strength of edge* (s, t)*. The log-partition function is computed as*

$$A(\theta) = \log \sum_{x \in \{0,1\}^{|V|}} \exp\left(\sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in \mathcal{E}} \theta_{st} x_s x_t \right).$$

Interestingly, not all distributions are regular exponential families. For example, the uniform distribution $X \sim \mathrm{Unif}(a, b)$ is not a regular exponential family, since the support of X depends on the parameters a and b.

The statistic $\phi(x)$ holds particular significance as under some conditions it can summarize the variable x without any loss of information. In that case, the statistic $\phi(x)$ is called a *sufficient statistic*, since inference in the distribution $p_X(x; \theta)$ is equivalent to inference in the distribution $p_\phi(\phi(x); \theta)$. The implication of this is that inference can be achieved in a much more efficient way (in terms of data storage) without any effect on the exactness of the solutions.

■ 2.8.1 Log-partition function

Log-partition function is important because derivatives of it can be used to generate cumulants of the sufficient statistics. That is the reason that log-partition function $A(\theta)$ is sometimes called *cumulant function* as well [73]. As we shall see later in Sec. 3.1, log-partition function also appears in common information measures such as the entropy. We will show below that the first and second derivatives of the log-partition function equal the mean $\mathbb{E}[\phi(x)]$ and covariance $\mathrm{cov}[\phi(x)]$ of the sufficient statistic $\phi(x)$, respectively.

Table 2.1: Well-known distributions represented as exponential families

Family	Alphabet $\mathcal{X}$	Parameter(s)	Base dist $q(x)$	Natural parameter(s) θ	Sufficient statistics $\phi(x)$	Log-partition function $A(\theta)$
Bernoulli	$\{0,1\}$	p	1	$\log \frac{p}{1-p}$	x	$\log(1+\exp(\theta))$
Multinoulli	$\{1,\dots,k\}$	$p_1,\dots,p_k$	$\frac{n!}{\prod_{i=1}^k x_i!}$	$\begin{bmatrix} \log \frac{p_1}{p_k} \\ \vdots \\ \log \frac{p_{k-1}}{p_k} \\ 0 \end{bmatrix}$	$\begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix}$	$n \log(1+\sum_{i=1}^{k-1} \exp(\theta_i))$
Poisson	$\{0,1,2,\dots\}$	λ	$\frac{1}{x!}$	$\log \lambda$	x	$\exp(\theta)$
Exponential	$[0,\infty)$	λ	1	$-\lambda$	x	$-\log(-\theta)$
Beta	$(0,1)$	α,β	1	$\begin{bmatrix} \alpha-1 \\ \beta-1 \end{bmatrix}$	$\begin{bmatrix} \log x \\ \log(1-x) \end{bmatrix}$	$\log\Gamma(\theta_1)+\log\Gamma(\theta_2)-\log\Gamma(\theta_1+\theta_2)$
Dirichlet	$(0,1)^k$	$\begin{bmatrix} \alpha_1-1 \\ \vdots \\ \alpha_k-1 \end{bmatrix}$	1	$\begin{bmatrix} \log x_1 \\ \vdots \\ \log x_k \end{bmatrix}$	$\begin{bmatrix} \log x_1 \\ \vdots \\ \log x_k \end{bmatrix}$	$\sum_{i=1}^k \log\Gamma(\theta_i+1)-\log\Gamma(\sum_{i=1}^k \theta_i+k)$
Gamma	$(0,\infty)$	α,β	1	$\begin{bmatrix} \alpha-1 \\ -\beta \end{bmatrix}$	$\begin{bmatrix} \log x \\ x \end{bmatrix}$	$\log\Gamma(\theta_1+1)-(\theta_1+1)\log(-\theta_2)$
Gaussian	$\mathbb{R}$	μ,σ^2	$\frac{1}{\sqrt{2\pi}}$	$\begin{bmatrix} \frac{\mu}{\sigma^2} \\ -\frac{1}{\sigma^2} \end{bmatrix}$	$\begin{bmatrix} x \\ x^2 \end{bmatrix}$	$-\frac{\theta_1^2}{4\theta_2}-\frac{1}{2}\log(-2\theta_2)$
Multivariate Gaussian	$\mathbb{R}^d$	h, J	$(2\pi)^{-d/2}$	$\begin{bmatrix} h \\ -\frac{1}{2}J \end{bmatrix}$	$\begin{bmatrix} x \\ xx^T \end{bmatrix}$	$-\frac{1}{4}\theta_1^T\Theta_2^{-1}\theta_1-\frac{1}{2}\log\lvert-2\Theta_2\rvert$

If we take the gradient of $A(\theta)$ with respect to θ, we have

$$\begin{aligned}\nabla_\theta A &= \nabla_\theta \left(\log \int q(x) \exp(\theta^T \phi(x)) \, \mathrm{d}x \right) = \frac{\int \phi(x) q(x) \exp(\theta^T \phi(x)) \, \mathrm{d}x}{\int q(x) \exp(\theta^T \phi(x)) \, \mathrm{d}x} \\ &\overset{(2.74)}{=} \frac{\int \phi(x) q(x) \exp(\theta^T \phi(x)) \, \mathrm{d}x}{\exp(A(\theta))} = \int \phi(x) q(x) \exp(\theta^T \phi(x) - A(\theta)) \, \mathrm{d}x \\ &= \mathbb{E}[\phi(x)]. \end{aligned} \tag{2.76}$$

Now, if we take the second derivative with respect to θ, we obtain

$$\begin{aligned}\nabla^2_{\theta\theta} A &= \nabla_\theta (\nabla_\theta A) = \nabla_\theta \left(\int \phi(x) q(x) \exp(\theta^T \phi(x) - A(\theta)) \, \mathrm{d}x \right) \\ &= \int \phi(x) (\phi(x) - \nabla A(\theta))^T q(x) \exp(\theta^T \phi(x) - A(\theta)) \, \mathrm{d}x \\ &= \int \phi(x) \phi(x)^T q(x) \exp(\theta^T \phi(x) - A(\theta)) \, \mathrm{d}x - \left(\int \phi(x) q(x) \exp(\theta^T \phi(x) - A(\theta)) \, \mathrm{d}x \right) \nabla A(\theta) \\ &\overset{(2.76)}{=} \mathbb{E}[\phi(x)\phi(x)^T] - \mathbb{E}[\phi(x)] \mathbb{E}[\phi(x)]^T = \mathrm{cov}(\phi(x)). \end{aligned} \tag{2.77}$$

Convexity of log-partition function

Since the second derivative $\nabla^2_{\theta\theta} A$ equals $\mathrm{cov}(\phi(x))$ and we know that covariance is a positive semidefinite matrix, it follows that the log-partition function $A(\theta)$ is convex with respect to θ.

We present below an example of the properties of first and second derivative of the log-partition function A for the multivariate Gaussian case.

Example 2.8.4 (Multivariate Gaussian)**.** *As we showed in Ex. 2.8.2, the sufficient statistic $\phi(x)$ is defined as $\phi(x) = \begin{bmatrix} x \\ xx^T \end{bmatrix}$. We know that $\mathbb{E}[x] = \mu$ and $\mathbb{E}[xx^T] = \Sigma + \mu\mu^T$. We will confirm below that taking the first derivative $\nabla_\theta A(\theta)$ will lead us to the familiar statistics of the Gaussian distribution. From Ex. 2.8.2, we have that*

$$A(\theta) = -\frac{1}{4} \theta_1^T \Theta_2^{-1} \theta_1 - \frac{1}{2} \log |-2\Theta_2|,$$

where $\theta_1 = h$ and $\Theta_2 = -\frac{1}{2} J$.

If we take the derivative with respect to θ_1, we have

$$\begin{aligned}\nabla_{\theta_1} A(\theta) &= \nabla_{\theta_1} \left(-\frac{1}{4} \theta_1^T \Theta_2^{-1} \theta_1 - \frac{1}{2} \log |-2\Theta_2| \right) = -\frac{1}{2} \Theta_2^{-1} \theta_1 = -\frac{1}{2}(-2J^{-1})h = J^{-1}h \\ &= \mu = \mathbb{E}[x]. \end{aligned}$$

Similarly,

$$\nabla_{\Theta_2} A(\theta) = \nabla_{\Theta_2} \left(-\frac{1}{4} \theta_1^T \Theta_2^{-1} \theta_1 - \frac{1}{2} \log |-2\Theta_2| \right) \overset{[83]}{=} -\frac{1}{4}(-\Theta_2^{-T} \theta_1 \theta_1^T \Theta_2^{-T}) - \frac{1}{2} \Theta_2^{-T}. \tag{2.78}$$

Since $\Theta_2 = -\frac{1}{2}J$, *it follows that* Θ_2 *is symmetric negative semidefinite and so* $\Theta_2^{-T} = \Theta_2^{-1}$. *In addition,* $\theta_1 = h$, *so Eq.* (2.78) *becomes*

$$\nabla_{\Theta_2} A(\theta) = -\frac{1}{4}(-\Theta_2^{-1}\theta_1\theta_1^T\Theta_2^{-1}) - \frac{1}{2}\Theta_2^{-1} = J^{-1}hh^TJ^{-1} + J^{-1} = \mu\mu^T + \Sigma = \mathbb{E}[xx^T].$$

The above derivations agree with our knowledge of the Gaussian moment functions.

■ 2.9 Gaussian Graphical models

A Gaussian graphical model is a graphical model representing jointly Gaussian variables. If we denote by $\boldsymbol{X}$ the random vector $\boldsymbol{X} = (X_1, \ldots, X_N)$, then it follows the multivariate Gaussian distribution, $\boldsymbol{X} \sim \mathcal{N}(x; \mu, \Sigma)$

$$\mathcal{N}(x; \mu, \Sigma) = (2\pi)^{-N/2}|\Sigma|^{-1/2} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right). \tag{2.79}$$

Here, μ is the mean and Σ the covariance of $\boldsymbol{X}$, $\mathbb{E}[\boldsymbol{X}] = \mu, \text{cov}(\boldsymbol{X}) = \Sigma$. Since, $\text{cov}(\boldsymbol{X}) = \mathbb{E}[(\boldsymbol{X}-\mu)(\boldsymbol{X}-\mu)^T]$, we can show easily that Σ is positive semidefinite, $\Sigma \succeq 0$. Eq. (2.79) is known as the *standard* form.

It is often more convenient to work with the *canonical* (*information*) form, which is defined as

$$\mathcal{N}^{-1}(x; h, J) = (2\pi)^{-N/2}|J|^{1/2} \exp\left(-\frac{1}{2}h^TJ^{-1}h\right) \exp\left(h^Tx - \frac{1}{2}x^TJx\right) \propto \exp\left(h^Tx - \frac{1}{2}x^TJx\right), \tag{2.80}$$

where h is the potential vector and J the precision (or else known as information) matrix. It also holds that $J \succeq 0$. Standard and canonical form parameters are linked as $\Sigma = J^{-1}, \mu = J^{-1}h$.[2] If there are N variables and each has dimension d, then h is a $Nd \times 1$ vector, while J an $Nd \times Nd$ matrix. The $d \times 1$ part of the potential vector that is related to node i is denoted by h_i, while J_{ij} is the $d \times d$ block of J that is related to variables $\boldsymbol{X}_i, \boldsymbol{X}_j$. There is an edge (i, j) between two jointly Gaussian variables $\boldsymbol{X}_i, \boldsymbol{X}_j$ if and only if $J_{ij} = 0$. The sufficient statistics in a Gaussian distribution are the mean and covariance (in standard form) or the potential vector and precision (in canonical form). In other words, knowledge of these parameters fully characterize the distribution.

It turns out that marginalization is easy in standard form, while conditioning is easy in canonical form. In more detail, if we have two disjoint sets of variables A, B such that $A \cup B = \{1, \ldots, N\}$, with $|A| = N_1, |B| = N_2$ and the joint distribution $\boldsymbol{X} = (\boldsymbol{X}_A, \boldsymbol{X}_B)$ is characterized by the parameters $\mu = \begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix}$, $\Sigma = \begin{bmatrix} \Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB} \end{bmatrix}$, with $\Sigma_{BA} = \Sigma_{AB}^T$, then

$$\mathbb{E}[\boldsymbol{X}_A] = \mu_A^m = \mu_A \tag{2.81}$$

$$\text{cov}(\boldsymbol{X}_A) = \Sigma_A^m = \Sigma_{AA}. \tag{2.82}$$

[2] As a reminder, the standard (moment) form is represented as $\mathcal{N}(x; \mu, \Sigma)$, while the canonical form as $\mathcal{N}^{-1}(x; h, J)$.

Similarly, if $X = (X_A, X_B)$ is given in canonical form as $h = \begin{bmatrix} h_A \\ h_B \end{bmatrix}$, $J = \begin{bmatrix} J_{AA} & J_{AB} \\ J_{BA} & J_{BB} \end{bmatrix}$, with $J_{BA} = J_{AB}^T$, and we are interested in characterizing the conditional distribution $X_A \mid X_B = x_B \sim \mathcal{N}^{-1}(x_A; h_{A|B}, J_{A|B})$, we have

$$h_{A|B} = h_A - J_{AB}x_B \tag{2.83}$$

$$J_{A|B} = J_{AA}. \tag{2.84}$$

Conversely, conditioning is hard in the standard form. More specifically, if we are in standard form and are interested in $\mathbb{E}[X_A \mid X_B = x_B], \mathrm{cov}(X_A \mid X_B = x_B)$, they are given by

$$\mathbb{E}[X_A \mid X_B = x_B] = \mu_{A|B} = \mu_A \quad + \Sigma_{AB}\Sigma_{BB}^{-1}(x_B - \mu_B) \tag{2.85}$$

$$\mathrm{cov}(X_A \mid X_B = x_B) = \Sigma_{A|B} = \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}. \tag{2.86}$$

This operation requires $\mathcal{O}(\max(N_1, N_2)^3)$ steps. Similarly, if X_A, X_B are expressed in canonical form, marginalization is hard. For instance, if we want to recover the distribution of $X_A \sim \mathcal{N}^{-1}(x_A; h_A^m, J_A^m)$ in canonical form, this would be

$$h_A^m = h_A \quad + J_{AB}J_{BB}^{-1}h_B \tag{2.87}$$

$$J_A^m = J_{AA} - J_{AB}J_{BB}^{-1}J_{BA}, \tag{2.88}$$

which leads to an $\mathcal{O}(\max(N_1, N_2)^3)$ operation.

It turns out that the content of precision matrix uniquely determines the graph structure. To elaborate further, it is useful to remind that zero correlation between jointly Gaussian variables implies independence and vice versa. Let us define the *partial correlation* ρ_{ij} between variables X_i, X_j given the rest of variables $X_{V\setminus\{i,j\}} = x_{V\setminus\{i,j\}}$ as

$$\rho_{ij} = \frac{\mathrm{cov}(X_i, X_j \mid X_{V\setminus\{i,j\}} = x_{V\setminus\{i,j\}})}{\mathrm{var}(X_i \mid X_{V\setminus\{i,j\}} = x_{V\setminus\{i,j\}})\mathrm{var}(X_j \mid X_{V\setminus\{i,j\}} = x_{V\setminus\{i,j\}})}. \tag{2.89}$$

The precision for the bivariate distribution of X_i, X_j given $X_{V\setminus\{i,j\}} = x_{V\setminus\{i,j\}}$ is

$$\mathrm{cov}(X_i, X_j \mid x_{V\setminus\{i,j\}})^{-1} \triangleq J_{ij|V\setminus\{i,j\}} = \begin{bmatrix} J_{ii} & J_{ij} \\ J_{ji} & J_{jj} \end{bmatrix},$$

which implies that

$$\mathrm{cov}\left(\begin{bmatrix} X_i \\ X_j \end{bmatrix} \mid x_{V\setminus\{i,j\}}\right) \triangleq \Sigma_{ij|V\setminus\{i,j\}} = \frac{1}{J_{ii}J_{jj} - J_{ij}J_{ji}} \begin{bmatrix} J_{jj} & -J_{ij} \\ -J_{ji} & J_{ii} \end{bmatrix}.$$

Therefore, $\mathrm{cov}(X_i, X_j \mid x_{V\setminus\{i,j\}}) = -J_{ij}/(J_{ii}J_{jj} - J_{ij}J_{ji})$, $\mathrm{var}(X_i \mid x_{V\setminus\{i,j\}}) = J_{jj}/(J_{ii}J_{jj} - J_{ij}J_{ji})$, $\mathrm{var}(X_j \mid x_{V\setminus\{i,j\}}) = J_{ii}/(J_{ii}J_{jj} - J_{ij}J_{ji})$. So, Eq. (2.89) becomes

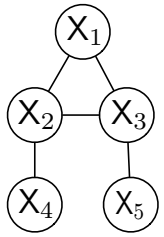
$$\rho_{ij} = \frac{-J_{ij}}{J_{ii}J_{jj}}. \tag{2.90}$$

If X_i, X_j are conditionally independent given the rest of variables, $X_i \perp\!\!\!\perp X_j \mid X_{V\setminus\{i,j\}}$, this implies that the partial covariance, $\mathrm{cov}(X_i, X_j \mid x_{V\setminus\{i,j\}})$ and partial correlation, ρ_{ij} are zero. From Eq. (2.90), we have that

$$X_i \perp\!\!\!\perp X_j \mid X_{V\setminus\{i,j\}} \Rightarrow \rho_{ij} = 0 \Rightarrow J_{ij} = 0. \tag{2.91}$$

Now, if the joint distribution of $X_1, \ldots, X_N$ is Markov with respect to graph G, this means that conditional independence between two variables given the remaining implies absence of an edge between them. Therefore, for a pair of nodes i, j such that $(i, j) \notin \mathcal{E}$, we have that $J_{ij} = 0$. In other words, $J_{ij} = 0$ when nodes i, j have no direct link, a property indicating conditional independence (given the rest of the graph),

Example 2.9.1. *Consider the Gaussian graphical model as depicted by*



The edge set $\mathcal{E}$ for this model is $\mathcal{E} = \{(1,2),(1,3),(2,3),(2,4),(3,5)\}$. The precision matrix J takes the form:

$$J = \begin{bmatrix} J_{11} & J_{12} & J_{13} & 0 & 0 \\ J_{21} & J_{22} & J_{23} & J_{24} & 0 \\ J_{31} & J_{32} & J_{33} & 0 & J_{35} \\ 0 & J_{42} & 0 & J_{44} & 0 \\ 0 & 0 & J_{53} & 0 & J_{55} \end{bmatrix}.$$

As we see, for every $i, j \in \{1, \ldots, 5\}$ such that $(i, j) \notin \mathcal{E}$, we have $J_{ij} = 0$.

■ 2.9.1 Entropy

The entropy of a Gaussian random vector $X = (X_1, \ldots, X_N)$ takes the form

$$H(X) = \frac{N}{2}(1 + \log(2\pi)) + \frac{1}{2}\log|\Sigma|, \tag{2.92}$$

where $X \sim \mathcal{N}(x; \mu, \Sigma)$. As we see entropy in the Gaussian case depends on the determinant of the covariance. We can express the entropy in terms of the precision matrix J as well

$$H(X) = \frac{N}{2}(1 + \log(2\pi)) - \frac{1}{2}\log|J|.$$

If we were to determine the pointwise conditional entropy, $H(X \mid Y = y)$, it would be

$$H(X \mid Y = y) = \frac{N}{2}(1 + \log(2\pi)) + \frac{1}{2}\log|\Sigma_{X|y}|, \tag{2.93}$$

where $\Sigma_{X|y} = \text{cov}(X \mid Y = y)$. Remarkably enough, as observed from Eqs. (2.84), (2.86), conditional covariance or conditional precision do not depend on the specific value of the conditioning variable. In other words, $\Sigma_{X|Y} = \Sigma_{X|y}, \forall y$, where $\Sigma_{X|Y}$ is the conditional covariance of X given Y that is the same for all values $Y = y$. This has important implications for planning, as it renders open-loop planning equivalent to closed loop (active planning as new observations are incorporated), when the problem is modeled with a Gaussian graphical model and the entropy is used as the reward function.

Since, the conditional covariance is the same for every value of the conditioning variable, this further implies that the conditional entropy is the same as the pointwise conditional entropy:

$$\begin{aligned} H(X \mid Y) &= \mathbb{E}_Y[H(X \mid Y = y)] = \mathbb{E}_Y\left[\frac{N}{2}(1 + \log(2\pi)) + \frac{1}{2}\log|\Sigma_{X|Y}|\right] \\ &= \frac{N}{2}(1 + \log(2\pi)) + \frac{1}{2}\log|\Sigma_{X|Y}|. \end{aligned} \tag{2.94}$$

■ 2.9.2 Mutual Information

Similarly, MI takes a simple form in a Gaussian model. The information gain of X from an observation Y is defined as

$$I(X; Y) = -\frac{1}{2}\log\frac{|\Sigma|}{|\Sigma_{X|Y}|} = -\frac{1}{2}\log\frac{|J_{X|Y}|}{|J|}.$$

As in the entropy case, MI does not depend on the exact value of observations for Gaussian models. This convenient property makes also MI an excellent candidate as a reward function for experiment designation.

■ 2.10 Inference in Graphical Models

Inference in graphical models often refers to the problem of finding the marginal distribution of the hidden nodes given observations, $p_i(x_i \mid y_1, \ldots, y_N)$, or retrieving the most likely hidden sequence (MAP assignment), $x^* \in \arg\max_x p(x \mid y_1, \ldots, y_N)$, where $x = (x_1, \ldots, x_N)$. It is often assumed that observations are *local*, or in other words, conditionally independent given the hidden variables. We will hold this assumption throughout this thesis as well. In other words, if we assume that each observation Y_i is obtained from hidden variable X_i, then the data likelihood is factorized as

$$p(y_1, \ldots, y_N \mid x_1, \ldots, x_N) = \prod_{i=1}^{N} p(y_i \mid x_i).$$

If in addition there is a prior on $X = (X_1, \ldots, X_N)$, $p(x)$, then the posterior distribution of X given the observations $Y = (Y_1, \ldots, Y_N)$ is

$$p(x \mid y) \propto p(x)p(y \mid x),$$

where $X = x, Y = y$.

Assume we have an MRF with the following node and edge potentials, $\varphi_i(x_i)$, $\psi_{ij}(x_i, x_j)$, $\xi_i(x_i, y_i)$, where $\varphi_i(x_i)$ is the node potential for hidden variable X_i, $\psi_{ij}(x_i, x_j)$ is the edge potential for any pair of hidden variable neighbors ($(i, j) \in \mathcal{E}$) and $\xi_i(x_i, y_i)$ is the edge potential between a hidden variable and its observation. W.l.o.g., we can assume that $\xi_i(x_i, y_i) = p(y_i \mid x_i)$. Then, the joint distribution $p(x, y)$ is

$$\begin{aligned} p(x, y) = p(y \mid x)p(x) &\propto p(x) \prod_{i=1}^{N} p(y_i \mid x_i) = \prod_{i=1}^{N} \varphi_i(x_i) \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j) \prod_{i=1}^{N} \xi_i(x_i, y_i) \\ &\propto \prod_{i=1}^{N} (\varphi_i(x_i)\xi_i(x_i, y_i)) \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j) \end{aligned} \tag{2.95}$$

Therefore, the conditional distribution $p(x \mid y)$ takes the form

$$p(x \mid y) \propto p(x, y) \propto \prod_{i=1}^{N} (\varphi_i(x_i)\xi_i(x_i, y_i)) \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j). \tag{2.96}$$

Now, consider the following MRF with potentials defined as $\tilde{\varphi}_i(x_i) = \varphi_i(x_i)\xi_i(x_i, y_i)$, $\forall i = 1, \ldots, N$, $\psi_{ij}(x_i, x_j), \forall (i, j) \in \mathcal{E}$. The (unconditional) joint distribution $\tilde{p}(x)$ would be

$$\tilde{p}(x) \propto \prod_{i=1}^{N} \tilde{\varphi}_i(x_i) \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j) = \prod_{i=1}^{N} (\varphi_i(x_i)\xi_i(x_i, y_i)) \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j). \tag{2.97}$$

By comparing Eqs. (2.96), (2.97) we see that $p(x \mid y) = \tilde{p}(x)$. Therefore, we see that the problems of finding conditional node marginals or MAP assignments given observations can be reduced to an unconditional one (containing only the hidden variables) as long as the node potentials of hidden variables are updated appropriately to take into account the "local" effect each observation Y_i has to its generating variable X_i. Thus, every time an observation $Y_i = y_i$ is made corresponding to variable X_i, we just need to multiply the edge potential $\xi_i(x_i, y_i)$ into the existing node potential of X_i, $\varphi_i(x_i) := \varphi_i(x_i)\xi_i(x_i, y_i)$.

If we were to evaluate the marginal at a node i naïvely, this would amount to $|\mathcal{X}|^N$ operations, where $\mathcal{X}$, is the (common) alphabet of hidden nodes as we would have to evaluate the following quantity

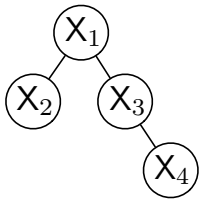
$$p_i(x_i) = \frac{Z_i(x_i)}{\sum_{x_i} Z_i(x_i)},$$

where $Z_i(x_i) = \sum_{x_{V\setminus i}} \prod_{i=1}^{N} \varphi_i(x_i) \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j)$. For every value $x_i \in \mathcal{X}$, $Z_i(x_i)$ is computed in $|\mathcal{X}|^{N-1}$ steps, resulting in an overall complexity of $|\mathcal{X}|^N$ (for the determination of one marginal). Same analysis holds for the MAP assignment. For a Gaussian graphical model of N variables (when we are in canonical form), finding the node marginals, translates to determining the marginal means and variances, which requires the inversion of the $N \times N$ precision matrix, an $\mathcal{O}(N^3)$ operation. If additionally, each hidden variable $\mathbf{X}_i$ is a vector of dimension d, the inversion of J would be an $\mathcal{O}(d^3N^3)$ operation. It is immediately obvious that completing the inference tasks in a brute-force way is infeasible. Fortunately, there exist algorithms that take into account the structure of the graph and result in a dramatic reduction in complexity. One such dynamic programing algorithm, which runs linearly in the number of nodes, N, is the belief propagation algorithm that we discuss below.

■ 2.10.1 Belief Propagation

Belief propagation (BP) is a message passing algorithm for performing inference on graphical models. The sum-product version of BP calculates the marginal distribution for each hidden node conditioned on the observed ones [82].[3] Two messages are transmitted on each edge (i, j), one from $i \to j$ and one from $j \to i$. A message from node i to node j essentially contains all the information from the subtree rooted at node i, plus the information enclosed on the node potential, $\varphi_i(x_i)$, and the pairwise potential, $\psi_{ij}(x_i, x_j)$. This message captures the effect of eliminating the subtree rooted at node i. Once we have all the incoming messages to node i correctly updated, the evaluation of its marginal is a trivial operation, which is at most linear in the number of nodes in the graph. A visualization of the above is given in Figs. 2.3a, 2.3b. For clarity of exposition, we will consider trees, where the algorithm is exact, but it can be generalized to any type of MRFs with often very satisfying results. Before we analyze the belief propagation algorithm further, we introduce a toy example to explain the logic behind it.

Example 2.10.1. *Consider the graphical model*



[3] There is also the *max-product* version of BP, where the result is the MAP sequence given the observations.

specified by the distribution

$$p(x_1, x_2, x_3, x_4) = \frac{1}{Z}\varphi_1(x_1)\varphi_2(x_2)\varphi_3(x_3)\varphi_4(x_4)\psi_{12}(x_1, x_2)\psi_{13}(x_1, x_3)\psi_{34}(x_3, x_4), \tag{2.98}$$

where Z is the partition function. Let us assume w.l.o.g. that we are interested in the marginal of node 1. Then , we have

$$\begin{aligned} p_1(x_1) &= \frac{1}{Z}\sum_{x_2}\sum_{x_3}\sum_{x_4}\varphi_1(x_1)\varphi_2(x_2)\varphi_3(x_3)\varphi_4(x_4)\psi_{12}(x_1, x_2)\psi_{13}(x_1, x_3)\psi_{34}(x_3, x_4) \\ &\propto \sum_{x_2}\sum_{x_3}\sum_{x_4}\varphi_1(x_1)\varphi_2(x_2)\varphi_3(x_3)\varphi_4(x_4)\psi_{12}(x_1, x_2)\psi_{13}(x_1, x_3)\psi_{34}(x_3, x_4). \end{aligned}$$

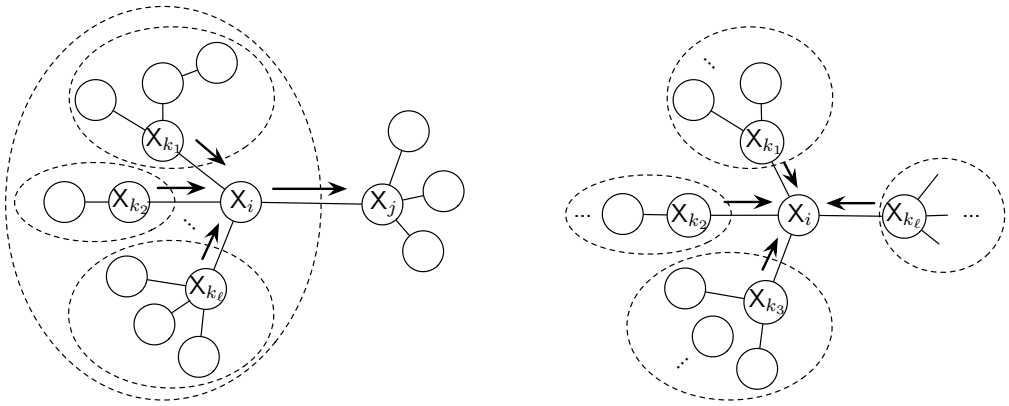
If we were to evaluate $p_1(x_1)$ naïvely, this would amount to $\mathcal{O}(|\mathcal{X}|^4)$ operations. However, if we push every sum operator as deep as possible in the expression, we can induce a significant reduction in computation. In fact, the marginal can be rewritten as

$$\begin{aligned} p_1(x_1) &\propto \sum_{x_2}\sum_{x_3}\sum_{x_4}\varphi_1(x_1)\varphi_2(x_2)\varphi_3(x_3)\varphi_4(x_4)\psi_{12}(x_1, x_2)\psi_{13}(x_1, x_3)\psi_{34}(x_3, x_4) \\ &\propto \varphi_1(x_1)\underbrace{\sum_{x_2}\varphi_2(x_2)\psi_{12}(x_1, x_2)}_{m_2(x_1)}\underbrace{\sum_{x_3}\varphi_3(x_3)\psi_{13}(x_1, x_3)\underbrace{\sum_{x_4}\varphi_4(x_4)\psi_{34}(x_3, x_4)}_{m_4(x_3)}}_{m_3(x_1)}. \end{aligned}$$

Notice, that we have pushed furthest sums that correspond to leaves (nodes $2, 4$) and we kept going upwards to the root, which we assumed to be the node of interest, that is, node 1. Message $m_4(x_3)$ is essentially the message from node 4 to 3, which is represented more clearly as $m_{4\to 3}(x_3)$ and conveys the information by the elimination of node 4 to node 3. For each value of x_3, $m_4(x_3)$ takes $\mathcal{O}(|\mathcal{X}|)$ to evaluate, thus amounting to $\mathcal{O}(|\mathcal{X}|^2)$ for all values of X_3. Similarly, $m_3(x_1)$ or else $m_{3\to 1}(x_1)$ transfers all the information that was generated from the elimination of nodes $3, 4$ to node 1 and completes in $\mathcal{O}(|\mathcal{X}|^2)$ time. Same for message $m_2(x_1)$. Since each message takes $\mathcal{O}(|\mathcal{X}|^2)$ time and three messages were propagated, the total time would be $\mathcal{O}(3|\mathcal{X}|^2)$ for the evaluation of one marginal, which is much more efficient than $\mathcal{O}(|\mathcal{X}|^4)$ of the naïve approach. The same process can be followed for the evaluation of all marginals. In fact, as we show later, the marginals of all nodes can be retrieved with a more clever bookkeeping.

As we hinted in the example, in the serial version of BP an arbitrary node is chosen as a root, then messages are passed from leaves to the root. This is sufficient to provide the marginal at the root. If we additionally propagate messages from root to the leaves, this allows for the evaluation of the marginals at all nodes. A message is passed from node i to node j, once all messages from the other neighbors of i are correctly updated. This is the intuition behind starting from the leaves and continuing to the root. When

a message propagates from a leaf to its parent, it is guaranteed to be correct since the leaf has no other neighbors (other than its parent). Then messages from leaves' parents to their parents would also be correct since the messages at the lower level have been correctly updated in the previous step. Continuing in this fashion, when we reach the root, which would mark the end of the first pass of the method, all messages pointing towards the root would be correct. Once all incoming messages to node i are correctly updated, which would be by the end of the first pass, its marginal can be obtained in $\mathcal{O}(|\mathcal{X}|)$ time as $p_i(x_i) \propto \varphi_i(x_i) \prod_{k \in N(i)} m_{k \to i}(x_i)$. In the second pass, we propagate messages from the root to the leaves. To see why this guarantees the correctness of these messages, suppose that the root sends a message to any of its children (for clarity, name this node j). This message is guaranteed to be correct, since all the incoming messages to the root from its other neighbors (all other children of the root excluding j) are correct. The same reasoning follows as we propagate messages towards the lower layers of the tree. At the end of the second pass, all incoming messages to all nodes would be correct, and hence the marginals at all nodes can be obtained (in linear time for each node). Since a tree has $N-1$ edges and two passes are required, a total of $2(N-1)$ messages are propagated, with each message resulting in $\mathcal{O}(|\mathcal{X}|^2)$ complexity. Hence, the total complexity of the serial version of BP is $\mathcal{O}(N|\mathcal{X}|^2)$. To summarize the



(a) Message passing (b) Marginal evaluation

Figure 2.3: Message passing and marginal evaluation. (a) Message $m_{i \to j}(x_j)$ contains all the information of the subtree rooted at node i. Consequently, node i gets all the information from the subtrees of its neighbors (except node j) as expressed in $m_{i \to j}(x_j) = \sum_{x_i} \phi_i(x_i) \psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{k \to i}(x_i)$. Essentially, message $m_{i \to j}(\cdot)$ represents the effect of eliminating the subtree rooted at i. (b) Once all incoming messages to node i are correctly updated, its marginal can be evaluated as $p_i(x_i) \propto \phi_i(x_i) \prod_{k \in N(i)} m_{k \to i}(x_i)$. Each incoming message captures the information from each subtree rooted at every neighbor of i.

serial version of BP, we provide its algorithm in 2.7.

Algorithm 2.7 Serial Belief Propagation

Initialization

Choose an arbitrary root r and set all messages to 1:
$m_{i\to j}(x_j) = 1, m_{j\to i}(x_i) = 1, \forall (i,j) \in \mathcal{E}, x_i, x_j \in |\mathcal{X}|$.

First pass

Generate messages starting from the leaves and going towards the root:
For every (i,j) such that $j = \text{pa}(i)$,

$$m_{i\to j}(x_j) = \sum_{x_i} \varphi_i(x_i)\psi_{ij}(x_i,x_j) \prod_{k\in N(i)\setminus j} m_{k\to i}(x_i). \tag{2.99}$$

The initial i's are the leaves of the tree.

Second pass

Generate messages starting from the root and going towards the leaves:
For every (i,j) such that $j = \text{ch}(i)$,

$$m_{i\to j}(x_j) = \sum_{x_i} \varphi_i(x_i)\psi_{ij}(x_i,x_j) \prod_{k\in N(i)\setminus j} m_{k\to i}(x_i). \tag{2.100}$$

The initial i is the root, $i = r$.

Marginal evaluation

For every i, evaluate the node and edge marginals as

$$p_i(x_i) \propto \varphi_i(x_i) \prod_{k\in N(i)} m_{k\to i}(x_i)$$

$$p_{ij}(x_i,x_j) \propto \varphi_i(x_i)\varphi_j(x_j)\psi_{ij}(x_i,x_j) \prod_{k\in N(i)\setminus j} m_{k\to i}(x_i) \prod_{\ell\in N(j)\setminus i} m_{\ell\to j}(x_j),$$

for every $i \in \{1,\ldots,N\}$, $(i,j) \in \mathcal{E}$.

Parallel version

The serial version of BP described above requires that all messages are updated in a certain manner, so that at the end all messages are correctly updated. However, due to the inherent locality of the information that each message conveys, the algorithm can be easily parallelized. More precisely, if we let $m_{i\to j}^{(t)}(x_j)$ to be the message from i to j at step t, the algorithm described in 2.8 forms the parallel version of BP. It can be shown that loopy BP converges to the exact solution in trees within a maximum number of iterations equal to the tree diameter.

Interestingly enough, parallel BP can be used for general loopy graphs, giving good approximations to hard problems [21, 70, 74, 98]. In the case of graphs with cycles, it is referred as *loopy BP*. However, loopy BP is not guaranteed to converge in general neither give correct solutions. Several works have studied the performance

Algorithm 2.8 PARALLEL BELIEF PROPAGATION

Initialization

Initialize all messages to 1:
$m^{(0)}_{i\to j}(x_j) = 1, m^{(0)}_{j\to i}(x_i) = 1, \forall (i,j) \in \mathcal{E}, x_i, x_j \in |\mathcal{X}|$.

Iteration

Apply the following update until convergence:

$$m^{(t+1)}_{i\to j}(x_j) = \sum_{x_i} \varphi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k\in N(i)\setminus j} m^{(t)}_{k\to i}(x_i). \tag{2.101}$$

Marginal evaluation

Assuming that the algorithm converges at step T , evaluate the node and edges marginals as

$$p_i(x_i) \propto \varphi_i(x_i) \prod_{k\in N(i)} m^{(T)}_{k\to i}(x_i)$$

$$p_{ij}(x_i, x_j) \propto \varphi_i(x_i)\varphi_j(x_j)\psi_{ij}(x_i, x_j) \prod_{k\in N(i)\setminus j} m^{(T)}_{k\to i}(x_i) \prod_{\ell\in N(j)\setminus i} m^{(T)}_{\ell\to j}(x_j),$$

for every i and $(i,j) \in \mathcal{E}$.

of loopy BP and considered the conditions of convergence [42, 69, 71, 99]. Yedidia et al. [105] showed that when BP converges on a loopy graph, it converges to a stationary point of an approximate free energy known as *Bethe free energy*. The Bethe free energy is defined as $\mathscr{F}(\mu) = -\sum_x \mu(x)E(x) - \sum_x \mu(x)\log\mu(x)$, where $E(x) = -\sum_{i=1}^N \varphi_i(x_i) - \sum_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j)$ is the energy representing distribution $p_{\boldsymbol{X}}(\cdot)$ and μ is any valid distribution supported in graph $G = (V, \mathcal{E})$.

With a little algebraic manipulation, $\mathscr{F}(\cdot)$ can we be rewritten as

$$\mathscr{F}(\mu) = \log Z - D(\mu \| p_{\boldsymbol{X}}) \leq \log Z,$$

where $\log Z$ is the log-partition function. We see that if we had a way to find the maximizing point for $\mathscr{F}(\cdot)$ (that sets the KL-divergence $D(\mu\|p_{\boldsymbol{X}})$ to zero, thus minimizing it), we would retrieve the log-partition function: $\max_\mu \mathscr{F}(\mu) = \log Z$.[4] This point is exactly $\mu = p_{\boldsymbol{X}}$. However, for complex distributions determining this point is NP-hard.

Therefore, they considered an approximation of the above problem by assuming only "tree-like" distributions for μ. In other words, distributions that can be expressed in the form:

$$\mu(x) = \prod_{i=1}^N \mu_i(x_i) \prod_{(i,j)\in\mathcal{E}} \frac{\mu_{ij}(x_i, x_j)}{\mu_i(x_i)\mu_j(x_j)},$$

[4] Once we know the value of the log-partition function, evaluating node and edge marginals is easy.

where $\mathcal{E}$ is the edge set of the original graph G. They demonstrated the following connection between the Bethe approximation and the fixed points of loopy BP:

Theorem 2.10.1 (Convergence of loopy BP to stationary points of the Bethe free energy)**.** *The fixed points of belief propagation message updates result in node and edge marginals that are stationary points of the Bethe variational problem defined as:*

$$
\begin{aligned}
\max_{\mu} \quad & \mathscr{F}(\mu) \triangleq -\sum_{x} \mu(x) E(x) - \sum_{x} \mu(x) \log \mu(x) \\
& = -\sum_{x} \mu(x) \left(-\sum_{i=1}^{N} \varphi_i(x_i) - \sum_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j) \right) - \sum_{x} \mu(x) \log \mu(x) \\
\text{s.t.} \quad & \mu(x) = \prod_{i=1}^{N} \mu_i(x_i) \prod_{(i,j)\in\mathcal{E}} \frac{\mu_{ij}(x_i, x_j)}{\mu_i(x_i)\mu_j(x_j)} \\
& \sum_{x_i} \mu_i(x_i) = 1 \\
& \sum_{x_j} \mu_{ij}(x_i, x_j) = \mu_i(x_i) \\
& \sum_{x_i} \mu_{ij}(x_i, x_j) = \mu_j(x_j) \\
& \mu_i(x_i) \geq 0 \\
& \mu_{ij}(x_i, x_j) \geq 0.
\end{aligned}
$$

Proof. A proof of this theorem in given in [105]. ■

However, these fixed points may not be unique nor the results would be exact. In addition, these fixed points might correspond to some other type of stationary points other than maximizing points. For Gaussian BP, it has been shown that when loopy BP converges, this results in the right means but generally incorrect variances [99].

Each iteration of the algorithm requires computing the messages associated with every edge, which leads to a total time of $\mathcal{O}(T_{\max} N |\mathcal{X}|^2)$, where $T_{\max}$ is the maximum number of iterations. For trees, $T_{\max}$ equals to the tree diameter, which is the length of the longest path in the tree. It is immediately obvious that for trees, the parallel procedure entails a significant overhead if implemented sequentially compared to the serial version. However, its parallelized nature can be exploited to generate a synchronous message schedule.

Efficient implementation

As we see in Eqs. (2.99)-(2.101), the complexity of each message depends on the number of neighbors of the source node and consequently on the type of graph. Thus, the worst case complexity can be linear in the number of nodes per message. For example, in a

star graph, where a central node is connected to the remaining $N-1$ nodes, each node (excluding the central node) has $\mathcal{O}(N)$ neighbors and thus complexity per message is $\mathcal{O}(N|\mathcal{X}|^2)$ and $\mathcal{O}(N^2|\mathcal{X}|^2)$ in total. In fact, we can make the complexity per message independent of the type of graph by doing some clever bookkeeping. Specifically, we consider the "pseudo-marginal" $b_i(x_i)$ as

$$b_i(x_i) \propto \varphi_i(x_i) \prod_{k \in N(i)} m_{k \to i}(x_i).$$

The computation of all pseudo-marginals takes $\mathcal{O}(|\mathcal{X}| \sum_{i=1}^{N} |N(i)|)$ time. For a tree, $\sum_{i=1}^{N} |N(i)| = 2(N-1)$ and hence the computation of all pseudo-marginals is linear in N. Each message can now be expressed as

$$m_{i \to j}(x_j) \propto \sum_{x_i} \psi_{ij}(x_i, x_j) \frac{b_i(x_i)}{m_{j \to i}(x_i)},$$

which is always of complexity $\mathcal{O}(|\mathcal{X}|^2)$ for trees and independent of the graph structure. For loopy graphs, complexity in the worst case per iteration is $\mathcal{O}\left(N|\mathcal{X}| \max(N, |\mathcal{X}|)\right)$.

Max-Product

We described above the sum-product version of BP, which returns the marginals at every node and pair of connected nodes. Another problem of interest in inference is the most likely sequence of hidden states given the observations:

$$x^* \in \arg\max_x \underbrace{p(x \mid y)}_{\tilde{p}(x)}.$$

As explained before, this problem is equivalent to an unconditional one as long as the node potentials of the hidden nodes are updated accordingly to capture the "local" effect from each observation, $\varphi_i(x_i) := \varphi_i(x_i) p(y_i \mid x_i)$. It turns out there is a very similar algorithm for this problem that runs in time linear to the number of nodes. More specifically, messages are flowing in both directions of every edge just like the sum-product version of BP. The only difference is that sums are replaced with max. In fact, to avoid numerical underflow, we consider the problem

$$x^* \in \arg\max_x \log p(x \mid y)$$

instead, which is equivalent to the original maximization problem. The algorithm for max-product is outlined in 2.9.

Obviously, the max-product can be implemented serially or in parallel as in the sum-product case. We see from Defn. (2.104), that we can recover the maximum value in $\mathcal{O}(|\mathcal{X}|)$ time from any max-marginal. However, what is often more useful is the maximizing point, x^*. For this, we need to introduce an additional type of

Algorithm 2.9 MAX-PRODUCT

Initialization

Set all messages to 0: $m_{i\to j}(x_j) = 0, m_{j\to i}(x_i) = 0, \forall (i,j) \in \mathcal{E}, x_i, x_j \in |\mathcal{X}|$.

Messages

For $(i,j) \in \mathcal{E}$:

$$m_{i\to j}(x_j) = \max_{x_i} \{\log \varphi_i(x_i) + \log \psi_{ij}(x_i, x_j) + \sum_{k \in N(i)\setminus j} m_{k\to i}(x_i)\}. \tag{2.102}$$

Max-marginal

For every i, evaluate the max-marginal as

$$\bar{p}_i(x_i) = \exp(\log \varphi_i(x_i) + \sum_{k \in N(i)\setminus j} m_{k\to i}(x_i)), \tag{2.103}$$

where the *max-marginal* is defined as

$$\bar{p}_i(x_i) \triangleq \max_{x_{V\setminus i}} p_{\mathsf{X}}(x). \tag{2.104}$$

"backtracking" messages, that do not increase the complexity in $\mathcal{O}$ terms, and are defined as

$$\delta_{i\to j}(x_j) = \arg\max_{x_i} \{\log \varphi_i(x_i) + \log \psi_{ij}(x_i, x_j) + \sum_{k \in N(i)\setminus j} m_{k\to i}(x_i)\}. \tag{2.105}$$

Message $\delta_{i\to j}(x_j)$ provides the value of X_i that maximizes message $m_{i\to j}(x_j)$, when $\mathsf{X}_j = x_j$. The way to recover the MAP sequence is by choosing arbitrarily a node r as a root. We find the value x_r^* that maximizes $\bar{p}_r(x_r)$ and then we backtrack by determining the values of its children that correspond to the maximizing value, x_r^*. In other words,

$$x_i^* = \delta_{i\to r}(x_r^*), \forall i \in \text{ch}(r).$$

We recurse until all nodes have been assigned a value.

■ 2.10.2 Gaussian Belief Propagation

It turns out, there is a straightforward extension of BP to Gaussian graphical models. Before we present the algorithm, we can sketch the intuition behind it with an example.

Example 2.10.2. *Let us consider the graphical model used in Ex. 2.10.1 and let us*

assume the variables are jointly Gaussian.

$$p(x_1,x_2,x_3,x_4) \propto \exp(h^T x - \frac{1}{2}x^T J x)$$
$$\propto \exp(h_1^T x_1 + h_2^T x_2 + h_3^T x_3 + h_4^T x_4 - \frac{1}{2}x_1^T J_{11} x_1 - \frac{1}{2}x_2^T J_{22} x_2$$
$$- \frac{1}{2}x_3^T J_{33} x_3 - \frac{1}{2}x_4^T J_{44} x_4 - x_1^T J_{12} x_2 - x_1^T J_{13} x_3 - x_3^T J_{34} x_4),$$

where $h = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix}$, $J = \begin{bmatrix} J_{11} & J_{12} & J_{13} & 0 \\ J_{21} & J_{22} & 0 & 0 \\ J_{31} & 0 & J_{33} & J_{34} \\ 0 & 0 & J_{43} & J_{44} \end{bmatrix}$, $x_1, x_2, x_3, x_4 \in \mathbb{R}^d$.

If we are interested in finding the marginal of X_1 *this would be:*

$$p_1(x_1) \propto \int_{x_2}\int_{x_3}\int_{x_4} \exp(h_1^T x_1 + h_2^T x_2 + h_3^T x_3 + h_4^T x_4 - \frac{1}{2}x_1^T J_{11} x_1 - \frac{1}{2}x_2^T J_{22} x_2$$
$$- \frac{1}{2}x_3^T J_{33} x_3 - \frac{1}{2}x_4^T J_{44} x_4 - x_1^T J_{12} x_2 - x_1^T J_{13} x_3 - x_3^T J_{34} x_4)\,\mathrm{d}x_2\mathrm{d}x_3\mathrm{d}x_4$$
$$\propto \exp(h_1^T x_1 - \frac{1}{2}x_1^T J_{11} x_1) \underbrace{\int_{x_2} \exp((h_2 - J_{21}x_1)^T x_2 - \frac{1}{2}x_2^T J_{22} x_2)\,\mathrm{d}x_2}_{m_2(x_1)} \cdot$$
$$\underbrace{\int_{x_3} \exp((h_3 - J_{31}x_1)^T x_3 - \frac{1}{2}x_3^T J_{33} x_3 \underbrace{\int_{x_4} \exp((h_4 - J_{43}x_3)^T x_4 - \frac{1}{2}x_4^T J_{44} x_4)\,\mathrm{d}x_4}_{m_4(x_3)})\,\mathrm{d}x_3}_{m_3(x_1)}$$
$$= \exp(h_1^T x_1 - \frac{1}{2}x_1^T J_{11} x_1) m_2(x_1) m_3(x_1). \tag{2.106}$$

Message $m_4(x_3)$ *(or better denoted by* $m_{4\to 3}(x_3)$*) equals to*

$$m_4(x_3) = \int_{x_4} \exp((h_4 - J_{43}x_3)^T x_4 - \frac{1}{2}x_4^T J_{44} x_4)\,\mathrm{d}x_4$$
$$= (2\pi)^{d/2}|J_{44}|^{-1/2} \exp(\frac{1}{2}(h_4 - J_{43}x_3)^T J_{44}^{-1}(h_4 - J_{43}x_3))$$
$$\propto \exp((-J_{34}J_{44}^{-1}h_4)^T x_3 + \frac{1}{2}x_3^T J_{34} J_{44}^{-1} J_{43} x_3)$$
$$= \exp(h_{4\to 3}^T x_3 - \frac{1}{2}x_3^T J_{4\to 3} x_3) \propto \mathcal{N}^{-1}(x_3; h_{4\to 3}, J_{4\to 3}),$$

where $h_{4\to 3} = -J_{34}J_{44}^{-1}h_4$, $J_{4\to 3} = -J_{34}J_{44}^{-1}J_{43}$.

Therefore, message $m_3(x_1)$ *takes the form*

$$m_3(x_1) = \int_{x_3} \exp((h_3 + h_{4\to 3} - J_{31}x_1)^T x_3 - \frac{1}{2}x_3^T (J_{33} + J_{4\to 3}) x_3)\,\mathrm{d}x_3.$$

On the same account, $m_3(x_1)$ simplifies to

$$m_3(x_1) \propto \exp((-J_{13}(J_{33}+J_{4\to3})^{-1}(h_3+h_{4\to3}))^T x_1 - \frac{1}{2}x_1^T(-J_{13}(J_{33}+J_{4\to3})^{-1}J_{31})x_1)$$
$$= \exp(h_{3\to1}^T x_1 - \frac{1}{2}x_1^T J_{3\to1}x_1) \propto \mathcal{N}^{-1}(x_1; h_{3\to1}, J_{3\to1}), \tag{2.107}$$

where $h_{3\to1} = -J_{13}(J_{33}+J_{4\to3})^{-1}(h_3+h_{4\to3})$, $J_{3\to1} = -J_{13}(J_{33}+J_{4\to3})^{-1}J_{31}$.

Similarly, one can show that $m_2(x_1)$ equals to

$$m_2(x_1) \propto \mathcal{N}^{-1}(x_1; h_{2\to1}, J_{2\to1}), \tag{2.108}$$

where $h_{2\to1} = -J_{12}J_{22}^{-1}h_2$, $J_{2\to1} = -J_{12}J_{22}^{-1}J_{21}$.

A careful observer would notice that messages are proportional to Gaussian distributions whose parameters are given by formulas which resemble the BP updates for the discrete case. That is, h_i and on-diagonal element J_{ii} resemble a node potential, while J_{ij} an edge potential. In addition, parameters of messages $(h_{k\to i}, J_{k\to i})$ pointing to a source node add up to form the message from node i to j. For example, in order to evaluate $J_{3\to1}$ all other neighbors of 3 (in this case, node 4) are added to the on-diagonal term of the source node, J_{33}. A similar operation is needed for the evaluation of $h_{3\to1}$.

The evaluation of marginal at node 1 from Eq. (2.106) due to Eqs. (2.107), (2.108) becomes

$$p_1(x_1) = \mathcal{N}^{-1}(x_1; \hat{h}_1, \hat{J}_1) \propto \exp((h_1+h_{2\to1}+h_{3\to1})^T x_1 - \frac{1}{2}x_1^T(J_{11}+J_{2\to1}+J_{3\to1})x_1) \Rightarrow$$
$$\hat{h}_1 = h_1+h_{2\to1}+h_{3\to1}, \quad \hat{J}_1 = J_{11}+J_{2\to1}+J_{3\to1}.$$

The computation of each message requires $\mathcal{O}(d^3)$ time, while the marginal at node 1 $\mathcal{O}(d^2)$ time. Compare this to the $\mathcal{O}((4d)^3)$ computation that the matrix inversion of the full precision matrix J would require to obtain the marginal means and covariances.

After we roughly outlined the BP method for the Gaussian case with an example, we proceed by giving the full algorithm in 2.10. Gaussian BP can be implemented either in serial or parallel form as its counterpart for discrete variables. If implemented serially, a root should be chosen randomly and then messages should flow from leaves to the root and then backwards from root to the leaves. If implemented in parallel, messages are exchanged locally until they converge. For trees, the convergence point would be at most within a number of steps equal to the tree diameter. The complexity of every message is $\mathcal{O}(d^3)$, where d is the dimension for every hidden variable X_i.[5] Since $2(N-1)$ are transmitted in the serial version, the total complexity would be $\mathcal{O}(Nd^3)$.[6] Therefore, all marginals can be recovered in $\mathcal{O}(Nd^3)$ time as opposed to $\mathcal{O}(N^3d^3)$ time that the inversion of the full J matrix would entail.

[5] We assume all hidden variables are of dimension d.

[6] It is $\mathcal{O}(T_{\max}Nd^3)$ for the parallel version.

Algorithm 2.10 GAUSSIAN BELIEF PROPAGATION

Initialization

Set all messages to 0: $h_{i\to j}=0, J_{i\to j}=0, h_{j\to i}=0, J_{j\to i}=0, \forall (i,j)\in\mathcal{E}$.

Messages

For $(i,j)\in\mathcal{E}$:

$$h_{i\to j}=-J_{ji}\left(J_{ii}+\sum_{k\in N(i)\setminus j}J_{k\to i}\right)^{-1}\left(h_i+\sum_{k\in N(i)\setminus j}h_{k\to i}\right) \tag{2.109}$$

$$J_{i\to j}=-J_{ji}\left(J_{ii}+\sum_{k\in N(i)\setminus j}J_{k\to i}\right)^{-1}J_{ij}. \tag{2.110}$$

Marginal

For every i, evaluate its marginal, $X_i\sim\mathcal{N}^{-1}(x_i;\hat{h}_i,\hat{J}_i)$ as

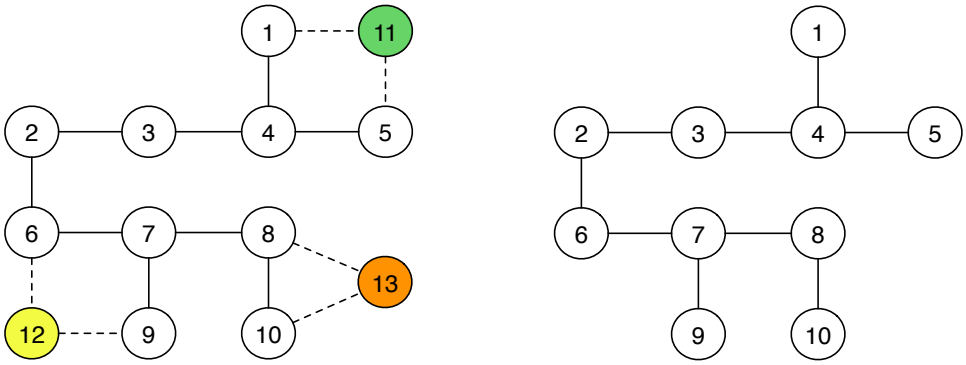
$$\hat{h}_i=h_i+\sum_{k\in N(i)}h_{k\to i} \tag{2.111}$$

$$\hat{J}_i=J_{ii}+\sum_{k\in N(i)}J_{k\to i}. \tag{2.112}$$

In jointly Gaussian variables, the mean is the most likely assignment. Hence, there is no need to come up with a max-product version for Gaussian models, as the most likely sequence can be derived by setting each $x_i^*=\hat{J}_i^{-1}\hat{h}_i$, where $X_i\sim\mathcal{N}^{-1}(x_i;\hat{h}_i,\hat{J}_i)$ is the output of Gaussian BP. Lastly, Gaussian BP can also be applied to loopy graphs as well, but there are no guarantees of convergence. As Weiss and Freeman [99] demonstrated, loopy Gaussian BP has no convergence guarantees, but when convergence is reached the estimated means equal to the true ones. However, there is no guarantee for the estimated covariances and in fact they can be far from the true ones.

■ 2.10.3 Feedback Message Passing (FMP)

Liu et al. [64] proposed the Feedback Message Passing (FMP) algorithm that breaks the potentially loopy graph in two parts; one cycle-free $\mathcal{T}$ and a set of nodes, called the *feedback vertex set* (FVS) $\mathcal{F}$, whose removal results in the cycle-free graph $\mathcal{T}$. An example of a graph that becomes acyclic after the removal of FVS nodes is shown in Fig. 2.4. This method provides a way to evaluate the exact means and variances in loopy Gaussian MRFs. Let us split the potential vector and information matrix in two parts corresponding to the FVS $\mathcal{F}$ and acyclic graph $\mathcal{T}$ as $h=\begin{bmatrix}h_{\mathcal{T}}\\h_{\mathcal{F}}\end{bmatrix}$ and



(a) Original graph (b) Acyclic graph $\mathcal{T}$

Figure 2.4: A graph with an FVS $\mathcal{F}$ of size 3. (a) Original graph G. (b) Acyclic-graph $\mathcal{T} = G \setminus \mathcal{F}$ after the removal of FVS $\mathcal{F} = \{11, 12, 13\}$.

$$J = \begin{bmatrix} J_{\mathcal{T}\mathcal{T}} & \overbrace{\begin{matrix} | & & | & & | \\ J_{\mathcal{T}p_1} & \cdots & J_{\mathcal{T}p} & \cdots & J_{\mathcal{T}p_K} \\ | & & | & & | \end{matrix}}^{h^p} \\ & \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{J_{\mathcal{T}\mathcal{F}}} \\ J_{\mathcal{F}\mathcal{T}} & J_{\mathcal{F}\mathcal{F}} \end{bmatrix}$$

Figure 2.5: Potential vector h^p for determining "feedback gains" $g_i^p, \forall i \in \mathcal{T}$.

$J = \begin{bmatrix} J_{\mathcal{T}\mathcal{T}} & J_{\mathcal{T}\mathcal{F}} \\ J_{\mathcal{F}\mathcal{T}} & J_{\mathcal{F}\mathcal{F}} \end{bmatrix}$. The FMP algorithm consists of several stages: On the first stage, an FVS $\mathcal{F}$ is determined from one of the existing algorithms. Then, the exact means and variances are obtained in two rounds. In the first round, BP runs on $\{h_{\mathcal{T}}, J_{\mathcal{T}\mathcal{T}}\}$, where $h_{\mathcal{T}}, J_{\mathcal{T}\mathcal{T}}$ correspond to the part of full potential vector and block of full information matrix that contains only nodes in $\mathcal{T} = V \setminus \mathcal{F}$ (the cycle-free graph after the removal of FVS nodes). This will produce messages $h_{i\to j}^{\mathcal{T}}, J_{i\to j}^{\mathcal{T}}, \forall i, j \in \mathcal{T}$ and $(i, j) \in \mathcal{E}$. Assuming the FVS $\mathcal{F}$ is of size K, we run BP K more times with parameters $\{h^p, J_{\mathcal{T}\mathcal{T}}\}_{p\in\mathcal{F}}$, where $h^p = J_{\mathcal{T}p}$. In other words, h^p is the column of information matrix J that is relevant to FV p and nodes in $\mathcal{T}$, as is shown in Fig. 2.5. Obviously, for every $i \in \mathcal{T} \setminus N(p)$, we have $[h^p]_i = 0$, since there is no direct link between FV p and i. This will generate messages $h_{i\to j}^p, \forall i, j \in \mathcal{T}, (i, j) \in \mathcal{E}$ and $p \in \mathcal{F}$. After the end of first round, we are provided with "partial" means and variances $\hat{\mu}_i^{\mathcal{T}}, \hat{\Sigma}_{ii}^{\mathcal{T}}, \forall i \in \mathcal{T}$ as well as "feedback gains" $g_i^p, \forall i \in \mathcal{T}, p \in \mathcal{F}$ that will be used in the second round of the method. It is

worth noting that the "partial" means and variances $\hat{\mu}_i^{\mathcal{T}}, \hat{\Sigma}_{ii}^{\mathcal{T}}, \forall i \in \mathcal{T}$ do not correspond to the true means and variances in these nodes as the contribution of feedback vertices is not taken into account yet.

Then, the feedback nodes collect the gains and partial means from their neighbors and update correspondingly the potential vector and information matrix at the FVS $\mathcal{F}$ as

$$[\hat{J}_{\mathcal{F}}]_{pq} = J_{pq} - \sum_{i \in N(p) \cap \mathcal{T}} J_{pi} g_i^q, \ \forall p, q \in \mathcal{F}$$
$$[\hat{h}_{\mathcal{F}}]_p = h_p - \sum_{i \in N(p) \cap \mathcal{T}} J_{pi} \hat{\mu}_i^{\mathcal{T}}, \ \forall p \in \mathcal{F}.$$

The (correct) means and variances of the FV nodes are recovered by solving a small inference problem

$$\Sigma_{\mathcal{F}} = \hat{J}_{\mathcal{F}}^{-1}$$
$$\mu_{\mathcal{F}} = \Sigma_{\mathcal{F}} \hat{h}_{\mathcal{F}}.$$

In the second round, the correct means $\mu_{\mathcal{F}}$ at the FVS are used to revise the potential vectors at the neighbors of FVS nodes as

$$\tilde{h}_i = h_i - \sum_{j \in N(i) \cap \mathcal{F}} J_{ij} [\mu_{\mathcal{F}}]_j, \ \forall i \in \mathcal{T}. \tag{2.113}$$

The potential vectors for $i \in \mathcal{T} \setminus N(p), \forall p \in \mathcal{F}$ remain the same,

$$\tilde{h}_i = h_i. \tag{2.114}$$

We retrieve the exact means on the cycle-free graph $\mathcal{T}$ by running BP one more time on $\mathcal{T}$ by propagating messages $\tilde{h}_{k \to i}^{\mathcal{T}}$ on $\{\tilde{h}_{\mathcal{T}}, J_{\mathcal{T}\mathcal{T}}\}$, where $\tilde{h}_{\mathcal{T}}$ is defined by Eqs. (2.113), (2.114). It is worth noting that messages $J_{i \to j}^{\mathcal{T}}, \forall i, j \in \mathcal{T}, (i,j) \in \mathcal{E}$ do not need to be recomputed as they are not affected by the revision of potential vectors. The end of BP provides the true means in $\mathcal{T}$

$$\mu_i = (J_{\mathcal{T}}^{-1} \tilde{h}_{\mathcal{T}}), \forall i \in \mathcal{T}.$$

Lastly, we estimate the variances in $\mathcal{T}$, $\hat{\Sigma}_{ii}^{\mathcal{T}}$ by adding correction terms computed from the first round as

$$\Sigma_{ii} = \hat{\Sigma}_{ii}^{\mathcal{T}} + \sum_{p \in \mathcal{F}} \sum_{q \in \mathcal{F}} g_i^p [\Sigma_{\mathcal{F}}]_{pq} g_i^q, \ \forall i \in \mathcal{T}.$$

A detailed description of FMP is provided in Alg. 2.11 and a flow of the algorithm in Fig. 2.6.

Theorem 2.10.2. *The FMP described in Alg. 2.11 outputs the exact means and variances for all nodes provided $\mathcal{F}$ is an FVS.*

Proof. A detailed proof can be found in Sec. III.C of [64]. ■

Algorithm 2.11 Feedback Message Passing (FMP)

1. Construct K potential vectors: $h^p = J_{\mathcal{T}p}, \forall p \in \mathcal{F}$.

2. Run BP $K+1$ times on $\mathcal{T}$ with parameters $\{h_{\mathcal{T}}, J_{\mathcal{T}\mathcal{T}}\}$, $\{h^p, J_{\mathcal{T}\mathcal{T}}\}_{p\in\mathcal{F}}$, which will produce messages $h^{\mathcal{T}}_{i\to j}, h^p_{i\to j}, J^{\mathcal{T}}_{i\to j}$ and marginals $\hat{\mu}^{\mathcal{T}}_i, g^p_i, \hat{\Sigma}^{\mathcal{T}}_{ii}, \forall i \in \mathcal{T}$.

3. Obtain graph of size K with updated parameters $\hat{h}_{\mathcal{F}}, \hat{J}_{\mathcal{F}}$ as

$$[\hat{J}_{\mathcal{F}}]_{pq} = J_{pq} - \sum_{i\in N(p)\cap\mathcal{T}} J_{pi} g^q_i, \ \forall p, q \in \mathcal{F} \tag{2.115}$$

$$[\hat{h}_{\mathcal{F}}]_p = h_p - \sum_{i\in N(p)\cap\mathcal{T}} J_{pi}\hat{\mu}^{\mathcal{T}}_i, \ \forall p \in \mathcal{F} \tag{2.116}$$

   and solve for $\Sigma_{\mathcal{F}} = \hat{J}^{-1}_{\mathcal{F}}$ and $\mu_{\mathcal{F}} = \Sigma_{\mathcal{F}}\hat{h}_{\mathcal{F}}$.

4. Revise the potential vector on $\mathcal{T}$ as

$$\tilde{h}_i = h_i - \sum_{j\in N(i)\cap\mathcal{F}} J_{ij}[\mu_{\mathcal{F}}]_j, \ \forall i \in \mathcal{T} \tag{2.117}$$

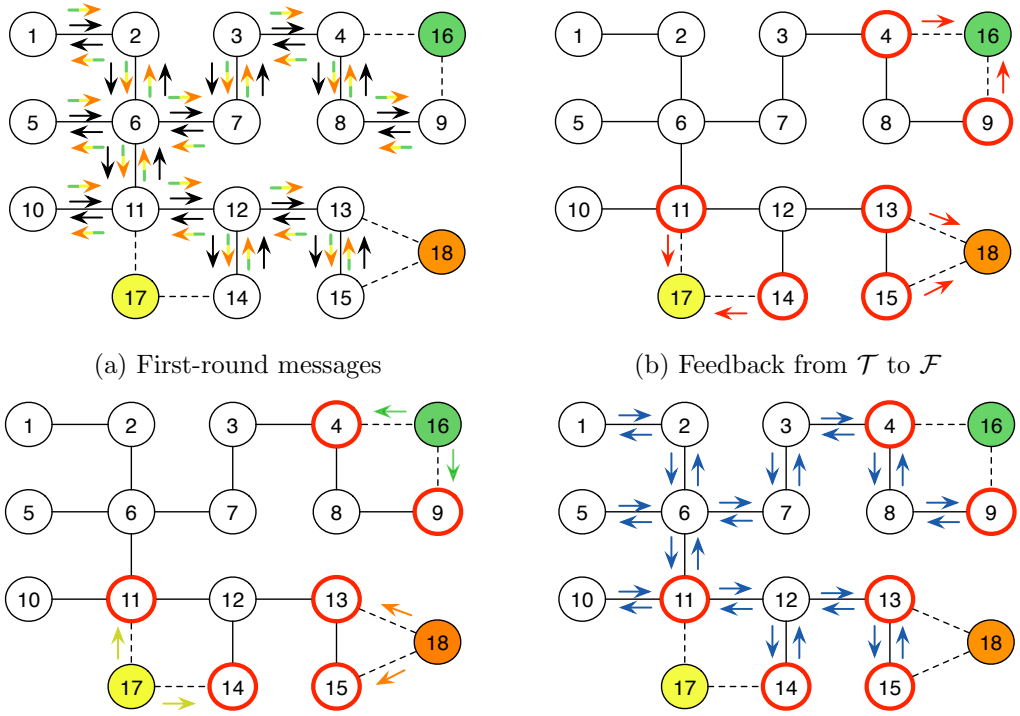
   and obtain the exact means by running BP one more time on the revised potential vector (the corresponding messages will be denoted by $\tilde{h}^{\mathcal{T}}_{i\to j}$).

5. Evaluate the means as

$$\hat{h}^{\mathcal{T}}_i = \tilde{h}_i + \sum_{k\in N(i)} \tilde{h}^{\mathcal{T}}_{k\to i}, \qquad \hat{\Sigma}^{\mathcal{T}}_{ii} = (J_{ii} + \sum_{k\in N(i)} J^{\mathcal{T}}_{k\to i})^{-1}, \qquad \mu_i = \hat{\Sigma}^{\mathcal{T}}_{ii}\hat{h}^{\mathcal{T}}_i. \tag{2.118}$$

6. Correct the variances with

$$\Sigma_{ii} = \hat{\Sigma}^{\mathcal{T}}_{ii} + \sum_{p\in\mathcal{F}}\sum_{q\in\mathcal{F}} g^p_i[\Sigma_{\mathcal{F}}]_{pq} g^q_i, \ \forall i \in \mathcal{T}. \tag{2.119}$$



(a) First-round messages

(b) Feedback from $\mathcal{T}$ to $\mathcal{F}$

(c) Feedback from $\mathcal{F}$ to $\mathcal{T}$

(d) Second-round messages

Figure 2.6: Flow of FMP algorithm. (a) In the first round, we run BP on $\mathcal{T}$ with parameters $\{h_{\mathcal{T}}, J_{\mathcal{T}\mathcal{T}}\}$. This generates messages $h^{\mathcal{T}}_{i\to j}, J^{\mathcal{T}}_{i\to j}$ depicted in black color. We also run BP K more times, where $K = |\mathcal{F}|$ with parameters $\{h^p, J_{\mathcal{T}\mathcal{T}}\}_{p\in\mathcal{F}}$ and produce messages $h^p_{i\to j}, \forall p \in \mathcal{F}$, which are depicted as multi-colored arrows. Each color shade corresponds to one of the three different FV nodes in this example. (b) After the messages are computed in the first round, the neighbors of FV nodes send feedback to the nodes in $\mathcal{F}$ in the form of "partial means" $\hat{\mu}^{\mathcal{T}}_i$ and "feedback gains" g^p_i. In more detail, the potential vector $h_{\mathcal{F}}$ and information matrix $J_{\mathcal{F}}$ at $\mathcal{F}$ are revised according to Eqs. (2.115), (2.116). (c) After we obtain the true means and variances at $\mathcal{F}$, we send feedback back to $\mathcal{T}$ in the form of revising the potential vectors at the neighbors of FV nodes. Each FV node sends feedback only to its immediate neighbors in $\mathcal{T}$ as illustrated with the choice of different arrow colors. The potential vectors at the FV neighbors are updated according to Eq. (2.117). The potential vectors at the remaining nodes in $\mathcal{T}$ stay unaffected. (d) In the last round, we run BP one more time on $\mathcal{T}$ with parameters $\{\tilde{h}_{\mathcal{T}}, J_{\mathcal{T}\mathcal{T}}\}$, where $\tilde{h}_{\mathcal{T}}$ are the revised potential vectors. Messages $J^{\mathcal{T}}_{i\to j}$ will not change from the previous round, since only potential vector $h_{\mathcal{T}}$ has been revised, something that does not affect the $J^{\mathcal{T}}_{i\to j}$ messages. Thus, in this round, we only need to propagate $\tilde{h}^{\mathcal{T}}_{i\to j}$ messages, depicted by blue color. At the end of this step, we can compute the true means and variances in $\mathcal{T}$ by Eqs. (2.118), (2.119).

Obtaining an FVS

The problem of deciding whether there exists an FVS with size at most K has been shown to be NP-complete by Karp [46]. In addition, finding the minimum FVS for general graphs is still an active research area. The fastest algorithm that exists so far runs in time $\mathcal{O}(1.7548^N)$, where N is the graph size [35]. However, approximate algorithms exist that produce an FVS whose size is only by a factor larger to the minimum. One of these methods by Bafna et al. [5] briefly works as follows. It starts by adding to FVS $\mathcal{F}$ all nodes with zero weight $w(u)$ and removing them from the graph G along with their incident edges. A node's weight is a non-negative number associated with each node. For example, in maximum flow problems it can relate to the node's capacity, or in other words, the maximum flow that can pass through a node. It continues by cleaning up the graph by recursively removing all edges with degree at most one along with their incident edges. During the main part of the algorithm, it finds the node u with the smallest weight to vertex degree ratio. As a reminder, the degree of a vertex u is the number of its neighbors $|N(u)|$. Then, it recalculates the weight of each vertex $w(u)$ as $w'(u) = w(u) - \gamma(|N(u)| - 1)$, where γ is the smallest weight to vertex degree ratio $\gamma = \min_{u \in V} w(u)/(|N(u)| - 1)$ and adds the nodes with zero weight to FVS, while removing them from G and cleaning it up, exactly in the same manner as in the initialization. The algorithm continues until there are not any nodes left in G. In the case of probabilistic undirected graphical models, the weights of nodes are initialized to one. The above algorithm runs in $\mathcal{O}(\min\{|\mathcal{E}| \log N, N^2\})$ and provides a 2-approximation ratio. In other words, the resulting FVS will not be more than two times larger than the minimum FVS. For example, a square grid graph with a side of size $\sqrt{N}$, which results in N hidden variables, has $2(\sqrt{N} - 1)\sqrt{N} = \mathcal{O}(N)$ edges. Therefore, finding an FVS with the above procedure takes $\mathcal{O}((\log N)^2)$ time.

Complexity of FMP

In step 2 of Alg. 2.11, we run BP $K + 1$ times on tree $\mathcal{T}$. The number of messages sent on each run is $\mathcal{O}(N - K)$. Therefore, for $K + 1$ runs, the complexity is $\mathcal{O}(K(N - K))$.[7] In step 3, the potential vector and information matrix at FVS $\mathcal{F}$ are revised in $\mathcal{O}(K^2 \max_{p \in \mathcal{F}} N(p))$. Also, the mean and covariance at $\mathcal{F}$ are evaluated in $\mathcal{O}(K^3)$ time. In step 4, it takes $\mathcal{O}(K \sum_{p \in \mathcal{F}} N(p)) = \mathcal{O}(K(N - K))$ time to revise the potential vectors at the neighbors of FV nodes. It also takes $\mathcal{O}(N - K)$ time to run BP one more time. Step 5 takes also $\mathcal{O}(N - K)$ time. Lastly, step 6 takes $\mathcal{O}(K^2(N - K))$ time.

The dominant term in the FMP algorithm is the one related to step 6, $\mathcal{O}(K^2(N - K))$, where we compute the variances at all nodes in $\mathcal{T}$. If our interest is knowing only a few variances in $\mathcal{T}$, then the dominant term comes from step 3 and the overall complexity is either $\mathcal{O}(K^2 \max_{p \in \mathcal{F}} N(p))$, if $\max_{p \in \mathcal{F}} \leq K$ or $\mathcal{O}(K^3)$, otherwise. In

[7] In this analysis, we ignore the inherent complexity to compute a message, which is $\mathcal{O}(|\mathcal{X}|^2)$ in the discrete and $\mathcal{O}(d^3)$ in the Gaussian case, where $|\mathcal{X}|$ and d are the alphabet size and hidden dimension, respectively.

other words, if the number of FVS nodes, $\mathcal{F} = K$, is larger than the maximum degree of an FV node, the complexity of FMP is dominated by the inversion of the $K \times K$ revised information matrix $\hat{J}_{\mathcal{F}}$ at $\mathcal{F}$.

Chapter 3

Theoretical Guarantees of Greedy Algorithms

INFORMATION gathering subject to resource constraints for estimation of an underlying phenomenon of interest poses several challenges. Information-driven methods seek to maximize information extraction while limiting resource expenditures via active control of the measurement process. Recent signal processing methods consider mutual information as the reward embedded in a dynamic sensing algorithm [30, 103, 107]. The problem of choosing an optimal subset of measurements is formulated as a combinatorial optimization problem which becomes intractable as the number of measurements grows. Fisher et al. [34], Nemhauser et al. [77] proposed, in their seminal work, approximating methods that run in polynomial time with nearly optimal guarantees to the optimal solution which is NP-hard [31]. Their work assumed general submodular monotone functions.

Krause and Guestrin [52, 54] were able to make use of the existing algorithms with provable bounds in information planning settings by recognizing that certain information rewards, such as the entropy and mutual information (under mild conditions) are submodular. Williams et al. [102] demonstrated that similar bounds hold for the more challenging sequential setting, where different constraints apply to disjoint observation sets. These previous results consider selection of measurements purely on utilizing monotone information rewards. In cases where measurements have non-uniform costs, one obvious drawback with choosing such rewards is that these are cost-unaware and always choose the measurement with the highest informational value regardless of the induced cost. In this chapter, we derive lower bounds of existing approximating algorithms when the reward is submodular but not necessarily monotone. This enables the introduction of penalized information rewards that take costs of measurements into account. In fact, we propose a penalized form of mutual information that takes costs of measurements into consideration and retains submodularity.

An additional challenge we face is to choose between the open-loop and closed-loop control policy. In the former, we perform planning based on the expected performance of measurements which can be done completely prior to accepting any measurements, while in the latter we perform planning in an online manner, where the next step is

determined upon the acquisition of actual measurement values. Obviously, in the first setting, there is no need to spend any resources to acquire actual measurements for proposing a plan, and therefore the generated plan might not reflect accurately the actual progression of a phenomenon of interest. It is of interest to characterize models where open-loop is equivalent to closed-loop control, so that the more cost-efficient open-loop approach is no different than the closed-loop one. It is well-known that Gaussian models satisfy this as information reward functions in this setting are related to the uncertainty of the underlying phenomenon of interest, which in turn does not depend on actual measurement values. In this work, we consider exponential families and derive sufficient and necessary conditions under which open-loop and closed-loop are equivalent.

Another interesting problem arises when there is a limited budget under which we select measurements. Previous works, provide approximating algorithms with either loose lower bounds [52] or optimal lower bounds that come at the expense of high complexity [90]. In this work, we derive upper bounds for the optimal solution by casting the budgeted problem in its dual form and making use of the recent algorithm by Buchbinder et al. [13], which runs in linear time with respect to the observation set size.

Lastly, another problem of interest we consider is that of focused planning [63]. Often, only a subset of the latent variables is of interest. In this case, the conditions under which mutual information is submodular do not hold anymore. We demonstrate that the use of the same approximating algorithm that applies to the submodular monotone case can be used in this case too with provable lower bounds under certain conditions. An earlier version of parts of this work was originally presented in [78].

We begin the chapter with Sec. 3.1 by discussing *value independent* models, where open-loop control planning is equivalent to closed-loop control planning and provide sufficient and necessary conditions for existence of such models. We show in Sec. 3.1.1 that Gaussian distributions satisfies the necessary condition. We continue with Sec. 3.2 where we derive lower bounds for greedy solutions when the reward function is submodular non-monotone. In Sec. 3.2.1, we introduce a penalized form of mutual information that takes into account measurement generating costs and is well-suited for settings where measurements have non-uniform costs. We also explore the case of varying costs in Sec. 3.3 that is interesting, when different consumers depending on their knowledge of the world are willing to "pay" different prices for the same measurement. The chapter continues with Sec. 3.4 by deriving upper bounds for the optimal solution of the submodular knapsack maximization problem (or in other words the budgeted batch setting). We resume in Sec. 3.5 by providing lower bounds for focused planning, where only a set of variables is of relevance $\mathcal{R}$ and the choice of reward is mutual information (MI). In this case, not all measurements are conditionally independent given the (latent) relevant set, a property that is essential for the submodularity of MI. Therefore, it is critical to determine an extended set $\hat{\mathcal{R}}$, that is a superset of the relevant set $\mathcal{R}$, which enforces conditional independence among all measurements so

that the known approximating algorithms are applied with lower-bound guarantees. We show in Sec. 3.5.1 how to generate such an extended set. The chapter continues by showing a synthetic example in Sec. 3.6, where the choice of penalized mutual information as reward function leads to solutions with higher informational value than the one generated by the mutual information, which does not consider the induced costs of measurements. We conclude with an outline of the contributions in Sec. 3.7

■ 3.1 Value Independent Models

In closed-loop control settings, the optimal policy is designed as new measurements become available. In more detail, as actual measurement values are drawn, optimal policy uses this new information to determine the next measurements to be considered. In open-loop control settings, a plan is devised based on the expected response of available measurements without utilizing or anticipating the availability of any future information (which would be in the form of actual measurement values). In the latter case, there is no need to actively draw measurement values for the construction of a plan. We are interested in settings where closed- and open-loop structures are equivalent. In other words, in settings where the information plan does not depend on the measurement values for a particular choice of reward function. This is the case with Gaussian models, since when the reward function is entropy or mutual information, it does not depend on the actual measurement values but rather the number of measurements. We provide analysis of conditions under which independence on measurement value can be generalized to exponential families and derive conditions under which an exponential family does not depend on measurement values. In Eq. (2.73), we presented the general form of an exponential family

$$p(x;\theta) = q(x)\exp(\langle\theta,\phi(x)\rangle - A(\theta)) = q(x)\exp(\theta^T\phi(x) - A(\theta)). \tag{3.1}$$

For our analysis, we will assume that the log-base distribution is a linear function of the natural statistic

$$\log q(x) = \gamma^T\phi(x), \tag{3.2}$$

where γ is a vector of constants. The above assumption is not an unrealistic one, since the log-base distribution is a linear combination of the natural statistic in many well-known distributions such as the Bernoulli, Exponential, Laplace, Gaussian, Gamma, Inverse Gamma, Beta, Dirichlet, Categorical, Wishart and Normal-Gamma.[1]

The independence of information planning on the measurement values is dictated by the reward function that guides the measurement plan. As we mentioned, we will focus on two information reward functions, mutual information (MI) and entropy. We first consider entropy, however, the results largely extend to mutual information since it can be expressed as a difference of entropies.

[1]There are, however, other distributions where this assumption does not hold such as the Binomial, Poisson, Chi-Squared and Lognormal.

For exponential families, entropy is expressed as a function of θ, γ as follows

$$H(\theta) = -\mathbb{E}[\log p(x;\theta)] \overset{(3.2)}{=} -\mathbb{E}[-A(\theta) + (\theta+\gamma)^T\phi(x)] \overset{(2.76)}{=} A(\theta) - (\theta+\gamma)^T\nabla A(\theta). \tag{3.3}$$

It is worth noting that when the log-base distribution $\log q$ is a constant[2], the entropy takes the form

$$H(\theta) = -\log q + A(\theta) - \theta^T\nabla A(\theta), \tag{3.4}$$

and therefore we can omit the constant term $-\log q$ for simplicity.

When new measurements arrive, the parameter θ is updated to incorporate this new information. A set of these parameters might depend only weakly on the measurements (e.g., dependence on the size of measurements), while the remaining set (of parameters) might depend strongly on the measurements (e.g., when it is a function of the measurement values). We denote the set of parameters that depend weakly on the measurements with $\theta_{\mathcal{I}}$, while the second subset that depends strongly with $\theta_{\mathcal{D}}$. Therefore, parameter θ can be expressed as $\theta = \begin{bmatrix} \theta_{\mathcal{I}} \\ \theta_{\mathcal{D}} \end{bmatrix}$. Now, we will define more formally parameters $\theta_{\mathcal{I}}$ and $\theta_{\mathcal{D}}$. Suppose a measurement Y is linked to latent variable X as follows

$$p_{\mathsf{Y}|\mathsf{X}}(y \mid x;\eta) = \psi(x,y,\eta),$$

where η is a set of parameters linking the two random variables. The posterior distribution of X given Y is given by

$$\begin{aligned} p_{\mathsf{X}|\mathsf{Y}}(x \mid y;\eta) &\propto p_{\mathsf{Y}|\mathsf{X}}(y \mid x;\eta)p_{\mathsf{X}}(x;\theta_{\mathcal{I}},\theta_{\mathcal{D}})\psi(x,y,\eta) \\ &= p_{\mathsf{X}}(x;\theta'_{\mathcal{I}},\theta'_{\mathcal{D}}), \end{aligned}$$

where $\theta'_{\mathcal{I}} = f(\theta_{\mathcal{I}},\eta), \theta'_{\mathcal{D}} = g(\theta_{\mathcal{D}},y)$. In other words, $\theta_{\mathcal{I}}$ depends only weakly on the measurements through the parameters η that link Y with X, while $\theta_{\mathcal{D}}$ depends on the actual measurement values y. To give a concrete example about the meaning of η, measurement Y in Gaussian models is usually represented as

$$\mathsf{Y} = C\mathsf{X} + \mathsf{W},$$

where $\mathsf{W} \sim \mathcal{N}(w;0,R)$ and $\mathsf{W} \perp\!\!\!\perp \mathsf{X}$. In that case, $\eta = (C,R)$.

We are interested in characterizing the conditions under which the entropy has dependence only on $\theta_{\mathcal{I}}$, since in that case the reward function would be independent of the actual measurement values resulting in the open-loop and closed-loop policies being equivalent.

Theorem 3.1.1 (Necessary and sufficient conditions for independence of entropy on the value of measurements)**.** *Given a distribution in the form of an exponential family*

$$p(x;\theta) = q(x)\exp\left(\theta^T\phi(x) - A(\theta)\right),$$

[2]For example, log-base distribution is a constant for Gaussian distributions.

where $\log q(x) = \gamma^T \phi(x)$*, log-partition function* $A(\cdot)$ *twice continuously differentiable, and two sets of parameters* $\mathcal{I}, \mathcal{D}$ *that depend weakly or strongly on the measurement values, respectively, the following condition is necessary and sufficient*

$$(\theta_\mathcal{I} + \gamma_\mathcal{I})^T \nabla^2_{\theta_\mathcal{I}\theta_\mathcal{D}} A(\theta_\mathcal{I}, \theta_\mathcal{D}) + (\theta_\mathcal{D} + \gamma_\mathcal{D})^T \nabla^2_{\theta_\mathcal{D}\theta_\mathcal{D}} A(\theta_\mathcal{I}, \theta_\mathcal{D}) \equiv 0, \tag{3.5}$$

such that entropy is independent on the measurement values.

Proof. First, we will prove the "necessary" condition. We separate the parameter θ into two sets, $\theta_\mathcal{I}, \theta_\mathcal{D}$ as described above. Similarly, γ is also divided in $\gamma_\mathcal{I}, \gamma_\mathcal{D}$. Using the fact that $\theta = \begin{bmatrix} \theta_\mathcal{I} \\ \theta_\mathcal{D} \end{bmatrix}$, $\gamma = \begin{bmatrix} \gamma_\mathcal{I} \\ \gamma_\mathcal{D} \end{bmatrix}$, Eq. (3.3) becomes

$$H(\theta_\mathcal{I}, \theta_\mathcal{D}) = A(\theta_\mathcal{I}, \theta_\mathcal{D}) - \begin{bmatrix} \theta_\mathcal{I} + \gamma_\mathcal{I} \\ \theta_\mathcal{D} + \gamma_\mathcal{D} \end{bmatrix}^T \begin{bmatrix} \nabla_{\theta_\mathcal{I}} A(\theta_\mathcal{I}, \theta_\mathcal{D}) \\ \nabla_{\theta_\mathcal{D}} A(\theta_\mathcal{I}, \theta_\mathcal{D}) \end{bmatrix}. \tag{3.6}$$

We see from Eq. (3.6), that the entropy is a function of $\theta_\mathcal{I}, \theta_\mathcal{D}$. From these two sets of parameters, only $\theta_\mathcal{D}$ depends strongly on the measurement values. Hence, the entropy will be independent of the measurement values, if it does not depend on $\theta_\mathcal{D}$. Formalistically, this means that the first derivative of H with respect to $\theta_\mathcal{D}$ would be zero.

$$\nabla_{\theta_\mathcal{D}} H(\theta_\mathcal{I}, \theta_\mathcal{D}) \equiv 0. \tag{3.7}$$

Due to Eq. (3.6), Eq. (3.7) expands to

$$(\theta_\mathcal{I} + \gamma_\mathcal{I})^T \nabla^2_{\theta_\mathcal{I}\theta_\mathcal{D}} A(\theta_\mathcal{I}, \theta_\mathcal{D}) + (\theta_\mathcal{D} + \gamma_\mathcal{D})^T \nabla^2_{\theta_\mathcal{D}\theta_\mathcal{D}} A(\theta_\mathcal{I}, \theta_\mathcal{D}) \equiv 0. \tag{3.8}$$

Now, we will show the "sufficiency" condition. Assume Eq. (3.5) holds. If we take the derivative of entropy H with respect to $\theta_\mathcal{D}$, Eq. (3.6) gives

$$\nabla_{\theta_\mathcal{D}} H(\theta_\mathcal{I}, \theta_\mathcal{D}) = (\theta_\mathcal{I} + \gamma_\mathcal{I})^T \nabla^2_{\theta_\mathcal{I}\theta_\mathcal{D}} A(\theta_\mathcal{I}, \theta_\mathcal{D}) + (\theta_\mathcal{D} + \gamma_\mathcal{D})^T \nabla^2_{\theta_\mathcal{D}\theta_\mathcal{D}} A(\theta_\mathcal{I}, \theta_\mathcal{D}). \tag{3.9}$$

Combining Eqs. (3.8), (3.9), we obtain

$$\nabla_{\theta_\mathcal{D}} H(\theta_\mathcal{I}, \theta_\mathcal{D}) \equiv 0.$$

Therefore, entropy is independent of $\theta_\mathcal{D}$ and consequently of measurement values as well. ■

Corollary 3.1.1 (Constant log-base distribution)**.** *In case where log-base distribution* $\log q(\cdot)$ *is only a constant and exponential family can be expressed as*

$$p(x; \theta) \propto \exp\left(\theta^T \phi(x) - A(\theta)\right),$$

the necessary and sufficient condition for independency of entropy on the value of measurements reduces to

$$\theta_\mathcal{I}^T \nabla^2_{\theta_\mathcal{I}\theta_\mathcal{D}} A(\theta_\mathcal{I}, \theta_\mathcal{D}) + \theta_\mathcal{D}^T \nabla^2_{\theta_\mathcal{D}\theta_\mathcal{D}} A(\theta_\mathcal{I}, \theta_\mathcal{D}) \equiv 0. \tag{3.10}$$

■ 3.1.1 Gaussian Models

From Eq. (2.92), we saw that the entropy in Gaussian models is expressed as

$$H(\mathsf{X}) = \frac{d}{2}(1 + \log(2\pi)) - \frac{1}{2}\log|J|, \tag{3.11}$$

where J is the precision matrix of X.

If a measurement Y_i is given by

$$\mathsf{Y}_i = C_i\mathsf{X} + \mathsf{W}_i,$$

where $\mathsf{W}_i \sim \mathcal{N}(w_i; 0, R_i)$ and $\mathsf{W}_i \perp\!\!\!\perp \mathsf{X}$, then the posterior precision matrix of X after the incorporation of N measurements $\mathsf{Y}_1, \ldots, \mathsf{Y}_N$, becomes

$$J_{\mathsf{X}|\mathsf{Y}_1,\ldots,\mathsf{Y}_N} = J + \sum_{i=1}^{N} C_i^T R_i^{-1} C_i, \tag{3.12}$$

where J is the prior precision matrix of X. It becomes evident that the posterior entropy does not depend on the measurement values $\mathsf{Y}_i = y_i$, but rather the parameters that define the relationship between the measurements Y_i and the latent variable X. Consequently, as we see from Eq. (3.11), entropy will not depend on the measurement values either. Therefore, Eq. (3.10) should hold as a necessary condition. As a reminder, we saw in Ex. 2.8.4 that the log-partition function of the multivariate Gaussian is equal to

$$A(\theta) = -\frac{1}{4}\theta_1^T \Theta_2^{-1} \theta_1 - \frac{1}{2}\log|-2\Theta_2|, \tag{3.13}$$

where $\theta_1 = h$ and $\Theta_2 = -\frac{1}{2}J$.

If we incorporate N measurements $\mathsf{Y}_1, \ldots, \mathsf{Y}_N$, the posterior potential vector $h_{\mathsf{X}|\mathsf{Y}_1,\ldots,\mathsf{Y}_N}$ becomes

$$h_{\mathsf{X}|\mathsf{Y}_1,\ldots,\mathsf{Y}_N} = h + \sum_{i=1}^{N} C_i^T R_i^{-1} y_i, \tag{3.14}$$

where h is the prior potential vector. By observing Eqs. (3.12), (3.14), we see that only the potential vector h depends on the measurement values y_i. Since $\theta_1 = h$, we immediately see that θ_1 plays the role of $\theta_{\mathcal{D}}$, while Θ_2 plays the role of $\theta_{\mathcal{I}}$. Cor. 3.1.1 should be satisfied as a necessary condition, since the entropy in Gaussian models is independent on the measurement values.

Corollary 3.1.2 (Satisfiability of necessary condition for independency on the measurement values in Gaussian models)**.** *For a Gaussian variable $\mathsf{X} \sim \mathcal{N}^{-1}(x; h, J)$ that is expressed in the form of an exponential family as*

$$p(x; h, J) \propto \exp(h^T x - \frac{1}{2}x^T J x) = \exp(\langle\theta_1, x\rangle + \langle\!\langle\Theta_2, xx^T\rangle\!\rangle - A(\theta_1, \Theta_2)),$$

where $\theta_1 = h, \Theta_2 = -\frac{1}{2}J$ *and* $A(\theta) = A(\theta_1, \Theta_2) = -\frac{1}{4}\theta_1^T\Theta_2^{-1}\theta_1 - \frac{1}{2}\log|-2\Theta_2|$, *the following relationship holds*

$$\Theta_2^T\nabla^2_{\Theta_2\theta_1}A(\theta_1,\Theta_2) + \theta_1^T\nabla^2_{\theta_1\theta_1}A(\theta_1,\Theta_2) \equiv 0. \tag{3.15}$$

Proof. Since $\theta_{\mathcal{I}} = \Theta_2, \theta_{\mathcal{D}} = \theta_1$, the following expression can be written as

$$\theta_{\mathcal{I}}^T\nabla^2_{\theta_{\mathcal{I}}\theta_{\mathcal{D}}}A(\theta_{\mathcal{I}},\theta_{\mathcal{D}}) + \theta_{\mathcal{D}}^T\nabla^2_{\theta_{\mathcal{D}}\theta_{\mathcal{D}}}A(\theta_{\mathcal{I}},\theta_{\mathcal{D}}) = \Theta_2^T\nabla^2_{\Theta_2\theta_1}A(\theta_1,\Theta_2) + \theta_1^T\nabla^2_{\theta_1\theta_1}A(\theta_1,\Theta_2). \tag{3.16}$$

Since $A(\theta) = -\frac{1}{4}\theta_1^T\Theta_2^{-1}\theta_1 - \frac{1}{2}\log|-2\Theta_2|$, we have that

$$\nabla_{\theta_1}A(\theta) = -\frac{1}{2}\Theta_2^{-1}\theta_1.$$

Therefore,

$$\nabla^2_{\theta_1\theta_1}A(\theta) = -\frac{1}{2}\Theta_2^{-1}, \tag{3.17}$$

$$\nabla^2_{\Theta_2\theta_1}A(\theta) = \frac{1}{2}\Theta_2^{-T}\theta_1^T\Theta_2^{-T}, \tag{3.18}$$

where the last property is obtained from [83].

Since $\Theta_2 = -\frac{1}{2}J$ is symmetric, we have that $\Theta_2^T = \Theta_2, \Theta_2^{-T} = \Theta_2^{-1}$ and so $\nabla^2_{\Theta_2\theta_1}A(\theta)$ becomes

$$\nabla^2_{\Theta_2\theta_1}A(\theta) = \frac{1}{2}\Theta_2^{-1}\theta_1^T\Theta_2^{-1}. \tag{3.19}$$

Due to Eqs. (3.17), (3.19), the RHS of (3.16) becomes

$$\Theta_2^T\nabla^2_{\Theta_2\theta_1}A(\Theta_2,\theta_1) + \theta_1^T\nabla^2_{\theta_1\theta_1}A(\Theta_2,\theta_1) = \Theta_2^T(\frac{1}{2}\Theta_2^{-1}\theta_1^T\Theta_2^{-1}) + \theta_1^T(-\frac{1}{2}\Theta_2^{-1}) \equiv 0.$$

■

Unfortunately, the only known distribution whose entropy is independent of the measurement values is the Gaussian distribution.

■ 3.2 Bounds on submodular non-monotone functions

In this section, we are going to discuss theoretical guarantees for the batch and sequential selection settings when the reward function is submodular and non-monotone. Nemhauser et al. [77] showed that by using a slightly modified version of Alg. 2.1 to solve problem $\mathcal{O} \in \arg\max_{|\mathcal{S}|\leq k} f(\mathcal{S})$, when f is submodular and non-monotone, it holds that

$$f(\mathcal{G}) \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) f(\mathcal{O}) - k\left(1 - \frac{1}{k}\right)^k\theta, \tag{3.20}$$

where θ is a non-negative number defined as $f(u \mid \mathcal{V} \setminus \{u\}) \geq -\theta$. We observe from Eq. (3.20) that as θ grows, the bound becomes more pessimistic. In case where $\theta = 0$ (monotone case), we obtain the familiar $1-1/e$ that applies to the submodular monotone case. The only necessary modification to Alg. 2.1 for the case of non-monotone f is that once we find a greedy element g_j, such that $f(g_j \mid \mathcal{G}_{j-1}) \leq 0$, the algorithm terminates, since by submodularity any future elements would result in smaller incremental reward. The above algorithm also runs in $\mathcal{O}(kN)$ time.

More recently, Lee et al. [61] provided a $\frac{1}{(1+\epsilon)(R+2+1/R)}$ approximation algorithm for maximizing a non-negative non-monotone submodular function subject to R matroid constraints. For the batch selection problem, which corresponds to a uniform matroid or the sequential selection problem, which corresponds to a partition matroid, $R = 1$ and so the approximating ratio is $1/4$. In that respect, they present an algorithm suitable for a much more generic set of problems, but one of the issues is that it runs in $\mathcal{O}(\frac{1}{\epsilon}(R+1)N^4 \log N)$ time. Therefore, even when $\frac{1}{\epsilon}$ is at most polynomial in N, the resulting complexity can be prohibitive even for moderate-sized observation sets.

In this section, we will attempt to generalize Alg. 2.5 by Williams et al. [102] for the case of non-monotone rewards. In more detail, we are interested in finding an approximation algorithm for the following problem

$$\mathcal{O} \in \underset{|\mathcal{S} \cap \mathcal{V}_t|_{t=1}^T}{\arg\max} f(\mathcal{S}),$$

where f is a submodular function.[3] We will assume, w.l.o.g. that each set $\mathcal{V}_t$ has k_t copies of a "slack" measurement e such that $f(e \mid \mathcal{S}) = 0$ for any set $\mathcal{S}$. We can think of measurement e as representing the option to take no measurement, if the incremental reward of the greedily selected element is non-positive. The approach that we propose is outlined in Alg. 3.1.

Theorem 3.2.1 (Performance bounds in the sequential selection setting)**.** *If the greedy method described in Alg. 3.1 is applied to problem* $\max_{\{|\mathcal{S} \cap \mathcal{V}_t| \leq k_t, t=1,\ldots,T\}} f(\mathcal{S})$*, where* f *is a submodular function, the following bound holds*

$$f(\mathcal{G}) \geq \frac{1}{2}(f(\mathcal{O}) - M\theta), \tag{3.21}$$

where $M = \sum_{t=1}^T k_t$ *and* θ *is a non-negative number such that* $f(u \mid \mathcal{V} \setminus \{u\}) \geq -\theta$*, for any* $u \in \mathcal{V} = \cup_{t=1}^T \mathcal{V}_t$*.*

Proof. Let us denote the optimal and greedy solutions by $\mathcal{O} = \{o_1, \ldots, o_M\}, \mathcal{G} = \{g_1, \ldots, g_M\}$, respectively. The optimal and greedy solutions have the same length, that is, $M = \sum_{t=1}^T k_t$, since we can always add as many copies of the "slack" measurement e, where $f(e \mid \mathcal{S}) = 0$ for any set $\mathcal{S}$ to both solutions such that they both attain size M. We consider the function $f'(\mathcal{S}) = f(\mathcal{S}) + |\mathcal{S}|\theta$. One can easily show that the

[3] As a reminder, Williams et al. [102] assumed that f is submodular monotone.

Algorithm 3.1 Sequential Selection Greedy Heuristic for Non-Monotone Functions

Initialization
Set $\mathcal{G}_0 = \emptyset$.
Define a visit walk $\boldsymbol{w} = \{w_1, \ldots, w_M\}$ such that $\sum_{j=1}^{M} \mathbb{1}(w_j = t) = k_t, \forall t$.

Iteration
for j in $1 : M$ **do**
  Select g_j s.t. $g_j \in \arg\max_{u \in \mathcal{V}_{w_j}} f(u \mid \mathcal{G}_{j-1})$.
  if $f(g_j \mid \mathcal{G}_{j-1}) > 0$ **then**
    Set $\mathcal{G}_j = \mathcal{G}_{j-1} \cup \{g_j\}$.
  else
    Set $\mathcal{G}_j = \mathcal{G}_{j-1} \cup \{e\}$.
  end if
  Set $\mathcal{V}_{w_j} = \mathcal{V}_{w_j} \setminus \{g_j\}$.
end for
$\mathcal{G} = \mathcal{G}_M$ is the greedy solution.

function retains submodularity. That is, for any measurement $u \notin \mathcal{B}$ and $\mathcal{A} \subseteq \mathcal{B}$, it holds that

$$f'(u \mid \mathcal{A}) = f(u \mid \mathcal{A}) + \theta \overset{(a)}{\geq} f(u \mid \mathcal{B}) + \theta = f'(u \mid \mathcal{B}),$$

where (a) holds by submodularity of f.

Interestingly, f' is also monotone. For any $u \notin \mathcal{S}$, we have

$$f'(u \mid \mathcal{S}) = f(u \mid \mathcal{S}) + \theta \geq 0,$$

where the last inequality holds since $f(u \mid \mathcal{S}) \geq f(u \mid \mathcal{V} \setminus \{u\}) \geq -\theta$.

Since f' is submodular and monotone, the bound by Williams et al. [102] applies (to f')

$$f'(\mathcal{G}) \geq \frac{1}{2} f'(\mathcal{O}) \Leftrightarrow f(\mathcal{G}) + |\mathcal{G}|\theta \geq \frac{1}{2}(f(\mathcal{O}) + |\mathcal{O}|\theta) \Leftrightarrow f(\mathcal{G}) + M\theta \geq \frac{1}{2}(f(\mathcal{O}) + M\theta)$$
$$f(\mathcal{G}) \geq \frac{1}{2}(f(\mathcal{O}) - M\theta).$$

■

It is worth mentioning that the bound in Thm. 3.2.1 is a pessimistic one, since it depends on the maximum negative reward for any measurement u, $-\theta$. As θ grows, the bound becomes looser and does not have any practical implications for really large values of θ or $k_t, \forall t$. However, for smaller values of θ, the bound basically says that the value of the submodular reward f for the greedy solution $\mathcal{G}$ cannot be worse than 50% of the optimal solution minus some quantity, which has to do with the worst (negative) contribution that a measurement can have in the existing set of chosen measurements. In fact, for $\theta = 0$ (monotone case), we have that $f(\mathcal{G}) \geq \frac{1}{2} f(\mathcal{O})$, which is the familiar bound by Williams et al. [102].

■ 3.2.1 Penalized Mutual Information

In information planning, acquiring measurements usually induces costs. Costs may be generated from a variety of different sources depending on the problem of interest. For example, costs can represent processing or communication costs. One may also be subject to limited budget b in many settings, under which measurements can be selected. This budget can be related, for example, to limited bandwidth of the channel through which measurements are transmitted. However, solely relying on MI as the reward function does not account for heterogeneous measurement costs. It would be better to just motivate the idea that accounting for heterogeneous measurement costs necessitate consideration of an alternative reward that incorporates both value of information and the cost of acquiring that information. We define the *penalized mutual information (PMI)* for this purpose as

Definition 3.2.1 (Penalized Mutual Information)**.**

$$f(\mathcal{A}) = I(\mathsf{X}; \mathsf{Y}_{\mathcal{A}}) - \lambda c(\mathcal{A}),$$

where $\lambda \geq 0$ is a regularization parameter transforming cost to information units.

The incremental reward of PMI is defined as

$$f(u \mid \mathcal{A}) = I(\mathsf{X}; \mathsf{Y}_u \mid \mathsf{Y}_{\mathcal{A}}) - \lambda c(u).$$

Lemma 3.2.1. *The penalized mutual information $f(\mathcal{A}) = I(\mathsf{X}; \mathsf{Y}_{\mathcal{A}}) - \lambda c(\mathcal{A})$ is a submodular function.*

Proof. For every $\mathcal{A}, \mathcal{B} \subseteq \mathcal{V}$ such that $\mathcal{A} \subseteq \mathcal{B}$, we have

$$\begin{gathered} I(\mathsf{X}; \mathsf{Y}_u \mid \mathsf{Y}_{\mathcal{A}}) \geq I(\mathsf{X}; \mathsf{Y}_u \mid \mathsf{Y}_{\mathcal{B}}) \\ I(\mathsf{X}; \mathsf{Y}_u \mid \mathsf{Y}_{\mathcal{A}}) - \lambda c(u) \geq I(\mathsf{X}; \mathsf{Y}_u \mid \mathsf{Y}_{\mathcal{B}}) - \lambda c(u) \\ f(u \mid \mathcal{A}) \geq f(u \mid \mathcal{B}). \end{gathered}$$

■

We can easily show that PMI is not monotone. A common selection scheme would be to select the measurement with the maximum incremental reward as it is dictated by standard greedy approaches we revised in the previous chapter. The issue with this approach, however, is that the plan might end up adopting measurements, which have negative incremental reward and thus reduce the total reward. A more natural approach is to be able to opt out from selecting a measurement if this measurement has non-positive incremental reward. We formulate this as

$$g_j = \begin{cases} \arg\max_{u \in \mathcal{V}_{w_j}} f(u \mid \mathcal{G}_{j-1}) & , f(u \mid \mathcal{G}_{j-1}) > 0 \\ \emptyset & , \text{otherwise.} \end{cases} \tag{3.22}$$

The above greedy rule, with slight abuse of notation, can be written as

$$g_j \in \arg\max_{u \in \mathcal{V}_{w_j}} \{f(u \mid \mathcal{G}_{j-1}), 0\}. \tag{3.23}$$

The problem with this reward though is that it is neither monotone nor submodular, therefore none of the previously presented bounds apply. To overcome this limitation, we can introduce k_t copies of a "slack" measurement e in each observation set $\mathcal{V}_t$ such that $f(e \mid \mathcal{S}) = 0$ for any set $\mathcal{S}$. This "slack" measurement plays the role of not selecting a measurement at a greedy step, if the greedily selected measurement results in non-positive incremental reward. Therefore, by choosing the submodular function $f(\mathcal{A}) = I(X; Y_\mathcal{A}) - \lambda c(\mathcal{A})$ as the reward and using the "slack" measurements to represent the option of not selecting a measurement, we can apply Alg. 3.1. Therefore, the bounds presented in Thm. 3.2.1 apply.

■ 3.3 Bounds on Varying Costs

In the previous section, we considered fixed costs for measurements. A more interesting scenario is when measurement costs change proportionally to the relative informational value they carry. From the perspective of an information consumer, one might be willing to obtain a measurement Y_u with a cost which is analogous to the relative information that the measurement conveys with respect to a specific consumer. We denote the existing knowledge of information consumer by $\mathcal{A}$. We can formulate the above as

$$c(u \mid \mathcal{A}) = r_u I(X; Y_u \mid Y_\mathcal{A}), \tag{3.24}$$

where $r_u \geq 0$ is a parameter related to the informational value of measurement u and $I(X; Y_u \mid Y_\mathcal{A})$ captures the relative informational value of measurement u based on the consumer's current knowledge as depicted by $\mathcal{A}$. When $\mathcal{A} = \emptyset$, we have $c(u) = c(u \mid \emptyset) = r_u I(X; Y_u)$. It would be natural to assume that an information consumer is only willing to incur a cost proportional to the informational utility of the measurement. Since this depends on $\mathcal{A}$ and $\mathcal{A}$ varies, the cost that one is willing to incur for a particular measurement will also vary and in fact decrease as $\mathcal{A}$ gets larger. This is captured from the fact that $\forall \mathcal{A} \subseteq \mathcal{B}$,

$$c(u \mid \mathcal{B}) = r_u I(X; Y_u \mid Y_\mathcal{B}) \leq r_u I(X; Y_u \mid Y_\mathcal{A}) = c(u \mid \mathcal{A}), \tag{3.25}$$

where the last inequality holds from the submodularity of MI. The inequality (3.25) also indicates that the cost function (as we define it in (3.24)) is submodular.

We can generalize Eq. (3.24), which considers the relative cost of a single measurement u, to sets of measurements $\mathcal{S}$. The relative cost of a set of measurements $\mathcal{S}$ given knowledge of $\mathcal{A}$ would be similarly defined as

$$c(\mathcal{S} \mid \mathcal{A}) = r_\mathcal{S} I(X; Y_\mathcal{S} \mid Y_\mathcal{A}). \tag{3.26}$$

The PMI is defined in the same way for the case of varying costs as well. That is,

$$f(\mathcal{A}) = I(X; Y_{\mathcal{A}}) - \lambda c(\mathcal{A})$$
$$f(u \mid \mathcal{A}) = I(X; Y_u \mid Y_{\mathcal{A}}) - \lambda c(u \mid \mathcal{A}),$$

where $c(u \mid \mathcal{A})$ is defined in Eq. (3.24).

Lemma 3.3.1. *There cannot be a set function $c(\cdot \mid \cdot)$ satisfying (3.26) with non-equal r.*

Proof. Consider the reward of set $\mathcal{S} \cup \{u\}$

$$f(\mathcal{S} \cup \{u\}) \equiv f(\mathcal{S} \cup \{u\})$$
$$\underbrace{f(\mathcal{S} \cup \{u\}) - f(\mathcal{S})}_{f(u|\mathcal{S})} + \underbrace{f(\mathcal{S}) - f(\emptyset)}_{f(\mathcal{S}|\emptyset)} = \underbrace{f(\mathcal{S} \cup \{u\}) - f(u)}_{f(\mathcal{S}|u)} + \underbrace{f(u) - f(\emptyset)}_{f(u|\emptyset)}$$
$$I(X; Y_u \mid Y_{\mathcal{S}}) - \lambda c(u \mid \mathcal{S}) + I(X; Y_{\mathcal{S}}) - \lambda c(\mathcal{S}) = I(X; Y_{\mathcal{S}} \mid Y_u) - \lambda c(\mathcal{S} \mid u) + I(X; Y_u) - \lambda c(u)$$
$$I(X; Y_u \mid Y_{\mathcal{S}}) + I(X; Y_{\mathcal{S}}) - \lambda(c(u \mid \mathcal{S}) + c(\mathcal{S})) = I(X; Y_{\mathcal{S}} \mid Y_u) + I(X; Y_u) - \lambda(c(\mathcal{S} \mid u) + c(u))$$
$$I(X; Y_u, Y_{\mathcal{S}}) - \lambda(c(u \mid \mathcal{S}) + c(\mathcal{S})) = I(X; Y_u, Y_{\mathcal{S}}) - \lambda(c(\mathcal{S} \mid u) + c(u)). \tag{3.27}$$

From Eq. (3.27), we immediately see that

$$c(u \mid \mathcal{S}) + c(\mathcal{S}) = c(\mathcal{S} \mid u) + c(u). \tag{3.28}$$

Using the fact that $c(u \mid \mathcal{S}) = r_u I(X; Y_u \mid Y_{\mathcal{S}})$ and $c(\mathcal{S} \mid u) = r_{\mathcal{S}} I(X; Y_{\mathcal{S}} \mid Y_u)$, Eq. (3.28) becomes

$$r_u I(X; Y_u \mid Y_{\mathcal{S}}) + r_{\mathcal{S}} I(X; Y_{\mathcal{S}}) = r_{\mathcal{S}} I(X; Y_{\mathcal{S}} \mid Y_u) + r_u I(X; Y_u). \tag{3.29}$$

Now, if we add terms $r_u I(X; Y_{\mathcal{S}}) + r_{\mathcal{S}} I(X; Y_u)$ in both sides of Eq. (3.29), we obtain

$$r_u I(X; Y_u, Y_{\mathcal{S}}) + r_{\mathcal{S}}(I(X; Y_{\mathcal{S}}) + I(X; Y_u)) = r_{\mathcal{S}} I(X; Y_u, Y_{\mathcal{S}}) + r_u(I(X; Y_{\mathcal{S}}) + I(X; Y_u))$$
$$(r_u - r_{\mathcal{S}}) I(X; Y_u, Y_{\mathcal{S}}) = (r_u - r_{\mathcal{S}})(I(X; Y_{\mathcal{S}}) + I(X; Y_u)).$$

Suppose $r_u \neq r_{\mathcal{S}}$, then we would have

$$I(X; Y_u, Y_{\mathcal{S}}) = I(X; Y_{\mathcal{S}}) + I(X; Y_u), \forall \mathcal{S} \subseteq \mathcal{V}, u \notin \mathcal{S}. \tag{3.30}$$

Due to submodularity of MI, we know that

$$I(X; Y_u) \geq I(X; Y_u \mid Y_{\mathcal{S}}) \Leftrightarrow$$
$$I(X; Y_u) + I(X; Y_{\mathcal{S}}) \geq I(X; Y_u \mid Y_{\mathcal{S}}) + I(X; Y_{\mathcal{S}}) = I(X; Y_u, Y_{\mathcal{S}}),$$

which contradicts (3.30). Eq. (3.30) holds only in the special case when $X, Y_{\mathcal{S}}, Y_u$ are independent. So, it must be that: $r_u = r_{\mathcal{S}} = r, \forall \mathcal{S} \subseteq \mathcal{V}, u \notin \mathcal{S}$. ∎

Since, $f(u \mid \mathcal{A}) = (1 - \lambda r) I(X; Y_u \mid Y_{\mathcal{A}})$, f is trivially submodular. For it to be monotone as well, we need to have $r \leq \frac{1}{\lambda}$. In this case

$$f(u \mid \mathcal{A}) = I(X; Y_u \mid Y_{\mathcal{A}}) - \lambda c(u \mid \mathcal{A}) = (1 - \lambda r) I(X; Y_u \mid Y_{\mathcal{A}}) \geq 0.$$

When $r \leq \frac{1}{\lambda}$, the known bounds presented in Sec. 2.6 for the submodular monotone case apply in the case of varying costs as well.

■ 3.4 Bounds on Submodular Knapsack Maximization

In Sec. 2.6, we explored approximation algorithms whose performance is nearly optimal compared to the optimal solution. We focused on two problems; the selection and budgeted setting. In the first one, we are interested in determining the measurements that maximize some specified reward under a constraint on the number of measurements that can be selected. In the second problem, each measurement is assigned a cost and we are interested in choosing the measurements that maximize a reward function subject to a budget constraint. The latter problem is also known as *submodular knapsack maximization (SKM)*. In this section, we revisit the SKM problem and determine upper bounds for the optimal solution by reducing the problem to an unconstrained submodular maximization (USM) one and using the bounds provided by Buchbinder et al. [13]. Even though previous methods have addressed this problem, they either propose algorithms whose complexity include high order terms of the observation set size, N or derive loose upper bounds for the optimal [52, 90]. In more detail, Sviridenko [90] proposes a heuristic that runs in $\mathcal{O}(N^5)$ time and provides a $1-1/e$ approximating ratio to the optimal, while Krause and Guestrin [52] present a method that runs in $\mathcal{O}(N^2)$ time with an approximating ratio of $(1 - 1/\sqrt{e}) \approx 0.394$. The first method can be impractical even for problems with moderately-sized observation sets, while the second one provides a loose upper bound for the optimal. In other words, if we denote by $\mathcal{G}$ the greedy solution generated by [52], the optimal solution cannot exceed $\frac{1}{1-1/\sqrt{e}} \approx 2.542$ times the greedy one.

Here, we will provide an upper bound for the optimal solution of the SKM problem by working on the dual domain and taking advantage of the algorithm by Buchbinder et al. [13], which runs in linear time with respect to the observation set size, N. In more detail, we consider the problem

$$\mathcal{O} \in \arg\max_{c(\mathcal{S}) \leq b} f(\mathcal{S}),$$

where f is monotone, submodular and c is a modular function. That is, $c(\mathcal{S}) = \sum_{u \in \mathcal{S}} c(u)$. We are interested in providing upper bounds for $f(\mathcal{O})$. Let us additionally, denote by h the penalized form of f as

$$h(\mathcal{S}; \lambda) = f(\mathcal{S}) - \lambda c(\mathcal{S}),$$

where λ is a non-negative penalty parameter. The larger λ is, the bigger role the cost (of a measurement) has in the designation of a measurement schedule. It can be

trivially shown that h is submodular but non-monotone. For a particular value of λ, the algorithm by Buchbinder et al. [13] provides a solution $\mathcal{G}_{\texttt{USM}}$ in $\mathcal{O}(N)$ time which is on average at least 50% close to the optimal, $\mathbb{E}[h(\mathcal{G}_{\texttt{USM}})] \geq 0.5 \max_{\mathcal{S}\subseteq\mathcal{V}} h(\mathcal{S};\lambda)$.

Theorem 3.4.1 (Upper bound for SKM)**.** *If we denote by $\mathcal{O}$ the optimal solution to the SKM problem*

$$\mathcal{O} \in \arg\max_{c(\mathcal{S})\leq b} f(\mathcal{S}),$$

where $c(\mathcal{S}) = \sum_{u\in\mathcal{S}} c(u)$, the value of the optimal solution is upper bounded by

$$f(\mathcal{O}) \leq \min_{\lambda\geq 0} \lambda b + 2\mathbb{E}[h(\mathcal{G}_{\texttt{USM}};\lambda)], \tag{3.31}$$

where $h(\mathcal{S};\lambda) = f(\mathcal{S}) - \lambda c(\mathcal{S})$, $b > 0$ and $\mathcal{G}_{\texttt{USM}}$ is the randomized greedy solution for the problem $\max_{\mathcal{S}\subseteq\mathcal{V}} h(\mathcal{S};\lambda)$ generated by Alg. 2.6.

Proof. By Lovász extension [65], we can transform the combinatorial problem to a convex optimization problem as

$$\max_{c(\mathcal{S})\leq b} f(\mathcal{S}) = \max_{x_i\in\{0,1\}} \min_{\lambda\geq 0} \; f_c(x) - \lambda(\sum_{i=1}^{n} c_i x_i - b) = \max_{x_i\in\{0,1\}} \min_{\lambda\geq 0} \; f_c(x) - \lambda(c^T x - b),$$

where f_c is the convex extension of submodular function f and $x_i \in \{0,1\}$ is an indicator variable determining whether item (measurement) i will be in the final solution.

Let us denote by $L(x,\lambda) = f_c(x) - \lambda(c^T x - b)$ the Lagrange function of the above problem. From the minimax inequality, we have that for any function L

$$\max_{x_i\in\{0,1\}} \min_{\lambda\geq 0} \; L(x,\lambda) \leq \min_{\lambda\geq 0} \max_{x_i\in\{0,1\}} \; L(x,\lambda)$$
$$\max_{x_i\in\{0,1\}} \min_{\lambda\geq 0} \; f_c(x) - \lambda(c^T x - b) \leq \min_{\lambda\geq 0} \max_{x_i\in\{0,1\}} \; f_c(x) - \lambda(c^T x - b).$$

Since $f_c, c^T x - b$ are convex with respect to x and there is a feasible solution, which is $x = 0$, belonging to the relative interior of $f_c, c^T x - b$, then Slater's condition holds and the above inequality becomes tight (*strong duality*).

Namely, we have

$$\max_{x_i\in\{0,1\}} \min_{\lambda\geq 0} \; f_c(x) - \lambda(c^T x - b) = \min_{\lambda\geq 0} \max_{x_i\in\{0,1\}} \; f_c(x) - \lambda(c^T x - b). \tag{3.32}$$

Therefore, we have

$$\max_{c(\mathcal{S})\leq b} f(\mathcal{S}) = \min_{\lambda\geq 0} \max_{x_i\in\{0,1\}} \; f_c(x) - \lambda(c^T x - b) = \min_{\lambda\geq 0} \lambda b + \max_{x_i\in\{0,1\}} \; f_c(x) - \lambda c^T x. \tag{3.33}$$

The last term $\max_{x_i\in\{0,1\}} \; f_c(x) - \lambda c^T x$ is equivalent to $\max_{\mathcal{S}} \; \underbrace{f(\mathcal{S}) - \lambda c(\mathcal{S})}_{h(\mathcal{S};\lambda)}$. If we denote by $\mathcal{O}_{\texttt{USM}} \in \arg\max_{\mathcal{S}\subseteq\mathcal{V}} h(\mathcal{S})$ the optimal solution of this unconstrained submodular

maximization problem, the application of Alg. 2.6 by Buchbinder et al. [13] generates a greedy solution $\mathcal{G}_{\texttt{USM}}$, which is on average at least 50% close to $h(\mathcal{O}_{\texttt{USM}}; \lambda)$:

$$\mathbb{E}[h(\mathcal{G}_{\texttt{USM}}; \lambda)] \geq \frac{1}{2} h(\mathcal{O}_{\texttt{USM}}; \lambda) \Rightarrow h(\mathcal{O}_{\texttt{USM}}; \lambda) = \max_{\mathcal{S} \subseteq \mathcal{V}} h(\mathcal{S}; \lambda) \leq 2\mathbb{E}[h(\mathcal{G}_{\texttt{USM}}; \lambda)]. \tag{3.34}$$

Due to Eq. (3.34), Eq. (3.33) becomes

$$f(\mathcal{O}) = \max_{c(\mathcal{S}) \leq b} f(\mathcal{S}) = \min_{\lambda \geq 0} \lambda b + \max_{\mathcal{S} \subseteq \mathcal{V}} h(\mathcal{S}; \lambda) \leq \min_{\lambda \geq 0} \lambda b + 2\mathbb{E}[h(\mathcal{G}_{\texttt{USM}}; \lambda)].$$

■

Finding a $\mathcal{G}_{\texttt{USM}}$ for a specific λ takes $\mathcal{O}(N)$ time. We can upper-bound $f(\mathcal{O})$ by discretizing λ to N values and choosing the λ that corresponds to the minimum upper bound. The total running time for determining an upper-bound would be $\mathcal{O}(N^2)$.

■ 3.5 Bounds for Focused Planning

It is often the case that only a specific set of latent variables is of interest. We will denote this set by $\mathcal{R}$ and call it *relevant* set. The remaining variables of the latent graph act as nuisances and are often used for modeling purposes. If we are interested in designing a plan with respect only to this relevant set $\mathcal{R}$, one solution is to marginalize the irrelevant latent variables. The issue with this approach is that marginalization would result in very expensive computations, since it would introduce direct dependencies between variables. Namely, every variable that is marginalized, introduces a clique between all the variables that directly connects to. Therefore, marginalization would result in a clique comprised of all the variables that are directly connected to the marginalized variables, thus making inference much harder. Another issue which is specific to information planning when the reward of choice is MI is that the conditional independency of measurements given $\mathsf{X}_{\mathcal{R}}$ may no longer hold and, as such, preclude the use of submodular analysis and associated bounds. In this section, we will propose an approximating method for selecting informative measurements with respect to the relevant latent variables $\mathsf{X}_{\mathcal{R}}$, that under mild assumptions provides lower bounds to the optimal.

In more detail, we will consider the following problem

$$\max_{|\mathcal{S}| \leq k} I(\mathsf{X}_{\mathcal{R}}; \mathsf{Y}_{\mathcal{S}}). \tag{3.35}$$

In Fig. 3.1, we see a graphical model with T observation sets and T latent variables $\mathsf{X}_1, \ldots, \mathsf{X}_T$. The relevant set $\mathcal{R}$ is the one filled with light green color. Unfortunately, if we apply Alg. 2.1 in problem (3.35), there are no guarantees that the greedy solution would be close to the optimal, since $I(\mathsf{X}_{\mathcal{R}}; \mathsf{Y}_{\mathcal{S}})$ is monotone but not submodular anymore. As a reminder, function $f(\mathcal{S}) = I(\mathsf{X}_{\mathcal{R}}; \mathsf{Y}_{\mathcal{S}})$ would be submodular, if $\mathsf{Y}_i \perp\!\!\!\perp \mathsf{Y}_j \mid \mathsf{X}_{\mathcal{R}}, \forall i \neq j \in \mathcal{V}$. One key idea introduced by Levine and How [62, 63]

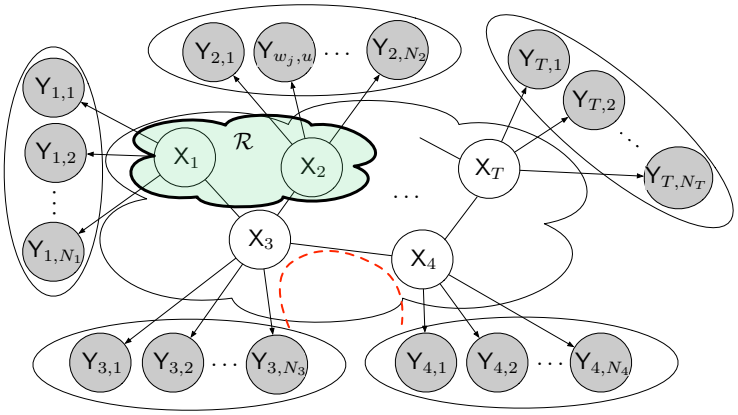


Figure 3.1: Focused inference setting. In this example, latent nodes 1 and 2 form the relevant set $\mathcal{R}$, which is depicted by light green color. There is at least one pair of measurements Y_i, Y_j such that $Y_i \not\perp\!\!\!\perp Y_j \mid X_\mathcal{R}$. For example, the red dashed line outlines a path between all measurements of observation set $\mathcal{V}_3$ and all measurements of observation set $\mathcal{V}_4$.

is the notion of extended set $\hat{\mathcal{R}}$ of $\mathcal{R}$. Extended set $\hat{\mathcal{R}}$ is a superset of $\mathcal{R}$ such that $Y_i \perp\!\!\!\perp Y_j \mid X_{\hat{\mathcal{R}}}, \forall i \neq j \in \mathcal{V}$. In this case, the bounds for the selection problem presented in Sec. 2.6 would hold for the function $I(X_{\hat{\mathcal{R}}}; Y_\mathcal{S})$. We will show below, how we can generalize the bounds to the reward of interest $I(X_\mathcal{R}; Y_\mathcal{S})$. As in Alg. 2.1, we select the measurement g_j at each step such that

$$g_j \in \underset{u \in \mathcal{V} \setminus \mathcal{G}_{j-1}}{\arg\max}\, I(X_\mathcal{R}; Y_u \mid Y_{\mathcal{G}_{j-1}}) \tag{3.36}$$

Theorem 3.5.1 (Relation between greedy and optimal solution in the focused setting)**.** *Let us denote by $\mathcal{R}$ the relevant set, and by $\hat{\mathcal{R}}$ the extended set, that is, the set such that $Y_i \perp\!\!\!\perp Y_j \mid X_{\hat{\mathcal{R}}}, \forall i \neq j \in \mathcal{V}$. If the greedy method described in Alg. 2.1 is applied to problem $\max_{|\mathcal{S}| \leq k} f(\mathcal{S})$, where $f(\mathcal{S}) = I(X_\mathcal{R}; Y_\mathcal{S})$, then the greedy solution is related to the optimal solution as follows*

$$f(\mathcal{G}) = I(X_\mathcal{R}; Y_\mathcal{G}) \geq f(\mathcal{O}) \left(1 - \left(1 - \frac{1}{k}\right)^k\right) - \sum_{j=1}^{k} f_j^U \left(1 - \frac{1}{k}\right)^{k-j}, \tag{3.37}$$

where $f_j^U = \max_{u \in \mathcal{V} \setminus \mathcal{G}_{j-1}} I(X_{\hat{\mathcal{R}} \setminus \mathcal{R}}; Y_u \mid X_\mathcal{R}, Y_{\mathcal{G}_{j-1}})$.

Proof. Let us denote by $\mathcal{O}$ the optimal solution for the problem $\max_{|\mathcal{S}| \leq k} I(X_\mathcal{R}; Y_\mathcal{S})$

and by $\mathcal{G}$ the greedy solution of Alg. 2.1 under the reward $f(\mathcal{S}) = I(X_\mathcal{R}; Y_\mathcal{S})$. We have

$$\begin{aligned}
I(X_\mathcal{R}; Y_\mathcal{O}) &\overset{(a)}{\leq} I(X_\mathcal{R}; Y_\mathcal{O}, Y_{\mathcal{G}_{j-1}}) = I(X_\mathcal{R}; Y_{\mathcal{G}_{j-1}}) + I(X_\mathcal{R}; Y_{\mathcal{O}\setminus\mathcal{G}_{j-1}} \mid Y_{\mathcal{G}_{j-1}}) \\
&\overset{(b)}{=} \sum_{i=1}^{j-1} I(X_\mathcal{R}; Y_{g_i} \mid Y_{\mathcal{G}_{i-1}}) + I(X_\mathcal{R}; Y_{\mathcal{O}\setminus\mathcal{G}_{j-1}} \mid Y_{\mathcal{G}_{j-1}}) \\
&\overset{(c)}{\leq} \sum_{i=1}^{j-1} I(X_\mathcal{R}; Y_{g_i} \mid Y_{\mathcal{G}_{i-1}}) + I(X_{\hat{\mathcal{R}}}; Y_{\mathcal{O}\setminus\mathcal{G}_{j-1}} \mid Y_{\mathcal{G}_{j-1}}) \\
&\overset{(d)}{\leq} \sum_{i=1}^{j-1} I(X_\mathcal{R}; Y_{g_i} \mid Y_{\mathcal{G}_{i-1}}) + \sum_{u\in\mathcal{O}\setminus\mathcal{G}_{j-1}} I(X_{\hat{\mathcal{R}}}; Y_u \mid Y_{\mathcal{G}_{j-1}}) \\
&= \sum_{i=1}^{j-1} I(X_\mathcal{R}; Y_{g_i} \mid Y_{\mathcal{G}_{i-1}}) + \sum_{u\in\mathcal{O}\setminus\mathcal{G}_{j-1}} (I(X_{\hat{\mathcal{R}}\setminus\mathcal{R}}; Y_u \mid Y_{\mathcal{G}_{j-1}}) + I(X_\mathcal{R}; Y_u \mid Y_{\mathcal{G}_{j-1}})) \\
&\overset{(e)}{\leq} \sum_{i=1}^{j-1} I(X_\mathcal{R}; Y_{g_i} \mid Y_{\mathcal{G}_{i-1}}) + k \max_{u\in\mathcal{O}\setminus\mathcal{G}_{j-1}} I(X_{\hat{\mathcal{R}}\setminus\mathcal{R}}; Y_u \mid X_\mathcal{R}, Y_{\mathcal{G}_{j-1}}) \\
&\qquad + k \max_{u\in\mathcal{V}\setminus\mathcal{G}_{j-1}} I(X_\mathcal{R}; Y_u \mid Y_{\mathcal{G}_{j-1}}),
\end{aligned} \tag{3.38}$$

where (a) is due to monotonicity of MI, (b) due to the chain rule of MI, (c) due to the monotonicity of MI ($\hat{\mathcal{R}} \supseteq \mathcal{R}$) and (d) due to submodularity of MI with respect to extended set $\hat{\mathcal{R}}$. As a reminder, submodularity does not hold for the function $I(X_\mathcal{R}; Y_\mathcal{S})$, but holds for $I(X_{\hat{\mathcal{R}}}; Y_\mathcal{S})$ since all measurements are conditionally independent given $X_{\hat{\mathcal{R}}}$. The last inequality in (e) holds because we expanded the constraint set of $I(X_\mathcal{R}; Y_u \mid Y_{\mathcal{G}_{j-1}})$ from $\mathcal{O}\setminus\mathcal{G}_{j-1}$ to $\mathcal{V}\setminus\mathcal{G}_{j-1}$.

Let us denote by o_j the element that maximizes $o_j \in \arg\max_{u\in\mathcal{O}\setminus\mathcal{G}_{j-1}} I(X_{\hat{\mathcal{R}}\setminus\mathcal{R}}; Y_u \mid X_\mathcal{R}, Y_{\mathcal{G}_{j-1}})$. From the greedy algorithm, we also have that $g_j \in \arg\max_{u\in\mathcal{V}\setminus\mathcal{G}_{j-1}} I(X_\mathcal{R}; Y_u \mid Y_{\mathcal{G}_{j-1}})$. Therefore, Eq. (3.38) becomes

$$I(X_\mathcal{R}; Y_\mathcal{O}) \leq \sum_{i=1}^{j-1} I(X_\mathcal{R}; Y_{g_i} \mid Y_{\mathcal{G}_{i-1}}) + kI(X_{\hat{\mathcal{R}}\setminus\mathcal{R}}; Y_{o_j} \mid X_\mathcal{R}, Y_{\mathcal{G}_{j-1}}) + kI(X_\mathcal{R}; Y_{g_j} \mid Y_{\mathcal{G}_{j-1}}). \tag{3.39}$$

Let us further denote by f_j^L, f_j^U the following quantities

$$f_j^L = \min_{u\in\mathcal{V}\setminus\mathcal{G}_{j-1}} I(X_{\hat{\mathcal{R}}\setminus\mathcal{R}}; Y_u \mid X_\mathcal{R}, Y_{\mathcal{G}_{j-1}}) \tag{3.40}$$

$$f_j^U = \max_{u\in\mathcal{V}\setminus\mathcal{G}_{j-1}} I(X_{\hat{\mathcal{R}}\setminus\mathcal{R}}; Y_u \mid X_\mathcal{R}, Y_{\mathcal{G}_{j-1}}). \tag{3.41}$$

Then, the following inequality holds

$$f_j^L \leq I(X_{\hat{\mathcal{R}}\setminus\mathcal{R}}; Y_{o_j} \mid X_\mathcal{R}, Y_{\mathcal{G}_{j-1}}) \leq f_j^U. \tag{3.42}$$

If we denote $y_j = I(X_{\mathcal{R}}; Y_{g_j} \mid Y_{\mathcal{G}_{j-1}}), z_j = I(X_{\hat{\mathcal{R}}\setminus\mathcal{R}}; Y_{o_j} \mid X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}})$, then Eqs. (3.39), (3.42) can be rewritten as

$$I(X_{\mathcal{R}}; Y_{\mathcal{O}}) \leq \sum_{i=1}^{j-1} y_i + kz_j + ky_j \tag{3.43}$$

$$f_j^L \leq z_j \leq f_j^U. \tag{3.44}$$

We have that $I(X_{\mathcal{R}}; Y_{\mathcal{G}}) = \sum_{j=1}^{k} y_j$ under the constraints (3.43), (3.44). Let us consider the following linear optimization problem

$$\begin{aligned} \min_{y_1,\ldots,y_k} \quad & \sum_{j=1}^{k} y_j \\ \text{s.t.} \quad & I(X_{\mathcal{R}}; Y_{\mathcal{O}}) \leq \sum_{i=1}^{j-1} y_i + kz_j + ky_j \\ & f_j^L \leq z_j \leq f_j^U, \forall j = 1, 2, \ldots, k. \end{aligned} \tag{3.45}$$

If we denote by $y^* = (y_1^*, \ldots, y_k^*)$ the optimal solution, then we have that $I(X_{\mathcal{R}}; Y_{\mathcal{G}}) = \sum_{j=1}^{k} y_j \geq \sum_{j=1}^{k} y_j^*$. Since our goal is to find a lower bound for the greedy solution, we are interested in solving the optimization problem (3.45).

By taking the dual of problem (3.45), we have

$$\begin{aligned} d^* = \max_{x_1,\ldots,x_k} \quad & f(\mathcal{O}) \sum_{j=1}^{k} x_j + \sum_{j=1}^{k} (f_j^L s_j - f_j^U t_j) \\ \text{s.t.} \quad & kx_j + \sum_{i=j+1}^{k} x_i = 1 \\ & kx_j + s_j - t_j = 0 \\ & x_j, s_j, t_j \geq 0, \forall j = 1, 2, \ldots, k. \end{aligned} \tag{3.46}$$

Dual variables x_j are related to the constraints $I(X_{\mathcal{R}}; Y_{\mathcal{O}}) \leq \sum_{i=1}^{j-1} y_i + kz_j + ky_j$, while dual variables s_j, t_j to the constraints $f_j^L \leq z_j \leq f_j^U$. The optimal solution to the dual problem is the following

$$x_j^* = \frac{1}{k}\left(1 - \frac{1}{k}\right)^{k-j} \tag{3.47}$$

$$t_j^* - s_j^* = \left(1 - \frac{1}{k}\right)^{k-j}, \forall j = 1, 2, \ldots, k. \tag{3.48}$$

The above problem is underdetermined since there are more variables ($3k$) than constraints ($2k$). Due to Eq. (3.48), the term $(f_j^L s_j - f_j^U t_j)$ in the objective of dual problem

(3.46) becomes

$$f_j^L s_j^* - f_j^U t_j^* = f_j^L s_j^* - f_j^U (s_j^* + (1-1/k)^{k-j}) = (f_j^L - f_j^U) s_j^* - f_j^U (1-1/k)^{k-j} \leq 0.$$

Obviously, we want this term to be as large as possible since it is a non-positive term and reduces the overall objective. The value of the non-negative s_j^* that maximizes the above term is $s_j^* = 0$, since $f_j^L \leq f_j^U$. For $s_j^* = 0$, we have

$$t_j^* = \left(1 - \frac{1}{k}\right)^{k-j}, \forall j = 1, 2, \ldots, k. \tag{3.49}$$

Therefore, the optimal dual solution (3.46) becomes

$$d^* = f(\mathcal{O}) \left(1 - \left(1 - \frac{1}{k}\right)^k\right) - \sum_{j=1}^{k} f_j^U \left(1 - \frac{1}{k}\right)^{k-j}. \tag{3.50}$$

Since this is a linear optimization problem, the duality gap is zero. Therefore

$$I(X_{\mathcal{R}}; Y_{\mathcal{G}}) = \sum_{j=1}^{k} y_j \geq \sum_{j=1}^{k} y_j^* = d^* = f(\mathcal{O}) \left(1 - \left(1 - \frac{1}{k}\right)^k\right) - \sum_{j=1}^{k} f_j^U \left(1 - \frac{1}{k}\right)^{k-j}.$$

■

Lemma 3.5.1 (Monotonicity of $I(X_{\hat{\mathcal{R}}\setminus\mathcal{R}}; Y_u \mid X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}})$ with respect to j)**.** *The MI term $I(X_{\hat{\mathcal{R}}\setminus\mathcal{R}}; Y_u \mid X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}})$ is decreasing with respect to j, for every Y_u.*

Proof. We have that

$$\begin{aligned} I(Y_u; X_{\hat{\mathcal{R}}\setminus\mathcal{R}}, Y_{g_j} \mid X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}}) &\overset{(a)}{=} I(Y_u; X_{\hat{\mathcal{R}}\setminus\mathcal{R}} \mid X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}}, Y_{g_j}) + I(Y_u; Y_{g_j} \mid X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}}) \\ &\overset{(b)}{=} I(Y_u; X_{\hat{\mathcal{R}}\setminus\mathcal{R}} \mid X_{\mathcal{R}}, Y_{\mathcal{G}_j}) + I(Y_u; Y_{g_j} \mid X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}}), \end{aligned} \tag{3.51}$$

where (a) is due to the chain rule of MI and (b) since $(Y_{\mathcal{G}_{j-1}}, Y_{g_j}) = Y_{\mathcal{G}_j}$.

The same term can also be written as

$$\begin{aligned} I(Y_u; X_{\hat{\mathcal{R}}\setminus\mathcal{R}}, Y_{g_j} \mid X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}}) &\overset{(a)}{=} I(Y_u; Y_{g_j} \mid X_{\hat{\mathcal{R}}\setminus\mathcal{R}}, X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}}) + I(Y_u; X_{\hat{\mathcal{R}}\setminus\mathcal{R}} \mid X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}}) \\ &= I(Y_u; Y_{g_j} \mid X_{\hat{\mathcal{R}}}, Y_{\mathcal{G}_{j-1}}) + I(Y_u; X_{\hat{\mathcal{R}}\setminus\mathcal{R}} \mid X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}}) \\ &\overset{(b)}{=} I(Y_u; X_{\hat{\mathcal{R}}\setminus\mathcal{R}} \mid X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}}), \end{aligned} \tag{3.52}$$

where (a) is due to the chain rule of MI and (b) due to the conditional independence of measurements given $X_{\mathcal{R}}$, $Y_u \perp\!\!\!\perp Y_{g_j} \mid X_{\mathcal{R}}$. If we combine Eqs. (3.51), (3.52), we have that

$$\begin{aligned} I(Y_u; X_{\hat{\mathcal{R}}\setminus\mathcal{R}} \mid X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}}) &= I(Y_u; X_{\hat{\mathcal{R}}\setminus\mathcal{R}} \mid X_{\mathcal{R}}, Y_{\mathcal{G}_j}) + I(Y_u; Y_{g_j} \mid X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}}) \\ &\geq I(Y_u; X_{\hat{\mathcal{R}}\setminus\mathcal{R}} \mid X_{\mathcal{R}}, Y_{\mathcal{G}_j}), \end{aligned}$$

where the last inequality holds due to the non-negativity of MI. ■

Theorem 3.5.2 (Performance bounds in the focused setting)**.** *Let us denote by* $f(\mathcal{S}) = I(X_{\mathcal{R}}; Y_{\mathcal{S}})$*,* $\mathcal{R}$ *the relevant set, and* $\hat{\mathcal{R}}$ *the extended set such that* $Y_i \perp\!\!\!\perp Y_j \mid X_{\hat{\mathcal{R}}}, \forall i \neq j \in \mathcal{V}$*. If the greedy method described in Alg. 2.1 is applied to problem* $\max_{|\mathcal{S}|\leq k} f(\mathcal{S})$ *and* $\max_u I(X_{\hat{\mathcal{R}}\setminus\mathcal{R}}; Y_u \mid X_{\mathcal{R}}) \leq \frac{1}{k}\max_u I(X_{\mathcal{R}}; Y_u)$*, the following bound holds*

$$f(\mathcal{G}) \geq \underbrace{\frac{1-1/e}{2-1/e}}_{\approx 0.387} f(\mathcal{O}). \tag{3.53}$$

Proof. Let us denote by h_j the measurement that maximizes $I(X_{\hat{\mathcal{R}}\setminus\mathcal{R}}; Y_u \mid X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}})$

$$h_j \in \underset{u\in\mathcal{V}\setminus\mathcal{G}_{j-1}}{\arg\max} \underbrace{I(X_{\hat{\mathcal{R}}\setminus\mathcal{R}}; Y_u \mid X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}})}_{f_j^U}. \tag{3.54}$$

It holds that

$$f_1^U = I(X_{\hat{\mathcal{R}}\setminus\mathcal{R}}; Y_{h_1} \mid X_{\mathcal{R}}) \overset{(a)}{\geq} I(X_{\hat{\mathcal{R}}\setminus\mathcal{R}}; Y_{h_j} \mid X_{\mathcal{R}}) \overset{(b)}{\geq} I(X_{\hat{\mathcal{R}}\setminus\mathcal{R}}; Y_{h_j} \mid X_{\mathcal{R}}, Y_{\mathcal{G}_{j-1}}), \tag{3.55}$$

where (a) holds due to the definition of greedy heuristic (3.54) and (b) due to Lem. 3.5.1. Due to Eq. (3.55), Thm. 3.5.1 can be simplified as

$$\begin{aligned} f(\mathcal{G}) = I(X_{\mathcal{R}}; Y_{\mathcal{G}}) = \sum_{j=1}^{k} y_j \geq \sum_{j=1}^{k} y_j^* = f(\mathcal{O})\left(1-\left(1-\frac{1}{k}\right)^k\right) - \sum_{j=1}^{k} f_j^U \left(1-\frac{1}{k}\right)^{k-j} \\ \geq f(\mathcal{O})\left(1-\left(1-\frac{1}{k}\right)^k\right) - f_1^U \sum_{j=1}^{k}\left(1-\frac{1}{k}\right)^{k-j} \\ = f(\mathcal{O})\left(1-\left(1-\frac{1}{k}\right)^k\right) - f_1^U k\left(1-\left(1-\frac{1}{k}\right)^k\right). \end{aligned} \tag{3.56}$$

If $f_1^U \leq \frac{1}{k}\underbrace{\max_u I(X_{\mathcal{R}}; Y_u)}_{f(g_1)}$, Eq.(3.56) becomes

$$f(\mathcal{G}) \geq f(\mathcal{O})\left(1-\left(1-\frac{1}{k}\right)^k\right) - f(g_1)\left(1-\left(1-\frac{1}{k}\right)^k\right)$$

$$\left(2-\left(1-\frac{1}{k}\right)^k\right) f(\mathcal{G}) \geq f(\mathcal{O})\left(1-\left(1-\frac{1}{k}\right)^k\right)$$

$$f(\mathcal{G}) \geq \frac{1-\left(1-\frac{1}{k}\right)^k}{2-\left(1-\frac{1}{k}\right)^k} f(\mathcal{O}) \geq \lim_{k\to\infty} \frac{1-\left(1-\frac{1}{k}\right)^k}{2-\left(1-\frac{1}{k}\right)^k} f(\mathcal{O}) = \underbrace{\frac{1-1/e}{2-1/e}}_{\approx 0.387} f(\mathcal{O}),$$

where the last inequality holds because the function $\frac{1-\left(1-\frac{1}{k}\right)^k}{2-\left(1-\frac{1}{k}\right)^k}$ is monotonically decreasing with respect to k and $\lim_{k\to\infty} \frac{1-\left(1-\frac{1}{k}\right)^k}{2-\left(1-\frac{1}{k}\right)^k} = \frac{1-1/e}{2-1/e}$.[4] ■

The above bound says that if the information content of the best measurement with respect to $X_{\hat{\mathcal{R}}\setminus\mathcal{R}}$ given $X_{\mathcal{R}}$ is not greater than $\frac{1}{k}\max_u I(X_{\mathcal{R}}; Y_u)$, the same algorithm that provides lower bounds for selection problems with submodular monotone rewards, gives theoretical guarantees for the focused setting as well.

■ 3.5.1 Obtaining an extended set $\hat{\mathcal{R}}$

From the previous section, it becomes apparent that the extended set $\hat{\mathcal{R}}$ needs to be a minimal superset of $\mathcal{R}$ so that the condition that makes the bound in Thm. 3.5.2 true is more likely to be satisfied. The extended set $\hat{\mathcal{R}}$ of latent nodes would basically block all the paths between every pair of measurements. The key idea is to transform the graph to a flow network, where the goal is to find the set of edges between latent nodes with minimum flow capacity that cut the graph in two sets of nodes and thus block the paths between two disjoint sets of measurements.

Before we proceed further, it would be helpful to explain briefly the min-cut problem. Let us assume a graph $G = (V, \mathcal{E})$ with N nodes and $|\mathcal{E}|$ edges. We further define two special nodes, the source s and the sink node t. Each edge (i, j) might have potentially a flow capacity c_{ij}. One main problem of interest is to find the maximum flow from the source s to the sink node t such that the amount of flow entering a node equals the amount of flow exiting assuming there are no other sources other than s. Also, the flow at every edge f_{ij} should not exceed the edge's capacity c_{ij}. This problem is also known in abbreviation as *max-flow*. An *s-t cut* $\mathcal{C}$ is a partition of V into two sets $\mathcal{S}, \mathcal{T}$ such that $s \in \mathcal{S}, t \in \mathcal{T}$. The min-cut problem is finding the edges with the minimum total capacity $c(\mathcal{S}, \mathcal{T}) = \sum_{(u,v)\in\mathcal{S}\times\mathcal{T}} c_{uv}$ whose removal would partition graph G in two sets $\mathcal{S}, \mathcal{T}$ containing nodes s, t, respectively. The min-cut problem is well-tied to the max-flow problem by the max-flow/min-cut theorem. In other words, the max-flow equals the min-cut capacity and there is a straightforward way to recover the min-cut edges from the max-flow problem. Ford-Fulkerson's algorithm is an established algorithm for recovering the max-flow of a network which runs in $\mathcal{O}(f_{\max}|\mathcal{E}|)$ time, where $f_{\max}$ is the max-flow [9].

We can transform the problem of finding an extended set $\hat{\mathcal{R}}$ that blocks all the paths between any pair of measurements to a min-cut problem as follows. Graph G would be comprised of the latent graph plus the observation nodes (measurements) that are connected to latent nodes. If we assume that the number of edges between latent nodes is $|\mathcal{E}|$ and a latent node is linked at most to m measurements, then the total number of edges for G would be on the order of $\mathcal{O}(|\mathcal{E}| + Nm)$, where $N = |\mathcal{V}|$ is the number of

[4] A proof of the monotonicity of $\frac{1-\left(1-\frac{1}{k}\right)^k}{2-\left(1-\frac{1}{k}\right)^k}$ can be found in Appx. A.1.

measurements. We then partition the set of measurements $\mathcal{V}$ into two equally-sized sets $\mathcal{V}_s, \mathcal{V}_t$ and connect set $\mathcal{V}_s$ with "fictitious" source node s and $\mathcal{V}_t$ with "fictitious" sink node t. We apply infinite flow capacities to all edges incident to the source and sink nodes and unit capacities to the remaining edges of the network that correspond to the edges of the original graphical model (see Fig. 3.2b). We run the Ford-Fulkerson to find the min cut-set $\mathcal{C}$ (see Fig. 3.2c). We arbitrarily remove the nodes of one side of the cut and their incident edges. This splits the graph in two disjoint graphs G_s, G_t that contain measurement sets $\mathcal{V}_s, \mathcal{V}_t$, respectively (see Fig. 3.2d). We continue this process recursively until we reach the base case where a graph contains only two measurement nodes ($|\mathcal{V}| = 2$). After each iteration, when we remove the nodes (and their incident edges) of one side of the cut, we block all the paths between each measurement in $\mathcal{V}_s$ and each measurement in $\mathcal{V}_t$. At the end, every pair of measurements would be blocked conditioned on all the nodes that were removed during the algorithmic process. A summary of the method is provided in Alg. 3.2. For a better understanding of the algorithm, that we name BlockingSet, we consider a case with $|\mathcal{V}| = 8$ measurements and demonstrate the flow of the algorithm in Figs. 3.2, 3.3, 3.4. Fig. 3.5 shows the original graph and the resulting extended set $\hat{\mathcal{R}}$ after the application of the proposed algorithm.

Algorithm 3.2 BlockingSet$(G, \mathcal{V})$

Partition set $\mathcal{V}$ in two equally-sized sets $\mathcal{V}_s, \mathcal{V}_t$.
Connect all nodes in $\mathcal{V}_s, \mathcal{V}_t$ to "fictitious" source and sink nodes s, t.
Assign infinite capacities on the edges incident to s, t and unit capacities to the remaining.
Run Ford-Fulkerson algorithm on the designed network.
Find the min cut-set $\mathcal{C}$.
Remove the nodes (and their incident edges) belonging to one side of the cut (this would partition the initial graph G into two disjoint graphs G_s, G_t).
if $|\mathcal{V}_s| \geq 2$ **then**
  BlockingSet$(G_s, \mathcal{V}_s)$
end if
if $|\mathcal{V}_t| \geq 2$ **then**
  BlockingSet$(G_t, \mathcal{V}_t)$
end if

Complexity

As we briefly mentioned in the previous section, Ford-Fulkerson algorithm runs in $\mathcal{O}(f_{\max}|\mathcal{E}|)$ time, where $|\mathcal{E}|$ is the number of edges in the flow network. The original graph contains $|\mathcal{E}|$ edges between latent nodes and at most Nm edges between latent and observed nodes. In addition, if source s is connected to N measurement nodes and since a latent node is connected at most to m measurements via unit capacity edges, the

max-flow $f_{\max}$ cannot exceed Nm. Therefore, the total complexity of Ford-Fulkerson algorithm for this type of network is $\mathcal{O}(Nm(|\mathcal{E}| + Nm))$. During the first iteration of BLOCKINGSET the set of N measurements $\mathcal{V}$ is partitioned into two sets $\mathcal{V}_s, \mathcal{V}_t$ of size $N/2$ each. Since the $N/2$ measurements of $\mathcal{V}_s$ are connected to latent nodes in the original graph with unit capacity edges and each latent node is connect at most to m measurement nodes, the max-flow is upper-bounded by $Nm/2$. The complexity of the algorithm in the first iteration is $\mathcal{O}(\frac{Nm}{2}(|\mathcal{E}| + Nm))$. In the second iteration, the graph is partitioned in two graphs G_s, G_t, each one holding $N/2$ measurements. We repeat the same procedure, where we split the measurement set of each graph in two equally-sized sets. The complexity in this iteration per graph is $\mathcal{O}(\frac{Nm}{4}(|\mathcal{E}| + \frac{Nm}{2}))$. Since, there are two graphs the total complexity in the second iteration is $\mathcal{O}(\frac{Nm}{2}(|\mathcal{E}| + Nm))$. If we continue in this fashion, we will see that the complexity per iteration is $\mathcal{O}(\frac{Nm}{2}(|\mathcal{E}| + Nm))$ per iteration and there is a total of $\log_2 N$ iterations, since every time we divide the remaining observation set by two. Therefore, the total complexity of BLOCKINGSET would be $\mathcal{O}(Nm \log_2 N(|\mathcal{E}| + Nm)) = \mathcal{O}(Nm|\mathcal{E}| \log_2 N)$ assuming $|\mathcal{E}| \geq Nm$.

■ 3.6 Experiments

We present an experiment whose purpose is to demonstrate that penalized MI (PMI) is better-suited for budgeted settings than MI. As a reminder, PMI is defined as

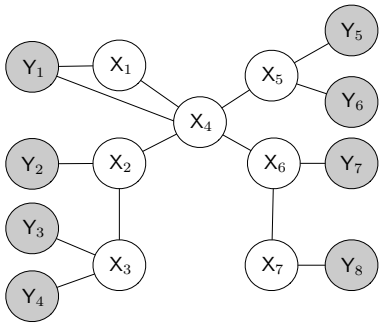
$$f(\mathcal{A}) = I(\mathsf{X}; \mathsf{Y}_{\mathcal{A}}) - \lambda c(\mathcal{A}).$$

We consider the following linear state-space model

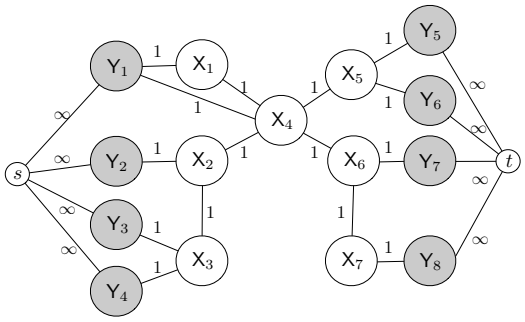
$$\begin{aligned} \mathsf{X}_{t+1} &= A_t \mathsf{X}_t + \mathsf{V}_t \\ \mathsf{Y}_t &= C_t \mathsf{X}_t + \mathsf{W}_t, \end{aligned}$$

where $t = \{1, \ldots, T\}, j = \{1, 2\}$, $\mathsf{V}_t \sim \mathcal{N}(v_t; 0, Q)$ is driving noise, and $\mathsf{W}_t \sim \mathcal{N}(w_t; 0, \sigma_t^2)$ measurement noise. We set $\lambda = 0.5$ and consider $T = 10$ time points. For odd time points, we obtain two measurements which extract the position in x and y. For even time points, we extract only the position in x. In addition, the variances of the noisy measurements are: $(\sigma_t^x)^2 = 16, (\sigma_t^y)^2 = 64$ for odd time points, while $(\sigma_t^x)^2 = 1$ for even time points. In addition, we set the costs such that $c(x) = 0.5, c(y) = 0.05, \forall t \in \{1, \ldots, T\}$.

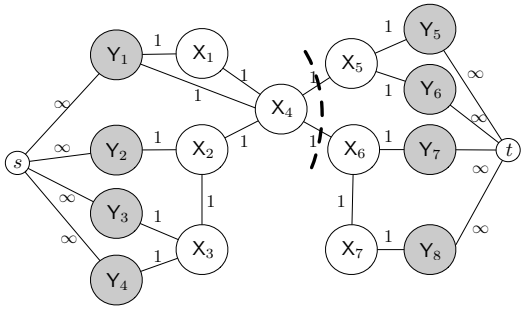
Let $\mathcal{G}_I$ denote the set obtained using MI as the reward function and $\mathcal{G}_P$ the set obtained using PMI. As expected, we see in Figs. 3.6a, 3.6b that the solution by choosing PMI as the reward, $\mathcal{G}_P$ results in lower accumulated cost and higher PMI than the solution when MI is the objective, $\mathcal{G}_I$. PMI is more conservative in selecting measurements that incur really high costs. Interestingly, the use of PMI results in higher cumulative MI at the end, despite the fact that $\mathcal{G}_I$ is optimized with respect to MI, as shown in Fig. 3.6c. The reason is twofold; first the greedy heuristic using MI as the reward selects measurements without regard to costs and second, the costs are structured in a way that the greedy choices for PMI prefer to measure the y-position of a latent variable when only the measurement of the x-position is available in



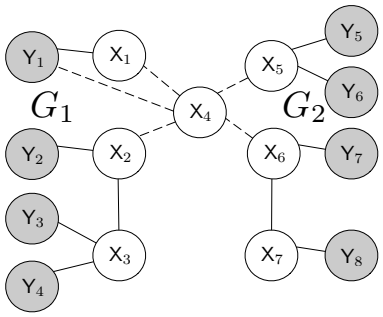
(a) Original graph.



(b) Conversion to a network structure.

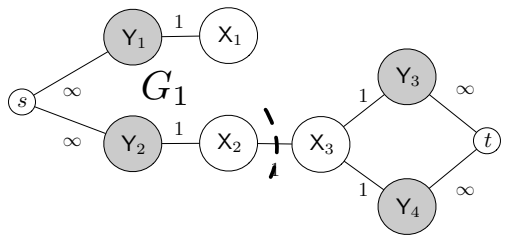


(c) Finding a minimum s-t cut.

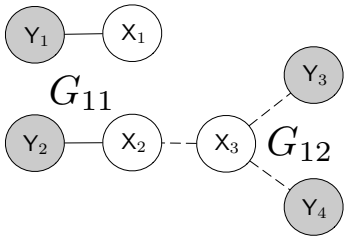


(d) Resulting graph.

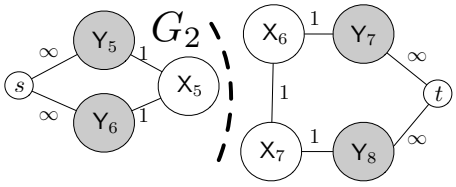
Figure 3.2: Extended set and conversion to a min-cut problem. (a) We are interested in finding an extended set $\hat{\mathcal{R}}$ such that $Y_i \perp\!\!\!\perp Y_j \mid X_{\hat{\mathcal{R}}}, \forall i \neq j$. (b) To convert the graph to a network for solving the min s-t cut problem, we do the following. We divide the set of measurements into two equally-sized sets arbitrarily. We connect the one set with a "fictitious" source node s and the other set with a "fictitious" sink node t. The edges connecting the source and sink nodes to measurements have infinite capacity. All the remaining edges that correspond to the original edges of the graph have unit capacity. (c) We find the min s-t cut by the Ford-Fulkerson algorithm. (d) We choose the nodes of one set of the min-cut as the *blocking* nodes. The edges of the cut in this case are $\{(4,5),(4,6)\}$. We remove arbitrarily node 4 (and its incident edges), since it belongs to one set of the min-cut (removed edges represented by dashed lines). This splits the original graph in two subgraphs G_1 and G_2. We continue this process recursively until we reach a base case, where the measurement set is of size 2.



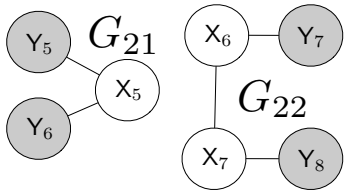
(a) Finding a minimum s-t cut in G_1.



(b) Resulting graph from the cut of G_1.

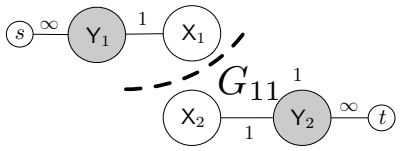


(c) Finding a minimum s-t cut in G_2.

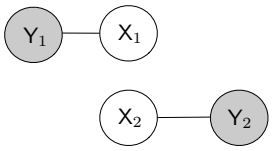


(d) Resulting graph from the cut of G_2.

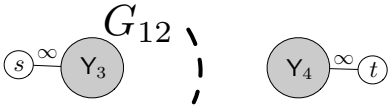
Figure 3.3: Finding the extended set in graphs G_1, G_2 (iteration 2). (a) We divide the measurements $Y_1, \ldots, Y_4$ of graph G_1 in two equally-sized sets, connect the one set with a "fictitious" source node s and the other with a "fictitious" sink node t. We assign infinite capacities to edges incident to s, t and unit costs to the remaining edges of G_1. (b) We pick node 3 arbitrarily as the blocking node and remove all its incident edges, since edge $\{(2, 3\}$ belongs in the cut-set. This results in two smaller graphs G_{11} and G_{12}. (c) We divide the measurements of graph G_2 in two equally-sized sets exactly as in (a). Here, the cut-set is the empty set since there is no flow from s to t. (d) Two smaller subgraphs G_{21} and G_{22} are created.



(a) Finding a minimum s-t cut in G_{11}.

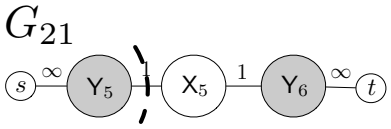


(b) Resulting graph from the cut of G_{11}.

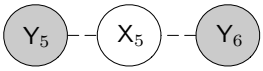


(c) Finding a minimum s-t cut in G_{12}.

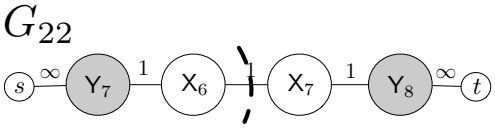
(d) Resulting graph from the cut of G_{12}.



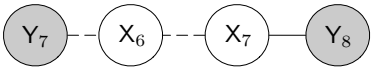
(e) Finding a minimum s-t cut in G_{21}.



(f) Resulting graph from the cut of G_{21}.

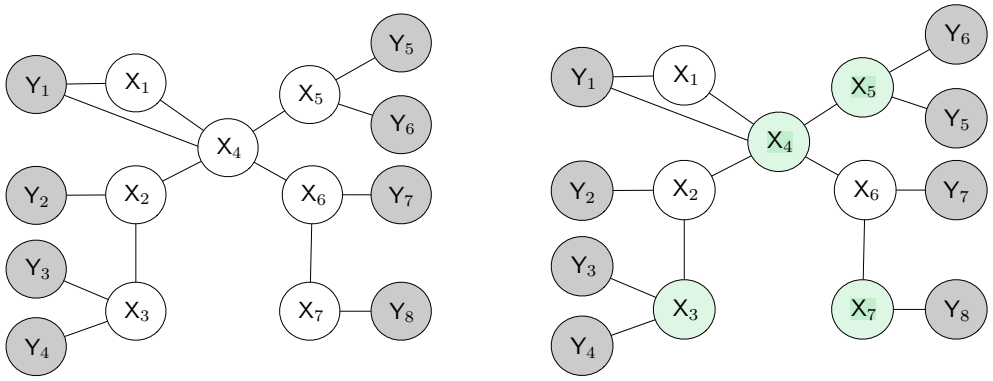


(g) Finding a minimum s-t cut in G_{22}.



(h) Resulting graph from the cut of G_{22}.

Figure 3.4: Finding the extended set in graphs $G_{11}, G_{12}, G_{21}, G_{22}$ (iteration 3). (a) The minimum cut-set in G_{11} is the empty set, since there is no flow between the source and sink nodes. (b) The resulting graph is the same as G_{11}. (c) The same holds for graph G_{12}. (d) As a result, the resulting graph is the same as G_{12}. (e) In graph G_{21}, the minimum cut-set is the edge connecting Y_5 and Y_5. (f) As a result, node 5 is the blocking node. (g) Similarly, for graph G_{22}, the minimum cut-set is $\{(6, 7\}$. (h) We pick arbitrarily node 6 as the blocking node.



(a) Original graph.

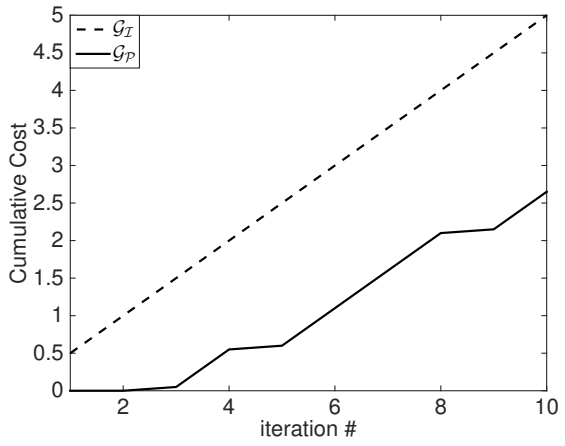
(b) Graph after determining the extended set $\hat{\mathcal{R}}$.

Figure 3.5: Determination of extended set $\hat{\mathcal{R}}$. (a) In the original graph, all measurements are unconditionally dependent. (b) With the previous procedure, we can find an extended set $\hat{\mathcal{R}}$, such that $Y_i \perp\!\!\!\perp Y_j \mid X_{\hat{\mathcal{R}}}, \forall i \neq j$. In this example, we have $\hat{\mathcal{R}} = \{3, 4, 5, 7\}$. The blocking nodes are depicted by light green color.

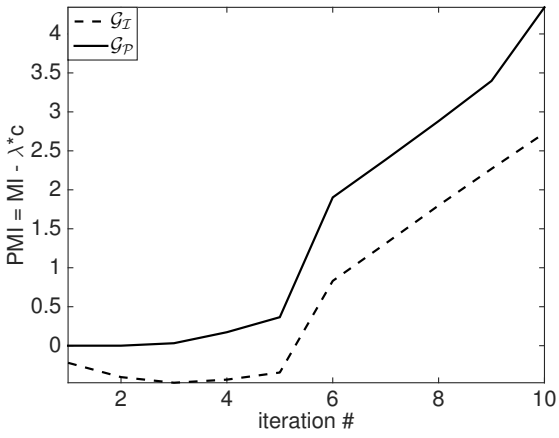
neighboring nodes. This is not to say that there is a general method for adapting cost structures so that greedy PMI outperforms MI, rather that the two, in general, are not comparable and that it is quite possible (as our example shows) that PMI may yield better information rewards than pure MI. Lastly, Fig. 3.6d shows that the posterior entropy of each latent variable is lower for $\mathcal{G}_P$ as compared to $\mathcal{G}_I$. As an additional observation, we see that the monotone behavior induced by the introduction of "slack" measurements (with no cost or informational value) results in non-negative incremental rewards for $\mathcal{G}_P$.

■ 3.7 Conclusion

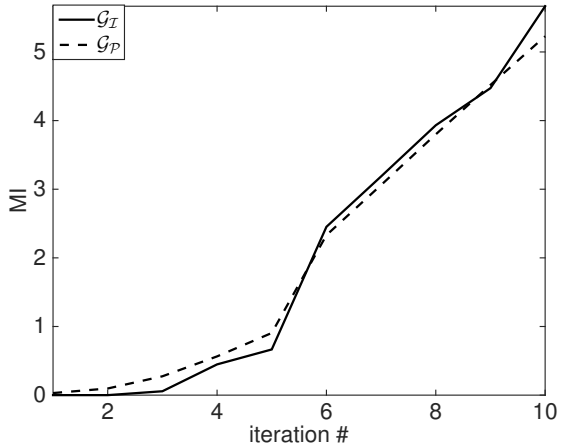
In this chapter, we derived sufficient and necessary conditions under which open-loop is equivalent to closed-loop control planning in exponential family distributions. As expected, we showed that Gaussian models satisfy the necessary condition since the entropy in this case does not depend on actual measurement values but rather on model parameters describing the dependence between measurements and latent variables. Unfortunately, to the best of our knowledge there is no other known distribution that satisfies this condition. We also considered the problem of finding nearly optimal solutions under different problem settings; when the reward is submodular non-monotone, measurements have varying costs and when only a part of latent variables is relevant. In more detail, we derived a lower bound for the greedy solution when the reward function is submodular, that depends on the minimum incremental value $-\theta$ that a measurement can have. In that respect the bound is very pessimistic as it depends on the measurement with the minimum incremental reward. However, the greedy solution might be much higher than the lower bound in cases where the incremental reward of



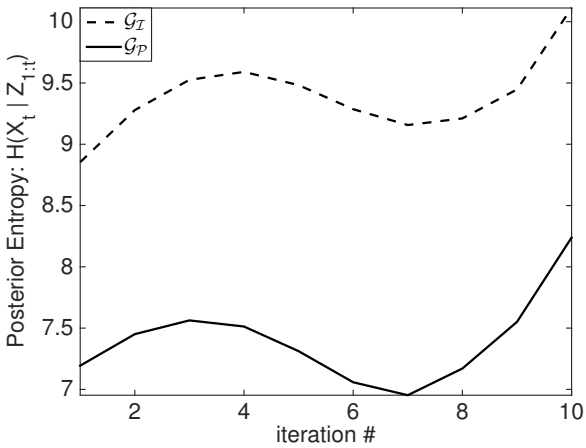
(a) Cumulative cost wrt iteration number.



(b) PMI wrt iteration number.



(c) MI wrt iteration number.



(d) Posterior entropy wrt iteration number.

Figure 3.6: Comparison of MI to PMI for measurement selection. We denote by $\mathcal{G}_I, \mathcal{G}_P$ the solutions generated by choosing MI and PMI as the rewards, respectively. (a) The cost of $\mathcal{G}_P$ is lower than the cost of $\mathcal{G}_I$, since PMI is more conservative in selecting measurements that incur high costs. (b) Since the set $\mathcal{G}_P$ is created by optimizing for PMI, the cumulative PMI at every step is larger than that of solution $\mathcal{G}_I$ as expected. (c) Interestingly, solution $\mathcal{G}_P$ results in higher total MI than $\mathcal{G}_I$, even though the latter set is generated by optimizing for MI. The reason is twofold; first the greedy heuristic using MI as the reward selects measurements without regard to costs and second, the costs are structured in a way that the greedy choices for PMI prefer to measure the y-position of a latent variable when only the measurement of the x-position is available in neighboring nodes. (d) The posterior entropy of each latent variable is lower for $\mathcal{G}_P$ as compared to $\mathcal{G}_I$.

the majority of measurements is much higher than $-\theta$. We additionally show that the use of a penalized form of mutual information as a reward in non-uniform cost settings retains submodularity and thus admits the bounds we derived for general submodular non-monotone functions. Furthermore, we show that when the costs of measurements vary based on the relative information they provide, we can still consider the penalized form of mutual information and use the same approximating algorithms with the same theoretical guarantees. We also consider the submodular knapsack maximization (SKM) problem and derive upper bounds for the optimal solution by casting the problem in its dual form and making use of the linear time algorithm by Buchbinder et al. [13]. Lastly, we derive lower bounds for the greedy solution in focused planning settings, where only a set of latent variables is of interest.

Chapter 4

Complexity Reduction of Information Planning in Gaussian Models

In Chap. 3, we proposed algorithms that serve as approximations to various settings where the optimal solution is intractable as the number of measurements grows. We measured complexity with respect to the graph size, the size of observation sets and the constraint sets. We made the implicit assumption that the complexity of evaluating the reward of a set is provided in constant time (*oracle value* model). Most previous works have assumed such models [51], but as already has been alluded in [38, 47, 54], the complexity of evaluating the reward of different measurement sets is nontrivial. For example, in Gaussian models, when mutual information serves as the reward function, the complexity of evaluating rewards depends on the latent dimension, the number of latent variables and the size of the observations sets. As a consequence, the assumption of oracle value models is not valid for large graphs and observation set sizes. The focus of this chapter is to suggest an approach that reduces substantially the complexity of information planning by taking advantage of sparsity in Gaussian graphical models. We additionally present a variant of Gaussian belief propagation (GaBP) that reduces the computational load significantly and show that it is much faster than standard Kalman filtering/smoothing techniques that are used during greedy selection. We show with experiments that taking sparsity into account and using the proposed variant of GaBP leads to major computational savings that become more significant as the latent dimension and observation size grow. Earlier versions of this work were originally presented in [79, 80].

The chapter commences with Sec. 4.1 by introducing the problem. Sec. 4.2 discusses related work that motivates the subsequent analysis. Previous works are cited that hint on the complexity of evaluating rewards. This motivates the problem of finding efficient ways to reduce this complexity by exploiting sparsity in the graph structure. We formulate the problem in Sec. 4.3 and reiterate briefly some of the necessary theory presented in Chap. 2. Sec. 4.4 presents an approach that takes advantage of sparsity in the graph structure to reduce the complexity of evaluating information rewards. As this

section shows, the complexity of evaluating mutual information is usually encountered in two forms; when we update the covariance of the current walk element after the incorporation of a new measurement and when we explore the current observation set to find the best measurement in a greedy sense. When each measurement depends only on a few latent variables, which translates to a sparse measurement matrix C, we show that significant reductions can be achieved by taking sparsity into account. Sec. 4.5 focuses on forward walks, in other words, on walks of non-decreasing order. When we incorporate a new measurement at the end of each greedy step, we need to update the information coming from the addition of this measurement to the (latent) variable that corresponds to the next walk element. In other words, we need to update the uncertainty (as expressed by the covariance or precision) of the latent variable corresponding to the next element in the walk after the addition of a new measurement. We introduce a variant of BP in Sec. 4.4.3 that sends only the minimal number of messages to update the precision of the next walk element. In Sec. 4.6 we outline how the above analysis can be extended to trees and loopy graphs, while in Sec. 4.7 we briefly present a generalization for the non-linear case. In this case, we can perform planning by using an extended Kalman filter, where we linearize the model around the mean value at that point [101]. As we note, even though linearization may be far from the true model, it can still be used for planning purposes. Once a measurement plan is obtained, inference can be achieved with more accurate techniques (e.g., sampling). We show with synthetic experiments in Sec. 4.8 that our analysis that takes advantage of sparsity results in planning which is orders of magnitude faster than standard approaches. We also demonstrate empirically that major computational savings are obtained by using the proposed variant of BP instead of standard Kalman filtering/smoothing techniques when our goal is to update the uncertainty of the next element in the walk (at each step of the greedy process). We provide an example in which the evaluation complexity and resulting information rewards are decoupled. Lastly, we end the chapter with a discussion in Sec. 4.9.

■ 4.1 Introduction

An important, often neglected, aspect of information-based approaches is the computational cost of evaluating a given plan. While the bounds for greedy selection presented in Sec. 2.6.3 hold for *any* plan subject to the same constraints, one is free to reorder the sequence in which subsets are considered. Some orderings have significantly higher information rewards than others. A simple example occurs in a Markov chain where at each node one may choose k out of N measurements. A naïve plan considers each node in increasing order, where k out of N measurements are selected at each node before we move on to the next node. Alternatively, one may consider nodes in random order selecting a single measurement (from those that have not already been selected), but ensuring each node is considered k times. We will refer to the order of visiting nodes as *visit walk*. One can see a depiction of a naïve plan (referred to also as *forward walk*) and

a random plan in Fig. 4.1b. Evaluating the information reward of the naïve plan has significantly lower computational complexity than the random plan, but the random plan will often yield significantly higher information reward. Thus, there is motivation to expend computational resources for exploring multiple plans subject to the same constraints. Furthermore, when exploring multiple plans, the plan with lowest reward provides the lowest upper bound on the optimal plan yielding a tighter performance guarantee as compared to the greedy plan with highest reward.

Here, we consider the computational complexity of evaluating information rewards for measurement selection in Gaussian models. In such models, complexity depends on the number of latent variables, the latent dimension, the number of measurements to be explored and the visitation order of observation sets, known also as *walk*. We show speedups up to a thousand times by taking advantage of sparsity in the measurement process without changing the outcome of the greedy algorithm. In addition, we demonstrate that by working with the information form of Gaussian, we can provide the sufficient statistics at every step with significantly reduced computation. We achieve that by deploying a variant of Gaussian belief propagation that is well-suited for adaptive inference settings. We present this method for Gaussian HMMs, but show in Chap. 5 its extension to the discrete case, to trees and Gaussian loopy graphs. This analysis is particularly useful for large-scale models, since the evaluation of information rewards poses a major computational bottleneck. Additionally, we demonstrate empirically in an example that both the information reward and evaluation complexity are largely decoupled and as such, exploration of low-complexity walks yields high information rewards.

■ 4.2 Related Work

Consider the problem of selecting an optimal k-element subset (measurements for our purposes) from a ground set $\mathcal{V}$ of size N that maximizes some reward f. As a reminder, we introduced the formulation of this problem in detail in Sec. 2.6. Due to combinatorial complexity, optimal solutions are intractable for even moderately large problems. However, Nemhauser et al. [77] showed in their seminal work that when function f is submodular and monotone the reward of the greedy solution obtain by Alg. 2.1 is no worse than $(1 - 1/e)$ of the optimal reward. However, this result implicitly assumes an *oracle value* model, *i.e.*, that the reward function for any given subset can be computed in constant time. Specifically, the *oracle value* model assumes a universal "oracle" that provides the function value for any input set. Subsequent work [34], which generalizes [77] to matroidal structures, also assumes an oracle value model. They show that if the same greedy approach is applied to matroidal structures, the reward cannot be worst than $1/2$ times the optimal one.

There have been a number of methods utilizing the preceding. As noted, Guestrin et al. [38], Krause and Guestrin [54] consider information planning in the batch setting while Kempe et al. [47] considers influential subset selection in social networks. The

original bound of $1/2$ in [34] is improved to $(1 - 1/e)$ in [15], but it is restricted to rewards that are sums of weighted rank functions of matroids. Online resource allocation networks [87, 88], stochastic submodular maximization [4], the submodular welfare problem [93], and additional extensions to submodular maximization [17, 32, 37] comprise just a small sampling of approaches and analyses which exploit the results of [34, 77]. All of the above works assume oracle value models.

However, as we discuss, evaluation of rewards presents a significant computational challenge. As such, Guestrin et al. [38] propose truncation methods as an approximation for Gaussian models. Similarly, Krause and Guestrin [54] note that the evaluation of conditional entropies, intrinsic to greedy selection, can be prohibitive, while Kempe et al. [47] acknowledge the complexity in evaluating the underlying influence function that guides the selection of the most influential nodes. We are not aware of previous results that exploit the structure of the latent graph to reduce the complexity of reward evaluations. Despite the inherent computational bottleneck, it is often overlooked. Here, we show for Gaussian models that exact computations are feasible by taking advantage of the graph structure and utilizing a variant of belief propagation (BP) more suited for active learning settings.

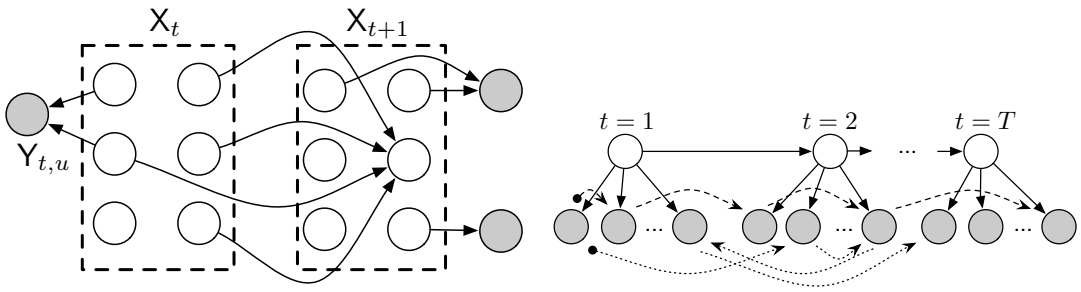
■ 4.3 Problem Statement

For brevity and clarity of discussion, we restrict ourselves to HMMs. However, the results easily extend to trees and polytrees. Our analysis can also be generalized to Gaussian loopy MRFs as we show in Sec. 5.9. In the case of Gaussian HMMs, the underlying dynamical system can be described by

$$X_t = A_{t-1}X_{t-1} + V_{t-1} \tag{4.1}$$
$$Y_t = C_t X_t + W_t, \tag{4.2}$$

where $X_t, Y_t, t \in \{1, \dots, T\}$ are the latent and observed variables, respectively. In addition, $V_t \sim \mathcal{N}(v_t; \mu_{1,t}, Q_t)$, $W_t \sim \mathcal{N}(w_t; \mu_{2,t}, R_t)$ are the respective process and measurement noises, with R_t being block-diagonal. Let $X = \{X_1, \dots, X_T\}$ denote the set of latent variables up to time T. Each X_t is a d-dimensional vector. For each X_t, we define an *observation set*, denoted by $\mathcal{V}_t$, where $|\mathcal{V}_t| = N_t$ (N_t comparable to d). Each measurement $Y_{t,u}$ from set $\mathcal{V}_t$ is an m-dimensional vector. In Eq. (4.2), Y_t represents the set of all N_t measurements of $\mathcal{V}_t$. Therefore, C_t is a $N_t m \times d$ matrix, where consecutive m-row patches correspond to one measurement from $\mathcal{V}_t$. W.l.o.g., we assume $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset, \forall i \neq j$. As shown in Fig. 4.1a, a measurement depends at most on q elements of X_t. As a consequence, C_t is a highly sparse matrix. Our goal is to characterize approximate solutions to the following (generally intractable) combinatorial optimization problem:

$$\mathcal{O} \in \underset{\{S \mid\mid S \cap \mathcal{V}_t| \leq k_t, \forall t\}}{\arg\max} f(\mathcal{S}), \tag{4.3}$$



(a) Sparsity in the measurement process. (b) Examples of different walks.

Figure 4.1: Sparsity and different walks. (a) Each measurement Y_t depends only on a few components of X_t. Dashed rectangles represent vectors of variables at a point, X_t, X_{t+1}. This structure can be collectively represented as an HMM, as Fig. 4.1b shows. (b) Two walks, that is, orders in which observation sets are visited, are visualized. A forward walk represented with dashed arrows is a walk with increasing order of visiting the different observation sets. In this type of walk, we greedily select the required number measurements for each observation set before we move on to the next one. Another (random) walk is depicted as well (represented with dotted arrows) for comparison. Arrows with a circle in one end denote the beginning of the walk. Different walks visit each observation set the same number of times, but in different orders.

where $f(\mathcal{S})$ is a set function and k_t are the selection constraints for the t-th set. We restrict ourselves to monotonic functions and thus it never "hurts" to obtain more measurements. Hence, the inequality constraint becomes tight. To give an indication of the hardness of problem (4.3), there are $\prod_t \binom{N_t}{k_t} = \binom{N}{k}^T$ feasible solutions, which is an extremely large number as N, k, T grow.

■ 4.3.1 Greedy Selection

Greedy methods select elements sequentially conditioned on the previous selection. A *walk* $\boldsymbol{w} = \{w_1, \ldots, w_M\}$ denotes the particular order in which observation sets are visited. A walk is depicted in Fig. 4.2. Greedy selection for a particular walk is defined as

$$g_j \in \underset{u \in \mathcal{V}_{w_j} \setminus \mathcal{G}_{j-1}}{\arg\max} \; f(u \mid \mathcal{G}_{j-1}), \tag{4.4}$$

where w_j is the observation set index corresponding to the j-th element of the walk and $\mathcal{G}_{j-1}$ is the greedy set obtained up to the previous iteration. The marginal increase in the reward (incremental reward) of incorporating a measurement u in a given set $\mathcal{G}_{j-1}$ is denoted as $f(u \mid \mathcal{G}_{j-1}) = f(\mathcal{G}_{j-1} \cup \{j\}) - f(\mathcal{G}_{j-1})$. Essentially, at each greedy step we choose the measurement that maximizes the incremental reward based on past selections $\mathcal{G}_{j-1}$. Since we explore at most all N measurements from a set and there are kT steps of the algorithm (assuming $k_t = k, \forall t$), its overall complexity is $\mathcal{O}(kTN)$

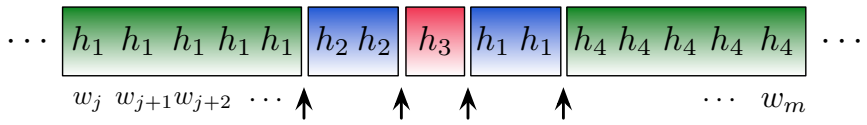


Figure 4.2: Composition of a walk. A walk, $\boldsymbol{w} = \{w_1, \ldots, w_M\}$, represents the particular order observation sets are visited during measurement selection and correspond to a feasible solution of problem (4.3). A walk segment is a part of the walk that consists of elements from the same observation set. Here, we have two segments of length 5 corresponding to sets $\mathcal{V}_{h_1}, \mathcal{V}_{h_4}$, two segments of length 2 corresponding to sets $\mathcal{V}_{h_1}, \mathcal{V}_{h_2}$, and one of length one corresponding to set $\mathcal{V}_{h_3}$. Vertical arrows indicate *transition points*, which are points of transition between different observation sets.

as compared to $\mathcal{O}(\binom{N}{k}^T)$ of the optimal approach. In information planning, a common choice for the reward function is MI, $f(\mathcal{S}) = I(\mathsf{X}; \mathsf{Y}_\mathcal{S})$, with the resulting incremental reward being $f(u \mid \mathcal{G}_{j-1}) = I(\mathsf{X}; \mathsf{Y}_u \mid \mathsf{Y}_{\mathcal{G}_{j-1}})$. A walk that corresponds to a feasible solution for problem (4.3) has length $M = \sum_{t=1}^{T} k_t$. A *forward walk* is a walk with non-decreasing order (referred as *naïve* walk in the introduction). W.l.o.g., we assume that each observation set has the same number of measurements $N_t = N$.

■ 4.3.2 Theoretical guarantees on greedy selection

As we showed in Sec. 2.6.3, a greedy solution in the sequential setting achieves in the worst case half of the optimal reward [102]:

$$f(\mathcal{G}) \geq \frac{1}{2} f(\mathcal{O}).$$

The above bound, seen differently, serves also as an upper bound on the optimal reward which cannot exceed twice the reward of an arbitrary walk. Depending on the problem structure, different walks yield significantly different rewards. Of most interest are the walks of highest and lowest reward, since the first category offers the good solutions, while the second gives the lowest bound on the optimal reward. Note that tighter on-line bounds are available with some computation [102].

■ 4.3.3 Gaussian HMMs

Gaussian models are appealing since information rewards can be expressed as a function of the covariance of the underlying process. In addition, obtaining values for the measurements does not have an effect in the selection of measurements, which makes planning completely independent of the actual measurement process. The incremental reward of a measurement u is defined as

$$f(u \mid \mathcal{G}_{j-1}) = I(\mathsf{X}; \mathsf{Y}_u \mid \mathsf{Y}_{\mathcal{G}_{j-1}}) = H(\mathsf{X} \mid \mathsf{Y}_{\mathcal{G}_{j-1}}) - H(\mathsf{X} \mid \mathsf{Y}_u, \mathsf{Y}_{\mathcal{G}_{j-1}}). \tag{4.5}$$

As we showed in Sec. 2.9.1, entropy can be analytically expressed as

$$H(X) = \frac{d}{2}(1 + \log 2\pi) + \frac{1}{2}\log|\Sigma_X|,$$

where d is the dimension of X.

Therefore, Eq. (4.5) becomes

$$f(u \mid \mathcal{G}_{j-1}) = \frac{1}{2}\log\frac{|\Sigma_{X|\mathcal{G}_{j-1}}|}{|\Sigma_{X|\{u\}\cup\mathcal{G}_{j-1}}|} = \frac{1}{2}\log\frac{|J_{X|\{u\}\cup\mathcal{G}_{j-1}}|}{|J_{X|\mathcal{G}_{j-1}}|}, \tag{4.6}$$

depending on whether we work with the moment (covariance) or information (precision) form.

In Sec. 2.1, we showed that the covariance updates in Gaussian HMMs take the following form

$$\Sigma_{t|t-1} = A_{t-1}\Sigma_{t-1|t-1}A_{t-1}^T + Q_{t-1} \tag{4.7}$$

$$\Sigma_{t|t} = \Sigma_{t|t-1} - G_t C_t \Sigma_{t|t-1} \tag{4.8}$$

$$G_t = \Sigma_{t|t-1}C_t^T(C_t\Sigma_{t|t-1}C_t^T + R_t)^{-1},$$

where $\Sigma_{t|t-1} = \mathrm{cov}(X_t \mid Y_1, \ldots, Y_{t-1}) = \mathrm{cov}(X_t \mid Y_{1:t-1})$, $\Sigma_{t|t} = \mathrm{cov}(X_t \mid Y_{1:t})$. Eqs. (4.7), (4.8) are referred to as the *propagation* and *update* steps, respectively. It is important to note that pursuant to propagation, the incremental reward depends only on the *local* update to the node whose entropy is of interest.[1] Consequently, updates to the local covariance (equivalently precision) matrix *fully* quantify the information reward with respect to the full set of latent nodes.

■ 4.4 Complexity Reduction in Information Planning

For a given walk, three primary sources of computational complexity arise when evaluating the information reward: (i) exploration, in which the information rewards of the remaining measurements of the current observation set are computed to find the best measurement in a greedy sense; (ii) update, in which the covariance matrix of the current walk element incorporates the selected measurement; and (iii) propagation, in which the uncertainty is propagated to the next element in the walk. Greedy algorithms proceed as follows; exploration, update and propagation: (i) explore the available candidate measurements in the observation set of that node choosing the measurement with highest incremental reward, as dictated by Eq. (4.4), (ii) update the uncertainty at that node after selecting the measurement, and lastly, (iii) propagate the uncertainty to the next node in the walk. Exploration and updates depend on the structure of C, while propagation depends on the composition of the walk.

[1] Note that this property is a consequence of conditional independence and is not restricted to Gaussian models.

With slight abuse of notation, we denote the m-row portion of matrix C_t corresponding to measurement $\mathsf{Y}_{t,u}$ as $C_t(u,:)$. After we select a measurement g_j at step j as dictated by Eq. (4.4), we need to update the uncertainty at the current node of the walk, X_{w_j} and then propagate this uncertainty to $\mathsf{X}_{w_{j+1}}$. For notational consistency with the analysis later in the text, we will denote the greedily selected measurement g_j by u in the discussion below. From Eq. (4.8), we see that the complexity of the update step is $\mathcal{O}(md^2)$, where d is the dimensionality of X_{w_j}, since the computation is dominated by the term $C_{w_j}(u,:)\Sigma_{w_j|\mathcal{G}_{j-1}}$ for $m \ll d$. Updates at each iteration yield overall complexity of $\mathcal{O}(\sum_{t=1}^{T} k_t m d^2) \overset{k_t=k}{=} \mathcal{O}(Tkmd^2)$. Exploration of one measurement takes $\mathcal{O}(d^3)$ time as shown in Eq. (4.6), since it requires the computation of the determinant of a $d \times d$ matrix. The number of measurements that should be considered is $\mathcal{O}(kTN)$. Therefore, the overall complexity of exploration is $\mathcal{O}(TkNd^3)$, which makes it the dominant term in the computational load. Lastly, every time we update the uncertainty at the current node in the walk we need to propagate it to the next walk element. We do this by propagating the uncertainty to the end of the chain through Kalman filtering and then smooth back to the next walk element via Kalman smoothing. The complexity of this is $\mathcal{O}(T)$. Since, we need to repeat this kT times assuming $k_t = k, \forall t$, the total complexity of propagation is $\mathcal{O}(kT^2)$.

The above analysis is agnostic to the sparsity of C. Here, we show that complexity of exploration and updates are dramatically reduced by taking advantage of this sparsity. Additionally, working with a minimal variant of belief propagation (instead of standard Kalman filtering/smoothing techniques) to propagate the uncertainty to the next walk element yields further efficiencies. We will start by first discussing reductions in the update step due to sparsity, then continue on discussing the exploration step and lastly cover the propagation step.

■ 4.4.1 Reductions during updates

Let I_c denote the indicator matrix of the non-zero elements of C. Computation of I_c requires $\mathcal{O}(Nmd)$ time or $\mathcal{O}(TNmd)$, if time-varying. We further assume that the largest parent set for an m-dimensional measurement is of size q, where $q \ll d$. A depiction of this is shown in Fig. 4.3. For an $m \times 1$ measurement u, denote I_u as $I_u = \vee_{i=1}^{m} I_c(u_i,:)$, where u_i is the i^{th} component of measurement u. That is, I_u is the $d \times 1$ indicator vector representing the nodes of latent graph X that generated measurement u. If latent variable X_i is linked to measurement $\mathsf{Y}_{t,u}$, $I_u(i) = 1$, while 0 otherwise. Since a measurement depends on at most q latent variables, we have that $\sum_{i=1}^{d} I_u(i) \leq q$. Making use of Eq. (4.8), the updated covariance Σ' with Σ as a prior is:

$$\Sigma' = \Sigma - \Sigma C(u,:)^T (C(u,:)\Sigma C(u,:)^T + R(u,u))^{-1} C(u,:)\Sigma. \tag{4.9}$$

By inverting (4.9), we obtain the updated precision matrix:

$$\begin{aligned} J' &= (\Sigma - \Sigma C(u,:)^T (C(u,:)\Sigma C(u,:)^T + R(u,u))^{-1} C(u,:)\Sigma)^{-1} \\ &\overset{(a)}{=} \Sigma^{-1} + C(u,:)^T R(u,u)^{-1} C(u,:) \\ &= J + C(u,:)^T R(u,u)^{-1} C(u,:), \end{aligned} \tag{4.10}$$

where in (a) we made use of the Woodbury matrix identity.

We denote by $\hat{C}_u = R(u,u)^{-1/2} C(u, I_u)$ the $m \times q$ matrix that takes only into account the part of C matrix that is relevant to the latent variables that generated measurement u. The matrix square root $R(u,u)^{-1/2}$ can be recovered in $\mathcal{O}(m^3)$ time. In addition, $\hat{C}_u$ is computed in $\mathcal{O}(m^2 q)$ time. Therefore, (4.10) can be rewritten as

$$J' = \begin{bmatrix} J(I_u, I_u) & J(I_u, \neg I_u) \\ J(\neg I_u, I_u) & J(\neg I_u, \neg I_u) \end{bmatrix} + \begin{bmatrix} \hat{C}_u^T \\ 0^T \end{bmatrix} \begin{bmatrix} \hat{C}_u & 0 \end{bmatrix}, \tag{4.11}$$

where $\neg I_u$ denotes all the latent variables that are not directly linked to measurement u. Eq. (4.11) can be written more concisely as

$$J'(I_u, I_u) = J(I_u, I_u) + \hat{C}_u^T \hat{C}_u,$$

since only the block of J dictated by I_u will be affected by $\hat{C}_u^T \hat{C}_u$.

The above calculation requires $\mathcal{O}(mq^2)$ steps. Summarizing, the square root $R(u,u)^{-1/2}$ takes $\mathcal{O}(m^3)$ time, the computation of $\hat{C}_u$ $\mathcal{O}(m^2 q)$ and the computation of $\hat{C}_u^T \hat{C}_u$ $\mathcal{O}(mq^2)$ time. Overall, the computation is on the order of $\mathcal{O}(m \max\{m,q\}^2)$. Compare this to the complexity of the standard calculation, which is $\mathcal{O}(md^2)$ per update, resulting in a speedup on the order of $(\frac{d}{\max\{m,q\}})^2$ per update.

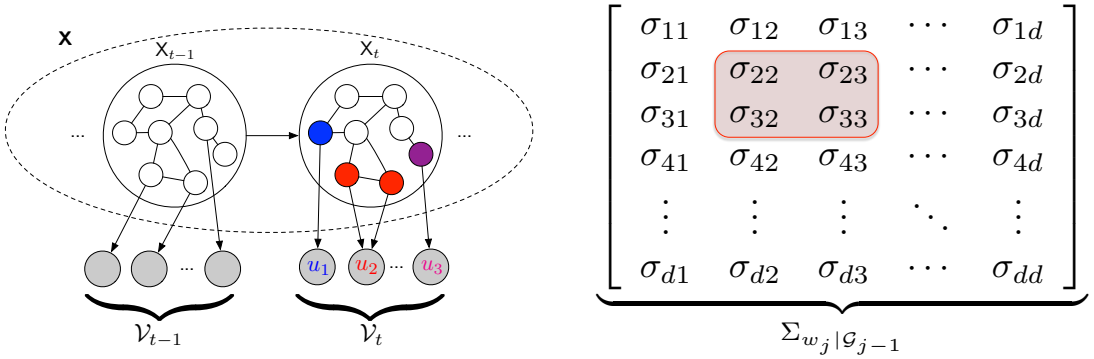
■ 4.4.2 Reductions during exploration

Greedy selection for Gaussian models simplifies to

$$g_j \in \underset{u \in \mathcal{V}_{w_j} \setminus \mathcal{G}_{j-1}}{\arg\max} \; I(\mathsf{X}; \mathsf{Y}_{w_j,u} \mid \mathsf{Y}_{\mathcal{G}_{j-1}}) \tag{4.12}$$

$$\begin{aligned} &\overset{(a)}{=} I(\mathsf{X}_{w_j}; \mathsf{Y}_{w_j,u} \mid \mathsf{Y}_{\mathcal{G}_{j-1}}) + \cancelto{0}{I(\mathsf{X}_{1:T \setminus w_j}; \mathsf{Y}_{w_j,u} \mid \mathsf{X}_{w_j}, \mathsf{Y}_{\mathcal{G}_{j-1}})} \\ &= \underset{u \in \mathcal{V}_{w_j} \setminus \mathcal{G}_{j-1}}{\arg\max} \; I(\mathsf{X}_{w_j}; \mathsf{Y}_{w_j,u} \mid \mathsf{Y}_{\mathcal{G}_{j-1}}) \\ &= \underset{u \in \mathcal{V}_{w_j} \setminus \mathcal{G}_{j-1}}{\arg\max} \; \log \frac{|J_{w_j|\{u\} \cup \mathcal{G}_{j-1}}|}{|J_{w_j|\mathcal{G}_{j-1}}|}, \end{aligned} \tag{4.13}$$

where (a) is due to the chain rule of MI and the fact that $\mathsf{X}_{1:T \setminus w_j} \perp\!\!\!\perp \mathsf{Y}_{w_j,u} \mid \mathsf{X}_{w_j}$. $J_{w_j|\mathcal{G}_{j-1}}$ is the precision of X_{w_j} given observations $\mathsf{Y}_{\mathcal{G}_{j-1}}$ and $J_{w_j|\{u\} \cup \mathcal{G}_{j-1}}$ is the precision after the incorporation of measurement u.



(a) Each measurement depends only on a few latent variables.

(b) Only the block of matrix linked to the measurement is necessary for information planning.

Figure 4.3: Sparsity in the measurement matrix. (a) Each measurement depends only on a few latent variables. In this example, measurements Y_{u_1}, Y_{u_3} from set $\mathcal{V}_t$ depend on one latent variable, while measurement Y_{u_2} depends on two latent variables. (b) If we assume that Y_{u_2} depends on variables $X_{w_j,2}, X_{w_j,3}$, then according to Eq. (4.15) we would only need the block of $\Sigma_{w_j|\mathcal{G}_{j-1}}$ that is related to $X_{w_j,2}, X_{w_j,3}$ for information planning purposes.

We can express $J_{w_j|\{u\}\cup\mathcal{G}_{j-1}}$ as a function of $J_{w_j|\mathcal{G}_{j-1}}$ by using Eq. (4.11):

$$J_{w_j|\{u\}\cup\mathcal{G}_{j-1}} = J_{w_j|\mathcal{G}_{j-1}} + \begin{bmatrix} \hat{C}_{w_j,u}^T \\ 0^T \end{bmatrix} \begin{bmatrix} \hat{C}_{w_j,u} & 0 \end{bmatrix}, \tag{4.14}$$

where $\hat{C}_{w_j,u} = R_{w_j}(u,u)^{-1/2} C_{w_j}(u, I_u)$.

As we observe in (4.13), the selection of next measurement (in a greedy sense) is determined by ratios of determinants. If we apply the Matrix Determinant Lemma on Eq. (4.14), we obtain

$$\begin{aligned} |J_{w_j|\{u\}\cup\mathcal{G}_{j-1}}| &= \left| J_{w_j|\mathcal{G}_{j-1}} + \begin{bmatrix} \hat{C}_{w_j,u}^T \\ 0^T \end{bmatrix} \begin{bmatrix} \hat{C}_{w_j,u} & 0 \end{bmatrix} \right| \\ &= |J_{w_j|\mathcal{G}_{j-1}}| \left| \mathbb{I}_{m\times m} + \begin{bmatrix} \hat{C}_{w_j,u} & 0 \end{bmatrix} J_{w_j|\mathcal{G}_{j-1}}^{-1} \begin{bmatrix} \hat{C}_{w_j,u}^T \\ 0^T \end{bmatrix} \right| \\ &= |J_{w_j|\mathcal{G}_{j-1}}| \left| \mathbb{I}_{m\times m} + \begin{bmatrix} \hat{C}_{w_j,u} & 0 \end{bmatrix} \Sigma_{w_j|\mathcal{G}_{j-1}} \begin{bmatrix} \hat{C}_{w_j,u}^T \\ 0^T \end{bmatrix} \right| \end{aligned}$$

Therefore the ratio of determinants can be re-written as,

$$\frac{|J_{w_j|\{u\}\cup\mathcal{G}_{j-1}}|}{|J_{w_j|\mathcal{G}_{j-1}}|} = |\mathbb{I}_{m\times m} + \hat{C}_{w_j,u} \Sigma_{w_j|\mathcal{G}_{j-1}}(I_u, I_u) \hat{C}_{w_j,u}^T|. \tag{4.15}$$

From a closer inspection, we see that the incremental reward of a measurement u, depends only the noise introduced in the measurement, the non-zero part of the measurement matrix $C_{w_j}(u,:)$ and the block of covariance $\Sigma_{w_j|\mathcal{G}_{j-1}}(I_u, I_u)$ that are related

Table 4.1: Speedups achieved by sparsity

Speedup/step	Arbitrary Walk	Forward Walk
propagation	–	$\mathcal{O}(d/p)$
exploration	$\mathcal{O}(N)$	$\mathcal{O}(1)$
update	$\mathcal{O}((d/\max\{m,q\})^2)$	$\mathcal{O}((d/\max\{m,q\})^3)$

to to measurement u. Evaluating $\hat{C}_{w_j,u}$ takes $\mathcal{O}(m^2 \max\{q,m\})$ time, while the term inside the determinant can be recovered in $\mathcal{O}(mq\max\{q,m\})$ time. In addition, evaluating the determinant of the above matrix is an $\mathcal{O}(m^3)$ operation. Overall the complexity of calculating the ratios is $\mathcal{O}(m\max\{m,q\}^2)$ as compared to that of the standard calculation which is $\mathcal{O}(d^3)$. However, Eq. (4.15) implies knowledge of $\Sigma_{w_j|\mathcal{G}_{j-1}}$. As we showed in Sec. 4.4.1, it is much more beneficial to evaluate the precision rather than the covariance matrix in an update step. Therefore, at the beginning of the exploration step, we need to recover $\Sigma_{w_j|\mathcal{G}_{j-1}}$ by inverting $J_{w_j|\mathcal{G}_{j-1}}$, which is accomplished in $\mathcal{O}(d^3)$ time. The complexity of an exploration step is thus $\mathcal{O}(d^3 + Nm\max\{m,q\}^2)$, while the complexity of an exploration step with the naïve approach is $\mathcal{O}(Nd^3)$, since we evaluate a $d \times d$ matrix for every of N measurements. The speedups that we obtain are on the order of $\left(N\frac{(d/\max\{m,q\})^3}{(d/\max\{m,q\})^3+N}\right)$. For $d/\max\{m,q\} \geq \sqrt[3]{N}$, which is the case, since our assumption is that d and N are comparable, the speedups we gain are on the order of N per exploration step. Since there is a total of kT steps of the greedy approach, the total speedups that we obtain through sparsity by our method are $\mathcal{O}(kT((d/\max\{m,q\})^2 + N)) = \mathcal{O}(kT(d/\max\{m,q\})^2)$. For example, if $d = N = 10^4, T = 1000, k = 10$ and $q = 10, m = 1$ the savings are on the order of 10^{10}. We summarize the gains we obtain in the update and exploration steps by making use of sparsity in Tab. 4.1.

■ 4.4.3 Reductions during propagation

The savings in the update and exploration steps assume the knowledge of $J_{w_j|\mathcal{G}_{j-1}}$ (or equivalently $\Sigma_{w_j|\mathcal{G}_{j-1}}$) at the beginning of each greedy step. The naïve way to compute $\Sigma_{w_j|\mathcal{G}_{j-1}}$ is to first propagate forward the covariance up to the node corresponding to the first element of the walk w_1 and greedily select the measurement from that node. In general, having selected a measurement, we need to propagate forward from the current node up to the maximum walk element encountered so far and then smooth back to the node corresponding to the next walk element. Each filtering and smoothing step requires $\mathcal{O}(d^3)$ time. Filtering is bounded by $\mathcal{O}(Td^3)$, while smoothing is bounded by $\mathcal{O}\left((T - w_{j+1})d^3\right)$. Filtering and smoothing occurs only at the points of the walk when we move to a new node, yielding overall complexity of $\mathcal{O}(\sum_{j=1}^{M-1} \mathbb{1}(w_j \neq w_{j+1})(2T - w_{j+1})d^3)$. In the worst case, $w_j \neq w_{j+1}, \forall j$ giving overall complexity of $\mathcal{O}(kT^2d^3)$, when $k_t = k, \forall t$. In the above approach, the farther the next walk element is from the node with the maximum index encountered so far, the greater the number of unnecessary computations. However, the increase in the total information reward only depends on

the covariance update to the node corresponding to the next walk element w_{j+1}

$$I(\mathsf{X}; \mathsf{Y}_{w_{j+1},u} \mid \mathsf{Y}_{\mathcal{G}_j}) = I(\mathsf{X}_{w_{j+1}}; \mathsf{Y}_{w_{j+1},u} \mid \mathsf{Y}_{\mathcal{G}_j}) = \frac{1}{2} \log \frac{|\Sigma_{w_{j+1}|\mathcal{G}_j}|}{|\Sigma_{w_{j+1}|\{u\}\cup\mathcal{G}_j}|} = \frac{1}{2} \log \frac{|J_{w_{j+1}|\{u\}\cup\mathcal{G}_j}|}{|J_{w_{j+1}|\mathcal{G}_j}|}. \tag{4.16}$$

Consequently, computations may be focused only on variable $\mathsf{X}_{w_{j+1}}$ that corresponds to the next walk element. In fact, as Eq. (4.16) suggests, we need the covariance (or equivalently precision) of $\mathsf{X}_{w_{j+1}}$ after the incorporation of the new measurement at step j, that changed the greedy solution to $\mathcal{G}_j = g_j \cup \mathcal{G}_{j-1}$. Obtaining the covariance $\Sigma_{w_{j+1}|\mathcal{G}_j}$ (through Kalman filtering and smoothing) would induce unnecessary computations, as it requires the propagation of uncertainty to the maximum walk element so far ($\max\{w_1, \ldots, w_j\}$) and then smoothing back to the next walk element w_{j+1}. As we show in the the next section the information form is more efficient as uncertainty propagation from the current to the next walk element can be achieved using a modified version of Gaussian belief propagation (GaBP).

Gaussian HMMs as MRFs

It is well known that a Gaussian HMM may be described by node and edge potentials as

$$\begin{aligned}
\varphi_t(x_t) &= \exp\left(-\frac{1}{2} x_t^T J_{t,t} x_t + x_t^T h_t\right) \\
\psi_{t,t+1}(x_t, x_{t+1}) &= \exp\left(-x_t^T J_{t,t+1} x_{t+1}\right), \text{ where} \\
h_t &= Q_{t-1}^{-1}\mu_{1,t-1} - A_t^T Q_t^{-1}\mu_{1,t} + C_t^T R_t^{-1}(y_t - \mu_{2,t}) && (4.17) \\
J_{t,t} &= Q_{t-1}^{-1} + A_t^T Q_t^{-1} A_t + C_t^T R_t^{-1} C_t && (4.18) \\
J_{t,t+1} &= -A_t^T Q_t^{-1}. && (4.19)
\end{aligned}$$

where node φ_t and edge potentials $\psi_{t,t+1}$ have information parameters $(h_t, J_{t,t})$, $(0, J_{t,t+1})$, respectively.

An alternative approach to Kalman filtering (that works with covariance matrices) is Gaussian BP (GaBP) (that works with precision matrices). In GaBP, the inference propagates in the form of "forward" and "backward" messages [82, 99]:

Forward Pass

$$\begin{aligned}
h_{t\to t+1} &= -J_{t,t+1}^T (J_{t,t} + J_{t-1\to t})^{-1}(h_t + h_{t-1\to t}) \ , \forall t && (4.20) \\
J_{t\to t+1} &= -J_{t,t+1}^T (J_{t,t} + J_{t-1\to t})^{-1} J_{t,t+1} && (4.21)
\end{aligned}$$

Backward Pass

$$\begin{aligned}
h_{t\to t-1} &= -J_{t-1,t} (J_{t,t} + J_{t+1\to t})^{-1}(h_t + h_{t+1\to t}) \ , \forall t && (4.22) \\
J_{t\to t-1} &= -J_{t-1,t} (J_{t,t} + J_{t+1\to t})^{-1} J_{t-1,t}^T && (4.23)
\end{aligned}$$

where $J_{0\to1} = 0, h_{0\to1} = 0$, $J_{T+1\to T} = 0, h_{T+1\to T} = 0$. The marginals X_t given all the measurements $\mathsf{X}_t \mid \mathsf{Y}_{1:T} \sim \mathcal{N}^{-1}(x_t; h_{t|T}, J_{t|T})$, are obtained as:

$$h_{t|T} = h_t + h_{t-1\to t} + h_{t+1\to t}$$
$$J_{t|T} = J_{t,t} + J_{t-1\to t} + J_{t+1\to t}.$$

Updating the node potential

Assuming noise is independent across different measurements within an observation set, R_t is an $N_t m \times N_t m$ block-diagonal matrix. Each block is of size $m \times m$. Incorporating an $m \times 1$ measurement u from set $\mathcal{V}_t$, only affects $h_t, J_{t,t}$ as

$$h'_t = h_t + C_t(u,:)^T R_t(u,u)^{-1}(y_t(u) - \mu_{2,t}(u))$$
$$J'_{t,t} = J_{t,t} + C_t(u,:)^T R_t(u,u)^{-1} C_t(u,:).$$

The above operation completes in $\mathcal{O}(m \max(m,q)^2)$ time.

Adaptive BP in Gaussian HMMs

We saw in Eq. (4.15) that the incremental reward of a measurement u only requires the covariance (precision) of the next walk element. We can further reduce the complexity by restricting the message updates to the nodes between the current and next walk element. In this manner, the precisions at the nodes between the current and next walk elements are correctly updated. The uncertainty of the remaining variables will be incorrect. However, this does not affect the planning process since the covariance of the node of interest corresponding to the next walk element will be correctly updated. The way that the algorithm works briefly is the following. If $w_j < w_{j+1}$, we only need to update the forward messages from w_j to w_{j+1}. Similarly, if $w_j > w_{j+1}$, we only need to update the backward messages from w_j to w_{j+1}. We call this modified version of belief propagation for HMMs, *adaptive BP* for HMMs.

To provide a more detailed description of the algorithm, we start by evaluating all node potentials assuming no measurements are available and propagate messages along the entire chain in both directions. As a new measurement g_j is selected from set $\mathcal{V}_{w_j}$, we compute and update messages from X_{w_j} to $\mathsf{X}_{w_{j+1}}$. This results in correct node marginals along that path. Within same walk segment, $(w_j = w_{j+1})$, propagation is unnecessary and we need only update the node potential (h_{w_j}, J_{w_j,w_j}) See Alg. 4.1 for details. Since relevant node potentials are correct at every iteration, and the message from $w_j - 1$ to w_j (if $w_j < w_{j+1}$), or $w_j + 1$ to w_j (if $w_j > w_{j+1}$) is correct, then the messages from w_j to w_{j+1} are guaranteed to be correct as well. Messages from $w_j - 1$ to w_j (or $w_j + 1$ to w_j) are also guaranteed to be correct, since during the previous walk step, from w_{j-1} to w_j, that message was either not included in the path from w_{j-1} to w_j and thus remained unchanged or has been correctly updated (as part of the directed message schedule from w_{j-1} to w_j). See Fig. 4.4 for an example flow of the algorithm.

At every iteration, we need to update exactly $|w_{j+1} - w_j|$ messages. Therefore, the overall complexity is $\mathcal{O}\left(\sum_{j=1}^{M-1} |w_{j+1} - w_j| d^3\right)$. If we denote by $\bar{\ell} \triangleq \frac{1}{M-1} \sum_{j=1}^{M-1} |w_{j+1} - w_j|$, the average length of the path connecting two nodes of neighboring walk elements, then the complexity term can be rewritten as $\mathcal{O}(kT\bar{\ell}d^3)$, where $M = kT$. Compare this with the complexity of the naïve approach which is $\mathcal{O}\left(kT^2d^3\right)$. Since $\bar{\ell} \leq T$, we achieve a speedup on the order of $\mathcal{O}(T/\bar{\ell})$ compared to Kalman filtering/smoothing or standard BP. Small $\bar{\ell}$ implies that there are only short jumps in the walk. In other words, it is "cheaper" to obtain the marginal of the node of the next walk element when there is a small distance from the current node. On the other hand, nodes farther from the current walk element generally give higher information gain at the cost of more intensive computation. In the best-case scenario, $\bar{\ell}$ is a small constant, making the speedup be on the order of $\mathcal{O}(T)$.

Algorithm 4.1 ADAPTIVE BELIEF PROPAGATION FOR GAUSSIAN HMMS

1: **Initialization**
2: Initialize the node, pairwise potentials and messages as described in Sec. 4.4.3.
3: **Iteration**
4: **for** $j = 1, \ldots, M-1$ **do**
5: Update the node potential at X_{w_ℓ} as

$$h_{w_j} = h_{w_j} \quad + C_{w_j}(g_j, :)^T R_{w_j}(g_j, g_j)^{-1} \mathsf{Y}_{w_j}(g_j) - \mu_{w,w_j}(g_j) \tag{4.24}$$

$$J_{w_j,w_j} = J_{w_j,w_j} + C_{w_j}(g_j, :)^T R_{w_j}(g_j, g_j)^{-1} C_{w_j}(g_j, :). \tag{4.25}$$

6: Send messages from w_j to w_{j+1}.
7: Compute the covariance at $\mathsf{X}_{w_{j+1}}$ as

$$J_{w_{j+1}|\mathcal{G}_j} = J_{w_{j+1},w_{j+1}} + J_{w_{j+1}-1 \to w_{j+1}} + J_{w_{j+1}+1 \to w_{j+1}}. \tag{4.26}$$

8: **end for**

■ 4.5 Forward Walks

Walks with non-decreasing orders, known as forward walks, are a special case since their computation relies only on forward propagation. While the evaluation complexity of such walks is low, they tend to produce significantly lower information rewards. However, such walks are still of use in that they provide tighter upper bounds on the optimal solution. Furthermore, forward walks benefit from additional computational reductions since Kalman filtering is a sufficient approach to update uncertainty of future nodes in the walk.

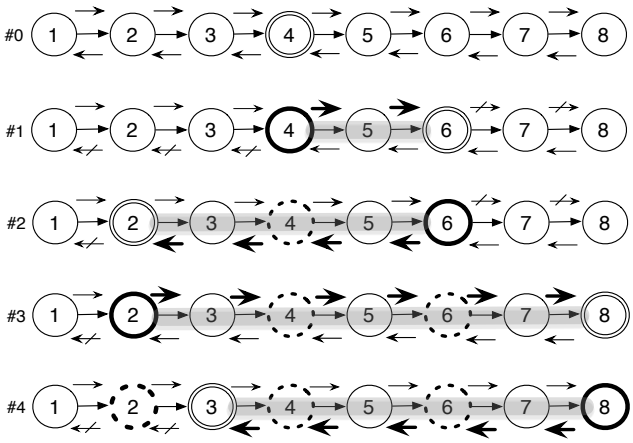


Figure 4.4: Adaptive message passing in Gaussian HMMs. Solid thick node represents the node w_j, while the node with double stroke the next one w_{j+1}. Dashed nodes represent measurements that have been obtained in the past. The potential of w_j is updated after the incorporation of measurement g_j. Thick arrows represent messages that are transmitted in the current iteration. Solid and strikethrough arrows represent correct messages and incorrect messages, respectively. Gray bands encompass all the nodes whose marginals are correctly computed. During initialization (iteration #0), all node potentials, forward and backward messages are computed. At each (greedy) iteration, we update the precision of the current walk element w_j and then send messages from w_j to w_{j+1}. This updates the precision at the next walk element $J_{w_{j+1}|\mathcal{G}_j}$. Having updated the precision at the next walk element correctly, we can then apply the greedy algorithm to find the best measurement for that step.

4.5.1 Reductions during propagation in forward walks

We assume that the dynamics matrix A is sparse. In other words, each row has at most p non-zero elements. The sparsity in the dynamics matrix A affords a computational advantage for such walks. It is straightforward to see that propagation requires $\mathcal{O}(d^3)$ time. We can take advantage of sparsity, by storing the indicator matrix I_a, which contains the non-zero elements of each row of A. The above operation requires $\mathcal{O}(d^2)$ time, or $\mathcal{O}(Td^2)$ for time-varying models. Assuming we know the covariance $\Sigma_{t-1|t-1}$, we can evaluate the elements of $\Sigma_{t|t-1}$ as follows:

$$\begin{aligned}
\Sigma_1(i,:) &= A_{t-1}(i, I_a(i,:))\Sigma_{t-1|t-1}(I_a(i,:),:), \ \forall i \in \{1,\ldots,d\} \\
\Sigma_2(:,\ell) &= \Sigma_1(:, I_a(\ell,:))A_{t-1}(\ell, I_a(\ell,:))^T \quad , \ \forall \ell \in \{1,\ldots,d\} \\
\Sigma_{t|t-1} &= \Sigma_2 + Q_{t-1},
\end{aligned}$$

where Σ_1, Σ_2 are temporary $d \times d$ matrices. The above evaluation completes in $\mathcal{O}(pd^2)$ time, where $p \ll d$ and so is much faster than $\mathcal{O}(d^3)$ of the standard calculation. The speedups per exploration step are thus $\mathcal{O}(d/p)$. It is important to note that we only need to propagate when the walk transits to a new node Since there is a maximum of T nodes, the overall speedups are on the order of $\mathcal{O}(Td/p)$. Interestingly, since the speedups depend on T, the worst-case complexity of a forward walk is not affected by the composition of the walk but rather the length of the chain T.

4.5.2 Reductions during updates in forward walks

As we saw in Sec. 4.4.1, the posterior covariance after the incorporation of a measurement u is

$$\Sigma' = \Sigma - \Sigma C(u,:)^T (C(u,:)\Sigma C(u,:)^T + R(u,u))^{-1} C(u,:)\Sigma, \tag{4.27}$$

which can be rewritten as

$$\Sigma' = \Sigma - \Sigma(:, I_u)C(u, I_u)^T (C(u, I_u)\Sigma(I_u, I_u)C(u, I_u)^T + R(u,u))^{-1} C(u, I_u)\Sigma(I_u, :), \tag{4.28}$$

if we take advantage of sparsity in C. Because we established in the beginning of Sec. 4.5 that Kalman filtering is optimal to propagate uncertainty to the next walk element for forward walks, it is more appropriate to work with the covariance form. In that case, it is preferable to incorporate the measurement (chosen in the greedy step) directly to the covariance. The dominant term in Eq. (4.28) is $C(u, I_u)\Sigma(I_u, :)$, which is a $m \times d$ matrix. Even though we can obtain significant gains by taking advantage of sparsity of C, the complexity of updating the covariance is an $\mathcal{O}(md^2)$ operation due to the presence of $C(u, I_u)\Sigma(I_u, :)$. Therefore, the update of covariance in forward walks does not provide any benefits compared to the standard solution in terms of $\mathcal{O}$ notation. However, the empirical benefits of exploiting sparsity in C when we evaluate the quantity $C(u, I_u)\Sigma(I_u, I_u)C(u, I_u)^T$ can still be significant, even though the theoretical complexity stays the same.

4.5.3 Reductions during exploration in forward walks

Regarding the exploration, if we denote by $N(u)$ the (latent) neighbors of measurement u we have

$$\begin{aligned} g_j &\in \underset{u \in \mathcal{V}_{w_j} \setminus \mathcal{G}_{j-1}}{\arg\max}\ I(\mathsf{X}_{w_j}; \mathsf{Y}_{w_j,u} \mid \mathsf{Y}_{\mathcal{G}_{j-1}}) \\ &\overset{(a)}{=} \underset{u \in \mathcal{V}_{w_j} \setminus \mathcal{G}_{j-1}}{\arg\max}\ I(\mathsf{X}_{w_j,N(u)}; \mathsf{Y}_{w_j,u} \mid \mathsf{Y}_{\mathcal{G}_{j-1}}) \\ &= \underset{u \in \mathcal{V}_{w_j} \setminus \mathcal{G}_{j-1}}{\arg\max}\ \log \frac{|\Sigma_{w_j|\mathcal{G}_{j-1}}(I_u, I_u)|}{|\Sigma_{w_j|\{u\}\cup\mathcal{G}_{j-1}}(I_u, I_u)|}, \end{aligned}$$

where (a) holds since the remaining hidden variables gain no information from u conditioned on the neighbors of measurement u, $\mathsf{X}_{w_j,N(u)}$.

From Eq. (4.15), we see that the selection of a measurement depends on $\hat{C}_{w_j,u}$ and the block of $\Sigma_{w_j|\mathcal{G}_{j-1}}$ that is related to the latent nodes that generated the measurement.

$$g_j \in \underset{u \in \mathcal{V}_{w_j} \setminus \mathcal{G}_{j-1}}{\arg\max}\ \log \frac{|J_{w_j|\{u\}\cup\mathcal{G}_{j-1}}|}{|J_{w_j|\mathcal{G}_{j-1}}|} = \log |\mathbb{I}_{m\times m} + \hat{C}_{w_j,u}\Sigma_{w_j|\mathcal{G}_{j-1}}(I_u, I_u)\hat{C}^T_{w_j,u}|. \quad (4.29)$$

The covariance $\Sigma_{w_j|\mathcal{G}_{j-1}}$ is computed during the propagation step when we move to the next walk element, while $\hat{C}_{w_j,u}$ is the $m \times q$ matrix defined in Sec. 4.4.2. The complexity of computing Eq. (4.29) is $\mathcal{O}(\max\{m,q\}^3)$ per measurement. If we compare this to the complexity of exploration of the standard approach $\mathcal{O}(d^3)$, the speedups that we gain are on the order of $\mathcal{O}((d/\max\{m,q\})^3)$ per measurement. Overall, we explore a maximum of N measurements per greedy step and there is a total of kT steps. So, the total speedups are on the order of $\mathcal{O}(kTN(d/\max\{m,q\})^3)$. Tab. 4.1 summarizes these results. By way of example, if $d = N = 10^4, T = 1000, k = 10$ and $q = 10, m = 1$ the savings are on the order of 10^{17}.

4.6 Extension to trees and loopy graphs

Our analysis on Sec. 4.4 applies directly to trees and loopy graphs as it depends only on the sparsity of measurement matrix C_{w_j} for iteration j, which models the dependence between the measurements of observation set $\mathcal{V}_{w_j}$ and the latent variable X_{w_j}. Therefore, the gains we receive from sparsity are not related to the latent graph structure. However, in case of trees or loopy graphs the notion of forward walk is not relevant anymore. Therefore, the findings of Sec. 4.5 apply only to Gaussian HMMs. Regarding the propagation of uncertainty to the next walk element, the algorithm presented in Sec. 4.4.3 is not applicable anymore. However, we extend the notion of adaptive BP to trees and loopy graphs in Chap. 5. We will see in this chapter that the propagation of uncertainty to the next walk element requires only as many messages as the length of the path connecting two consecutive walk elements. We can determine

instantly the connecting path of two nodes by keeping a structure that provides the lowest common ancestor of two nodes in constant time. In case of loopy graphs, we break the loops by determining a set of nodes, called *Feedback Vertex Set (FVS)*. We then apply the variant of BP presented in Chap. 5 to the remaining acyclic graph.

■ 4.7 Complexity Reduction in Non-Linear Models

In the previous sections, we presented results for linear Gaussian models. Usually more complex phenomena need to be expressed by non-linear models. We can generalize to the non-linear case by taking first-order approximations. Assume $X_1 \sim p$, where $p \propto p^*$, and p^* is a tractable, continuous, second-order differentiable function, strictly positive on its domain. If this distribution has a local maximum at x^*, we can approximate p around x^* with a Gaussian through Laplace's method [67]

$$p_{X_1}(x^*) \simeq \mathcal{N}(x^*; \mu_1, \Sigma_1),$$

where $\mu_1 = x^*, \Sigma_1 = -\nabla^2 \ln p^*(x)\big|_{x=x^*}$. In addition, we assume that each X_t is given as

$$X_t = f_{t-1}(X_{t-1}) + V_{t-1},$$

where $f_{t-1} : \mathbb{R}^d \to \mathbb{R}^d$, $V_{t-1} \sim \mathcal{N}(v_{t-1}; 0, Q_{t-1})$ and measurements Y_t are given by

$$Y_t = h_t(X_t) + W_t,$$

where $h_t : \mathbb{R}^d \to \mathbb{R}^N$, $W_t \sim \mathcal{N}(w_t; 0, R_t)$. In this case, the propagation and update are given by the Extended Kalman Filter equations [28].

Propagation

$$\begin{aligned} \hat{x}_{t|t-1} &= f_{t-1}(\hat{x}_{t-1|t-1}) \\ \Sigma_{t|t-1} &= A_{t-1}\Sigma_{t-1|t-1}A_{t-1}^T + Q_{t-1} \end{aligned}$$

Update

$$\begin{aligned} \hat{x}_{t|t} &= \hat{x}_{t|t-1} + G_t(y_t - h_t(\hat{x}_{t|t-1})) \\ \Sigma_{t|t} &= \Sigma_{t|t-1} - G_t C_t \Sigma_{t|t-1} \\ G_t &= \Sigma_{t|t-1}C_t^T(C_t\Sigma_{t|t-1}C_t^T + R_t)^{-1}, \end{aligned}$$

where $A_t = (\nabla f_t(\hat{x}_{t|t}))^T, C_t = (\nabla h_t(\hat{x}_{t|t-1}))^T$. That is, $[A_t]_{ij} = \frac{\partial f_{ti}}{\partial x_j}, [C_t]_{uj} = \frac{\partial h_{tu}}{\partial x_j}$. In this case, sparsity in the dynamics function f_t is expressed through the i-th component of that function. That is, if the i-th latent variable at point $t+1$ $X_{t+1,i}$ is linked to a few neighbors from X_t, the i-th component of function f_t is written as $f_{ti}(x_{t,1}, \ldots, x_{t,d}) = f_{ti}(x_{t,N(t+1,i)})$, where $N(t+1, i)$ are the neighbors of variable $X_{t+1,i}$ at point t. Similarly, for h_{tu}. For each row of A_t and C_t, the only non-zero components would be the ones corresponding to variables that participate in the generation of X_{t+1}, Y_t, respectively. For example, if variable $X_{t,j}$ does not participate in the generation of measurement $Y_{t,u}$, we would have that $\partial h_{tu}/\partial x_j = 0$. We should note that if functions f_t, h_t are

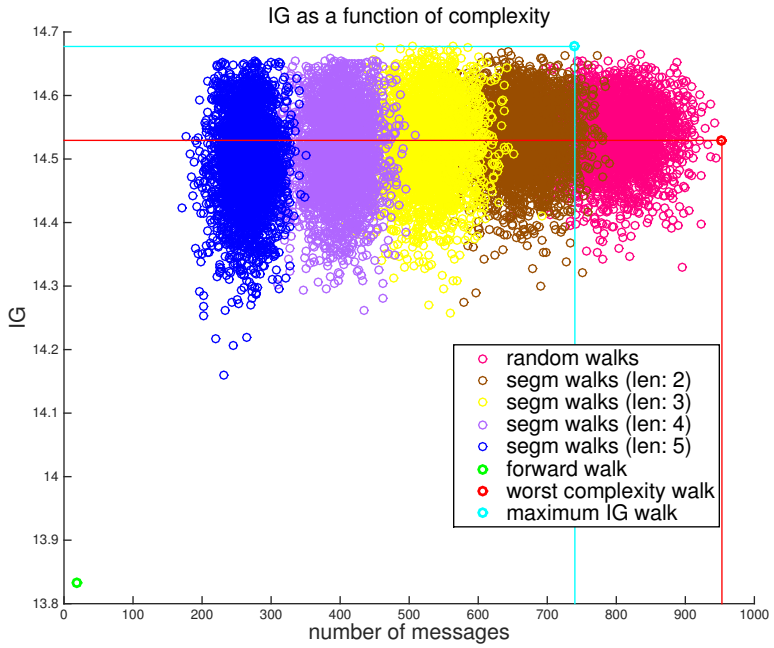


Figure 4.5: Empirical analysis of structured and unstructured walks. The figure compares information rewards versus computation complexity (as quantified by number of messages) for walks of different minimum size segment (magenta:1, brown:2, yellow:3, purple:4, blue:5). As we see, the distribution of rewards is similar in all cases, but the evaluation complexity is significantly lower for walks with harder constraints on the minimum segment length.

highly non-linear, the first-order approximations for inference purposes can be extremely inaccurate. However, for planning purposes we can still design a planning schedule by using non-linear approximations and perform inference later with more accurate techniques (e.g., sampling).

■ 4.8 Experiments

We consider synthetic tracking experiments, where the primary goal is to demonstrate the utility of the method from a computational perspective. We additionally observe that in some cases, information rewards - depending on the structure of the problem - may be decoupled from the complexity of walk. In such cases, exploration may be restricted to low-complexity walks while yielding high information rewards. The properties under which this condition arises remains an open question. The previous analysis provides a tool by which such questions can be examined.

In our setup, there are three objects. Two of the objects move away from the third one. Each object has a 6-dimensional states, $p_x, p_y, p_z, v_x, v_y, v_z$ representing the positions and velocities along the three axes. We consider the following linear state-

space model

$$X_t = A_{t-1}X_{t-1} + V_{t-1}$$
$$Y_t = C_tX_t + W_t,$$

where A_{t-1} captures linear dynamics, $V_{t-1} \sim \mathcal{N}(v_{t-1}; 0, Q_{t-1})$ is driving noise and $W_t \sim \mathcal{N}(w_t; 0, R_t)$ is measurement noise. Potential measurements are available for each latent variable (position, velocity), which amounts to 18 measurements per time point (6 per object) of which we may select six (at each time point). We consider five different types of walks with the following walk segment minimum sizes; $\ell_{\min} \in \{1, 2, 3, 4, 5\}$.[2] By minimum size, we mean that for every set $\mathcal{V}_t$, there is a walk segment with size at least $\ell_{\min}$. In Fig. 4.5 we compare rewards of different walks to their complexities. As expected, the forward walk (green circle) has lowest complexity and lowest reward. Interestingly enough, even though walks with larger minimum segment sizes (blue, purple) result in much lower complexities, they have comparable rewards to those of lower minimum segment sizes (brown, magenta) and higher complexity. Additionally, the walk with the maximum reward which belongs in one of the highest complexity clusters is not significantly higher than the maximum from lower complexity clusters.

We also examine the speedup due to sparsity by considering 200 moving objects with different degrees of correlated motion. The latent dimension in this case is $d = 1200$. We consider different observation sizes, constituting $\{10\%, 25\%, 50\%, 75\%, 100\%\}$ of the latent dimension and different degrees of sparsity in the measurement model. Fig. 4.6 shows the efficiency gains as a function of sparsity q and observation size N, as we discussed in Sec. 4.4. Here, color indicates speedup factor, the maximum being 1400 and the lower being 88.

Lastly, we examine the advantage of adaptive BP compare to standard Kalman filtering and smoothing. We construct 10 Markov chains of varying length (from $T = 10$ to $T = 300$) and compare adaptive BP to the standard Kalman filtering and smoothing. In Fig. 4.7a, certain speedups are obtained when the walk elements are sampled uniformly across the T different observation sizes. We also construct multiple walks with specified average distance between consecutive walk elements, (1–5, 5–10, 15–20, 50–60) for a chain of length $T = 100$ to better demonstrate the sensitivity of the proposed method to the average distance between consecutive walk elements. As the average distance between consecutive walk elements decreases, the greater the advantage of Adaptive BP.

■ 4.9 Conclusion

We have considered the problem of efficient evaluation of information rewards in Gaussian HMMs. Naïve evaluation of such rewards is generally prohibitive for all but forward walks. There are generally three sources of complexity; exploration, update and propagation. We propose an approach that takes advantage of sparsity in the measurement

[2] Walk segment is a segment of the walk where all elements belong to the same observation set.

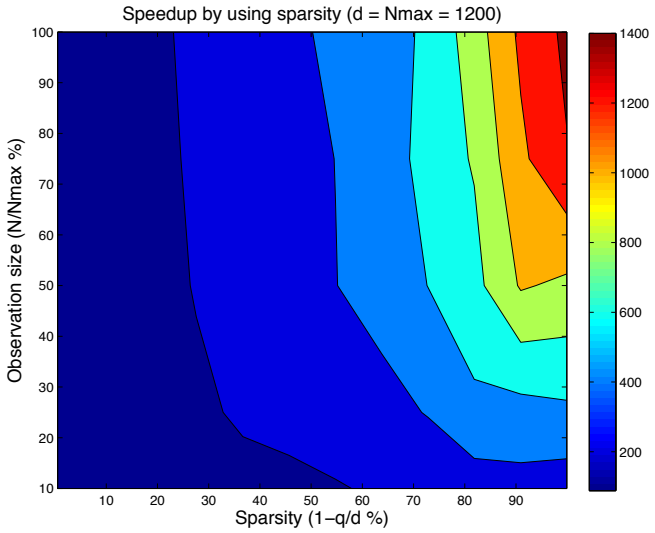
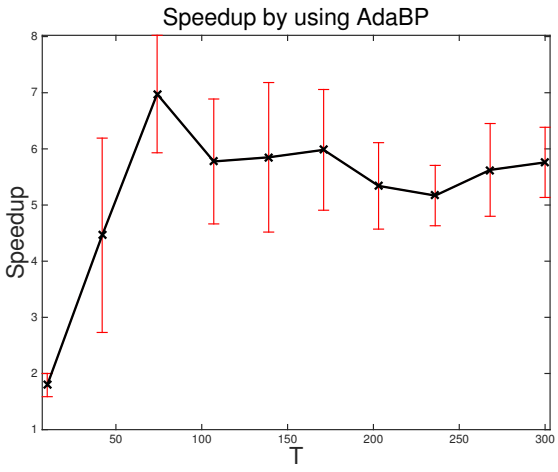
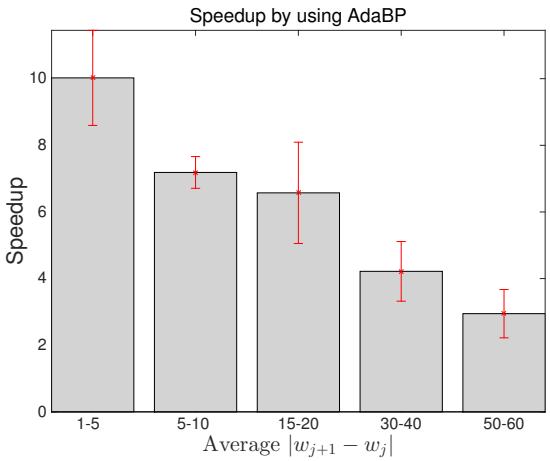


Figure 4.6: Speedups by taking advantage of sparsity during exploration and updates. The figure shows the speedup we gain by taking sparsity into account. We explore the speedup for different degrees of sparsity defined as $1 - q/d$ and different observation sizes N (as expressed by ratios to a maximum size $N_{\max}$). As expected, we see that the gains are more imminent as observation size and sparsity grows.

process to reduce substantially the complexity in exploration and update steps. We obtain a $\mathcal{O}(N)$ speedup per exploration step, where N is the size of an observation set. Furthermore, we obtain $\mathcal{O}((d/\max\{m,q\})^2)$ speedup per update step. In addition, we introduce a variant of Gaussian belief propagation that only sends messages from the current w_j to the next element of the walk w_{j+1} to evaluete the uncertainty $\Sigma_{w_{j+1}|\mathcal{G}_j}$ and is much more efficient than standard Kalman filtering and smoothing techniques. If the average distance between consecutive walk elements is $\bar{\ell}$, then we obtain speedup on the order $\mathcal{O}(T/\bar{\ell})$. The significant reductions in the complexity of evaluating information rewards allow for the exploration of multiple plans that might lead to improved rewards. Furthermore, our experimental results reveal that in some cases the information reward of walk can be decoupled from its complexity. As a result, exploration can be restricted to low-complexity walks while still yielding high information rewards that are guaranteed to be within a computable factor of the (intractable) optimal solution.



(a) Distance between consecutive walk elements is uniform.



(b) Varying average distance between consecutive walk elements.

Figure 4.7: Speedups from adaptive message passing during propagation. (a) Efficiency gains as the length of the chain T increases. Gains stabilize around a single number due to the construction of the walks. Since we choose the same number of measurements from every observation set, and observation sets are sampled uniformly with probability $1/T$, the mean distance between two consecutive points would be $T/3$. Kalman filtering and smoothing presents an $\mathcal{O}(T)$ asymptotic complexity. In reality though, we send $2T$ messages on average at every iteration, since we need to propagate to the end of the chain and then smooth back to the node that corresponds to the next walk element. Therefore, we would expect the speedup to converge to a number close to $\frac{2T}{T/3} = 6$. (b) The figure shows how gains change as the average distance between consecutive elements increases. As expected, we see that when the average distance between consecutive walk elements is low, we gain significant speedups.

Chapter 5

Adaptive Belief Propagation

GRAPHICAL models are widely used in inference problems. They can represent relationships between random variables and span a wide range of applications. The proliferation of data and the desire to build more accurate models has given rise to graphs with ten of thousands or even millions of variables. However, users of such models are often interested in only a particular set of variables which might change over time depending on the particular task at hand. In practice, one may construct a single large-scale model to explain a phenomenon of interest, which may be utilized in a variety of settings. The latent variables of interest, which can differ in each setting, may only represent a small subset of all variables. A query that arises often is evaluating the marginals at specific nodes. The marginals at these nodes may change after the addition of measurements at different time points. In such adaptive settings, naïve algorithms, such as standard belief propagation (BP), may utilize many unnecessary computations by propagating messages over the entire graph.

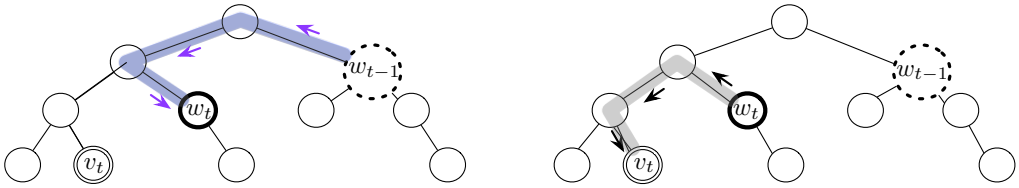
In this chapter, we formulate an efficient inference procedure, termed *adaptive BP* (AdaBP), suitable for adaptive inference settings. In other words, for settings where new observations are added sequentially and there is a need of evaluating statistics such as marginals while avoiding computing recurring quantities. This work is closely tied to work in Chap. 4, since after the selection of a measurement at a greedy step, we need to propagate the uncertainty at the next walk element. In that respect, we have measurements that are added sequentially (after the end of each greedy selection step) and there is a need of evaluating the covariance matrix (or else the precision matrix) of the latent node that corresponds to the next walk element.

We show that AdaBP gives exact results for trees in discrete and Gaussian Markov Random Fields (MRFs), and provide an extension to Gaussian loopy graphs. We additionally demonstrate that when the inference problem is finding the most likely sequence, the solution corresponds to the full latent graph, rather than just a small subset. Furthermore, we show that the problem of finding a nearly optimal schedule of measurements can be cast as a Traveling Salesman Problem (TSP). We compare the proposed method to standard BP and to that of Sümer et al. [89], which tackles the same problem. We show in synthetic and real experiments that it outperforms standard BP by orders of magnitude and explore the settings that it is advantageous over Sümer et al. [89]. An earlier version of this work was originally presented in [81].

We start the chapter with Sec. 5.1 by introducing the problem and motivating the need for an algorithm that would be better suited for adaptive inference settings, when there is a need for performing inference over a small subset of the hidden variables. We continue by presenting related work in Sec. 5.2. In Sec. 5.3, we formulate the problem and introduce necessary notation. Sec. 5.4 explains the concept of the lowest common ancestor of two nodes and a way to retrieve it in constant time by reducing this problem to the Range Minimum Query (RMQ) problem. We show in Sec. 5.5 how node potentials are updated upon the reception of a new measurement. Sec. 5.6 discusses the algorithm focusing on tree MRFs. A complexity analysis is given in Sec. 5.6.2. We analyze the case of multiple measurements and marginals per iteration in Sec. 5.7. Sec. 5.8 extends the method to finding the MAP sequence. It is worth noting that this extension recovers the MAP sequence on the full latent graph and not just a small subset of interest, as is the case when evaluating marginals. Sec. 5.9 generalizes the method to Gaussian loopy MRFs. We show that by using the algorithm proposed by Liu et al. [64], the solutions are still exact. Sec. 5.10 studies the reverse problem. In other words, when there is a fixed budget on the measurements we can retrieve and there is no preference on the order of obtaining them. Then, the goal is to design a measurement schedule that results in the minimum number of computations. We continue the chapter with Sec. 5.11 by demonstrating the strengths and weaknesses of our method on both synthetic and real data. Lastly, we conclude by summarizing the contributions in Sec. 5.12.

■ 5.1 Introduction

We consider the problem of inference in large-scale models. It is often the case that only a subset of latent variables is of interest for different applications which may vary from instance to instance. Additionally, the set of available measurements may vary with use or become available at different points in time. The latter is common for any sequential estimation problem. In such situations, general-purpose inference algorithms, such as BP may utilize many unnecessary computations when only a small subset is desired. There exist several examples that fall into this category of problems. Patient monitoring provides one such practical example [16]. Large-scale systems may monitor the health status of many patients; however, different physicians limit their interest to patients under their immediate care. In funding allocation, a funding agency might be interested in the expected impact that funding a particular research group has on a certain scientific topic [100]. Temperature monitoring sensors provide data over time and space, but sensitive areas (e.g., server room) may require more careful examination for the timely response in case of abnormal behavior. In computer vision, when image segmentation is performed on video frames, the data from frame to frame change slightly which might make possible the reuse of previous quantities to avoid recurring computations [49]. Lastly, in computational biology, the effects of mutations are explored (*computational mutagenesis*), with each putative mutation resulting in a



(a) Send messages from $w_{\ell-1}$ to w_ℓ. (b) Send messages from w_ℓ to v_ℓ.

Figure 5.1: Outline of AdaBP. (a) At every iteration, we send messages along the path between the previous measurement node $w_{\ell-1}$ and the current one w_ℓ. The path between $w_{\ell-1}$ and w_ℓ is depicted in purple color. (b) In the second phase, we send messages in the path between nodes w_ℓ and v_ℓ. This path is depicted in gray color. As we will show later, this schedule guarantees that all marginals on the path between w_ℓ and v_ℓ will be correct.

very similar problem [89].

This motivates methods for problems where measurements are added incrementally and the interest is in a subset of node marginals at a given time point or the MAP sequence of the full latent graph. This is the problem of *adaptive inference*, where the goal is to take advantage of previously computed quantities instead of performing inference from scratch. In these cases, standard BP results in many redundant computations. Consequently, we develop an adaptive inference approach which avoids redundant computations and whose average-case performance shows significantly lower complexity compared to BP. The main idea is to send only messages between the node where a measurement has been obtained from, w_ℓ, and the node whose marginal is of interest, v_ℓ.[1] The correctness of this approach is guaranteed by propagating messages between consecutive measurement nodes $w_{\ell-1}, w_\ell$ at every iteration. As a result, we only send the absolutely necessary messages to guarantee that the incoming messages to the node of interest v_ℓ are correct. We call this minimal messaging schedule *adaptive BP* or AdaBP in short. A brief outline of this algorithm is shown in Fig. 5.1. We show that it gives exact results on trees (as standard BP) and provide an extension for Gaussian loopy graphs that still guarantees exactness in the evaluation of marginals.

The proposed method requires a preprocessing step of $\mathcal{O}(N \log N)$ time, where N is the number of latent nodes. In the worst case, when relative distance between consecutive "measurement" nodes is approximately the diameter of the tree and the diameter is on the order of N (highly unbalanced tree), the performance is comparable – yet still faster than– standard BP. However, for height-balanced trees, worst-case performance results in $\mathcal{O}(\log N)$ messages per update as compared to $\mathcal{O}(N)$ for standard BP. If nodes w_ℓ, v_ℓ are close to each other, the computation of the node marginal is obtained in constant time per iteration. We provide an extension of the method for MAP inference and for Gaussian loopy MRFs and show how it can be used to suggest nearly optimal measurement schedules. We compare the proposed method to Sümer et al. [89]

[1] We will refer to w_ℓ as "measurement" node for abbreviation.

and examine settings under one approach may have advantages over the other. Lastly, we empirically demonstrate the performance of our method in a variety of synthetic datasets, as well in two real applications.

■ 5.2 Related Work

Kohli and Torr [49] consider an adaptive inference setting where nodes and edges can be deleted or added at any time. They only consider the MAP inference problem and show that a solution can be obtained in polynomial time by solving a dynamic version of the st-mincut problem. However, their method is restricted to discrete variables and to submodular pairwise factors. Komodakis et al. [50] also analyze the problem of MAP inference in dynamic settings. They cast the MAP problem as a primal-dual optimization problem, and solve a series of max-flow problems, where the number of augmenting paths per max-flow runs decreases over time. This last attribute guarantees efficiency of inference. Even though their method is applied to a wider range of MRFs, it only addresses the MAP problem and does not generalize to the Gaussian case. Nath and Domingos [76] propose a variant of BP, termed *expanding frontier belief propagation*, where messages are only propagated in a small subset of nodes in the close region around a node whose potential has changed. They additionally provide guarantees on performance of their method relative to standard loopy BP.

Chechetka and Guestrin [16] examine the problem of inference over a fixed set of nodes, $\mathcal{Q}$, called the *query set*. They create a prioritized message schedule weighted towards messages to which set $\mathcal{Q}$ is most sensitive. Their approach is limited to discrete graphs, the query set is fixed and the preprocessing time depends on the number of edges and neighbors of the nodes. Wick and McCallum [100] consider the same problem of focused inference but from an MCMC perspective. They use Metropolis Hastings to draw samples that come more frequently from the query variables. They achieve this by creating variable selection distribution that favors the selection of query variables as well as those variables that are highly influential to the query variables.

Lastly, Sümer et al. [89] analyze the problem of adaptive exact inference in the context of factor graphs utilizing the factor elimination algorithm to evaluate node marginals [23]. They construct a balanced representation of the elimination tree in $\mathcal{O}(|\mathcal{X}|^{3w}N)$ time, which allows for computation of a node marginal in $\mathcal{O}(|\mathcal{X}|^{2w}\log N)$, where N is the number of nodes, w is the elimination tree width (size of the largest clique in the chordal graph minus one) and $|\mathcal{X}|$ the alphabet size. However, the preprocessing step becomes prohibitive as the alphabet $|\mathcal{X}|$ and treewidth w grow large, thus making this method inappropriate for dense loopy graphs. For trees, the width of the elimination tree is one and the complexity of updating the model reduces to $\mathcal{O}(|\mathcal{X}|^3\log N)$ as compared to $\mathcal{O}(|\mathcal{X}|N)$ for standard BP. Note that they address the discrete case only. As we later show, the computational complexity is impacted significantly by not taking into account the relative distances between consecutive nodes of interest. In contrast, our proposed method extends to loopy Gaussian models, has a

much reduced pre-processing time, and allows for varying sets of interest.

■ 5.3 Problem Statement

We consider the Markov Random Field (MRF) which represents a graph $G = (V, \mathcal{E})$ of N latent variables, $\mathsf{X} = \{\mathsf{X}_1, \ldots, \mathsf{X}_N\}$, whose direct dependencies are represented by edge set $\mathcal{E}$. The neighbors of latent node X_t are denoted by $N(t)$ and each latent node X_t is linked to m_t measurements $\{\mathsf{Y}_{t,1:m_t}\}$. The set $\{\mathsf{Y}_{t,1:m_t}\}$ will be called *observation set* and denoted by $\mathcal{V}_t$. In addition, each $X_t \in \mathcal{X}$. A feedback vertex set (FVS) $\mathcal{F}$ is a set of nodes whose removal results in a cycle–free graph $\mathcal{T} = V \setminus \mathcal{F}$ (forest). Obviously, $\mathcal{F} = \emptyset$ in the case of trees. We denote $|\mathcal{F}| = K$ to be the size of FVS. We would also call all nodes in $\mathcal{T}$ that are neighbors to an FV node as *anchors* and denote them by $\mathcal{A}$. That is, $\mathcal{A} = \{i \mid i \in \mathcal{T}, i \in N(p), \forall p \in \mathcal{F}\}$. For the purpose of our analysis, we focus on discrete MRFs, but the proposed method generalizes straightforwardly to Gaussian MRFs. The only difference is in the inherent complexity of a message; $\mathcal{O}(|\mathcal{X}|^2)$ for discrete, $\mathcal{O}(d^3)$ for Gaussian, where $|\mathcal{X}|$ is the alphabet size and d the dimension of X, respectively. Lastly, a common assumption is that measurements Y are conditionally independent on X. We focus on pairwise MRFs as MRFs with larger cliques can be reduced to pairwise ones [97]. We are interested in problems where a measurement is added at a time and only one or a few marginals are of interest at any point. The total number of available measurements is M. We are given a *measurement plan* $\boldsymbol{w} = \{w_1, \ldots, w_M\} = \{w_{1:M}\}$, which provides the order of taking measurements from each set. That is, a measurement is obtained from set $\mathcal{V}_{w_1}$, then from $\mathcal{V}_{w_2}$, and so on. We call *marginal order* $\boldsymbol{v} = \{v_{1:M}\}$, the sequence of the latent nodes whose marginal is of interest at each step.

■ 5.4 Lowest Common Ancestor (LCA)

One efficient approach to evaluate marginals in graphical models is the belief propagation algorithm, which is exact for trees. When we acquire a new measurement, which updates a node's potential, we can find the marginal of a node of interest by treating this node as a root and then propagating messages from the leaves to the root. When a node's potential is updated, this changes all the outgoing messages of this node, which in effect change the messages of this node's neighbors and so on. The question that arises is if we can avoid repeated computations by running standard BP every time a new measurement arrives. A potentially optimal approach is to pass messages from the measurement node to the node of interest, but without any bookkeeping this will not take into account the effect that past measurements had in the propagated messages. We will show in Sec. 5.6 how we can tweak the above idea and produce correct marginal estimates after a change in a node's potential. For now, we will assume that we need to send messages from the measurement node to the node of interest. This would require knowledge of the path that connects these two nodes. The lowest common ancestor (lca) of two nodes is the shared ancestor of these nodes that is located farthest from the root. The lca is directly related to the path between two nodes as the path can be

determined by traversing from a node up to the lca and then down to the other node. Since we consider problems where node potentials are updated frequently, we need a method that determines paths (or in other words, lcas) between nodes in a very efficient way. It turns out that the determination of the lowest common ancestor between any two nodes can occur in $\mathcal{O}(1)$ time by reducing the LCA problem to the Minimum Range Query (RMQ) problem as we describe below.

The RMQ solves the problem of finding the index of the minimum element between two specified indices of an array $\boldsymbol{A}[i \dots j]$. Interestingly, they provide an answer in constant time by building a structure $\boldsymbol{M}$ of size $N \times L$, where $L = \lceil \log_2 N \rceil + 1$ [39]. The element $[\boldsymbol{M}]_{i,j}$ gives the index of the minimum element in $\boldsymbol{A}[i \dots i + 2^{j-1} - 1]$. The RMQ is extremely well-suited for problems with a large number of queries, R, where $R \gg N$, since it is linear in R. It turns out that the LCA can be reduced to the RMQ problem [22, 33]. For a specified root, we assume that each node is labeled in a breadth-first manner. That is, the root is assigned label 1, and all other nodes are labeled accordingly in a top-down, left-right approach (cf. Fig. 5.2). Now, suppose we recover the Euler tour $\boldsymbol{E}$ of the tree starting from the root. As a reminder, the *Euler tour* of a strongly connected, directed graph $\mathcal{G}$ is a cycle that traverses each edge of $\mathcal{G}$ exactly once, although it may visit a node more than once [19]. Since we are dealing with undirected graphs here, we assume for the purposes of analysis that each undirected edge is equivalent to two directed edges of opposing direction. The number of edges arriving at a node is called the *in-degree*, while the number of edges leaving a node is called the *out-degree*. Since by construction each node has equal out- and in-degree, the Euler tour is always a cycle, that is, it starts and ends on the same node (here, the root). Now, if we denote by $\boldsymbol{H}$ the vector which stores the index of the first occurrence of each node in $\boldsymbol{E}$, the lca of two nodes u, v would be somewhere in $\boldsymbol{E}[\boldsymbol{H}_u, \dots, \boldsymbol{H}_v]$ due to the way the Euler tour is constructed (depth-first manner). Since the nodes are labeled in a breadth-first manner, the lca of u, v would be the one with the smallest label and hence the smallest depth in the range $\boldsymbol{E}[\boldsymbol{H}_u, \dots, \boldsymbol{H}_v]$. It becomes apparent that we need to introduce a vector $\boldsymbol{D}_e$ which would store the depth of the corresponding nodes in the Euler tour. For example, the depth of the first node in the Euler tour, which is the root by construction, is $[\boldsymbol{D}_e]_1 = 0$. Since the lca of u, v is the node with the smallest depth in $\boldsymbol{E}[\boldsymbol{H}_u, \dots, \boldsymbol{H}_v]$, the index of the minimum element of subarray $\boldsymbol{D}_e[\boldsymbol{H}_u, \dots, \boldsymbol{H}_v]$ would give us the $\text{lca}(u, v)$. We explain the quantities $\boldsymbol{E}, \boldsymbol{D}_e, \boldsymbol{H}$ with an example in Fig. 5.2.

It remains now to build a matrix $\boldsymbol{M}$ that would provide answers to queries of the type $\arg\min \boldsymbol{D}_e[\boldsymbol{H}_u, \dots, \boldsymbol{H}_v]$ in constant time. The size of this matrix would be $N \times L$, where $L = \lceil \log_2 N \rceil + 1$, while element $[\boldsymbol{M}]_{i,j}$ would represent the index of the minimum element of the subarray $\boldsymbol{D}_e$ that starts at i and has length 2^{j-1}:

$$[\boldsymbol{M}]_{i,j} = \begin{cases} [\boldsymbol{M}]_{i,j-1} & , [\boldsymbol{D}_e]_{[\boldsymbol{M}]_{i,j-1}} \leq [\boldsymbol{D}_e]_{[\boldsymbol{M}]_{r,j-1}} \\ [\boldsymbol{M}]_{r,j-1} & , \text{otherwise,} \end{cases}$$

where $i = 1, \dots, N$, $j = 1, \dots, L$, $r = \min\{i + 2^{j-2}, N\}$ and $[\boldsymbol{M}]_{i,1} = i$.

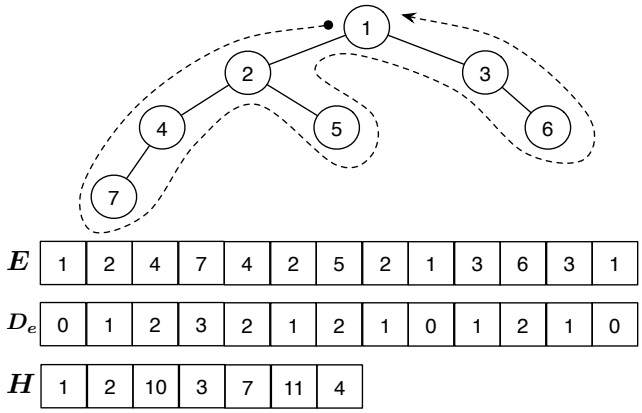


Figure 5.2: Reduction from LCA to RMQ problem. The dashed arrows denote the Euler tour. The black circle denotes the beginning and the arrow the end of the tour. Note that the Euler tour begins and ends in the root. The array $\boldsymbol{E}$ gives the order in which nodes are encountered in the Euler tour. The array $\boldsymbol{D}_e$ gives the depth of each node in the Euler tour, while array $\boldsymbol{H}$ gives the index of the first occurrence of a node in the Euler tour. For example, the index of node's 3 first occurrence in the Euler tour is 10. In other words, $\boldsymbol{E}_{10} = 3$.

The absolute index of the minimum value of the subarray $\boldsymbol{D}_e[i, \ldots, j]$ is recovered in constant time as

$$\mathrm{RMQ}_{\boldsymbol{D}_e}(i,j) = \begin{cases} [\boldsymbol{M}]_{i,k+1} & , [\boldsymbol{D}_e]_{[\boldsymbol{M}]_{i,k+1}} \leq [\boldsymbol{D}_e]_{[\boldsymbol{M}]_{s,k+1}} \\ [\boldsymbol{M}]_{j-2^k+1,k+1} & , \text{otherwise,} \end{cases}$$

where $k = \lfloor \log_2(j - i + 1) \rfloor$.

The lca of u, v is simply

$$\mathrm{lca}(u,v) = \begin{cases} \boldsymbol{E}_{\mathrm{RMQ}_{\boldsymbol{D}_e}(\boldsymbol{H}_u, \boldsymbol{H}_v)} & , \boldsymbol{H}_u < \boldsymbol{H}_v \\ \boldsymbol{E}_{\mathrm{RMQ}_{\boldsymbol{D}_e}(\boldsymbol{H}_u, \boldsymbol{H}_v)} & , \text{otherwise.} \end{cases} \tag{5.1}$$

■ 5.5 Updating node potentials

Updating a node's potential is really straightforward. We consider three types of potentials; node potentials of latent variables $\varphi_t^{(0)}(x_t)$, pairwise potentials between latent and observed variables $\chi_{t\ell}(x_t, y_\ell)$, and pairwise potentials between latent variables $\psi_{ij}(x_i, x_j)$. In the discrete case, an observed variable $\mathsf{Y}_{t\ell} = y_\ell$ is embedded into the node potential of the latent variable it links to as

$$\varphi_t(x_t) = \varphi_t^{(0)}(x_t)\chi_{t\ell}(x_t, y_\ell) \tag{5.2}$$

In the gaussian case, the node potential of variable X_t has the initial form $\chi_t(x_t) = \exp(x_t^T h_t^{(0)} - \frac{1}{2} x_t^T J_{tt}^{(0)} x_t)$, while pairwise potentials between latent variables take the

form $\psi_{ij}(x_i, x_j) = \exp(-x_i^T J_{ij} x_j)$. If a measurement Y_t is obtained from X_t as

$$Y_t = C_t X_t + W_t,$$

where $W_t \sim N(0, R_t)$, then the posterior distribution of the vector of variables $X_1, \ldots, X_N$ given Y_t takes the form

$$\begin{aligned}
p(x \mid y_t) &\propto p(y_t \mid x)p(x) = \exp(-\frac{1}{2}(y_t - C_t x_t)^T R_t^{-1}(y_t - C_t x_t)) \exp(-\frac{1}{2}x^T J x + x^T h) \\
&= \exp(-\frac{1}{2}(y_t - C_t x_t)^T R_t^{-1}(y_t - C_t x_t)) \exp(-\frac{1}{2}x_t^T J_{tt}^{(0)} x_t + x_t^T h_t^{(0)}) \\
&\quad \cdot \exp(-\frac{1}{2}\sum_{i \neq t} x_i^T J_{ii}^{(0)} x_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j \in N(i)} x_i^T J_{ij} x_j + \sum_{i \neq t} x_i^T h_i^{(0)}) \\
&\overset{(a)}{=} \exp(-\frac{1}{2}x_t^T (J_{tt}^{(0)} + C_t^T R_t^{-1} C_t) x_t + x_t^T (h_t^{(0)} + C_t^T R_t^{-1} y_t) \\
&\quad \cdot \exp(-\frac{1}{2}\sum_{i \neq t} x_i^T J_{ii}^{(0)} x_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j \in N(i)} x_i^T J_{ij} x_j + \sum_{i \neq t} x_i^T h_i^{(0)}),
\end{aligned} \tag{5.3}$$

where (a) has been obtained after isolating from $p(y_t \mid x)$ the terms that contain x_t. It becomes clear from Eq. (5.3), that a measurement which is drawn from a variable X_t affects only the potential of that variable, as in the discrete case.

$$J_{tt} = J_{tt}^{(0)} + C_t^T R_t^{-1} C_t \tag{5.4}$$

$$h_t = h_t^{(0)} + C_t^T R_t^{-1} y_t \tag{5.5}$$

The update of a node potential in the discrete case takes $\mathcal{O}(|\mathcal{X}|)$ time, while $\mathcal{O}(d^2 m)$ in the Gaussian case, where d is the dimension of X_t and m the dimension of Y_t (assuming $m < d$).

■ 5.6 Adaptive BP

For the purpose of analysis, we would delay the discussion to general Gaussian MRFs. We will consider trees here and present an extension later to loopy graphs in Sec. 5.9. As a reminder, we obtain one measurement at every step and are interested in characterizing the belief at a given node. Recall that a measurement order $\boldsymbol{w} = \{w_1, \ldots, w_M\}$ is the order that measurements are obtained. In addition, sequence $\boldsymbol{v} = \{v_1, \ldots, v_M\}$ determines the marginals of interest at any time. The key idea is to propagate messages in the paths $(w_{\ell-1}, w_\ell)$ and $(w_\ell, v_\ell), \forall \ell$. We show that by propagating messages between consecutive measurement nodes, the messages take into account the information of all past measurements. The discovery of these paths is directly related to finding the *lowest common ancestor* (lca) of pairs $(w_{\ell-1}, w_\ell)$ and $(w_\ell, v_\ell), \forall \ell$, which we discussed in Sec. 5.4.

5.6.1 Method Description

With a careful inspection, we observe that after the incorporation of a measurement at node X_{w_ℓ}, the evaluation of the messages along the unique path from node w_ℓ to node v_ℓ is sufficient for the determination of node v_ℓ's marginal. The above procedure guarantees to give the correct marginals along this path as long as all the incoming messages to node w_ℓ are correct. This is possible, if we additionally propagate messages from $w_{\ell-1}$ to w_ℓ at every iteration. The algorithm is described as follows. During initialization, all the node potentials are evaluated assuming no measurements are available. If measurements are already available, they are absorbed in the corresponding latent node potentials, as described above. At this point, we propagate messages along the entire graph in both directions. As a new measurement arrives from set $\mathcal{V}_{w_1}$, the messages from w_1 to v_1 are computed. This way, the marginals of the nodes in the path (incl. w_1, v_1) are correctly updated. Then, we propagate messages from w_1 to w_2, update the node potential of X_{w_2} and send messages from w_2 to v_2. We continue with this procedure for each ℓ. If $w_\ell = w_{\ell-1}$, no messages are propagated from $w_{\ell-1}$ to w_ℓ, while if $w_\ell = v_{\ell-1}$, only the node potential X_{w_ℓ} is updated. Obviously, the path from node $w_{\ell-1}$ to w_ℓ is directly related to the $\text{lca}(w_{\ell-1}, w_\ell)$. Similarly, for the pair (w_ℓ, v_ℓ). Therefore, at every iteration, we need to determine the lcas of these two pairs, which is accomplished in constant time, with the reduction to the RMQ problem. Once we find the lca of pair $(w_{\ell-1}, w_\ell)$, we can trivially determine the directed path from $w_{\ell-1}$ to w_ℓ. We will denote the messages in this path by $\mathcal{M}(w_{\ell-1} \to w_\ell)$. Similarly, we denote the messages of the directed path from w_ℓ to v_ℓ by $\mathcal{M}(w_\ell \to v_\ell)$. Note here that both of the above schedules contain only the single-direction messages from one node to another. The update is done in the same manner as in the serial version of BP, that is, we propagate messages from $w_{\ell-1}$ to the $\text{lca}(w_{\ell-1}, w_\ell)$ and then down to w_ℓ. The procedure is the same for the pair (w_ℓ, v_ℓ). The flow of the algorithm is depicted in Fig. 5.3 (see Alg. 5.1 for details).

Messages are updated as:

$$m_{i\to j}(x_j) = \sum_{x_i} \varphi_i(x_i)\psi_{ij}(x_i, x_j) \prod_{k\in N(i)\setminus j} m_{k\to i}(x_i), \quad \forall (i,j) \in \mathcal{M}(w_{\ell-1} \to w_\ell) \text{ and } \mathcal{M}(w_\ell \to v_\ell), \tag{5.6}$$

while the marginal of node of interest v_ℓ is computed as

$$p_{\mathsf{X}_{v_\ell}}(x_{v_\ell}) \propto \varphi_{v_\ell}(x_{v_\ell}) \prod_{k\in N(v_\ell)} m_{k\to v_\ell}(x_{v_\ell}). \tag{5.7}$$

Obviously, if the model is Gaussian, we use the formulas for Gaussian BP as indicated in Eqs. (2.109), (2.110), (2.111), (2.112). If the latent graph is a chain, there is no need to determine the lca at every step: we simply propagate from $w_{\ell-1}$ to w_ℓ, update the node potential at w_ℓ and propagate messages to node v_ℓ.

By applying this algorithm, we guarantee exactness in the marginals of nodes on the path $\mathcal{M}(w_\ell \to v_\ell)$.

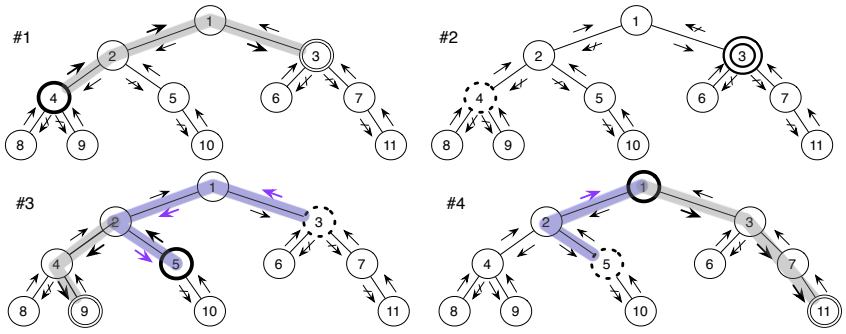


Figure 5.3: Adaptive BP flow. The solid thick node represents node w_ℓ, double stroke node v_ℓ, while dashed node previous node $w_{\ell-1}$. Purple bands encompass the messages sent between $w_{\ell-1}$ and w_ℓ, while gray bands messages between w_ℓ and v_ℓ. Thick arrows represent messages at the current iteration (purple ones transmitted between $w_{\ell-1}, w_\ell$ and black ones between w_ℓ, v_ℓ). Solid and strikethrough arrows represent correct and incorrect messages, respectively, computed from previous iterations. At iteration #1, measurement node $w_1 = 4$ sends messages to node of interest $v_1 = 3$. At iteration #2, since $w_2 = v_2 = 3$, no messages need to be sent. At iteration #3, messages are sent from past measurement node $w_2 = 3$ to current measurement node $w_3 = 5$ to guarantee consistency and then messages are passed from $w_3 = 5$ to node of interest $v_3 = 9$. Similarly, for iteration #4.

Theorem 5.6.1. *The marginals of all nodes in $\mathcal{M}(w_\ell \to v_\ell)$ are correct.*

To prove the above statement, we need first to prove a few intermediate lemmas as that we include below.

Lemma 5.6.1. *Messages in path $\mathcal{M}(w_{\ell-1} \to w_\ell)$ are correct.*

Proof. Base case: The messages in the path $w_1 \to w_2$ are correct. This is trivially true since all the incoming messages to w_1 and to the nodes in the path $\mathcal{M}(w_1 \to w_2)$ have been correctly evaluated during initialization. Therefore, after we absorb the measurement in the potential of X_{w_1}, propagating from $w_1 \to w_2$ will give us the correct messages. *Induction step:* We will assume now that the messages in $\mathcal{M}(w_{j-1} \to w_j)$, $j \in \{2, \ldots, \ell-1\}$ are correct and we will show that the messages in $\mathcal{M}(w_{\ell-1} \to w_\ell)$ will be correct as well. W.l.o.g. assume the tree is rooted at $w_{\ell-1}$ as shown in Fig. 5.4 and i is one of $w_{\ell-1}$'s neighbors. We need to show that all the incoming messages to $w_{\ell-1}$ as well as the incoming messages to the other nodes in $\mathcal{M}(w_{\ell-1} \to w_\ell)$ are correct. Let us first show that the incoming messages to $w_{\ell-1}$ are correct. There are three cases for the subtree $\mathcal{T}_i$ rooted at i (if we ignore the branch containing the edge $(i, w_{\ell-1})$): (a) there are no previous measurements $\{w_1, \ldots, w_{\ell-2}\}$ from it, (b) the last measurement from it was taken at time $t_i < \ell - 2$, or (c) at time $t_i = \ell - 2$. In the first case (a), since there are no previous measurements, the incoming message $m_{i \to w_{\ell-1}}$ stayed intact since initialization and thus is correct. In the second case (b), since $t_i < \ell - 2$, this

Algorithm 5.1 Adaptive Belief Propagation

Preprocessing
Determine Euler tour $\boldsymbol{E}$, depths of elements in the Euler tour $\boldsymbol{D}_e$, vector $\boldsymbol{H}$, which stores the index of the first occurrence of node i in $\boldsymbol{E}$, and matrix $\boldsymbol{M}$, which stores the index of the minimum value of the subarray of $\boldsymbol{D}_e$ starting at i and having length 2^{j-1}.
Initialization
Initialize the node, pairwise potentials and messages.
Iteration
for $\ell = 1, \ldots, M$ **do**
  Find $\text{lca}(w_{\ell-1}, w_\ell)$ from Eq. (5.1).
  Determine schedule $\mathcal{M}(w_{\ell-1} \to w_\ell) : w_{\ell-1} \to \text{lca}(w_{\ell-1}, w_\ell) \to w_\ell$
  Compute messages $m_{i\to j}(x_j)$ in $\mathcal{M}(w_{\ell-1} \to w_\ell)$ from Eq. (5.6).
  Update the node potential at X_{w_ℓ}.
  Find $\text{lca}(w_\ell, v_\ell)$ from Eq. (5.1).
  Determine schedule $\mathcal{M}(w_\ell \to v_\ell) : w_\ell \to \text{lca}(w_\ell, v_\ell) \to v_\ell$
  Compute messages $m_{i\to j}(x_j)$ in $\mathcal{M}(w_\ell \to v_\ell)$ from Eq. (5.6).
  Compute the marginal of interest $p_{\mathsf{X}_{v_\ell}}(x_{v_\ell})$ from Eq. (5.7).
end for

means that at point $t_i + 1$, we moved to a subtree of another neighbor of $w_{\ell-1}$ through $w_{\ell-1}$. Due to our assumption, that all messages from previous paths $\mathcal{M}(w_{j-1} \to w_j)$, $j < \ell$, are correct, this also implies that the messages in the path $\mathcal{M}(w_{t_i} \to w_{t_i+1})$ are correct and this includes message $m_{i\to w_{\ell-1}}$ as well. Lastly, if $t_i = \ell - 2$, this means that the previous measurement, at time $\ell - 2$, was taken from the subtree rooted at i (c). By assumption, all messages in $\mathcal{M}(w_{\ell-2} \to w_{\ell-1})$ are correct. So, in all cases, the incoming message from i to $w_{\ell-1}$ is correct. We follow similar logic for all neighbors of $w_{\ell-1}$. Lastly, we should demonstrate that the incoming messages to the other nodes in the path $\mathcal{M}(w_{\ell-1} \to w_\ell)$ are correct. The logic is similar as before. Let us refer to the subtrees that are attached to the path $\mathcal{M}(w_{\ell-1} \to w_\ell)$ as *tree branches*. Take a node (call it j) attached to $\mathcal{M}(w_{\ell-1} \to w_\ell)$ and consider the subtree $\mathcal{T}_j$ rooted at it. Let us denote by k the node in the path $\mathcal{M}(w_{\ell-1} \to w_\ell)$ that j links to, as show in Fig. 5.4. As before, we have three cases: (a) there are no previous measurements taken from $\mathcal{T}_j$, (b) the last measurement was taken at time $t_j < \ell - 2$, or (c) at time $t_j = \ell - 2$. If there are no previous measurements (a), this means that the message $m_{j\to k}$ stayed intact since initialization. If $t_j < \ell - 2$ (b), then at point t_{j+1} we "exited" subtree $\mathcal{T}_j$ through node k and moved either to another branch of that path or to another subtree of $w_{\ell-1}$. In either case, due to our assumption, the messages in $\mathcal{M}(w_{t_j} \to w_{t_j+1})$ are correctly updated including message $m_{j\to k}$. Lastly, if $t_j = \ell - 2$ (c), then due to our assumption, the messages $\mathcal{M}(w_{\ell-2} \to w_{\ell-1})$ are correct, including the message $m_{j\to k}$. We reason similarly for all nodes which are part of $\mathcal{M}(w_{\ell-1} \to w_\ell)$. Therefore, since all incoming messages to $w_{\ell-1}$ and nodes in $\mathcal{M}(w_{\ell-1} \to w_\ell)$ are correct, the messages in

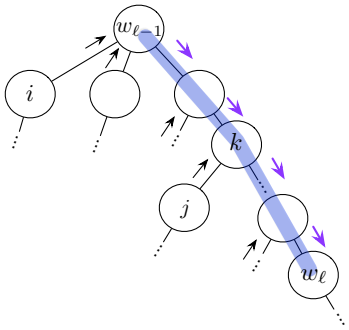


Figure 5.4: Correctness of message updates. Purple thick arrows represent the messages that will be propagated in the current iteration from $w_{\ell-1} \to w_\ell$, while solid black arrows the incoming messages to $\mathcal{M}(w_{\ell-1} \to w_\ell)$ which have been evaluated correctly from previous iterations.

$\mathcal{M}(w_{\ell-1} \to w_\ell)$ would also be correct. ■

Lemma 5.6.2. *The incoming messages of each node in $\mathcal{M}(w_{\ell-1} \to w_\ell)$ are correct.*

Proof. We denote by k a node in $\mathcal{M}(w_{\ell-1} \to w_\ell)$ and by j one of its neighbors, $j \in N(k)$, as shown in Fig. 5.5. Denote further by $\mathcal{T}_j$ the tree that is rooted at j if we exclude the tree branch that contains the edge (j,k). We define as $t(k,j)$ the most recent time that a measurement has been obtained from tree $\mathcal{T}_j$. By default, if no measurement has been obtained from $\mathcal{T}_j$, we set $t(k,j) = \infty$. Also, if $\mathcal{T}_j$ includes node $w_{\ell-1}$, then obviously $t(k,j) = \ell - 1$. From the definition of $t(k,j)$, which indicates the time that the last measurement has been obtained from $\mathcal{T}_j$, we have that at time $t(k,j)+1$ we exited the tree $\mathcal{T}_j$ through edge (j,k). Due to Lem. 5.6.1, all messages in $\mathcal{M}(w_{t(k,j)} \to w_{t(k,j)+1})$, including message $m_{j\to k}$ are correct. We follow the same logic for all neighbors of k. ■

Lemma 5.6.3. *The incoming messages of each node in $\mathcal{M}(w_\ell \to v_\ell)$ are correct.*

Proof. We will first start by showing that the incoming messages of the neighbor of w_ℓ in path $\mathcal{M}(w_\ell \to v_\ell)$ are correct. Then, we can show with a similar logic that all the incoming messages of the remaining nodes in $\mathcal{M}(w_\ell \to v_\ell)$ are correct as well. From Lem. 5.6.2, we showed that the incoming messages of all nodes in $\mathcal{M}(w_{\ell-1} \to w_\ell)$ are correct. This includes node w_ℓ. Let us denote by k the neighbor of w_ℓ in $\mathcal{M}(w_\ell \to v_\ell)$. Since, the incoming messages of w_ℓ are correct, after the update of w_ℓ's potential, it follows from relation $m_{w_\ell \to k}(x_k) = \sum_{x_{w_\ell}} \varphi_{w_\ell}(x_{w_\ell}) \psi_{w_\ell,k}(x_{w_\ell}, x_k) \prod_{s \in N(w_\ell)\setminus k} m_{s\to w_\ell}(x_{w_\ell})$, that message $m_{w_\ell \to k}$ is correct as well. Now, let us denote by j a neighbor of k (other than w_ℓ), and by $\mathcal{T}_j$ the tree rooted at j that does not include the tree branch that contains edge (j,k), as shown in Fig. 5.6. Again, $t(k,j)$ denotes the most recent time a measurement has been obtained from tree $\mathcal{T}_j$. If $w_{\ell-1}$ is contained in tree $\mathcal{T}_j$,

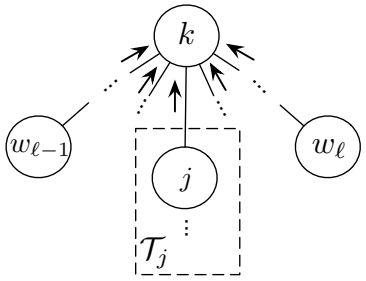


Figure 5.5: Correctness of incoming messages in $\mathcal{M}(w_{\ell-1} \to w_\ell)$. The incoming messages of every node in $\mathcal{M}(w_{\ell-1} \to w_\ell)$ are correct. Here, k is a node in $\mathcal{M}(w_{\ell-1} \to w_\ell)$ and $j \in N(k)$, while $\mathcal{T}_j$ is the tree rooted at j, if exclude the tree branch that contains edge (j, k).

then $t(k, j) = \ell - 1$ (Fig. 5.6a), if no measurement has been obtained from $\mathcal{T}_j$, then $t(k, j) = \infty$ (Fig. 5.6c), and $t(k, j) = t < \ell - 1$, otherwise (Fig. 5.6b). If $t(k, j) = \ell - 1$, then during propagation $\mathcal{M}(w_{\ell-1} \to w_\ell)$, message $m_{j\to k}$ has been correctly updated as part of the schedule $\mathcal{M}(w_{\ell-1} \to w_\ell)$. If $t(k, j) = t < \ell - 1$, this means that at time $t(k, j) + 1$, we exited tree $\mathcal{T}_j$ through edge (j, k). Hence, message $m_{j\to k}$ has been correctly updated during schedule $\mathcal{M}(w_{t(k,j)} \to w_{t(k,j)+1})$. Lastly, if $t(k, j) = \infty$, this means that no measurement has been obtained from tree $\mathcal{T}_j$, and hence message $m_{j\to k}$ stayed intact since initialization. This obviously holds for every neighbor j of k. We have established that all incoming messages to k, with k being the direct neighbor of w_ℓ in $\mathcal{M}(w_\ell \to v_\ell)$ are correct. Therefore, the message from k to its other neighbor in $\mathcal{M}(w_\ell \to v_\ell)$ would also be correct, since all the incoming messages to k are correct. We argue that all the incoming messages to k's neighbor in $\mathcal{M}(w_\ell \to v_\ell)$ are correct in exactly the same fashion we argued for k. By following this logic, we show that the incoming messages of all nodes in $\mathcal{M}(w_\ell \to v_\ell)$ are correct. ■

Theorem 5.6.1. *The marginals of all nodes in $\mathcal{M}(w_\ell \to v_\ell)$ are correct.*

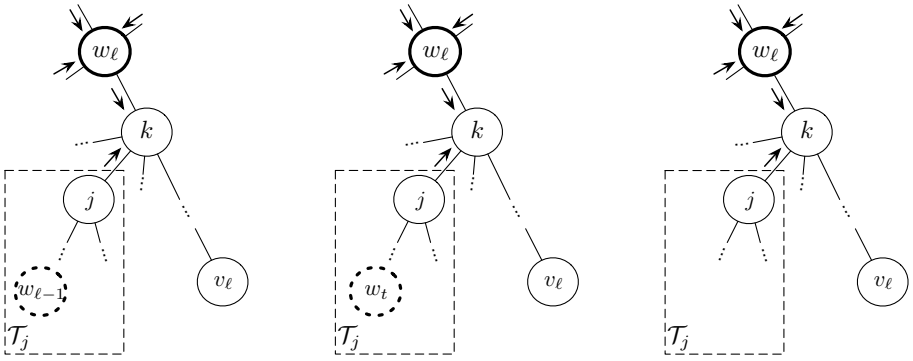
Proof. Since the marginal at a node i is given by

$$p_{\mathsf{X}_i}(x_i) \propto \varphi_i(x_i) \prod_{k \in N(i)} m_{k\to i}(x_i),$$

and by Lem. 5.6.3 all incoming messages to a node in $\mathcal{M}(w_\ell \to v_\ell)$ are correct, then the marginal at node $i \in \mathcal{M}(w_\ell \to v_\ell)$ will also be correct. ■

■ 5.6.2 Complexity

If the depth of each node (D) is not known in advance, it can be retrieved in $\mathcal{O}(N)$ time, in a depth-first approach. Similarly, the Euler tour is also retrievable in linear time. The same holds for vectors $\boldsymbol{D}_e$ and $\boldsymbol{H}$. Lastly, the creation of matrix $\boldsymbol{M}$, takes $\mathcal{O}(N \log_2 N)$

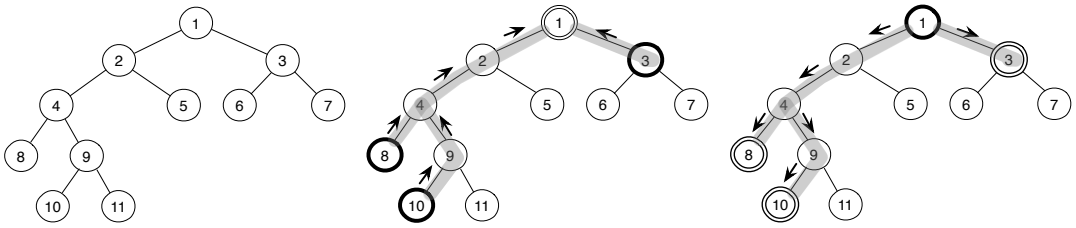


(a) $t(k,j) = \ell - 1$ (b) $t(k,j) = t$ (c) $t(k,j) = \infty$

Figure 5.6: Correctness of incoming messages of nodes in $\mathcal{M}(w_\ell \to v_\ell)$. The incoming messages of every node in $\mathcal{M}(w_\ell \to v_\ell)$ are correct. Tree $\mathcal{T}_j$ represents the tree rooted at node j, if we exclude the brach that contains edge (j,k). (a) Node $w_{\ell-1}$ is included in $\mathcal{T}_j$. (b) The most recent measurement from $\mathcal{T}_j$ has been taken at time $t(k,j) < \ell - 1$. (c) No measurements have been received from tree $\mathcal{T}_j$.

time and space. Therefore, the overall complexity of preprocessing is $\mathcal{O}(N \log_2 N)$. For adaptive BP, we only need to send messages along the directed paths $\mathcal{M}(w_{\ell-1} \to w_\ell)$ and $\mathcal{M}(w_\ell \to v_\ell)$. The number of messages to be sent in step ℓ is $\text{dist}(w_{\ell-1}, w_\ell) + \text{dist}(w_\ell, v_\ell)$.[2] The overall complexity, is $\mathcal{O}(\sum_{\ell=1}^{M}(\text{dist}(w_{\ell-1}, w_\ell) + \text{dist}(w_\ell, v_\ell))|\mathcal{X}|^2)$. Compare this with standard BP, where $N-1$ messages are sent at each iteration resulting in an overall complexity of $\mathcal{O}\left(\sum_{k=1}^{N} m_k N|\mathcal{X}|^2\right) = \mathcal{O}\left(mN^2|\mathcal{X}|^2\right)$, assuming the number of measurements from each set is the same, $m_k = m, \forall k$. As we see, the complexity of adaptive BP directly depends on the context of the measurement and marginal order, while standard BP has a fixed cost per iteration. We will analyze the worst, best and average complexity of adaptive BP for balanced and unbalanced trees. In the worst-case, when the tree is highly unbalanced (tree diameter on the order of N) and the relative distance between $(w_{\ell-1}, w_\ell)$, (w_ℓ, v_ℓ) is comparable to the diameter for all ℓ, we need to transmit $\mathcal{O}(N)$ messages at every iteration. In this case, the order of the number of messages to be sent is the same with standard BP. If, instead, the latent graph is a balanced tree, with each node having approximately q children, $\mathcal{O}\left(\lfloor \log_q N \rfloor\right)$ messages are propagated at every iteration in the worst case. In the best-case scenario, if $w_{\ell-1}, w_\ell, v_\ell$ are akin (e.g., parent-child or siblings) for every ℓ, then only one or two messages are propagated at every iteration, which reduces the overall complexity to just $\mathcal{O}\left(mN|\mathcal{X}|^2\right)$. As expected, when there is small distance between pairs of nodes $(w_{\ell-1}, w_\ell)$, (w_ℓ, v_ℓ), the complexity is substantially reduced. Complexity only depends on the relative distance between consecutive terms. Structure comes only into consideration, in the worst case, when the relative distance between $(w_{\ell-1}, w_\ell)$ and (w_ℓ, v_ℓ)

[2]The distance between nodes w, v is the length of the path connecting them and equals $\text{dist}(w,v) = D_v + D_w - 2D_{\text{lca}(w,v)}$.



(a) Original graph. (b) Multiple w_ℓ, one v_ℓ. (c) One w_ℓ, multiple v_ℓ.

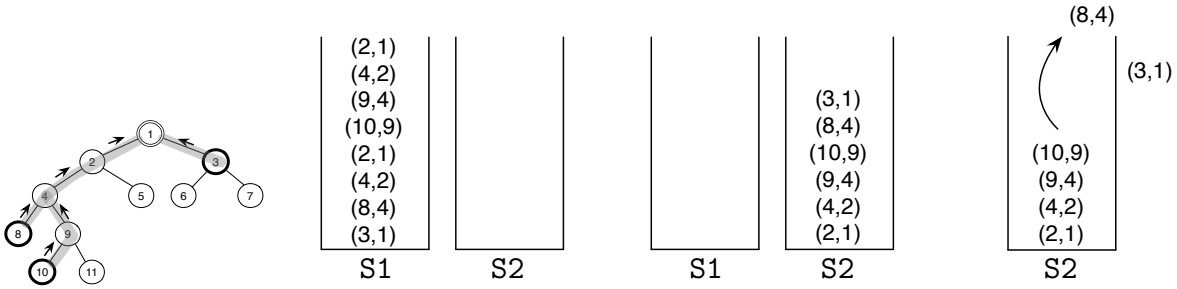
Figure 5.7: Extension to multiple measurements/marginals. (a) Original graph. (b) Multiple w_ℓ, one v_ℓ. Measurements at node 3, 8, 10 are obtained, while node 1's marginal is sought. (c) One w_ℓ, multiple v_ℓ. Measurement at node 1 is obtained, while marginals at nodes 3, 8, 10 are of interest.

are consistently comparable to the tree diameter.

■ 5.7 Extension to Multiple Measurements/Marginals

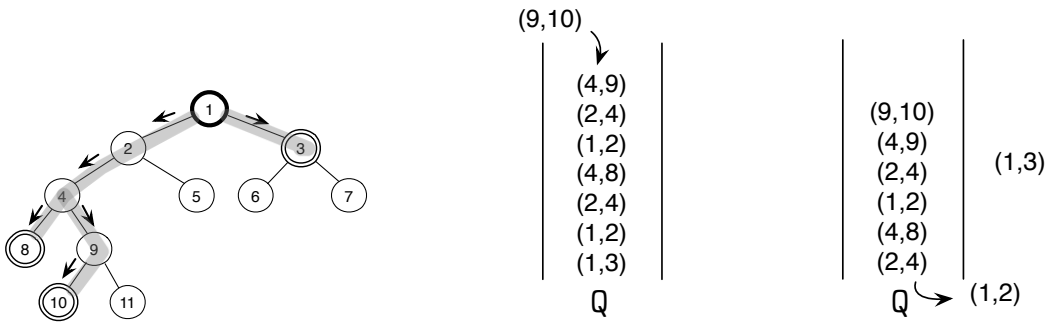
We have made the assumption that w_ℓ, v_ℓ are scalars. That is, we have assumed we obtain one measurement and are interested in just one marginal at a time. We can easily relax this assumption by extending to multiple measurements or marginals at a time (cf. Fig. 5.7). Let us start with the case of multiple measurements and one marginal (Fig. 5.8a). That is, w_ℓ is a vector and v_ℓ a scalar. A naïve approach would be to propagate messages in $\mathcal{M}(u \to v_\ell)$, for each $u \in w_\ell$, but this would result in the re-evaluation of many messages that are found in overlapping paths $\mathcal{M}(u \to v_\ell)$, for all $u \in w_\ell$. Ideally, we would like to send messages on the gray band just once in the right order (Fig. 5.8a). In that case, for each $u \in w_\ell$, we retrieve the messages in the path $\mathcal{M}(u \to v_\ell)$ that need to be evaluated and push them into a stack (`S1`) (see Fig. 5.8b). In order to place the messages in the right order of evaluation, we pop the messages and push them into a second stack, `S2`. To avoid duplicates, we keep a hash table with messages as a key. We evaluate messages by popping elements from stack `S2` one-by-one (see Fig. 5.8c).

In the case of one measurement and multiple marginals (Fig. 5.9a), w_ℓ is a scalar while v_ℓ a vector. For each $u \in v_\ell$, we retrieve the messages in the path $\mathcal{M}(w_\ell \to u)$ that need to be evaluated and push them into a queue (`Q`) (see Fig. 5.9b). We poll messages from `Q` (retrieve and remove the head of the queue), while we avoid duplicates. To avoid duplicates, we keep a hash table with messages as a key (Fig. 5.9c). If a message already exists in the hash table, it will not be considered in the messaging schedule. Lastly, we treat the multiple measurements/multiple marginals case by applying the procedure of the multiple measurements/one marginal case to each different marginal.



(a) Original graph. (b) Multiple w_ℓ, one v_ℓ. (c) One w_ℓ, multiple v_ℓ. (d) One w_ℓ, multiple v_ℓ.

Figure 5.8: (a) Multiple w_ℓ, one v_ℓ. We need only propagate messages on the gray band from measurement nodes (in bold face) to the node of interest v_ℓ. Here, $w_\ell = \{3, 8, 10\}$, $v_\ell = 1$. (b) For each $u \in \mathcal{M}(u \to v_\ell)$, we push the messages that need to be evaluated on stack `S1`. This also contains duplicate messages due to the overlap of paths (e.g., messages $(2, 1), (4, 2)$). In this example, messages were pushed into the stack in the following order; first messages in $\mathcal{M}(3 \to 1)$, then in $\mathcal{M}(8 \to 1)$ and lastly messages in $\mathcal{M}(10 \to 1)$. (c) Pop each element from stack `S1` and push it to stack `S2`, while keeping a hash table to avoid duplicates. For instance, at the beginning message $(2, 1)$ is pushed to stack `S2`, then $(4, 2), (9, 4), (10, 9)$. When element $(2, 1)$ is encountered again, it will be skipped since it already exists in the hash table. (d) After we pop all elements from stack `S1` and push them to stack `S2` (avoiding duplicates), we form the messaging schedule by popping messages from the top of stack `S2`.



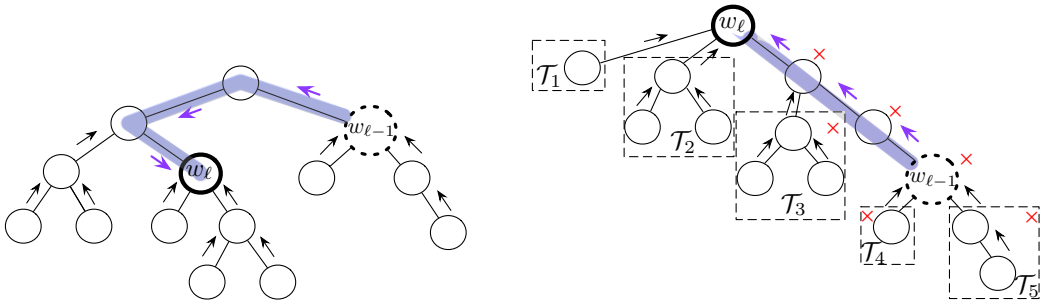
(a) One w_ℓ, multiple v_ℓ. (b) Multiple w_ℓ, one v_ℓ. (c) One w_ℓ, multiple v_ℓ.

Figure 5.9: (a) One w_ℓ, multiple v_ℓ. We need only propagate messages on the gray band from measurement node (in bold face) to the nodes of interest v_ℓ. Here, $w_\ell = 1$, $v_\ell = \{3, 8, 10\}$. (b) For each $u \in \mathcal{M}(w_\ell \to u)$, we push the messages that need to be evaluated on queue `Q`. This also contains duplicate messages due to the overlap of paths (e.g., messages $(1, 2), (2, 4)$). In this example, messages were pushed into the queue in the following order; first messages in $\mathcal{M}(1 \to 3)$, then in $\mathcal{M}(1 \to 8)$ and lastly messages in $\mathcal{M}(1 \to 10)$. (c) We generate the messaging schedule by polling each element from the (head of) queue, while keeping a hash table to avoid duplicates. For instance, at the beginning message $(1, 3)$ is polled from the queue and then message $(1, 2)$. The second time that message $(1, 2)$ will be encountered, it will be skipped since it already exists in the hash table.

■ 5.8 Extension to Max-Product

In case of max-product, we just need to replace sum with max and introduce a new type of messages, called *delta messages*, that will be used for the recovery of the MAP sequence. A delta message $\delta_{i \to j}(x_j)$ indicates the value of the source node that corresponds to the MAP sequence of the subtree rooted at the source node (excl. the branch containing the target node) for a specific value of the target node. That is, if we denote by $\mathcal{T}_i$ the subtree rooted at i excluding the branch that contains j, then $[x^*_{\mathcal{T}_i}]_i = \delta_{i \to j}(x_j)$. That is, it provides the maximizing value at node i of the MAP subsequence $x^*_{\mathcal{T}_i}$ if node j had value x_j. In order to recover the MAP sequence, we need to propagate delta messages from $w_{\ell-1}$ to w_ℓ and then backtrack from w_ℓ down to the leaves (considering w_ℓ as the root).

In general, obtaining the MAP sequence is a linear operation in the number of nodes. However, local changes in node potentials (via the introduction of measurements) might induce only small changes in the MAP sequence. We should also note that the only delta messages pointing towards the root w_ℓ that have changed, are the ones in path $\mathcal{M}(w_{\ell-1} \to w_\ell)$, which were correctly updated at iteration ℓ. See Fig. 5.10a for details. This observation can help us recover the MAP sequence in a more efficient way. In more detail, we can create an indicator sparse matrix, where rows would represent the source



(a) Message updates (max-product version) (b) Savings in MAP computations

Figure 5.10: Message updates in max-product and computational savings. (a) Message updates (max-product version). Purple thick arrows represent the messages that will be propagated in the current iteration, while solid black messages that have been evaluated correctly from previous iterations. (b) Savings in MAP sequence computations. During the ℓ-th step, the node potential at w_ℓ (bold-faced node) as well as delta messages $\delta_{i\to j}(x_j)$ in $\mathcal{M}(w_{\ell-1} \to w_\ell)$ (purple arrows) change. Let us assume the maximizing value at w_ℓ remained the same compared to the previous iteration, while the maximizing values of the remaining nodes in path $\mathcal{M}(w_{\ell-1} \to w_\ell)$ changed. We visualize this change with a red $\times$. Since the maximizing value at w_ℓ stayed intact, the maximizing values of all its subtrees (not including the one containing path $\mathcal{M}(w_{\ell-1} \to w_\ell)$) will remain the same (here, trees $\mathcal{T}_1, \mathcal{T}_2$). Therefore, there is no need to backtrack down to a node whose maximizing value did not change since the last iteration. On the other hand, since the maximizing values of the remaining nodes in $\mathcal{M}(w_{\ell-1} \to w_\ell)$ changed, their subtrees' maximizing values $(\mathcal{T}_3, \mathcal{T}_4, \mathcal{T}_5)$ would also potentially change and hence backtracking on these trees is necessary. Usually, a change in a node's maximizing value results only in local changes in the MAP sequence. Therefore, this scheme might practically lead to a lot of computational savings.

and columns the target of a delta message. We can assign the value 1 to any delta message that became "dirty" (changed) in the most recent iteration. That is, every message in path $\mathcal{M}(w_{\ell-1} \to w_\ell)$ (purple arrows in Fig. 5.10a). Therefore, when we backtrack from w_ℓ down to the leaves, we must consider the effect that these changed messages can have in the MAP sequence. Nevertheless, if the value of a node remains the same (with the previous iteration), then the sequences of the subtrees rooted at the neighbors of this node will remain the same. Therefore, there is no need to backtrack further down to a subtree once a node's maximizing value remained the same and the subtree is linked to that node via a "clean" message. A visual explanation is provided in Fig. 5.10b.

■ 5.9 Extension to Gaussian Loopy MRFs

Adaptive BP can be extended to Gaussian loopy graphs using the ideas from [64]. In the case of Gaussian loopy graphs, we should observe that in order to obtain a marginal at a node, we need to send two types of messages; first–round messages $J^{\mathcal{T}}_{i\to j}, h^{\mathcal{T}}_{i\to j}$ corresponding to the acyclic part of the graph $\mathcal{T}$ after the removal of FVS $\mathcal{F}$, "feedback" messages $h^{p}_{i\to j}$ for every feedback vertex p and second–round messages $\tilde{h}^{\mathcal{T}}_{i\to j}$, which are revised potential messages after the update of the potential vector h at the anchors (neighbors of feedback vertices). This change in potential vector requires the knowledge of the updated means $\mu_{\mathcal{F}}$ and covariance $\Sigma_{\mathcal{F}}$ of the FVS, which requires in turn the knowledge of all "partial" means $\hat{\mu}^{\mathcal{T}}_i$ and "feedback gains" g^p_i at the anchors.

This observation leads to a natural extension of adaptive BP to Gaussian loopy graphs. First, let us denote the node from set $\mathcal{T}$, where a measurement has been obtained most recently as $w^{\mathcal{T}}_\ell$. It obviously holds

$$w^{\mathcal{T}}_\ell = \begin{cases} w_\ell & , w_\ell \in \mathcal{T} \\ w^{\mathcal{T}}_{\ell-1} & , \text{otherwise.} \end{cases}$$

If $w_{\ell-1}, w_\ell \in \mathcal{T}$, we send messages $J^{\mathcal{T}}_{i\to j}, h^{\mathcal{T}}_{i\to j}, h^{p}_{i\to j}$ from $w_{\ell-1} \to w_\ell$ exactly as we did in the acyclic case. However, when $w_{\ell-1} \in \mathcal{F}$, we need to send messages from node $w^{\mathcal{T}}_{\ell-1}$, which is the node where a measurement has been obtained most recently, to propagate the effects of the past changes in the current node w_ℓ. In summary, when $w_\ell \in \mathcal{T}$, we send messages from $w^{\mathcal{T}}_{\ell-1}$ to w_ℓ, while no action is necessary when $w_\ell \in \mathcal{F}$. By this procedure, we ensure that all incoming messages $J^{\mathcal{T}}_{i\to j}, h^{\mathcal{T}}_{i\to j}, h^{p}_{i\to j}$ to w_ℓ are correct.

Now let us assume we want to update $v_\ell \in \mathcal{F}$. This would require the knowledge of partial means $\hat{\mu}^{\mathcal{T}}_i$ and "feedback gains" $g^p_i, \forall p \in \mathcal{F}$ at the anchors, $i \in \mathcal{A}$. As a reminder, anchors are the neighbors of FVS nodes which belong in $\mathcal{T}$, $\mathcal{A} = \{i \mid i \in \mathcal{T}, i \in N(p), \forall p \in \mathcal{F}\}$. The correct update of $\hat{\mu}^{\mathcal{T}}_i, g^p_i, \forall i \in \mathcal{A}$ leads to the correct evaluation of $\hat{h}_{\mathcal{F}}, \hat{J}_{\mathcal{F}}$ which in turn provides the correct mean and variance for v_ℓ since by our assumption belongs to the FVS. Partial means and "feedback gains" at the anchors would be correct if the change in the potential of the most recent measurement node $w^{\mathcal{T}}_\ell$ is propagated at the anchors after the update of node's w_ℓ potential. We achieve this by sending messages $J^{\mathcal{T}}_{i\to j}, h^{\mathcal{T}}_{i\to j}, h^{p}_{i\to j}$ from $w^{\mathcal{T}}_\ell$ to all anchors $\mathcal{A}, \forall i$. This guarantees that all incoming messages at the anchors are correct.

If $v_\ell \in \mathcal{T}$, we need to propagate a second–round of messages to account for the feedback provided by the updated parameters $\mu_{\mathcal{F}}, \Sigma_{\mathcal{F}}$ of the FVS nodes. In other words, we revise the potential vectors as $\tilde{h}_i = h_i + \sum_{j\in N(i)\cap\mathcal{F}} J_{ij}[\mu_{\mathcal{F}}]_j$. From an inspection of the above relationship, we can easily see that the only potential vectors which would change are the ones at the anchors, since the sum involves the intersection of the FVS nodes $\mathcal{F}$ and the neighbors of a node in $i \in \mathcal{T}$. This means we need to propagate messages $\tilde{h}^{\mathcal{T}}_{i\to j}$ from the anchors $\mathcal{A}$ to node v_ℓ. We obtain the right mean at v_ℓ from $(\tilde{h}_{\mathcal{T}}, J_{\mathcal{T}\mathcal{T}}, \tilde{h}^{\mathcal{T}}_{i\to j}, J^{\mathcal{T}}_{i\to j})$. Lastly, we correct the variance from $\sigma^2_{v_\ell} = (\hat{J}^{\mathcal{T}}_{v_\ell,v_\ell})^{-1} + \sum_{p,q\in\mathcal{F}} g^p_i [\Sigma_{\mathcal{F}}]_{pq} g^q_i$, where $\hat{J}^{\mathcal{T}}_{v_\ell,v_\ell}$ is obtained from the previous run

Table 5.1: Messages between $w_{\ell-1}^{\mathcal{T}}, w_\ell$ in loopy adaptive BP

	$w_\ell \in \mathcal{T}$	$w_\ell \in \mathcal{F}$
$w_{\ell-1} \in \{\mathcal{F}, \mathcal{T}\}$	Send $J_{i\to j}^{\mathcal{T}}, h_{i\to j}^{\mathcal{T}}, h_{i\to j}^{p}$ in $\mathcal{M}(w_{\ell-1}^{\mathcal{T}} \to w_\ell)$	–

Table 5.2: First- and second-round messages between $w_\ell^{\mathcal{T}}, \mathcal{A}, v_\ell$ in loopy adaptive BP

	$v_\ell \in \{\mathcal{F}, \mathcal{T}\}$
$w_\ell \in \{\mathcal{F}, \mathcal{T}\}$	Send $J_{i\to j}^{\mathcal{T}}, h_{i\to j}^{\mathcal{T}}, h_{i\to j}^{p}$ in $\mathcal{M}(w_\ell^{\mathcal{T}} \to \mathcal{A})$

	$v_\ell \in \mathcal{T}$	$v_\ell \in \mathcal{F}$
$w_\ell \in \{\mathcal{F}, \mathcal{T}\}$	Send $\tilde{h}_{i\to j}^{\mathcal{T}}$ in $\mathcal{M}(\mathcal{A} \to v_\ell)$ Send $h_{i\to j}^{p}$ in $\mathcal{M}(w_\ell^{\mathcal{T}} \to v_\ell)$	– –

of BP. As we observe, the "feedback gains" at node v_ℓ are essential for the correct evaluation of variance at v_ℓ. As a last step, we need to propagate messages $h_{i\to j}^{p}, \forall p \in \mathcal{F}$ from $w_\ell^{\mathcal{T}}$ to v_ℓ. This concludes the algorithm. Tables 5.1, 5.2 summarize the messaging protocol. A more detailed description is provided in Alg. 5.2.

Theorem 5.9.1. *The marginal at v_ℓ $(\mu_{v_\ell}, \sigma_{v_\ell}^2)$ is correct.*

In order to prove this theorem, we need to prove the following intermediate lemmas first.

Lemma 5.9.1. *If $w_\ell \in \mathcal{T}$, messages $h_{i\to j}^{\mathcal{T}}, J_{i\to j}^{\mathcal{T}}, h_{i\to j}^{p}, \forall p \in \mathcal{F}$ in path $\mathcal{M}(w_{\ell-1}^{\mathcal{T}} \to w_\ell)$ are correct.*

Proof. The proof follows the same logic with that of Lem. 5.6.1. ■

The only difference here is that $w_{\ell-1}$ is substituted by $w_{\ell-1}^{\mathcal{T}}$, which is defined as

$$w_{\ell-1}^{\mathcal{T}} = \begin{cases} w_{\ell-1} & , w_{\ell-1} \in \mathcal{T} \\ w_{\ell-2}^{\mathcal{T}} & , \text{otherwise.} \end{cases}$$

In other words, $w_{\ell-1}^{\mathcal{T}}$ represents the most recent measurement that has been obtained from $\mathcal{T}$. The reason for propagating from $w_{\ell-1}^{\mathcal{T}}$ to w_ℓ is that we need to propagate the effect of the most recent measurement in $\mathcal{T}$ to w_ℓ. Obviously, when $w_\ell \in \mathcal{F}$, schedule $\mathcal{M}(w_{\ell-1}^{\mathcal{T}} \to w_\ell) = \emptyset$.

Lemma 5.9.2. *If $w_\ell \in \mathcal{T}$, the incoming messages $h_{i\to j}^{\mathcal{T}}, J_{i\to j}^{\mathcal{T}}, h_{i\to j}^{p}, \forall p \in \mathcal{F}$ of each node in $\mathcal{M}(w_{\ell-1}^{\mathcal{T}} \to w_\ell)$ are correct.*

Algorithm 5.2 Adaptive Belief Propagation for Gaussian Loopy Graphs

Preprocessing

Find FVS $\mathcal{F}$ using one of known algorithms (e.g., [5]).

Build the RMQ structure on tree $\mathcal{T} = V \setminus \mathcal{F}$ as described in Sec. 3 of main paper.

Initialization

Before incorporating any measurements, run BP on tree $\mathcal{T}$ using parameters $(h_{\mathcal{T}}, J_{\mathcal{T}\mathcal{T}})$, $(J_{\mathcal{T}p}, J_{\mathcal{T}\mathcal{T}}), \forall p \in \mathcal{F}$. This will generate first–round $J^{\mathcal{T}}_{i\to j}, h^{\mathcal{T}}_{i\to j}, h^{p}_{i\to j}, \forall p \in \mathcal{F}$, and second–round messages $\tilde{h}^{\mathcal{T}}_{i\to j} \equiv h^{\mathcal{T}}_{i\to j}$. Also, initialize $w^{\mathcal{T}}_0 = 0$.

Iteration

for $\ell = 1, \ldots, M$ **do**

  if $\ell > 1 \wedge w^{\mathcal{T}}_{\ell-1} \neq 0 \wedge w_\ell \in \mathcal{T}$ **then**

    Send $J^{\mathcal{T}}_{i\to j}, h^{\mathcal{T}}_{i\to j}, h^{p}_{i\to j}, \forall p \in \mathcal{F}$ in $\mathcal{M}(w^{\mathcal{T}}_{\ell-1} \to w_\ell)$.

  end if

  Update the node potential at X_{w_ℓ}: this changes $h_{w_\ell}, J_{w_\ell, w_\ell}$.

  Send messages $J^{\mathcal{T}}_{i\to j}, h^{\mathcal{T}}_{i\to j}, h^{p}_{i\to j}, \forall p \in \mathcal{F}$ in $\mathcal{M}(w^{\mathcal{T}}_{\ell} \to \mathcal{A})$.

  Evaluate partial means $\hat{\mu}^{\mathcal{T}}_i$ from $(h_{\mathcal{T}}, J_{\mathcal{T}\mathcal{T}}, h^{\mathcal{T}}_{i\to j}, J^{\mathcal{T}}_{i\to j})$ and "feedback gains" g^p_i from $(J_{\mathcal{T}p}, J_{\mathcal{T}\mathcal{T}}, h^{p}_{i\to j}, J^{\mathcal{T}}_{i\to j})$, for all $i \in \mathcal{A}, p \in \mathcal{F}$.

  Obtain the K–sized FVS graph with updated parameters $\hat{h}_{\mathcal{F}}, \hat{J}_{\mathcal{F}}$ as

$$[\hat{J}_{\mathcal{F}}]_{pq} = J_{pq} - \sum_{i \in N(p) \cap \mathcal{T}} J_{pi} g^q_i, \ \forall p, q \in \mathcal{F} \tag{5.8}$$

$$[\hat{h}_{\mathcal{F}}]_{p} = h_{p} \ - \sum_{i \in N(p) \cap \mathcal{T}} J_{pi} \hat{\mu}^{\mathcal{T}}_i, \ \forall p \in \mathcal{F} \tag{5.9}$$

  and solve for $\Sigma_{\mathcal{F}} = \hat{J}^{-1}_{\mathcal{F}}$ and $\mu_{\mathcal{F}} = \Sigma_{\mathcal{F}} \hat{h}_{\mathcal{F}}$.

  if $v_\ell \in \mathcal{F}$ **then**

    $\mu_{v_\ell} = [\mu_{\mathcal{F}}]_{v_\ell}$, $\sigma^2_{v_\ell} = [\Sigma_{\mathcal{F}}]_{v_\ell, v_\ell}$.

  else

    Revise potential vectors as $\tilde{h}_i = h_i + \sum_{j \in N(i) \cap \mathcal{F}} J_{ij} [\mu_{\mathcal{F}}]_j$.

    Send messages $\tilde{h}^{\mathcal{T}}_{i\to j}$ in $\mathcal{M}(\mathcal{A} \to v_\ell)$.

    Send messages $J^{\mathcal{T}}_{i\to j}, h^{p}_{i\to j}, \forall p \in \mathcal{F}$ in $\mathcal{M}(w^{\mathcal{T}}_{\ell} \to v_\ell)$.

    Evaluate $\mu_{v_\ell} = (\hat{J}^{\mathcal{T}}_{v_\ell, v_\ell})^{-1} \hat{h}^{\mathcal{T}}_{v_\ell}$, where

$$\hat{h}^{\mathcal{T}}_{v_\ell} = \tilde{h}_{v_\ell} \quad + \sum_{k \in N(v_\ell)} \tilde{h}^{\mathcal{T}}_{k \to v_\ell} \tag{5.10}$$

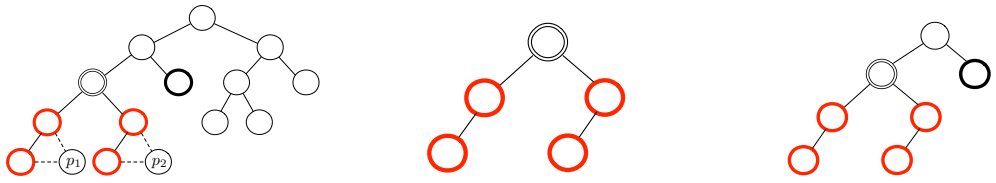
$$\hat{J}^{\mathcal{T}}_{v_\ell, v_\ell} = J_{v_\ell, v_\ell} + \sum_{k \in N(v_\ell)} J^{\mathcal{T}}_{k \to v_\ell} \tag{5.11}$$

$$\text{and } \sigma^2_{v_\ell} = (\hat{J}^{\mathcal{T}}_{v_\ell, v_\ell})^{-1} + \sum_{p,q \in \mathcal{F}} g^p_{v_\ell} [\Sigma_{\mathcal{F}}]_{pq} g^q_{v_\ell}. \tag{5.12}$$

    Reset messages $\tilde{h}^{\mathcal{T}}_{i\to j}$ in $\mathcal{M}(\mathcal{A} \to v_\ell)$.

  end if

end for



(a) Original graph. (b) $\mathcal{T}_\ell^v$. (c) $\mathcal{T}_\ell^w$.

Figure 5.11: (a) Original loopy graph. The graph $G = (V, \mathcal{E})$ is divided in FVS nodes $\mathcal{F}$ (nodes p_1, p_2) and the acyclic part $\mathcal{T} = V \setminus \mathcal{F}$. The black bold-faced node indicates the node from $\mathcal{T}$ where a measurement has been taken most recently, $w_\ell^{\mathcal{T}}$, the double-stroke node represents the node of interest, v_ℓ, and the red bold-faced nodes represent the anchor nodes $\mathcal{A}$, that is, nodes in $\mathcal{T}$ that are neighbors to FVS nodes. (b) $\mathcal{T}_\ell^v$ tree. Tree $\mathcal{T}_\ell^v$ is the subtree of $\mathcal{T}$ that has node v_ℓ as a root and passes through all anchor nodes $\mathcal{A}$. (c) $\mathcal{T}_\ell^w$ tree. Tree $\mathcal{T}_\ell^v$ is the subtree of $\mathcal{T}$ that that has node $w_\ell^{\mathcal{T}}$ as a root and passes through all anchor nodes $\mathcal{A}$.

Proof. The proof follows the same logic with that of Lem. 5.6.2. For every node k in the path $\mathcal{M}(w_{\ell-1}^{\mathcal{T}} \to w_\ell)$, we consider one of its neighbors in tree $\mathcal{T}$. Let us denote it by j. We are interested in showing that the messages $h_{j\to k}^{\mathcal{T}}, J_{j\to k}^{\mathcal{T}}, h_{j\to k}^p, \forall p \in \mathcal{F}$ are correct. Again, we denote by $\mathcal{T}_j$ the tree rooted at j, if we exclude the branch that contains edge (j,k) and by $t(j,k)$ the most recent time that a measurement has been obtained from tree $\mathcal{T}_j$. Then, at time $t(j,k)+1$, we exited tree $\mathcal{T}_j$ through the edge (j,k), and by Lem. 5.9.1 messages in $\mathcal{M}(w_{t(j,k)}^{\mathcal{T}} \to w_{t(j,k)+1})$ are correct, which includes messages $h_{j\to k}^{\mathcal{T}}, J_{j\to k}^{\mathcal{T}}, h_{j\to k}^p, \forall p \in \mathcal{F}$. This holds for every neighbor of k in $\mathcal{T}$. ■

Let us denote the (minimal) subtree of $\mathcal{T}$ rooted at $w_\ell^{\mathcal{T}}$ that passes through all the anchor nodes $\mathcal{A}$ by $\mathcal{T}_\ell^w$ and by $\mathcal{T}_\ell^v$ the (minimal) subtree that is rooted at $v_\ell \in \mathcal{T}$ and passes through all the anchor nodes $\mathcal{A}$. See Fig. 5.11 for a visualization of trees $\mathcal{T}_\ell^w, \mathcal{T}_\ell^v$. Messages from $w_\ell^{\mathcal{T}}$ to $\mathcal{A}$ in the tree $\mathcal{T}_\ell^w$ represent the messages in $\mathcal{M}(w_\ell^{\mathcal{T}} \to \mathcal{A})$. Equivalently, messages from all $u \in \mathcal{A}$ to $v_\ell \in \mathcal{T}$ in the tree $\mathcal{T}_\ell^v$ represent the messages in $\mathcal{M}(\mathcal{A} \to v_\ell)$.

Proposition 5.9.1. *The incoming messages $h_{i\to j}^{\mathcal{T}}, J_{i\to j}^{\mathcal{T}}, h_{i\to j}^p, \forall p \in \mathcal{F}$ of each node in $\mathcal{T}_\ell^w$ are correct.*

Proof. We should show that the messages from the "root" $w_\ell^{\mathcal{T}}$ towards the leaves of the minimal subtree $\mathcal{T}_\ell^w$ that contains all the nodes in $\mathcal{A}$ are correct. If $w_\ell \in \mathcal{T}$, then $w_\ell^{\mathcal{T}} = w_\ell$ and we showed in Lem. 5.9.2 that the incoming messages of each node in $\mathcal{M}(w_{\ell-1}^{\mathcal{T}} \to w_\ell)$ are correct. This includes the incoming messages to node w_ℓ. If $w_\ell \in \mathcal{F}$, and τ was the last time a measurement was obtained from $\mathcal{T}$, then $w_\ell^{\mathcal{T}} = w_\tau$ and by Lem. 5.9.2 all incoming messages to every node in $\mathcal{M}(w_{\tau-1}^{\mathcal{T}} \to w_\tau)$ are correct, which includes those of node w_τ. Since, by assumption all remaining measurements (from $\tau+1$ to ℓ have been taken from $\mathcal{F}$), the incoming messages $h_{i\to j}^{\mathcal{T}}, J_{i\to j}^{\mathcal{T}}, h_{i\to j}^p, \forall p \in \mathcal{F}$

to w_τ would reflect the correct value up to iteration ℓ. Therefore, we established that whether $w_\ell \in \mathcal{T}$ or $w_\ell \in \mathcal{F}$, the incoming messages to $w_\ell^{\mathcal{T}}$ are correct. Consequently, messages to its children would be correct. We show that the incoming messages of the remaining nodes in $\mathcal{T}_\ell^w$ are correct in exactly the same fashion as in Lem. 5.6.3. That is, if we denote by k a child of $w_\ell^{\mathcal{T}}$ and by j one of k's neighbors, we show that messages $h_{j\to k}^{\mathcal{T}}, J_{j\to k}^{\mathcal{T}}, h_{j\to k}^{p}, \forall p \in \mathcal{F}$ are correct by claiming that they have been part of a past message schedule $\mathcal{M}(w_{t(k,j)} \to w_{t(k,j)+1})$, where $t(k,j)$ is the most recent time a measurement has been obtained from subtree $\mathcal{T}_j$.[3] We continue this reasoning in a top-down approach, from the "root" $w_\ell^{\mathcal{T}}$ to the nodes in $\mathcal{A}$. ■

Corollary 5.9.1. *The "partial" means $\hat{\mu}_i^{\mathcal{T}}$ of all nodes in $\mathcal{T}_\ell^w$ are correct.*

Proof. This follows trivially from Prop. 5.9.1, since all incoming messages $h_{i\to j}^{\mathcal{T}}, J_{i\to j}^{\mathcal{T}}$ to every node in $\mathcal{T}_\ell^w$ are correct. ■

Corollary 5.9.2. *The "feedback gains" g_i^p of all nodes in $\mathcal{T}_\ell^w$ are correct.*

Proof. This follows trivially from Prop. 5.9.1, since all incoming messages $h_{i\to j}^{p}, J_{i\to j}^{\mathcal{T}}, \forall p \in \mathcal{F}$ to every node in $\mathcal{T}_\ell^w$ are correct. ■

Corollary 5.9.3. *The mean $\mu_{\mathcal{F}}$ and covariance $\Sigma_{\mathcal{F}}$ are correct.*

Proof. From Eqs. (5.8), (5.9), we see that $\hat{h}, \hat{J}$ are correct, since by Cor. 5.9.1, 5.9.2 "partial" means $\hat{\mu}_i^{\mathcal{T}}$ and "feedback gains" $g_i^p, \forall p \in \mathcal{F}$ at the anchors are correct and the node potential at X_{w_ℓ} has been already updated (l. 11, Alg. 5.2). ■

Corollary 5.9.4. *The revised potentials $\tilde{h}_i$ for every node $i \in \mathcal{T}$ are correct.*

Proof. From l. 18, Alg. 5.2, the revised potential is defined as

$$\tilde{h}_i = h_i + \sum_{j \in N(i) \cup \mathcal{F}} J_{ij}[\mu_{\mathcal{F}}]_j.$$

Since by Cor. 5.9.3, we showed that $\mu_{\mathcal{F}}$ is correct, then the revised potentials would be correct as well. ■

From the summation, it is clear that the only potential vectors that are revised are the ones at the anchors.

Proposition 5.9.2. *If $v_\ell \in \mathcal{T}$, the incoming messages $\tilde{h}_{i\to j}^{\mathcal{T}}$ of each node in $\mathcal{T}_\ell^v$ are correct.*

[3] As a reminder, subtree $\mathcal{T}_j$ is defined as the subtree rooted at j that excludes the branch which contains edge (j,k).

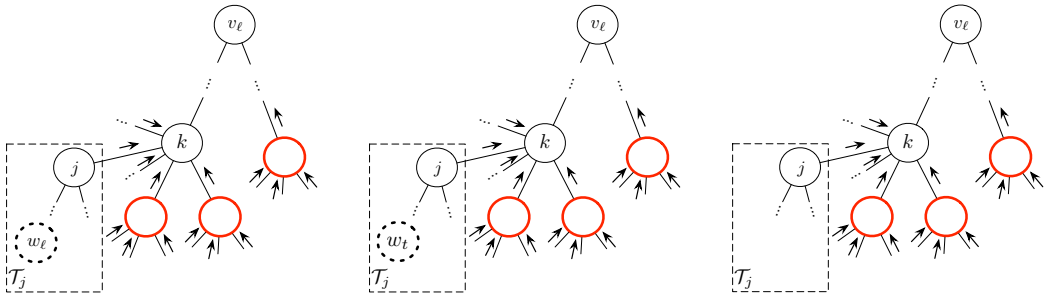
Proof. Let us start with the first iteration, $\ell = 1$. Messages $\tilde{h}^{\mathcal{T}}_{i\to j}$ to nodes in $\mathcal{A}$ are identical to $h^{\mathcal{T}}_{i\to j}$, since no other node potential has been revised yet. Initially, the incoming messages $h^{\mathcal{T}}_{i\to j}, J^{\mathcal{T}}_{i\to j}$ to the anchors $\mathcal{A}$ are correct by Prop. 5.9.1. The only potentials that are revised after we learn $\mu_{\mathcal{F}}, \Sigma_{\mathcal{F}}$ are the ones at the anchors, which by Cor. 5.9.4 are correct. This implies, that revised messages $\tilde{h}^{\mathcal{T}}_{i\to j}$ from the anchors to their parent nodes would also be correct. Let us denote by k a parent of an anchor node and by j one of its neighbors, as shown in Fig. 5.12. If j is one of the anchors, since we assumed that k is parent node to an anchor node, we just argued above that the message $\tilde{h}^{\mathcal{T}}_{j\to k}$ is correct. Now, if j does not belong to the tree $\mathcal{T}^v_\ell$, we denote by $\mathcal{T}_j$ the tree rooted at j excluding the tree branch that contains edge (j,k), as shown in Fig. 5.12. There are three scenarios, the last measurement that has been received from tree $\mathcal{T}_j$ is in time $t(j,k) = \ell$ (Fig. 5.12a), $t(j,k) = t < \ell$ (Fig. 5.12b) or $t(j,k) = \infty$ (Fig. 5.12c), which means that no measurement has been obtained from that tree yet. For the first two cases, message $\tilde{h}^{\mathcal{T}}_{j\to k}$, which is identical to $h^{\mathcal{T}}_{j\to k}$, is correct as it is part of the schedule $\mathcal{M}(w_{t(k,j)\to t(k,j)+1})$. By Prop. 5.9.1, all incoming messages $h^{\mathcal{T}}_{j\to k}, J^{\mathcal{T}}_{j\to k}, h^{p}_{j\to k}$, of each node in $\mathcal{M}(w_{t(k,j)\to t(k,j)+1})$ are correct. For the third case, when there is no measurement from subtree $\mathcal{T}_j$, message $h^{\mathcal{T}}_{j\to k}$ has stayed intact since initialization. So, in all three cases message $\tilde{h}^{\mathcal{T}}_{j\to k}$ is correct. Hence, when node k sends a message to its own parent it will also be correct. Obviously, here because we start with the first iteration $\ell = 1$, $t(k,j)$ can only be $t(k,j) = 1$ or $t(j,k) = \infty$. Continuing in this logic, we show that all incoming messages to node v_ℓ, $\tilde{h}^{\mathcal{T}}_{j\to v_\ell}$, are correct as well. As a last step of the algorithm, after we evaluate the marginal at the node of interest, we reset all messages $\tilde{h}^{\mathcal{T}}_{i\to j}$ in T^v_ℓ to their previous values as the revised potentials $\tilde{h}_i$ at the anchors reflect "imaginary" changes produced by the feedback of FVS nodes, rather than real changes that would be an outcome of obtaining a new measurement. By doing so, we guarantee that messages $\tilde{h}^{\mathcal{T}}_{i\to j}$ coincide with messages $h^{\mathcal{T}}_{i\to j}$ at the end of the first iteration. Therefore, when we move to the second iteration, we follow an identical logic to show that messages $\tilde{h}^{\mathcal{T}}_{i\to j}$ in T^v_ℓ would be correct. Similarly, messages $\tilde{h}^{\mathcal{T}}_{i\to j}$ in T^v_ℓ for every iteration ℓ would be correct. ■

Corollary 5.9.5. *If $v_\ell \in \mathcal{T}$, the incoming messages $J^{\mathcal{T}}_{i\to j}, h^{p}_{i\to j}, \forall p \in \mathcal{F}$ of each node in $\mathcal{M}(w^{\mathcal{T}}_\ell \to v_\ell)$ are correct.*

Proof. The proof follows exactly the same logic as that of Prop. 5.9.1. ■

Theorem 5.9.1. *The marginal at v_ℓ $(\mu_{v_\ell}, \sigma^2_{v_\ell})$ is correct.*

Proof. If $v_\ell \in \mathcal{F}$, the marginal (mean and variance) have been correctly estimated in Eqs. (5.8), (5.9) as shown in Cor. 5.9.3. If $v_\ell \in \mathcal{T}$, by Cor. 5.9.4, Prop. 5.9.2, and Cor. 5.9.5, the revised potential at v_ℓ, $\tilde{h}_{v_\ell}$, and the incoming messages to node v_ℓ, $\tilde{h}^{\mathcal{T}}_{k\to v_\ell}$, $J^{\mathcal{T}}_{k\to v_\ell}$ are correct. Therefore, by Eqs. (5.10), (5.11), $\hat{h}^{\mathcal{T}}_{v_\ell}, \hat{J}^{\mathcal{T}}_{v_\ell,v_\ell}$ are correct, which makes the mean at v_ℓ, μ_{v_ℓ} correct. Lastly, since by Cor. 5.9.5, messages $h^{p}_{i\to j}, J^{\mathcal{T}}_{i\to j}, \forall p \in \mathcal{F}$ in



(a) $t(k,j) = \ell$. (b) $t(k,j) = t$. (c) $t(k,j) = \infty$.

Figure 5.12: Correctness of incoming messages of nodes in T_ℓ^v. The incoming messages of every node in T_ℓ^v are correct. Red bold faced nodes represent the anchors and the fact that their potential vectors have been revised (changed). Tree $\mathcal{T}_j$ represents the tree rooted at node j, if we exclude the brach that contains edge (j,k). (a) Node w_ℓ is included in $\mathcal{T}_j$. (b) The most recent measurement from $\mathcal{T}_j$ has been taken at time $t(k,j) < \ell$. (c) No measurements have been received from tree $\mathcal{T}_j$.

$\mathcal{M}(w_\ell^{\mathcal{T}} \to v_\ell)$ are correct, it follows that the "feedback gain" at v_ℓ, g_{v_ℓ} is also correct and hence variance at v_ℓ, $\sigma_{v_\ell}^2$ as estimated by Eq. (5.12) will also be estimated correctly. ■

■ 5.9.1 Complexity

In terms of complexity, we first need to determine the FVS. Even though, finding the minimum FVS is NP-complete [46], there are approximate algorithms that find an FVS with size comparable to the optimal. For example, Bafna et al. [5] provide a 2-approximation, which runs in $\mathcal{O}(\min\{|\mathcal{E}|\log N, N^2\})$ time. At every iteration we need to send $(K+2)\text{dist}(w_{\ell-1}^{\mathcal{T}}, w_\ell)$ messages between $w_{\ell-1}^{\mathcal{T}}$ and w_ℓ, if $w_\ell \in \mathcal{T}$ and $(K+2)(|\mathcal{T}_\ell^w| - 1)$ messages between $w_\ell^{\mathcal{T}}$ and nodes in $\mathcal{A}$. If, in addition, $v_\ell \in \mathcal{T}$, the propagation of $(|\mathcal{T}_\ell^v| - 1)$ $\tilde{h}_{i\to j}^{\mathcal{T}}$ messages between the anchors $\mathcal{A}$ and v_ℓ is necessary, plus $K\text{dist}(w_\ell^{\mathcal{T}}, v_\ell)$ messages $h_{i\to j}^p$ from $w_\ell^{\mathcal{T}}$ to v_ℓ. Therefore, we send $\mathcal{O}(K(\text{dist}(w_{\ell-1}^{\mathcal{T}}, w_\ell) + \text{dist}(w_\ell^{\mathcal{T}}, v_\ell) + |\mathcal{T}_\ell^w|) + |\mathcal{T}_\ell^v|)$ messages per iteration. Compare this to the $\mathcal{O}(K|\mathcal{T}|)$ messages per iteration of standard FMP. To understand the difference in complexity, let us assume for the shake of exposition that $|\mathcal{T}_\ell^w| \geq \text{dist}(w_{\ell-1}^{\mathcal{T}}, w_\ell), \text{dist}(w_\ell^{\mathcal{T}}, v_\ell), |\mathcal{T}_\ell^v|$. This means that the complexity of adaptive BP is $\mathcal{O}(K|\mathcal{T}_\ell^w|)$, which results in a speedup on the order of $\mathcal{O}(|\mathcal{T}|/|\mathcal{T}_\ell^w|)$, since it always holds that $|\mathcal{T}_\ell^w| \leq |\mathcal{T}|$. Therefore, adaptive BP is consistently faster than standard FMP.

■ 5.10 Determining a nearly optimal measurement schedule

We have made the assumption that the measurement order is not known to us in advance. An equally interesting problem arises when we are given constraints on the number of measurements we can draw from each latent node and the task is to construct an optimal schedule of obtaining them. More formally, suppose we can draw

k_t measurements from X_t and we draw measurements from M distinct latent nodes.[4] Obviously, the schedule should be designed in such a way that it would result in the minimum number of propagated messages. Since there is no propagation of messages when measurements are taken consecutively from the same node, we can reduce this problem to one where there is one measurement vector (of size k_t) for each of the M nodes. In other words, once we reach a node X_t (dictated by the measurement schedule), we will draw k_t measurements from that node. Even though we can find the optimal solution to the above problem for small M, the exhaustive search becomes intractable as M grows, since there are $M!$ possible solutions. The problem of determining an optimal schedule of measurements that visits each of the M nodes exactly once, which corresponds to finding a schedule with the minimum number of computations, can be reduced to the shortest Hamiltonian path problem. As a reminder, a *Hamiltonian path* is a path that visits each node exactly once. A *Hamiltonian cycle* is a cycle that visits each node exactly once except for the starting node, which is visited twice. A graph that contains a Hamiltonian cycle is called a Hamiltonian graph. A graph that has a Hamiltonian cycle has trivially a Hamiltonian path as well, since the edge between the last node in the visitation order and the starting node can be removed. The shortest Hamiltonian path problem has shown to be NP-complete [3].

To formulate the shortest Hamiltonian path problem, we are given a set of nodes $\mathsf{X}_1, \ldots, \mathsf{X}_M$ that form an edge set $\mathcal{E}$. For every edge $(i, j) \in \mathcal{E}$ linking two nodes, there is a non-negative distance (cost) $\text{dist}(i, j)$ associated with them. The goal is to find an ordering $\boldsymbol{w}$, where each node is visited exactly once, that minimizes the total distance traveled

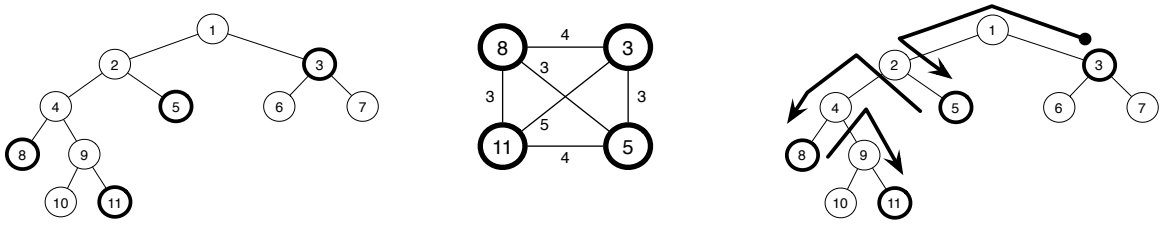
$$\max_{\boldsymbol{w}} \sum_{j=1}^{M-1} \text{dist}(w_j, w_{j+1}).$$

When the triangle inequality holds, that is, for every triplet $(i, j), (i, k), (j, k) \in \mathcal{E}$, $\text{dist}(i, j) \leq \text{dist}(i, k) + \text{dist}(j, k)$, there are approximate techniques with nice theoretical guarantees that provide nearly optimal solutions. One algorithm that runs in polynomial time $\mathcal{O}(M^3)$ is a variant of Christofides' algorithm, which was initially designed for the Traveling Salesman Problem (TSP) [18]. The TSP is very related to the shortest Hamiltonian path, since the objective is the same with the additional constraint that at the end of the visitation order, we return to the starting point. In other words, it is a shortest Hamiltonian cycle problem. The variant of Christofides' algorithm that gives an approximate solution for the shortest Hamiltonian path problem is proposed in [41]. This algorithm serves as a 3/2 approximation in the worst case.

We convert the problem of finding a schedule of minimum computations to a shortest Hamiltonian path as follows. We concatenate all k_t measurements of variable X_t into one vector of measurements. Since we draw measurements from M latent nodes, we compute the distance between every pair of latent nodes as

$$\text{dist}(i, j) = D_i + D_j - 2D_{\text{lca}(i,j)}, \tag{5.13}$$

[4] As a reminder, there are N latent nodes in total.



(a) Original graph. (b) Full graph of measurement nodes. (c) Minimum computational schedule.

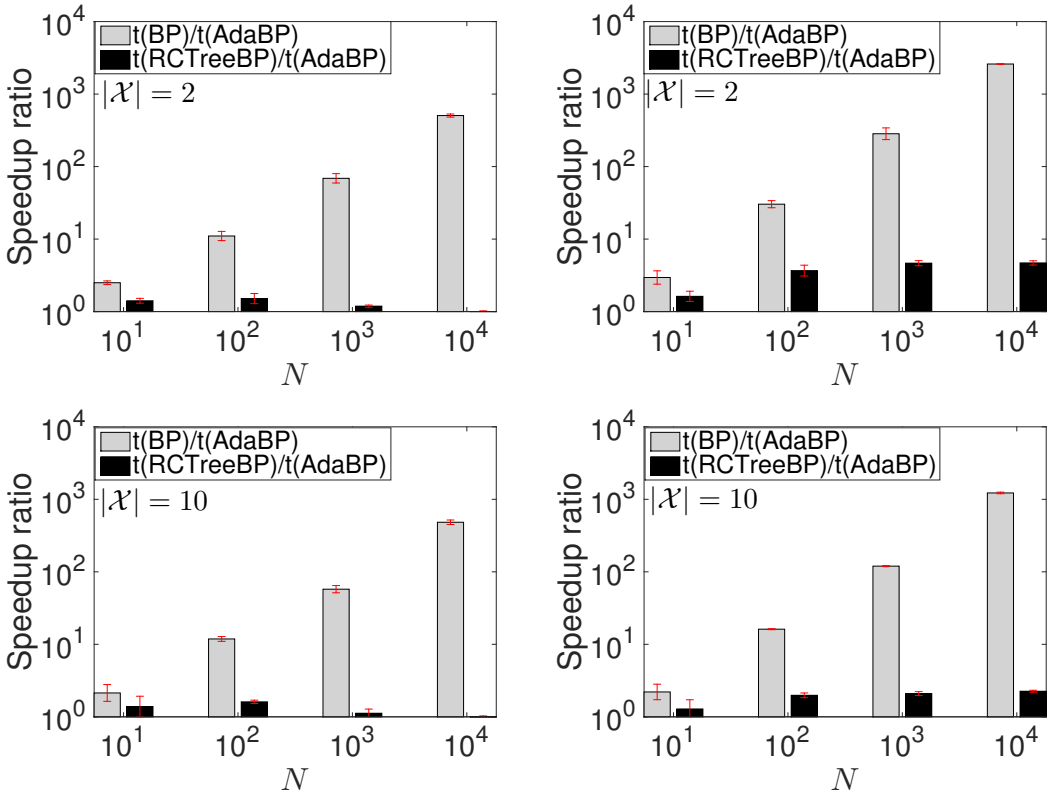
Figure 5.13: Reduction of finding optimal schedules to shortest Hamiltonian path. (a) The nodes where we would obtain measurements from are $3, 5, 8, 11$ (depicted as boldface). Our task is to design a measurement plan with the minimum number of messages for inference purposes. (b) We form a full undirected graph comprised of the measurement nodes. The weight of each edge would be the distance between these two nodes in the original graph, calculated by Eq. (5.13). (c) The path shown is one possible optimal solution. The arrow with a circle in one end indicates the starting node of sequence $\boldsymbol{w}$.

where D is the depth of a node and $\text{lca}(i, j)$ is the lowest common ancestor of i, j, which is recovered in constant time through the reduction to the RMQ problem, as we showed in Sec. 5.4. With this approach, we form a full undirected graph of M nodes, where each edge is weighted by the distance between the incident nodes. This graph is guaranteed to have a Hamiltonian path, since Dirac [24] showed that a simple graph with M vertices with $M \geq 3$ is Hamiltonian if every node has degree $M/2$ or greater, which applies to full graphs. You can see a visualization of the measurement plan designation in Fig. 5.13.

If we denote the length of the nearly shortest Hamiltonian path by ℓ_H, then in the Gaussian case, the overall complexity of message passing would be $\mathcal{O}(\ell_H d^3)$, where d is the dimension of latent variables. If, in addition, the dimension d is comparable to the number of latent variables N, the complexity of finding a shortest Hamiltonian path $\mathcal{O}(M^3)$ does not affect the overall complexity (since $M \leq N$).

■ 5.11 Experiments

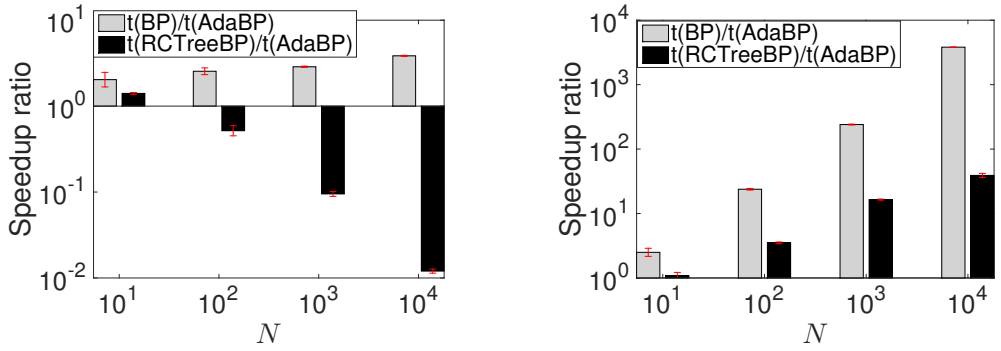
Henceforth, we refer to the proposed algorithm as AdaBP, the method of [89] as RC-TreeBP, and standard BP as BP. We use a publicly available version of RCTreeBP. In addition, when we make use of the term "consecutive elements", we mean consecutive measurement elements $w_{\ell-1}, w_\ell$ and concurrent measurement and marginal elements w_ℓ, v_ℓ. Recall that updates per iteration in RCTreeBP have complexity $\mathcal{O}(|\mathcal{X}|^3 \log N)$ (for trees), while complexity is $\mathcal{O}(|\mathcal{X}|^2(\text{dist}(w_{\ell-1}, w_\ell) + \text{dist}(w_\ell, v_\ell)))$ for AdaBP. Our experiments demonstrate that AdaBP is consistently orders of magnitude faster than standard BP (except in the worst case), and outperforms RCTreeBP when the average



(a) $\mathbb{E}[\text{dist}(w_{\ell-1}, w_\ell)]$ unconstrained.

(b) $\mathbb{E}[\text{dist}(w_{\ell-1}, w_\ell)] \leq |\mathcal{X}| \log N$

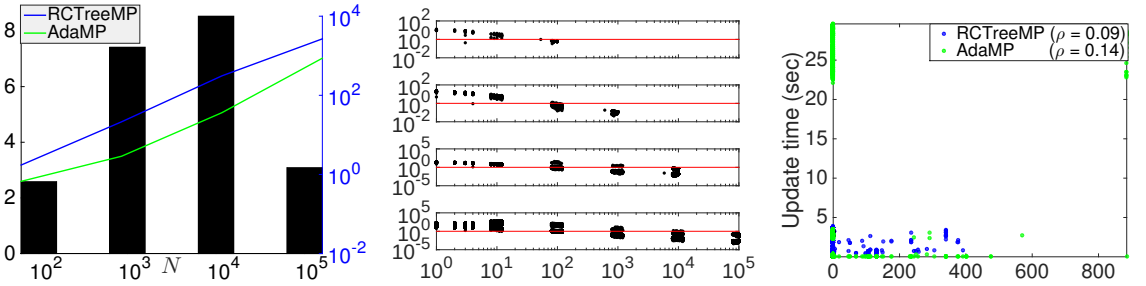
Figure 5.14: Comparison of total running time ratios between AdaBP and standard BP (gray bars) and AdaBP and standard RCTreeBP (black bars) over different alphabet sizes, $|\mathcal{X}| \in \{2, 10\}$. (a) Distance between consecutive elements $\mathbb{E}[\text{dist}(w_{\ell-1}, w_\ell)]$ is unconstrained. (b) $\mathbb{E}[\text{dist}(w_{\ell-1}, w_\ell)] \leq |\mathcal{X}| \log N$. For average distance $\mathbb{E}[\text{dist}(w_{\ell-1}, w_\ell)]$ smaller than $|\mathcal{X}| \log N$, AdaBP is 1.3–4.7 faster than RCTreeBP.



(a) $\mathbb{E}[\text{dist}(w_{\ell-1}, w_\ell)] \sim N$. (b) $\mathbb{E}[\text{dist}(w_{\ell-1}, w_\ell)] = 2$.

Figure 5.15: (a) Worst case. Distance between consecutive elements is on the order of N. AdaBP is comparable to standard BP (still being 2–4 times faster) and hence orders of magnitude slower than RCTreeBP. However, AdaBP is remarkably slower than RCTreeBP (nearly 80 times), since average distance between consecutive elements is close to N. (b) Best case. Consecutive elements are very close to each other. Therefore, only a constant number of updates is required per step for AdaBP. In contrast, RCTreeBP is insensitive to the distance between consecutive elements.

distance between consecutive elements is less than $|\mathcal{X}| \log N$ (see Fig. 5.14b). Conversely, if the tree diameter is much greater than $|\mathcal{X}| \log N$ and the average distance between consecutive elements is comparable to the tree diameter, AdaBP yields worse performance than RCTreeBP. We consider the following synthetic experiment where we construct unbalanced trees of sizes $N \in \{10, 10^2, 10^3, 10^4\}$. We repeat the above procedure $R = 10$ times for each N, by randomly constructing a new tree. For each tree, we randomly generate different $\boldsymbol{w}$ orders of size N and for simplicity of analysis we set $\boldsymbol{v} = \boldsymbol{w}$, so that only the distance between consecutive measurement nodes affects the computation. Figs. 5.14a and 5.14b compare the ratios of running times of AdaBP against standard BP and RCTreeBP (different rows correspond to different alphabet sizes). In all cases both AdaBP and RCTreeBP significantly outperform standard BP. Fig. 5.14a considers the case of randomly generated $\boldsymbol{w}$. When there is no restriction on the distance between consecutive elements, both AdaBP and RCTreeBP are comparable. However, for average distance between consecutive elements less than $|\mathcal{X}| \log N$, AdaBP is 1.3–4.7 times faster than RCTreeBP. Figs. 5.15a and 5.15b consider worst and best case performance of AdaBP, respectively. In the former, we generate several different instances of Markov chains of varying sizes and construct the measurement and marginal orders, $\boldsymbol{w}$ and $\boldsymbol{v}$ such that there is at least $2N/3$ distance between consecutive elements. In the latter case, we consider different instances of a star graph (tree diameter: 2) of varying sizes and randomly create measurement and marginal orders (which by construction do not have consecutive elements of more than 2 nodes apart). As expected, in Fig. 5.15a, RCTreeBP outperforms AdaBP for worst-case $\boldsymbol{w}$ (those with large distances between consecutive elements), yet still outperforms BP by



(a) Speedup of AdaMP over RCTreeMP.

(b) Sensitivity to $\mathbb{E}[\text{dist}(w_\ell, w_{\ell-1})]$.

(c) Sensitivity to changes in the MAP sequence.

Figure 5.16: Speedups of AdaMP over RCTreeMP for varying-size stretches of chromosome 21 (10^2–10^5 bp). (a) Left y-axis shows the speedup over RCTreeMP, while right y-axis the actual running times in sec (represented as lines). (b) Ratios of update times of AdaMP over RCTreeMP for different values of $\text{dist}(w_{\ell-1}, w_\ell)$ (x-axis: $\text{dist}(w_{\ell-1}, w_\ell)$, y-axis: speedup). The four log-log plots correspond to four different DNA stretches of $10^2, 10^3, 10^4, 10^5$ bp size, respectively. For smaller distances, AdaMP outperforms, but for distances closer to the graph size N, RCTreeMP is preferable. Red line indicates ratio of 1. (c) Both methods are not very sensitive to changes in the MAP sequence between consecutive iterations (x-axis: # of bp that differ between consecutive MAP sequences).

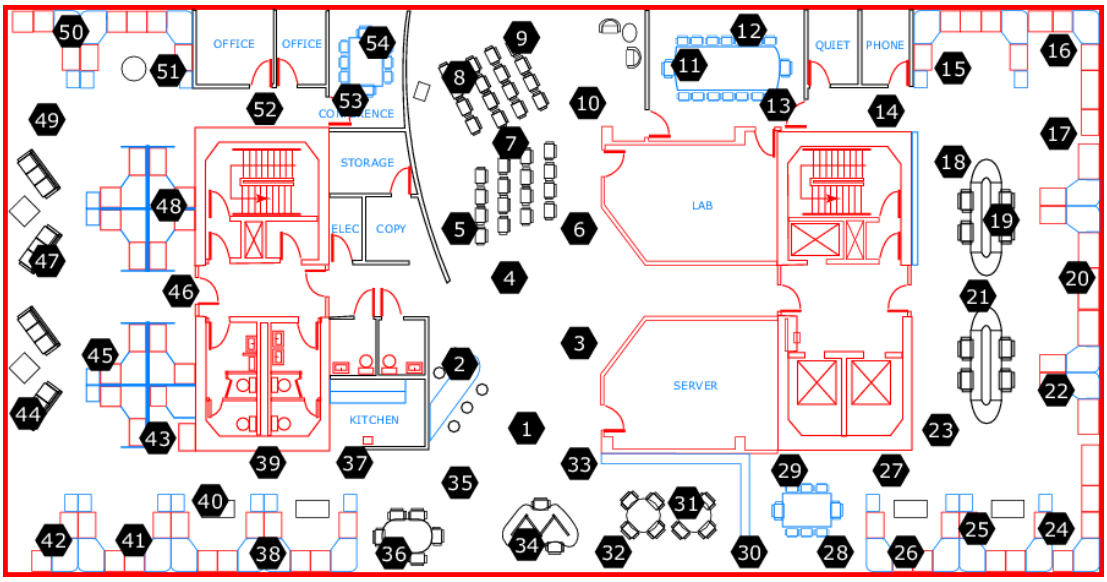
a factor of 2–4. However, in Fig. 5.15b we see that AdaBP is 4–49 times faster than RCTreeBP and hundred to thousand times faster than standard BP.

Next, we consider application of AdaMP (MP denotes max-product) to biological data. Specifically, we explore the effects of pointwise mutations in DNA sequences to the birth or disappearance of CpG islands. CpG islands are regions of DNA with high percentage of cytosine (C) occurring next to guanine (G) nucleotides and are believed to be responsible for upstream gene regulation. Usually, CpG island detection is modeled as an HMM problem where hidden nodes are binary variables which indicate the presence (or absence) of a CpG region and observed variables correspond to the observed DNA sequence comprised of the four nucleotides {A,T,C,G}. The goal is to find the MAP sequence (CpG regions) that best explains the observed data (DNA sequence). In *computational mutagenesis*, changes in the location of CpG islands are of interest due to mutations in the DNA sequence [1]. We compare AdaMP and RCTreeMP on varying-size stretches (10^2–10^5 bp) of human chromosome 21 obtained from the NCBI database. We train the parameters of HMM with one of the standard CpG prediction tools, CpG Island Searcher [91]. We perform a mutation every other nucleotide for each DNA-pair stretch and compare the running times of both methods under different criteria in Fig. 5.16. In this experiment, $v_\ell = w_\ell, \forall \ell$. Fig. 5.16a shows the speedup of AdaMP over RCTreeMP for varying sizes of DNA sequence. For medium to large sequences, AdaMP exhibits better performance, however, for very large sequences of size $\sim 10^5$, the computational cost of determining the MAP sequence is nearly linear with

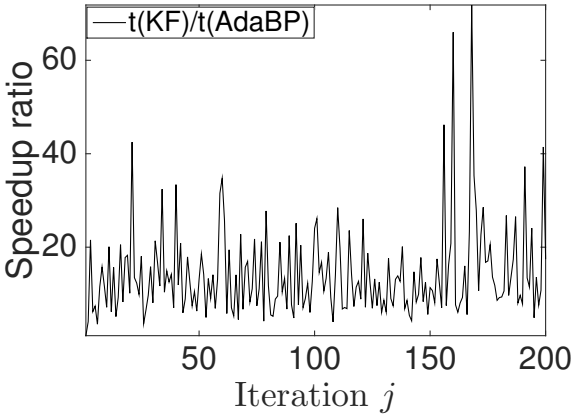
the graph size (even though the cost of updating the delta messages remains remarkably low). In contrast, RCTreeMP depends only on the number of variables which changed since the previous iteration. Fig. 5.16b examines the relationship in performance to the distance between consecutive elements for DNA stretches of varying size (10^2–10^5 bp). As expected, AdaMP is very sensitive to the distance between consecutive elements $\text{dist}(w_{\ell-1}, w_\ell)$. On the contrary, RCTreeMP depends only on the graph size N. AdaMP is preferred for measurement schedules with low average $\text{dist}(w_{\ell-1}, w_\ell)$ (points above the red line), while RCTreeMP average distance comparable to graph size (points below the red line). Lastly, Fig. 5.16c shows that both methods are not very sensitive to changes in the MAP sequence between consecutive iterations.

As a second experiment, we analyzed temperature measurements collected from 53 wireless sensors at 30 sec intervals from the Intel Berkeley Research lab. We model the latent temperatures in the various locations of the lab (Fig. 5.17a) as a grid graph. We assume that sensor measurements are a noisy representation of the temperatures around its close vicinity. We further assume that temperatures evolve over time following linear dynamics as $X_t = AX_{t-1} + V_{t-1}$, where $V_{t-1} \sim \mathcal{N}(v_{t-1}; 0, Q)$ and X_t represents the temperatures of the lab at time t. That is, we model the problem as a Gaussian HMM. We learn parameters A and Q by training the data between Feb 28 and Mar 7, 2004 on a Normal-inverse-Wishart model. We are interested in estimating the mean and covariance of temperatures in various lab locations which constantly change after the incremental incorporation of new measurements. One of the primary goals is to estimate the temperatures around sensitive areas with some certainty. The standard approach is to use Kalman filter/smoothing (KF) updates to compute means and variances. We use measurements in a 6-hour window on Feb 28, 2014 on a random order and compare the update times of AdaBP versus standard Kalman filter/smoothing updates (RCTreeBP is not included for comparison here, since it is not applicable to Gaussian models). We see in Fig. 5.17b, that AdaBP is consistently (1–42 times) faster than Kalman smoothing (green dots vs red dots). Also, in Fig. 5.17c, we observe the direct dependence of AdaBP to distance between consecutive elements. We see that AdaBP is much more appropriate to use when distance between consecutive measurement nodes is small.

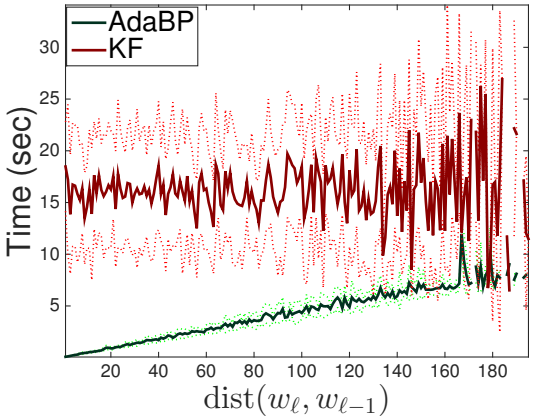
Lastly, we show the exactness of AdaBP in Gaussian loopy graphs by comparing the solution to one obtained from naïve inference. We consider the Gaussian loopy graph in Fig. 5.18a We create a random measurement $\boldsymbol{w}$ and marginal order $\boldsymbol{v}$ of size 1000. In the naïve inference approach, we retrieve the marginal v_ℓ after incorporating measurement w_ℓ as $\sigma^2_{v_\ell} = [J^{-1}]_{v_\ell, v_\ell}, \mu_{v_\ell} = [J^{-1}h]_{v_\ell}$, which has cubic complexity in the number of hidden nodes. The marginal at v_ℓ in the case of loopy AdaBP is calculated as described by Alg. 5.2. We observe in Fig. 5.18b that both methods gives the same results, which demonstrates empirically that loopy AdaBP makes exact inference.



(a) Lab diagram

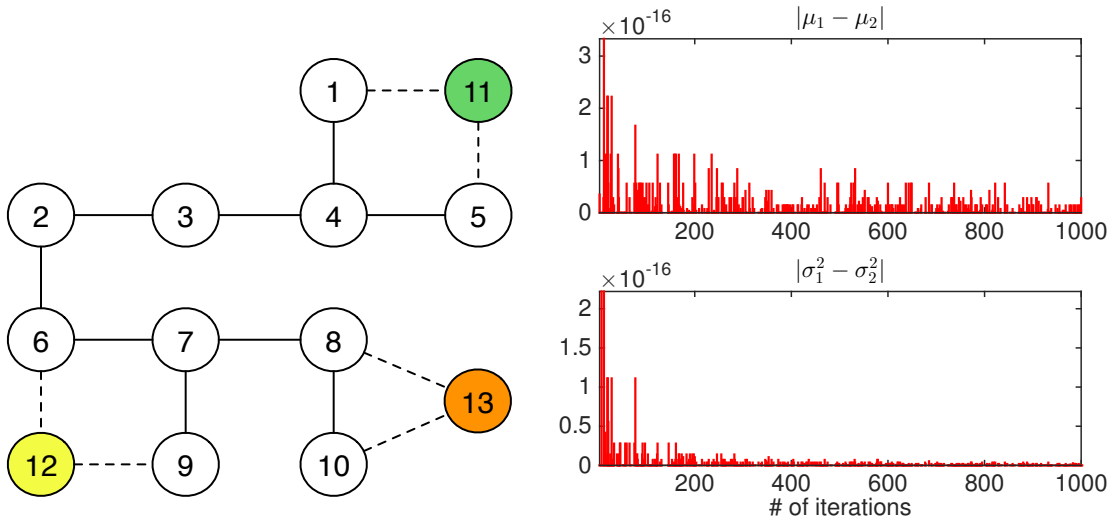


(b) Update time of AdaBP versus Kalman filter.



(c) Dependence of AdaBP and KF to dist$(w_\ell, w_{\ell-1})$.

Figure 5.17: Speedups of AdaBP over Kalman filtering in temperature monitoring data. (a) Lab diagram. The polygons show the locations where the 53 sensors are placed. The latent temperatures in the lab are modeled as a grid graph. (b) This figure shows the speedup over Kalman filter (KF). AdaBP is 1–42 times faster than standard Kalman filtering/smoothing techniques. (c) Running time per iteration of AdaBP and KF as a function of consecutive distance between elements. AdaBP is much more sensitive to dist$(w_\ell, w_{\ell-1})$ and as the figure suggests it is much faster than KF when dist$(w_{\ell-1}, w_\ell)$ is small. Dotted plots represents deviation due to different runs of the experiment.



(a) Loopy graph.

(b) Difference of loopy AdaBP to naïve inference.

Figure 5.18: Exact inference of AdaBP in Gaussian loopy graphs. (a) Loopy graph. The FVS nodes are 11, 12 and 13. For a random measurement $\boldsymbol{w}$ and marginal order $\boldsymbol{v}$, we compare AdaBP against naïve inference, which requires the inversion of the information matrix J at every step. (b) Results between AdaBP and naïve inference are the same. We denote by μ_1, σ_1^2, the sufficient statistics for node v_ℓ at each iteration produced by naïve inference, while by μ_2, σ_2^2, the sufficient statistics computed by AdaBP.

■ 5.12 Conclusion

We presented a new algorithm, AdaBP, which is particularly suited to sequential inference problems, when there is little or no knowledge of the measurement schedule in advance. In addition, when we can design the measurement order, we propose a nearly optimal schedule by casting it as a shortest Hamiltonian path problem We compare our method to standard BP and RCTreeBP [89]. In the case of trees, standard BP incurs a prohibitive cost of sending $\mathcal{O}(N)$ messages per iteration, while AdaBP sends only the necessary messages between consecutive elements. It is also much faster than RCTreeBP when the mean distance between consecutive elements is much smaller than $\mathcal{O}(|\mathcal{X}| \log N)$. We provided an extensive analysis of the algorithmic complexity with respect to the measurement $\boldsymbol{w}$ and marginal schedule $\boldsymbol{v}$. Lastly, we show extensions in the case where we have multiple measurements or marginals of interest per iteration. We provide the max-product version of the algorithm and extend to Gaussian loopy graphs, where inference is still exact by using the FMP method by Liu et al. [64].

Chapter 6

Conclusion

THIS thesis has addressed some of the fundamental problems encountered in information planning. We focused on proposing approximating algorithms that provide theoretical guarantees for different settings: when the reward is non-monotone; when measurements induce costs and there is a limited budget; when costs change based on the relative information value of measurements and when the set of interest is only a subset of the full latent graph. In addition, we have shown for Gaussian models that the complexity of information planning can be substantially reduced by taking sparsity in the measurement process into account. We have also designed a variant of belief propagation, called adaptive belief propagation that is well-suited for settings where model parameters change constantly and inference is made only on a set of relevant (latent) variables. Information planning can be seen as a special case of adaptive inference, as at the end of each greedy step we obtain a new measurement and this new information needs to be propagated to a fixed (latent) variable. We now summarize the contributions of this work to each of the addressed problems.

6.1 Contributions to information planning

The following sections highlight some of the contributions made to the problem of information planning.

6.1.1 Theoretical guarantees for greedy heuristics

We begin the analysis in Chap. 3 by discussing value independent models and providing necessary and sufficient conditions for existence of such models. Determining the models where planning is independent on the values of the selected measurements is important, since in this case open-loop control planning, which can be done completely in advance of measurement sampling, is equivalent to closed-loop control planning. We additionally present bounds for non-monotone rewards in the sequential setting for a slightly modified version of the greedy algorithm that is used for the monotone case, with the same complexity. Usage of such rewards is more natural in budgeted settings, where measurements induce different costs. In that respect, we propose a penalized form of mutual information, that we refer to as PMI, that retains submodularity and takes into account not only the informational value of a measurement, but its cost as

well. We then consider the case of varying costs, where measurement costs change based on the relative informational value they provide to the planning process. This scenario can be encountered in cases where different information consumers have access to the same pool of measurements. Different consumers might possess different knowledge of the underlying quantity of interest and hence might be willing to obtain a new measurement at different costs (given the relative information it carries). We provide conditions under which this case accepts the same bounds that hold in the unconstrained setting when the reward is a submodular monotone function. We additionally demonstrate upper bounds for the optimal solution of the submodular knapsack maximization (SKM) problem. The objective in the submodular knapsack maximization problem is to find the set of measurements that maximizes a submodular monotone reward under a budget constraint. Even though, Sviridenko [90] presented a greedy algorithm with a $1 - 1/e$ approximating ratio, its applicability can be prohibitive even for problems with moderate observation sizes N due to its high complexity: $\mathcal{O}(N^5)$. This complexity for can be prohibitive. We show that by converting the original problem in its dual form and using the algorithm by Buchbinder et al. [13], which is only linear in the number of measurements N, we can obtain upper bounds for the optimal. Lastly, we consider focused planning when the reward is MI, where only a subset of the latent variables are of interest. In this case, conditional independence between measurements breaks and thus submodularity of MI does not hold anymore. We apply the same greedy algorithm that is used for the unconstrained case of submodular monotone rewards and show (under certain conditions) worst-case lower bounds for the greedy solution with respect to the optimal.

■ 6.1.2 Complexity reduction of reward evaluations

In Chap. 4, we take advantage of sparsity in the measurement process to reduce the complexity of evaluating information rewards. We focus on Gaussian models and mutual information (as the reward function). We highlight the inappropriateness of the oracle-value model assumption, since the complexity of evaluating the information gain of a given measurement set (to the latent variable) depends on the dimension of the latent variable and the size of the measurement set. We focus our analysis on Gaussian HMMs, but it can be trivially extended to Gaussian tree MRFs. We propose an alternative approach of evaluating rewards, that under the assumption that a measurement depends only on a few latent variables, it provides speedups several orders of magnitude larger than standard Kalman filtering estimation. Additionally, we propose a variant of belief propagation that sends only messages from the current to the next node of interest (next element in the walk), thus avoiding unnecessary computations without compromising the accuracy of estimation. The dramatic reduction in complexity of evaluating rewards opens up the space for exploring more walks under the same time constraints. The exploration of more walks is important, because as we remarked in Sec. 4.3.2 different walks might lead to very different solutions.

6.2 Contributions to adaptive inference

The following section outlines some of the contributions to the problem of adaptive inference.

6.2.1 Adaptive Belief Propagation

Chap. 5 focuses on adaptive inference settings, that is, in settings where there are sequential changes in the parameters of the graphical model. The two inference problems that we consider is that of evaluating the marginals at given nodes and determining the most likely (MAP) sequence of all latent variables (given measurements) after each change in the model parameters. We are interested in giving answers to such queries without performing inference from scratch. We concentrate on focused inference settings, where only a few marginals are of interest (at any given point). We present a variant of BP, termed *adaptive BP* (AdaBP), that is well-suited for such settings. We show that the algorithm is exact for trees and it applies both to discrete and Gaussian variables. Interestingly, we demonstrate that this algorithm is exact for Gaussian loopy graphs, when combined with the method by Liu et al. [64]. We provide a thorough complexity analysis and show that adaptive BP is always faster than standard BP in adaptive inference settings. We include extensions when multiple nodes are of interest or multiple observations arrive at a time and extend the method to the MAP sequence problem. Furthermore, we consider the reverse problem where instead of being given a measurement plan, we have constraints on the number of measurements and our goal is to determine a feasible measurement plan of minimum complexity. We show how we can obtain a nearly optimal measurement plan by a reduction to the shortest Hamiltonian path problem. Lastly, we present experiments on both synthetic and real data and empirically show that AdaBP is orders of magnitude faster than standard BP and outperforms state-of-the-art method by Sümer et al. [89], when the average distance between consecutive elements is less than $\mathcal{O}(|\mathcal{X}| \log N)$.

6.3 Future Work

We will list below some promising areas for future work.

6.3.1 Theoretical guarantees of Greedy Algorithms

In Chap. 3 we derived the conditions under which open-loop is equivalent to closed-loop planning and we provided worst-case guarantees of one-step look-ahead greedy algorithms for different settings: submodular non-monotone rewards, budgeted settings and focused planning. In this section, we provide future directions for each of the above settings.

Conditions for weak dependence on measurement values

In Sec. 3.1, we provided conditions for exponential families under which entropy is independent on the measurement values. Unfortunately, the only known distribution that satisfies these conditions is the Gaussian. One interesting direction is to explore conditions under which models weakly depend on values of measurements. Even though the notion of "weak" dependence on measurement values has not been introduced formally, the goal of this analysis is to characterize models where planning (and consequently rewards of selected measurements) is robust to the acquirement of new measurement values.

Tighter lower-bound guarantees for stochastic sequential settings

In Sec. 3.2, we provided a very pessimistic bound for submodular non-monotone rewards that was based on the assumption that the minimum incremental of any measurement is greater than a negative value $-\theta$. It might be worth to explore the case where the minimum incremental value of each measurement follows a distribution (with fixed mean and variance) rather than being a fixed negative value. In that matter, we might be able to derive average-case performance bounds that can be less pessimistic than the derived worst-case guarantees.

Closed-loop guarantees

Williams [101] showed that the closed-loop greedy heuristic can be arbitrarily worse than the optimal closed-loop policy. Nevertheless, it might be possible to obtain weak guarantees by introducing additional structure in the graph. One such structural constraint as formulated in [101] could be that the information that a measurement conveys about a latent node of interest should be larger than a certain factor as compared to the information that the same measurement conveys about the neighbors of this latent node.

Worst-case bounds for discounted rewards in sequential settings

In Sec. 3.2, we derived lower-bound guarantees for submodular non-monotone functions in sequential settings. We implicitly made the assumption that as we build the greedy policy, the incremental rewards are not affected by how far into the planning horizon we project. However, it might be beneficial to incorporate discount factors in the objective that would represent the fact that measurements obtained later in the process would be less valuable [101]. If we make the assumption that every time we move on to the next iteration of the greedy process the incremental value is discounted by a factor α_j, which is related to the current step, the greedy heuristic can be expressed as

$$g_j \in \underset{u \in \mathcal{V}_{w_j} \setminus \mathcal{G}_{j-1}}{\arg\max} \prod_{i=1}^{j-1} \alpha_i f(u \mid \mathcal{G}_{j-1}).$$

It would be interesting if we can derive similar worst-case bounds for the discounted reward case.

Budgeted settings with different resource constraints for each observation set

In Sec. 3.4, we derived upper bounds for the optimal solution of the budgeted batch problem, known also as submodular knapsack maximization (SKM) problem. The underlying assumption in this case is that observations share the same resources. It would be interesting to consider the sequential setting where different budget constraints apply to different observation sets and explore whether meaningful upper bounds can be derived in this case as well.

Stochasticity of parameters in Gaussian models

Krause and Guestrin [52] showed that under mild assumptions MI is a submodular function. Later on, they considered the use of MI in Gaussian processes [38]. The implicit assumption is that the model parameters are known. Often, there is some uncertainty around the model parameters and hence it is more realistic that these are expressed as random variables. A useful extension would be to explore whether submodularity of MI holds for the case where model parameters are random variables.

Characterization of graph structures that satisfy the worst-case bounds for focused planning

In Sec. 3.5 we considered the focused planning problem, where a set of the latent variables is of interest $\mathcal{R}$ (relevant set). Because conditional independency of measurements given $X_{\mathcal{R}}$ no longer holds, we had to introduce an extended set $\hat{\mathcal{R}}$ that enforces conditional independencies among all pairs of measurements so that submodularity of MI holds. In this section, we provided a 39% worst-case bound under the assumption that the maximum information that a measurement conveys about $X_{\hat{\mathcal{R}}\setminus\mathcal{R}}$ given $X_{\mathcal{R}}$ is less than a factor of the maximum information that a measurement conveys about $X_{\mathcal{R}}$. Intuitively, this condition would be satisfied for cases where relevant set $X_{\mathcal{R}}$ enforces conditional independence to almost all pairs of measurements. It would be interesting though to characterize the graph structures that satisfy this condition and consequently the 39% lower bound applies.

■ 6.3.2 Complexity Reduction of Information Planning in Gaussian Models

In Chap. 4 we provided an analysis that achieves substantial reductions in the complexity of information rewards in Gaussian models when there is sparsity in the measurement process. Sparsity is expressed via sparsity of matrix C which is assumed to be known. Here, we will suggest future work that might generalize to cases where matrix C is stochastic and to non-Gaussian models.

Stochasticity of measurement matrix C

The results in Chap. 4 are based in the assumption that matrix C is known. It might be beneficial to explore the setting where there is some uncertainty on the value of C, that is, the level of sparsity in each row is a random variable. The last assumption would model the fact that there are times where the graph structure is not entirely known. So, even though we might have a rough idea about the links connecting different nodes, it might not be possible to identify the exact variables and precisely determine the strength of the links between nodes.

Extension to discrete graphs

In our work, we focused on Gaussian models, because the sparsity in the graph structure can be precisely described by the composition of matrix C. In addition, there is a closed-form expression that ties the entropy with the uncertainty (covariance) at each latent node and covariance updates after the incorporation of measurements can be derived in closed form. It would be interesting to extend the ideas of sparsity in the measurement process to discrete MRFs. The challenge in this case would be to be able to specify how model parameters change in sparse graphs and express information rewards in terms of the model parameters in a way that the sparsity pattern would emerge similarly to the Gaussian case.

Sparsity in the measurement process in focused planning settings

In Chap. 4, all the latent variables $\mathsf{X}_1, \ldots, \mathsf{X}_T$ were of interest. This conveniently allowed us at each greedy step to focus only on the latent variable that was related to the observation set corresponding to the current walk element. This is due to the fact that conditioned on the latent variable X_{w_j} linked to the observation set $\mathcal{V}_{w_j}$ of the current walk element w_j, each measurement from set $\mathcal{V}_{w_j}$ is independent on the remaining latent variables. A question that arises is how to take advantage of sparsity in focused planning settings, where only a set of latent variables $\mathsf{X}_{\mathcal{R}}$ is of interest. In this case, we can no longer quantify the information content of a measurement based only on the latent variable it links to, because this latent variable might not belong to the relevant set. It might be beneficial to use the notion of diffusive rewards as introduced in [101] to be able to use relaxations of MI (over the relevant set) that contain the latent variables that link to a measurement under consideration at each greedy step.

Connection of walk complexity to value of walks

In Sec. 4.8 we hinted upon the connection of a walk's computational complexity and its informational value. We also presented an example where rewards of different walks are not closely tied to their intrinsic complexities. An interesting area of exploration would be to formally quantify the tradeoffs between complexity and value of rewards across different walks.

6.3.3 Adaptive Belief Propagation

In Chap. 5 we presented AdaBP, a variant of belief propagation well-suited for adaptive inference settings. In Sec. 5.9, we provided an extension of AdaBP for Gaussian loopy graphs that returns the true marginals using the FMP algorithm by Liu et al. [64]. Two promising future directions as outlined below would be to extend these ideas to discrete loopy and Gaussian loopy graphs with many cliques.

Extension to Gaussian loopy graphs with large cliques

The complexity of AdaBP in Gaussian loopy graphs depends on the number of FV nodes K. In fact it grows linearly to K. See Sec. 5.9.1 for a detailed discussion of AdaBP's complexity in Gaussian loopy graphs. Unfortunately, dense graphs with many loops would result in a really large FVS $\mathcal{F}$. In that case, the dominating term in the complexity would be K, where $K = |\mathcal{F}|$ and asymptotically AdaBP would be no better than the FMP method by Liu et al. [64]. To make the inference problem to dense loopy graphs tractable it is helpful to introduce the notion of *pseudo-FVS* instead (as discussed in [64]), that is a set of nodes whose removal breaks loops in many parts of the graph but does not guarantee that the entire remaining graph $\mathcal{T} = V \setminus \mathcal{F}$ would be acyclic. Since the resulting graph $\mathcal{T}$ would not be acyclic, we know that AdaBP will not provide exact results. However, an interesting research direction would be to derive conditions under which AdaBP would provide similar solutions to loopy BP.

Extension to loopy discrete graphs

It is well established that convergence of loopy BP is not guaranteed in loopy graphs and even if convergence is reached the solutions might not correspond to the true values [42, 99]. Therefore, we already know that application of AdaBP in loopy graphs would result in incorrect marginals. A natural extension would be to focus the computation on the paths connecting the measurement w_ℓ and node of interest v_ℓ and apply a loopy version of AdaBP. Previous works that center the computation around certain parts of the graph by weighting accordingly BP messages based on their proximity to areas of interest have already been proposed [16, 29]. It would be worthwhile to study whether similar techniques could be applied to the AdaBP algorithm as well and derive conditions under which results would be similar to existing methods.

Appendix A

Derivations

A.1 Monotonicity of $f(k) = \frac{1-\left(1-\frac{1}{k}\right)^k}{2-\left(1-\frac{1}{k}\right)^k}$

Let us consider the functions $f(k) = \frac{1-\left(1-\frac{1}{k}\right)^k}{2-\left(1-\frac{1}{k}\right)^k}$ and $g(k) = \left(1-\frac{1}{k}\right)^k$. Trivially, $g(k) \geq 0, \forall k$. We have that

$$\log g(k) = k \log\left(1-\frac{1}{k}\right). \tag{A.1}$$

If we take the derivatives (with respect to k) on both sides of the above expression, Eq. (A.1) becomes

$$\begin{aligned} \frac{g'(k)}{g(k)} &= \log\left(1-\frac{1}{k}\right) + \frac{1}{k-1} \\ g'(k) &= g(k)\left[\log\left(1-\frac{1}{k}\right) + \frac{1}{k-1}\right]. \end{aligned} \tag{A.2}$$

From [66], we draw the standard logarithm inequality

$$\log(1+x) \geq \frac{x}{1+x}, \forall x > -1.$$

For $x = -1/k$, the above inequality becomes

$$\log\left(1-\frac{1}{k}\right) \geq -\frac{1}{k-1}. \tag{A.3}$$

Due to Eq. (A.3), we obtain from (A.2) that

$$g'(k) \geq 0, \tag{A.4}$$

and hence $g(\cdot)$ is monotonically increasing. Function f can be expressed in terms of function g as

$$f(k) = \frac{1-g(k)}{2-g(k)}.$$

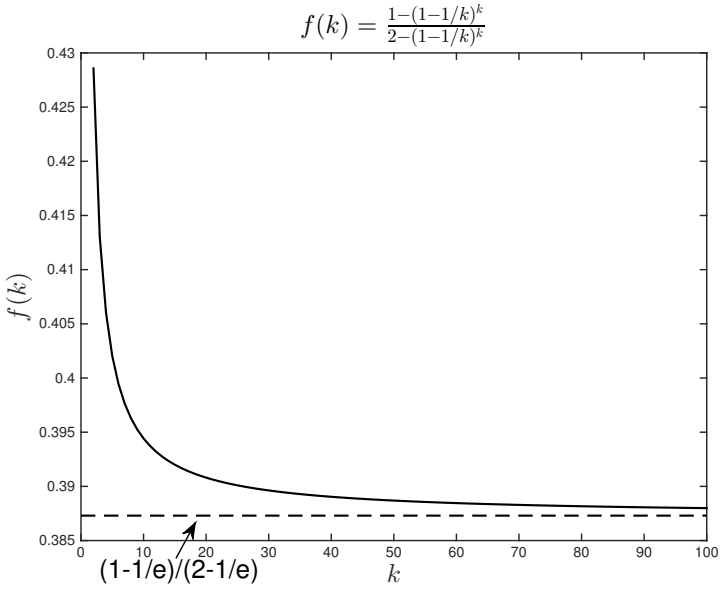


Figure A.1: Function $f(k) = \frac{1-\left(1-\frac{1}{k}\right)^k}{2-\left(1-\frac{1}{k}\right)^k}$. The function is monotonically decreasing and lower-bounded by $\frac{1-1/e}{2-1/e} \approx 0.387$.

Therefore,

$$f'(k) = -\frac{g'(k)}{(2-g(k))^2} \overset{(A.4)}{\leq} 0.$$

Therefore, f is monotonically decreasing with respect to k. Function f is depicted in Fig. A.1.

Bibliography

[1] U. A. Acar, A. T. Ihler, R. R. Mettu, and Ö. Sümer. Adaptive Updates for MAP Configurations with Applications to Bioinformatics. In *IEEE/SP 15th Workshop on Statistical Signal Processing (SSP)*, August 2009.

[2] S. M. Ali and S. D. Silvey. A General Class of Coefficients of Divergence of One Distribution from Another. *Journal of the Royal Statistical Society. Series B (Methodological)*, 28(1):131–142, 1966.

[3] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.

[4] A. Asadpour, H. Nazerzadeh, and A. Saberi. Stochastic Submodular Maximization. In *Proceedings of the 4th International Workshop on Internet and Network Economics, WINE*, pages 477–489, December 2008.

[5] V. Bafna, P. Berman, and T. Fujito. A 2-Approximation Algorithm for the Undirected Feedback Vertex Set Problem. *SIAM Journal on Discrete Mathematics*, 12(3):289–297, 1999.

[6] O. Barinova, V. Lempitsky, and P. Kohli. On the Detection of Multiple Object Instances using Hough Transforms. In *Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR)*, 2010.

[7] P. Bernhard, J. C. Engwerda, B. Roorda, J. M. Schumacher, V. Kolokoltsov, P. Saint-Pierre, and J-P. Aubin. *The Interval Market Model in Mathematical Finance*. Static & Dynamic Game Theory: Foundations & Applications. Springer, New York, 2013.

[8] D. P. Bertsekas and J. N. Tsitsiklis. *Introduction to Probability, 2nd Edition*. Athena Scientific, 2nd edition, July 2008.

[9] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1st edition, 1997.

[10] J. Bilmes. Deep Mathematical Properties of Submodularity with Applications to Machine Learning. NIPS Conference 2013 Tutorial, December 2013.

[11] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, New York, 2006. ISBN 0387310738.

[12] M. B. Blaschko. Branch and Bound Strategies for Non-maximal Suppression in Object Detection. In *Proceedings of the 8th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, pages 385–398, Berlin, Heidelberg, 2011. Springer-Verlag.

[13] N. Buchbinder, M. Feldman, J. Naor, and R. Schwartz. A Tight Linear Time (1/2)-Approximation for Unconstrained Submodular Maximization. In *Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, FOCS '12, pages 649–658, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-0-7695-4874-6.

[14] N. Buchbinder, M. Feldman, and R. Schwartz. Comparing Apples and Oranges: Query Tradeoff in Submodular Maximization. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1149–1168, 2015.

[15] G. Calinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a Submodular Set Function Subject to a Matroid Constraint. In *Proceedings of the 12th International Conference on Integer Programming and Combinatorial Optimization*, IPCO '07, pages 182–196, Berlin, Heidelberg, 2007. Springer-Verlag.

[16] A. Chechetka and C. Guestrin. Focused Belief Propagation for Query-Specific Inference. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, May 2010.

[17] C. Chekuri, J. Vondrák, and R. Zenklusen. Submodular Function Maximization via the Multilinear Relaxation and Contention Resolution Schemes. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 783–792, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0691-1.

[18] N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.

[19] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 3rd edition, 2009.

[20] T. M. Cover and J. A. Thomas. *Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, July 2006.

[21] C. Crick and A. Pfeffer. Loopy Belief Propagation As a Basis for Communication in Sensor Networks. In *Proceedings of the 19th International Conference on Uncertainty in Artificial Intelligence (UAI)*, UAI'03, pages 159–166, San Francisco, CA, USA, August 2003. Morgan Kaufmann Publishers Inc. ISBN 0-127-05664-5.

[22] A. Czumaj, M. Kowaluk, and A. Lingas. Faster algorithms for finding lowest common ancestors in directed acyclic graphs. *Theor. Comput. Sci.*, 380(1-2): 37–46, July 2007.

[23] A. Darwiche and M. Hopkins. Using recursive decomposition to construct elimination orders, jointrees, and dtrees. In *Trends in Artificial Intelligence, Lecture Notes in AI*, pages 180–191. Springer-Verlag, 2001.

[24] G. A. Dirac. Some theorems on abstract graphs. *Proceedings of the London Mathematical Society*, 3(1):69–81, 1952.

[25] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[26] Rauch H. E., Tung F., and Striebel C. T. Maximum Likelihood Estimates of Linear Dynamic Systems. *Journal of the American Institute of Aeronautics and Astronautics (AIAA)*, 3(8):1445–1450, Aug 1965.

[27] F. Echenique, D. Golovin, and A. Wierman. A Revealed Preference Approach to Computational Complexity in Economics. In *Proceedings of the 12th ACM Conference on Electronic Commerce*, pages 101–110, New York, NY, USA, 2011. ACM.

[28] G. A. Einicke and L. B. White. Robust Extended Kalman Filtering. *IEEE Transactions on Signal Processing*, 47(9):2596–2599, 1999.

[29] G. Elidan. Residual Belief Propagation: Informed Scheduling for Asynchronous Message Passing. In *Proceedings of the 22nd International Conference on Uncertainty in Artificial Intelligence (UAI)*, 2006.

[30] E. Ertin, J. W. Fisher III, and L. C. Potter. Maximum Mutual Information Principle for Dynamic Sensor Query Problems. In *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN)*, pages 405–416, February 2003.

[31] U. Feige. A Threshold of ln N for Approximating Set Cover. *Journal of the ACM*, 45(4):634–652, July 1998.

[32] Y. Filmus and J. Ward. Monotone Submodular Maximization over a Matroid via Non-Oblivious Local Search. *SIAM J. Comput.*, 43(2):514–542, 2014.

[33] J. Fischer and V. Heun. A New Succinct Representation of RMQ-Information and Improvements in the Enhanced Suffix Array. In *Proceedings of the 1st International Conference on Combinatorics, Algorithms, Probabilistic and Experimental Methodologies*, ESCAPE'07, pages 459–470. Springer-Verlag, 2007.

[34] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An Analysis of Approximations for Maximizing Submodular Set Functions - II. *Polyhedral combinatorics*, pages 73–87, 1978.

[35] F. V. Fomin, S. Gaspers, A. V. Pyatkin, and I. Razgon. On the Minimum Feedback Vertex Set Problem: Exact and Enumeration Algorithms. *Algorithmica*, 52 (2):293–307, 2008. ISSN 0178-4617.

[36] S. Fujishige. *Submodular Functions and Optimization.* Annals of Discrete Mathematics. Elsevier, 2005.

[37] S. O. Gharan and J. Vondrák. Submodular Maximization by Simulated Annealing. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1098–1116, January 2011.

[38] C. Guestrin, A. Krause, and A. Singh. Near-optimal sensor placements in Gaussian processes. In *International Conference on Machine Learning (ICML)*, August 2005.

[39] D. Harel and R. E. Tarjan. Fast Algorithms for Finding Nearest Common Ancestors. *SIAM J. Comput.*, 13(2):338–355, May 1984.

[40] K. J. Hintz and E. S. McVey. Multi-process constrained estimation. *IEEE Transactions on Systems, Man and Cybernetics*, 21(1):237–244, 1991. ISSN 0018-9472. doi: 10.1109/21.101154.

[41] J. A. Hoogeveen. Analysis of Christofides' heuristic: Some paths are more difficult than cycles. *Operations Research Letters*, 10(5):291–295, 1991.

[42] A. T. Ihler, J. W. Fisher III, A. S. Willsky, and M. Chickering. Loopy Belief Propagation: Convergence and Effects of Message Errors. *Journal of Machine Learning Research (JMLR)*, 6:905–936, 2005.

[43] F. Jelinek. *Statistical Methods for Speech Recognition.* MIT press, 1997.

[44] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[45] C. Karlof and D. Wagner. Hidden Markov Model Cryptanalysis. Technical report, EECS Department, University of California, Berkeley, 2003.

[46] R. M. Karp. Reducibility among Combinatorial Problems. In *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Springer US, 1972.

[47] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the Spread of Influence Through a Social Network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 137–146, New York, NY, USA, 2003. ACM. ISBN 1-58113-737-0.

[48] D. Kempe, A. Mu'Alem, and M. Salek. Envy-Free Allocations for Budgeted Bidders. In *Proceedings of the 5th International Workshop on Internet and Network Economics*, pages 537–544, 2009.

[49] P. Kohli and P. H. S. Torr. Dynamic Graph Cuts for Efficient Inference in Markov Random Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2079–2088, Dec 2007. ISSN 0162-8828.

[50] N. Komodakis, G. Tziritas, and N. Paragios. Performance vs Computational Efficiency for Optimizing Single and Dynamic MRFs: Setting the State of the Art with Primal-dual Strategies. *Computer Vision and Image Understanding*, 112(1):14–29, Oct 2008. ISSN 1077-3142.

[51] A. Krause and D. Golovin. Submodular function maximization. In *Tractability: Practical Approaches to Hard Problems (to appear)*. Cambridge University Press, February 2014.

[52] A. Krause and C. Guestrin. A Note on the Budgeted Maximization of Submodular Functions. Technical Report CMU-CALD-05-103, Carnegie Mellon University, June 2005.

[53] A. Krause and C. Guestrin. Optimal Nonmyopic Value of Information in Graphical Models – Efficient Algorithms And Theoretical Limits. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1339–1345, July 2005.

[54] A. Krause and C. Guestrin. Near-optimal Nonmyopic Value of Information in Graphical Models. In *Proceedings of the 21st International Conference on Uncertainty in Artificial Intelligence (UAI)*, July 2005.

[55] A. Krause and C. Guestrin. Submodularity and its Applications in Optimized Information Gathering. *ACM Transactions on Intelligent Systems and Technology*, 2(4), 2011.

[56] A. Krause, A. Singh, and C. Guestrin. Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *Journal of Machine Learning Research (JMLR)*, 9:235–284, June 2008.

[57] A. Krause, C. Guestrin, A. Gupta, and J. Kleinberg. Robust Sensor Placements at Informative and Communication–Efficient Locations. *ACM Transactions on Sensor Networks (TOSN)*, 7, February 2011.

[58] C. Kreucher, K. Kastella, and A. Hero. Sensor management using an active sensing approach. *Signal Processing*, 85(3):607–624, 2005.

[59] A. Kulik, H. Shachnai, and T. Tamir. Maximizing Submodular Set Functions Subject to Multiple Linear Constraints. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 545–554, 2009.

[60] A. Kulik, H. Shachnai, and T. Tamir. Approximations for Monotone and Non-monotone Submodular Maximization with Knapsack Constraints. *Mathematics of Operations Research*, 38(4):729–739, 2013.

[61] J. Lee, V. S. Mirrokni, V. Nagarajan, and M. Sviridenko. Maximizing Non-monotone Submodular Functions under Matroid or Knapsack Constraints. *SIAM Journal of Discrete Mathematics*, 23(4):2053–2078, 2010.

[62] D. S. Levine and J. P. How. Sensor Selection in High-Dimensional Gaussian Trees with Nuisances. In *Advances in Neural Information Processing Systems 26*, pages 2211–2219. Curran Associates, Inc., 2013.

[63] D. S. Levine and J. P. How. Quantifying Nonlocal Informativeness in High-Dimensional, Loopy Gaussian Graphical Models. In *Proceedings of the 30th International Conference on Uncertainty in Artificial Intelligence (UAI)*, July 2014.

[64] Y. Liu, V. Chandrasekaran, A. Anandkumar, and A. S. Willsky. Feedback Message Passing for Inference in Gaussian Graphical Models. *IEEE Transactions on Signal Processing*, 60(8):4135–4150, Aug 2012.

[65] L. Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pages 235–257. Springer Berlin Heidelberg, 1983.

[66] E. R. Love. Some Logarithm Inequalities. *The Mathematical Gazette*, 64(427): 55–57, 1980.

[67] D. J. C. MacKay. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002. ISBN 0521642981.

[68] D. M. Malioutov. *Approximate Inference in Gaussian Graphical Models*. PhD thesis, Massachusetts Institute of Technology, May 2008.

[69] D. M. Malioutov, J. K. Johnson, and A. S. Willsky. Walk-Sums and Belief Propagation in Gaussian Graphical Models. *Journal of Machine Learning Research (JMLR)*, 7:2031–2064, Oct 2006.

[70] C. C. Moallemi and B. Van Roy. Consensus propagation. *IEEE Transactions on Information Theory*, 52(11):4753–4766, 2006.

[71] J. M. Mooij and H. J. Kappen. Sufficient Conditions for Convergence of the Sum-Product Algorithm. *IEEE Transactions on Information Theory*, 53(12): 4422–4437, Dec 2007.

[72] E. Mossel and S. Roch. On the submodularity of influence in social networks. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 128–134, New York, NY, USA, 2007. ACM.

[73] K. P. Murphy. *Machine learning: a probabilistic perspective*. The MIT Press, Cambridge, MA, 2012.

[74] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy Belief Propagation for Approximate Inference: An Empirical Study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 467–475, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1-55860-614-9.

[75] R. Nag, K. Wong, and F. Fallside. Script Recognition using Hidden Markov Models. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 11, pages 2071–2074, Apr 1986.

[76] A. Nath and P. Domingos. Efficient Belief Propagation for Utility Maximization and Repeated Inference. In *Proceedings of the 24th Conference on Artificial Intelligence (AAAI)*, July 2010.

[77] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An Analysis of Approximations for Maximizing Submodular Set Functions - I. *Mathematical Programming*, 14(1): 265–294, 1978.

[78] G. Papachristoudis and J. W. Fisher III. Theoretical Guarantees on Penalized Information Gathering. In *Statistical Signal Processing Workshop (SSP)*, August 2012.

[79] G. Papachristoudis and J. W. Fisher III. Efficient Information Planning in Gaussian MRFs. In *Proceedings of the 18th International Conference on Information Fusion*, July 2015.

[80] G. Papachristoudis and J. W. Fisher III. On the Complexity of Information Planning in Gaussian Models. In *Proceedings of the 40th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, April 2015.

[81] G. Papachristoudis and J. W. Fisher III. Adaptive Belief Propagation. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, July 2015.

[82] J. Pearl. Reverend Bayes on inference engines: A distributed hierarchical approach. In *Proceedings of the American Association of Artificial Intelligence National Conference (AAAI)*, pages 133–136, 1982.

[83] K. B. Petersen and M. S. Pedersen. The Matrix Cookbook, Nov 2012.

[84] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, volume 77, pages 257–286, San Francisco, CA, USA, February 1989.

[85] A. Rényi. On measures of information and entropy. In *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, pages 547–561, 1961.

[86] M. Salek and D. Kempe. Auctions for Share-Averse Bidders. In *Proceedings of the 4th International Workshop on Internet and Network Economics*, pages 609–620, 2008.

[87] M. J. Streeter and D. Golovin. An Online Algorithm for Maximizing Submodular Functions. In *Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems*, pages 1577–1584, December 2008.

[88] M. J. Streeter, D. Golovin, and A. Krause. Online Learning of Assignments. In *Advances in Neural Information Processing Systems 22*, pages 1794–1802, 2009.

[89] Ö. Sümer, U. A. Acar, A. T. Ihler, and R. R. Mettu. Adaptive exact inference in graphical models. *Journal of Machine Learning Research*, 12:3147–3186, Nov 2011.

[90] M. Sviridenko. A Note on Maximizing a Submodular Set Function Subject to a Knapsack Constraint. *Operations Research Letters*, 32(1):41–43, 2004.

[91] D. Takai and P. A. Jones. Comprehensive analysis of CpG islands in human chromosomes 21 and 22. *Proceedings of the National Academy of Sciences (PNAS)*, 99(6):3740–3745, March 2002.

[92] J. Vondrák. *Submodularity in Combinatorial Optimization.* PhD thesis, Charles University, 2007.

[93] J. Vondrák. Optimal Approximation for the Submodular Welfare Problem in the Value Oracle Model. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 67–74, New York, NY, USA, 2008. ACM.

[94] J. Vondrák. Submodularity and curvature: the optimal algorithm. *RIMS Kokyuroku Bessatsu B*, 23:253–266, 2010.

[95] M. J. Wainwright. Graphical models and message-passing algorithms: Some introductory lectures. Machine Learning Summer School, Kyoto, Japan, September 2012.

[96] M. J. Wainwright and M. I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, Jan 2008.

[97] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. MAP estimation via agreement on trees: Message-passing and linear programming. *IEEE Transactions on Information Theory*, 51:2005, 2005.

[98] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree-based Reparameterization Framework for Analysis of Sum-product and Related Algorithms. *IEEE Transactions on Information Theory*, 49(5):1120–1146, September 2006. ISSN 0018-9448.

[99] Y. Weiss and W. T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation*, 13:2173–2200, 2001.

[100] M. L. Wick and A. McCallum. Query-Aware MCMC. In *Advances in Neural Information Processing Systems 24*, pages 2564–2572, 2011.

[101] J. L. Williams. *Information Theoretic Sensor Management*. PhD thesis, Massachusetts Institute of Technology, February 2007.

[102] J. L. Williams, J. W. Fisher III, and A. S. Willsky. Performance Guarantees for Information Theoretic Active Inference. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS)*, March 2007.

[103] J. L. Williams, J. W. Fisher III, and A. S. Willsky. Approximate Dynamic Programming for Communication-Constrained Sensor Network Management. *IEEE Transactions on Signal Processing*, 55(8):3995–4003, August 2007.

[104] J. Yang and Y Xu. Hidden Markov Model for Gesture Recognition. Technical report, Robotics Institute, May 1994.

[105] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Generalized Belief Propagation. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 689–695, 2001.

[106] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding Belief Propagation and Its Generalizations. In *Exploring Artificial Intelligence in the New Millennium*, chapter 8, pages 239–236. Morgan Kaufmann Publishers, Jan 2003.

[107] F. Zhao, J. Shin, and J. Reich. Information-driven Dynamic Sensor Collaboration. *IEEE Signal Processing Magazine*, 19(2):61–72, March 2002.