DISS. ETH
NO. 25840

Alkis Gotovos

Sampling from Probabilistic Submodular Models

2019

DISS. ETH NO. 25840

Sampling from Probabilistic Submodular Models

A thesis submitted to attain the degree of

Doctor of sciences of ETH Zurich
(Dr. sc. ETH Zurich)

presented by

Alkis Gotovos
MSc ETH CS

born on Feb 12, 1987
citizen of Greece

accepted on the recommendation of

Prof. Dr. Andreas Krause,	examiner
Prof. Dr. Alexandros Dimakis,	co-examiner
Prof. Dr. Stefanie Jegelka,	co-examiner
Prof. Dr. Gunnar Rätsch,	co-examiner

2019

Abstract

Practical problems of discrete nature are very common in machine learning; application domains include computer vision (e.g., image segmentation), sequential decision making (e.g., active learning), social network analysis (e.g., influence maximization), and natural language processing (e.g., document summarization). Submodular set functions have found wide applicability in such problems for their ability to capture notions of coverage, diversity, or exclusivity; analogously, supermodular set functions have been used to capture notions of regularity, smoothness, or co-occurrence.

While the topic of submodular optimization has received much attention, these functions can also be used to define expressive discrete probabilistic models, called probabilistic submodular models. Going beyond optimization, these models allow us to quantify predictive uncertainty, and suggest a maximum likelihood approach for learning such functions from noisy data. Prominent examples of probabilistic submodular models include Ising and Potts models, as well as determinantal point processes, but the general class is much richer and little studied.

It is well known, though, that performing probabilistic inference in such models is computationally intractable in general. In this thesis, we investigate the use of Markov chain Monte Carlo sampling as a means of performing approximate inference in probabilistic submodular models.

We start with analyzing the Gibbs sampler, and establish theoretical conditions that guarantee efficient convergence of this sampler in probabilistic submodular models. We next propose a novel sampling procedure that makes use of discrete semigradients to perform efficient global moves, so as to avoid so-called state-space bottlenecks, and thus lead to improved convergence behavior. Finally, we employ the aforementioned sampling methods to approximate the likelihood gradients, and learn such models from data. We apply our learning procedure to the problem of modeling interactions between genetic mutations in cancer patients, and demonstrate considerable improvement over the state of the art in many of our experimental results on both synthetic and real cancer data.

Zusammenfassung

Praktische Probleme diskreter Natur sind weit verbreitet im maschinellen Lernen. Anwendungsbereiche umfassen unter anderen Computer Vision (z.B. Bildsegmentierung), sequentielle Entscheidungsfindung (z.B. aktives Lernen), soziale Netzwerkanalyse (z.B. Einflussmaximierung), und maschinelle Sprachverarbeitung (z.B. Dokumentzusammenfassung). Submodulare Mengenfunktionen werden in diesen Bereichen häufig aufgrund ihrer Fähigkeit, Überdeckungsprobleme, Diversität oder Exklusivität zu modellieren, eingesetzt. Analog werden Supermodulare Mengenfunktionen verwendet, um Regularität, Glattheit oder das gemeinsame Auftreten verschiedener Elemente zu modellieren.

Insbesondere der Themenbereich Submodulare Optimierung hat einige Aufmerksamkeit erregt, dabei können submodulare Funktionen ebenso der Definition diskreter probabilistischer Modelle dienen, auch bekannt unter dem Namen Probabilistische Submodulare Modelle. Über ihre Optimierung hinaus ermöglichen sie uns auch, stochastische Unsicherheit zu quantifizieren und legen eine Maximum-Likelihood-Methode nahe, um sie auf Basis verrauschter Daten zu lernen. Berühmte Beispiele Probabilistischer Submodularer Modelle sind vor allem Ising und Potts Modelle sowie Determinantal Point Processes. Doch die generelle Modellklasse ist um einiges reichhaltiger und weniger gut untersucht.

Allerdings ist allgemeint bekannt, dass die Inferenz dieser Modelle im Allgemeinen rechnerisch unmöglich ist. Deshalb untersuchen wir in dieser Dissertation die Markov-Ketten-Monte-Carlo (MCMC) Verfahren zur approximativen Inferenz Probabilistischer Submodularer Modelle.

Zunächst analysieren wir Gibbs Sampling und etablieren theoretische Voraussetzungen, unter denen dieser Algorithmus bei Probabilistischen Submodularen Modellen effizient konvergiert. Als nächstes stellen wir ein neuartiges Verfahren zur Stichprobenentnahme vor, die diskrete Halbgradienten gebraucht, um effiziente globale Schritte zu unternehmen. Dies vermeidet sogenannte Zustandsraumengpässe (bottlenecks) und führt zu verbessertem Konvergenzverhalten. Letztendlich benutzen wir zuvor erwähnte Sampling Methoden, um Likelihood-Gradienten zu approximieren und unsere Modelle aus Daten zu lernen. Konkret verwenden wir unser Lernverfahren, um die Interaktionen genetischer Mutationen in Krebspatienten zu modellieren. Sowohl für synthetische als auch echte Daten erzielen wir erhebliche Verbesserungen in vielen unserer Experimente im Vergleich zum neuesten Stand der Forschung.

Acknowledgements

I would like to thank my advisor, Andreas Krause, for his support over the years, always providing valuable feedback and suggesting new interesting directions to pursue. I would also like to thank Alex Dimakis, Stefanie Jegelka, and Gunnar Rätsch for agreeing and taking the time to participate in my thesis committee.

I am grateful to all my collaborators who gave me the opportunity to work with them. In particular, big thanks to Hamed Hassani and Stefanie Jegelka for our numerous discussions, which helped me get a deeper intuition into many of the topics related to this thesis.

Thanks to the members of the LAS group and the ML institute for all the good times, both within and outside academia.

Rebekka, thank you for making my life brighter over the past year, and for giving me the courage to complete this thesis. Finally, my gratitude goes out to my parents for their unending support and encouragement—I would not be here without you.

Contents

1 Introduction **1**
- 1.1 Probabilistic Submodular Models . . . 3
- 1.2 Thesis Topic & Contributions . . . 4
- 1.3 Collaborators . . . 4

2 Background **5**
- 2.1 Submodularity . . . 5
  - 2.1.1 Basics . . . 5
  - 2.1.2 Submodular Maximization . . . 7
- 2.2 Discrete Probabilistic Models . . . 7
  - 2.2.1 Inference . . . 12
- 2.3 Sampling . . . 14

3 Gibbs Sampling in Prob. Submodular Models **21**
- 3.1 Introduction . . . 21
- 3.2 Hardness of Inference . . . 22
  - 3.2.1 Example: Log-submodular Grid . . . 22
- 3.3 Polynomial-time Mixing . . . 25
  - 3.3.1 Examples . . . 25
  - 3.3.2 Mixing Time Bound . . . 26
  - 3.3.3 Proof of Theorem 3.2 . . . 26
- 3.4 Fast Mixing . . . 31
  - 3.4.1 Proof of Theorem 3.6 . . . 32
  - 3.4.2 Additively Decomposable Functions . . . 35
- 3.5 Experiments . . . 37
- 3.6 Conclusion . . . 39

4 Improved Mixing using Semigradients **43**
- 4.1 Introduction . . . 43
- 4.2 The Mixture Chain . . . 44
- 4.3 Ising Model on the Complete Graph . . . 47
- 4.4 Constructing the Mixture . . . 53

4.5 Experiments . . . 56
4.6 Conclusion . . . 61

5 Learning Prob. Submodular Models **63**
5.1 Introduction . . . 63
5.2 Approximate Maximum Likelihood Learning . . . 63
5.3 Modeling Interactions of Gene Mutations in Cancer . . . 67
5.3.1 Our Approach . . . 67
5.3.2 Experimental Setup . . . 68
5.4 Synthetic Data . . . 71
5.4.1 Learning . . . 71
5.4.2 Single Mutually Exclusive Group . . . 71
5.4.3 Multiple Mutually Exclusive Groups . . . 73
5.5 Real Cancer Data . . . 76
5.5.1 Acute Myeloid Leukemia (AML) . . . 76
5.5.2 Breast Cancer (BRCA) . . . 77
5.6 Conclusion . . . 88

6 Conclusion **89**
6.1 Future Work . . . 89

A Additional Experimental Results **91**
A.1 Results from Chapter 4 . . . 91
A.2 Results from Chapter 5 . . . 92

Bibliography **95**

1 Introduction

To introduce the main concepts of this thesis, we begin with a motivating application from the field of cancer genomics. One of the major undertakings in large-scale cancer genomics research projects, such as The Cancer Genome Atlas (TCGA, 2008), is obtaining and analyzing genetic data from cancer patients. Beyond investigating the occurrence of genetic mutations one by one, it is of particular interest to discover meaningful interactions between groups of mutations.

For example, it has been observed that, depending on the type of cancer, there are groups of specific mutations that are approximately mutually exclusive, that is, most of the time no more than one mutation from a particular group occurs in the same patient (Yeang et al., 2008). Biologically this is explained by the fact that so-called driver mutations, i.e., mutations that are crucial in cancer development, often occur in a limited number of biological pathways, and mutations that affect a specific pathway tend to not occur in the same patient. Conversely, discovering groups of mutually exclusive mutations may be helpful in uncovering the structure of cancer-related pathways, and identifying important groups of driver mutations.

More concretely, assume that we are given a data set of n mutations and m patients. In the simplest case, the data set contains only binary information about whether or not each mutation $i \in \{1, \ldots, n\}$ occurs in each patient $j \in \{1, \ldots, m\}$, which can be encoded using a binary matrix, as shown at the top of Figure 1.1. At the bottom of the same figure we show a permuted version of the previous matrix, which illustrates that the first four mutations are approximately mutually exclusive. Searching for such groups in data sets containing hundreds or thousands of mutations is a combinatorially daunting task. Crucially, the available data is quite limited—TCGA data sets range from a few hundred to a couple of thousand patients—and contains significant noise introduced by the employed measurement and preprocessing procedures.

Many other practical machine learning problems are of similar nature, that is, like the problem described above, they consist in choosing one or more subsets out of a set of finite elements. Examples include sensor place-

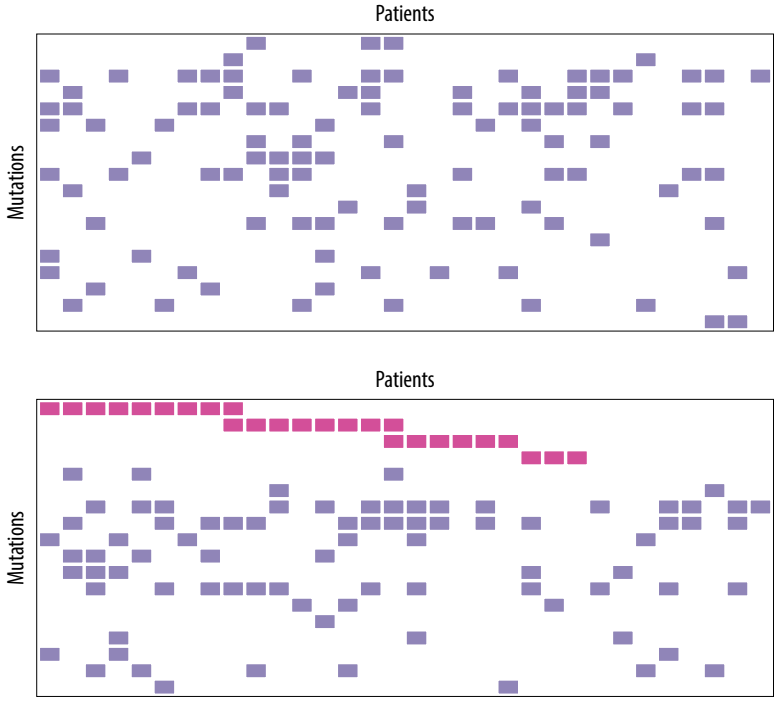


Figure 1.1: (top) An example binary mutation matrix, in which each shaded entry (i, j) indicates that mutation i occured in patient j. (bottom) The same matrix with permuted rows and columns to illustrate the mutual exclusivity between the first four mutations.

ment (Krause et al., 2006), active learning (Golovin & Krause, 2011), influence maximization (Kempe et al., 2003), image segmentation (Jegelka & Bilmes, 2011), and document summarization (Lin & Bilmes, 2011). While discrete optimization methods have been successful in many of these applications, it is often advantageous to go beyond optimization, and consider discrete probabilistic models.

The probabilistic nature of such models offers a way to deal with noisy data, and provides a flexible framework to robustly answer queries pertaining to the problem at hand. Rather than obtaining a single optimum as the solution to our problem, we have a way to quantify our uncertainty about

the most likely configurations, and make robust decisions based on computing various marginal and conditional probabilities of interest. In addition, the use of probabilistic models suggests a principled approach for learning the potentially complex interactions present in the data, namely maximizing the likelihood of the model parameters. Finally, constraining ourselves to specific model classes allows us to incorporate prior assumptions about the problem structure, and alleviate the scarce data issue.

1.1 Probabilistic Submodular Models

One one hand, past research on discrete probabilistic models has primarily focused on models defined by pairwise interactions, such as Markov random fields (Koller & Friedman, 2009). In many applications, however, it is of importance to directly capture higher-order dependencies between larger groups of variables. For example, in our aforementioned application, being able to directly encode larger groups of mutually exclusive mutations provides a potentially sparser and easier to interpret representation, while at the same time it allows for a richer structure of interactions.

On the other hand, in the context of discrete optimization, there has been extensive research on submodular set functions. Submodularity is a diminishing returns property that has been used to encode repulsiveness, diversity, or exclusivity. Analogously, its counterpart, supermodularity, has been used to encode attractiveness, cooperation, or co-occurence. Notably, there exist well-known efficient algorithms for both approximate submodular maximization as well as submodular minimization.

Merging these two directions naturally leads us to consider *probabilistic submodular models* (Djolonga & Krause, 2014; Gotovos et al., 2015), a class of discrete probabilistic models defined by submodular (or supermodular) functions. More concretely, given a ground set $V = \{1, \dots n\}$, a probabilistic submodular model is a distribution over subsets of V of the form

$$p(S; \boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left(F(S; \boldsymbol{\theta})\right),$$

for all $S \subseteq V$, where F is a submodular or supermodular function parameterized by $\boldsymbol{\theta}$, and $Z(\boldsymbol{\theta})$ is the normalizer of the distribution. Distributions of this form generalize some well-studied model classes, such as Ising models and determinantal point processes.

1.2 Thesis Topic & Contributions

Both learning the model parameters $\boldsymbol{\theta}$ from data, as well as quantifying uncertainty and making decisions with the learned distribution, boil down to the fundamental task of probabilistic inference, that is, computing the normalizer Z or various marginal probabilities of such distributions, a problem that is known to be computationally intractable in general. The main topic of this thesis is to investigate the use of Markov chain Monte Carlo sampling as a means of performing approximate inference in probabilistic submodular models.

The primary contributions of this thesis can be summarized as follows.

Chapter 3 We analyze the Gibbs sampler in probabilistic submodular models, and prove sufficient theoretical conditions for polynomial-time, and fast—$O(n \log n)$—mixing.

Chapter 4 We propose a novel sampler that makes use of discrete semigradients to perform efficient global moves in the state space to avoid bottlenecks, thus leading to improved mixing compared to the Gibbs sampler.

Chapter 5 We use sampling to learn probabilistic submodular models via approximate likelihood maximization, and apply this procedure to the problem of modeling interactions between genetic mutations in cancer patients. Many of our results demonstrate considerable improvement over the state of the art.

1.3 Collaborators

The topic of sampling from probabilistic submodular models was conceived by my advisor, Prof. Andreas Krause, who has also contributed to most parts of this thesis by providing large amounts of input and feedback over the years. Parts of the theoretical analysis in Chapter 3 and Chapter 4 were done in collaboration with Prof. Hamed Hassani. The work of Chapter 4 was done under the guidance of Prof. Stefanie Jegelka, who also contributed to the theoretical analysis of this chapter. Finally, regarding the application presented in Chapter 5, I have had several fruitful discussions with Gideon Dresdner, Dr. Kjong Lehmann, and Prof. Gunnar Rätsch.

2 Background

2.1 Submodularity

Modeling notions such as coverage, representativeness, or diversity is an important challenge in many machine learning problems. These notions are well captured by submodular set functions. Analogously, supermodular functions capture notions of smoothness, regularity, or cooperation. As a result, submodularity and supermodularity have found numerous applications in machine learning problems of discrete nature, akin to concavity and convexity in continuous optimization.

2.1.1 Basics

We consider set functions $F : 2^V \to \mathbb{R}$, where V is a finite ground set of size $|V| = n$. Without loss of generality, if not otherwise stated, we will hereafter assume that $V = [n] := \{1, 2, \ldots, n\}$. Adding an element i to a set S results in a difference in the value of F that is called marginal gain, and is defined as follows.

Definition 2.1 (Marginal gain). *For any $i \in V$, and $S \subseteq V$, the marginal gain of adding i to S is*

$$F(i \mid S) := F(S \cup \{i\}) - F(S).$$

Intuitively, submodularity expresses a notion of diminishing returns; that is, adding an element to a larger set provides less benefit than adding it to a smaller one.

Definition 2.2 (Submodularity). *F is submodular if, for any $S \subseteq T \subseteq V$, and any $v \in V \setminus T$, it holds that*

$$F(v \mid T) \leq F(v \mid S).$$

The following is an equivalent definition of submodularity that will also be useful later in the thesis.

Definition 2.3 (Submodularity). *F if submodular if, for any $A, B \subseteq V$, it holds that*

$$F(A \cup B) + F(A \cap B) \leq F(A) + F(B).$$

Supermodularity is defined analogously by reversing the sign of the above inequalities.

Definition 2.4 (Supermodularity). *A function F is supermodular if and only if $-F$ is submodular.*

If a function m is both submodular and supermodular, then it is called modular. Modular functions can be seen as the discrete analogue of linear continuous functions, and can be defined using a sum over real-numbered weights.

Definition 2.5 (Modularity). *A function m is called modular if it is both submodular and supermodular; it can be written as*

$$F(S) = c + \sum_{i \in S} m_i,$$

where $c \in \mathbb{R}$, and $m_i \in \mathbb{R}$, for all $i \in V$.

A function is called monotone when adding an element never decreases its value.

Definition 2.6 (Monotonicity). *A function F is monotone if, for any $i \in V$, and $S \subseteq V$, it holds that*

$$F(i \mid S) \geq 0.$$

Furthermore, a function F is called normalized if $F(\varnothing) = 0$. In some of our results we will use the fact that we can separate the non-normalized, and non-monotone parts of any submodular function according to the following decomposition.

Definition 2.7 (Submodular decomposition). *Any submodular function F can be decomposed as*

$$F(S) = c + m(S) + f(S),$$

for all $S \subseteq V$, where $c \in \mathbb{R}$ is a constant, m is a normalized modular function, and f is a normalized monotone submodular function.

An analogous decomposition using a monotone supermodular function f is possible for any supermodular function F as well.

Algorithm 2.1: Greedy submodular maximization

Input: Set function F, cardinality constraint k

$S^* \leftarrow \varnothing$

for $j = 1$ **to** k **do**

  Select $i^* \in \operatorname{argmax}_{i \in V \setminus S^*} F(i \mid S^*)$

  $S^* \leftarrow S^* \cup \{i^*\}$

return S

2.1.2 Submodular Maximization

Perhaps the most celebrated result pertaining to submodular functions is the approximation guarantee for maximizing a monotone submodular function under a cardinality constraint. Although the maximization problem itself is NP-hard, Nemhauser et al. (1978) showed that the simple greedy Algorithm 2.1, which repeatedly adds the element with the maximum marginal gain, identifies a solution that is within a factor of $1 - 1/e$ of the optimal value.

Theorem 2.8 (Nemhauser et al., 1978). *For any normalized monotone submodular function F, the solution S^* returned by Algorithm 2.1 satisfies*

$$F(S^*) \geq \left(1 - \frac{1}{e}\right) \max_{S \subseteq V, |S| \leq k} F(S).$$

Numerous extensions and generalizations of this result have been studied, including approximation guarantees for the non-monotone setting (Feige et al., 2011; Buchbinder et al., 2014); for different kinds of constraints, such as matroid (Lee et al., 2009; Calinescu et al., 2011) and knapsack (Chekuri et al., 2011); and for the adaptive setting (Golovin & Krause, 2011; Gotovos et al., 2015).

2.2 Discrete Probabilistic Models

As stated in the introduction, in the interest of venturing beyond discrete optimization, we consider discrete probabilistic models, that is, distributions

over finite subsets of the ground set V defined as

$$p(S;\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left(F(S;\boldsymbol{\theta})\right),$$

for all $S \subseteq V$. The function F is parameterized by a vector $\boldsymbol{\theta}$, and $Z(\boldsymbol{\theta})$ denotes the normalizing constant of the distribution, which is also often referred to as the partition function, and defined as

$$Z(\boldsymbol{\theta}) := \sum_{S \subseteq V} \exp\left(F(S;\boldsymbol{\theta})\right).$$

An alternative and equivalent way of defining distributions of the above form is via binary random vectors $X \in \{0,1\}^n$. If we define the transformation from vectors to sets, $V(X) := \{v \in V \mid X_v = 1\}$, it is easy to see that the distribution $p_X(X) \propto \exp(F(V(X)))$ over binary vectors is isomorphic to the above distribution over sets. With a slight abuse of notation, we will use $F(X)$ to denote $F(V(X))$, and use p to refer to both distributions.

For large parts of this thesis, we will focus on such distributions with F being submodular or supermodular.

Definition 2.9 (Probabilistic submodular model). *A probabilistic submodular model (Djolonga & Krause, 2014; Gotovos et al., 2015) is a distribution of the form*

$$p(S;\boldsymbol{\theta}) \propto \exp\left(F(S;\boldsymbol{\theta})\right),$$

for all $S \subseteq V$, where F is a submodular or supermodular function.

The resulting distributions of this form are also referred to as log-submodular and log-supermodular respectively. Note that the most likely configurations of these distributions directly correspond to the maximizers of the sub- or supermodular function F. Some commonly used discrete models fall under these categories; for example, the standard Ising and Potts models are log-supermodular, while determinantal point processes are log-submodular. We now present some examples models in more detail.

Example 2.10 (Product distribution). *Product or log-modular distributions describe a collection of n independent binary random variables. The corresponding function F is modular, that is, $F(S) = c + \sum_{i \in S} m_i$, and the partition function can be derived in closed form as*

$$Z = \exp(c) \prod_{i \in V} \left(1 + \exp(m_i)\right).$$

Consequently, a log-modular distribution can be written as

$$p(S) = \frac{\exp\left(\sum_{i \in S} m_i\right)}{\prod_{i \in V} (1 + \exp(m_i))}.$$

Note that the constant c does not appear in the distribution. More generally, the discrete models we consider are invariant to adding a constant to F, since that constant gets canceled by the partition function Z.

Example 2.11 (Ising model). *In its simplest form, the (ferromagnetic) Ising model (Ising, 1925) is defined via an undirected graph (V, E), and a set of "attractive" pairwise potentials*

$$\sigma_{i,j}(S) := 4\,([\![\{i \in S\}]\!] - 0.5)\,([\![\{j \in S\}]\!] - 0.5),$$

for all $\{i, j\} \in E$. We use $[\![\cdot]\!]$ to denote the Iverson bracket, which has value 1 when the enclosed condition is true, and 0 otherwise. We can see that $\sigma_{i,j}$ takes value 1 if S contains both or neither of i, j, and value -1 if it contains only one of i or j. It follows that each $\sigma_{i,j}$ is a supermodular set function. The Ising distribution is defined as

$$p(S) \propto \exp\left(\sum_{\{i,j\} \in E} \sigma_{i,j}(S)\right).$$

It is log-supermodular, since each $\sigma_{i,j}$ is supermodular, and supermodular functions are closed under addition.

We can also define the anti-ferromagnetic Ising model by a different set of "repulsive" pairwise potentials $\hat{\sigma}_{i,j}(S) := -\sigma_{i,j}(S)$. In this case, each $\hat{\sigma}_{i,j}$ is a submodular set function, and the resulting distribution is log-submodular.

Ising models, and Potts models (Potts, 1952), which generalize Ising models from binary to k-state variables, originate in statistical physics, but have also found numerous applications in computer vision (Wang et al., 2013).

Example 2.12 (Determinantal point process). *A determinantal point process (Lyons, 2003; Kulesza & Taskar, 2012) is defined via a positive semidefinite matrix $\boldsymbol{L} \in \mathbb{R}^{n \times n}$, and has a distribution of the form*

$$p(S) = \frac{\det(\boldsymbol{L}_S)}{\det(\boldsymbol{L} + \boldsymbol{I})},$$

where $\boldsymbol{L}_S$ denotes the square submatrix indexed by set S, and I is the $n \times n$ identity matrix. (We only describe here the form known as an $\boldsymbol{L}$-ensemble.) Since $F(S) = \log\det(\boldsymbol{L}_S)$ is a submodular function, determinantal point processes (DPPs) are log-submodular distributions. Interestingly, as we can see from the above equation, the partition function $Z = \det(\boldsymbol{L} + \boldsymbol{I})$ can be easily computed, which makes DPPs one of very few known tractable higher-order models.

DPPs originate in statistical physics, but have been used to encourage diversity in various machine learning applications, such as image and video summarization (Kulesza & Taskar, 2012; Gong et al., 2014).

Example 2.13 (FLiD). *Tschiatschek et al. (2016) defined the class of facility location diversity (FLiD) models by means of facility location functions, that is, functions of the form*

$$F(S) = \sum_{i \in S} u_i + \sum_{j=1}^{L} \left(\max_{i \in S} w_{ij} - \sum_{i \in S} w_{ij} \right),$$

where $w_{ij} \geq 0$. This is a submodular set function, therefore the resulting distribution $p(S) \propto \exp(F(S))$ is log-submodular.

The above function F is parameterized by a utility vector $\boldsymbol{u} \in \mathbb{R}^n$, and a diversity matrix $\boldsymbol{w} \in \mathbb{R}^{n \times L}$. Increasing the utility u_i of an element $i \in S$ intuitively increases the probability of all sets containing that element, therefore also increases its marginal probability. The diversity matrix $\boldsymbol{w}$ can be thought of as consisting of L latent dimensions (columns). Elements of the ground set that have large value in the same column j will tend to appear together less frequently, since the term $\max_{i \in S} w_{ij} - \sum_{i \in S} w_{ij}$ will be negative for sets S that contain combinations of such items. The structure of these models make them ideal for capturing mutual exclusivity, since the high-valued entries of each column of the $\boldsymbol{w}$ matrix intuitively encode a group of approximately mutually exclusive elements.

In Figure 2.1, we show an example FLiD model on ground set $V = \{1, 2, 3\}$. Note how the high value of $u_3 = 1$ increases the probability of sets containing element 3, but the high values of $w_{22} = w_{32} = 2$ significantly decrease the probability of sets $\{2, 3\}$ and V. Similar observations can be made about the lower probability of set $\{1, 2\}$ due to $w_{11} = w_{21} = 1$. In this sense, the model encodes two approximately mutually exclusive groups, namely $\{1, 2\}$, and $\{2, 3\}$.

Example 2.14 (FLDC). *Djolonga et al. (2016b) extended the FLiD model described above to include an additional matrix $\boldsymbol{v}$ that encodes "attraction" between groups of elements. The additional term is analogous to the original facility*

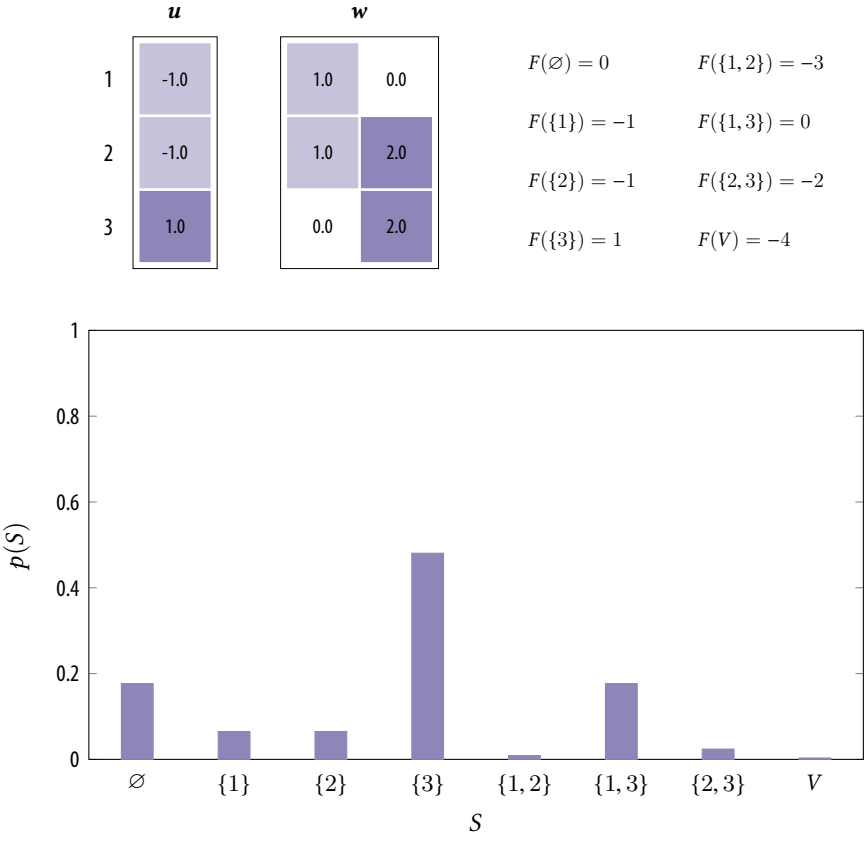


Figure 2.1: An example FLiD model with $L = 2$ dimensions on ground set $V = \{1, 2, 3\}$, the corresponding values of the facility location function $F(S)$, for all $S \subseteq V$, and the resulting distribution $p(S) \propto \exp(F(S))$.

location function, except that its sign is reversed. The corresponding function of the resulting facility location diversity and complements (FLDC) model is of the following form,

$$F(S) = \sum_{i \in S} u_i + \sum_{j=1}^{L} \left(\max_{i \in S} w_{ij} - \sum_{i \in S} w_{ij} \right) - \sum_{j=1}^{K} \left(\max_{i \in S} v_{ij} - \sum_{i \in S} v_{ij} \right).$$

We have now L latent dimensions encoding diversity or mutual exclusivity, and K latent dimensions encoding complementarity or co-occurrence. Note that the added term is a supermodular function, therefore $F(S)$ is neither submodular nor supermodular anymore. It follows that the distribution $F(S) \propto \exp(F(S))$ is not a probabilistic submodular model.

In Figure 2.2, we show an example FLDC model, which has the same $\boldsymbol{u}$ and $\boldsymbol{w}$ as our previous FLiD example, but additionally contains an attractive matrix $\boldsymbol{v}$ of dimension $K = 1$. The single latent dimension encodes co-occurrence between elements 1 and 3. While the only function values that change from before are $F(\{1, 3\})$ and $F(V)$, note that all probabilities $p(S)$ are different than those of Figure 2.1, because of the new partition function Z.

2.2.1 Inference

The basic tasks we would like to perform in a given discrete probabilistic model are computing various marginal probabilities of interest, and computing the partition function Z. These two tasks are more often that not tightly related to each other, and many algorithms that accomplish one of them can also accomplish the other with minor modifications. We therefore refer to them jointly as *probabilistic inference*.

There are very few classes of discrete probabilistic models that are known to be amenable to tractable exact inference. The most prominent such class is that of determinantal point processes that we described above. In fact, it is well known that performing exact inference in general probabilistic submodular models is a computationally intractable problem. Even for Ising models, Jerrum & Sinclair (1993) showed that exactly computing the partition function is a #P-hard problem. Worse than that, they showed that there can be no FPRAS for these models unless RP = NP. However, they also proposed a sampling-based procedure for performing approximate inference in ferromagnetic Ising models under some conditions. Several other conditional results for approximate sampling-based inference are known for Ising models (Ch. 15, Levin et al., 2008b). A generalization of determinantal point processes, called strongly Rayleigh distributions (Borcea et al., 2008), are the

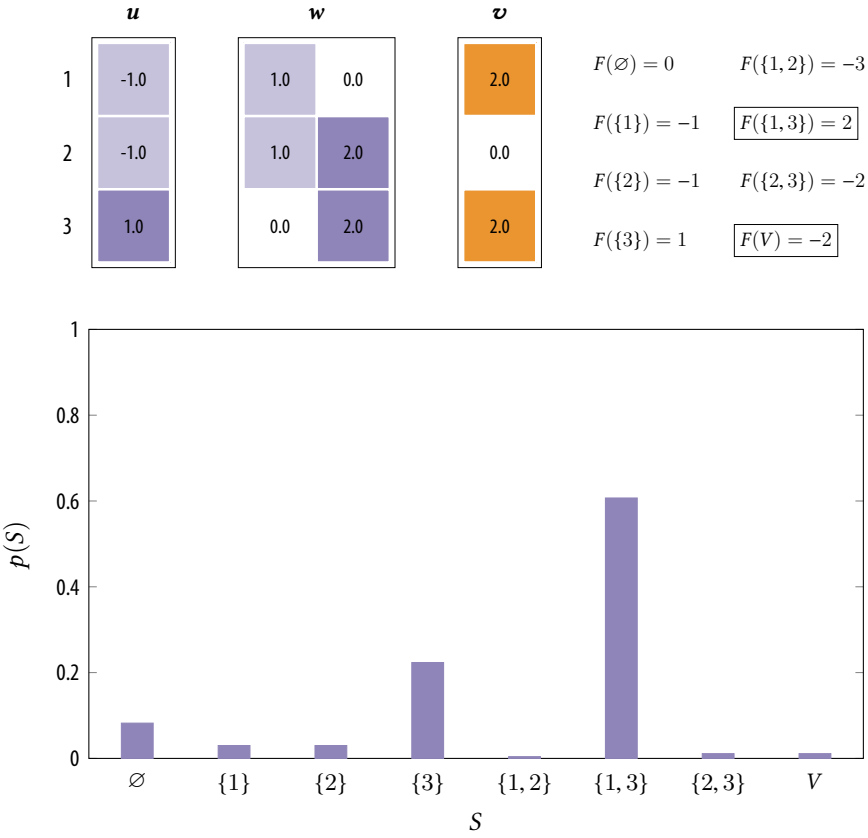


Figure 2.2: An example FLDC model with $L = 2$ and $K = 1$ dimensions on ground set $V = \{1, 2, 3\}$, the corresponding values of $F(S)$, for all $S \subseteq V$, and the resulting distribution $p(S) \propto \exp(F(S))$. The two function values $F(\{1, 3\})$ and $F(V)$ are the only ones that change compared to the previous FLiD example.

result of a long line of research into the notion of negative dependence between random variables (Pemantle, 2000; Liggett, 2002; Wagner, 2008). It has been shown that one can efficiently sample from strongly Rayleigh distributions using a simple Metropolis sampler (Anari et al., 2016; Li et al., 2016).

Iyer & Bilmes (2015) considered a different class of probabilistic models, called submodular point processes, which are also defined via submodular functions, and are of the form $p(S) \propto F(S)$. They showed that inference in these models is, in general, also a hard problem, and provided approximations and closed-form solutions for some subclasses.

Besides sampling, the primary alternative for performing approximate inference in discrete models have been variational methods. The fundamental idea of these methods is to choose a distribution among a tractable class to approximate the true distribution at hand. There has been extensive research on variational methods for low-order models, particularly for exponential families; many well-known algorithms, such as belief propagation and mean-field methods, fall under this category. (For an introductory treatment we refer to the monograph by Wainwright & Jordan (2008).) More recently, Djolonga & Krause (2014) proposed a variational approach for performing approximate probabilistic inference in probabilistic submodular models based on log-modular approximations of the distribution.

2.3 Sampling

In this thesis, we focus on Markov chain Monte Carlo sampling algorithms, which are based on performing randomly selected moves in a state space Ω to approximate probabilistic quantities of interest. The visited states $(X_0, X_1, \ldots)$ form a Markov chain, which under mild conditions converges to a stationary distribution p (see Theorem 4.9, Levin et al., 2008b). Crucially, the probabilities of transitioning from one state to another are carefully chosen to ensure that the stationary distribution is identical to the distribution of interest. In our case, the state space is the powerset of our ground set, that is, $\Omega := 2^V$, and we want to construct a chain over subsets of V that has stationary distribution p. We denote by $P : \Omega \times \Omega \to \mathbb{R}$ the transition matrix of a Markov chain, that is,

$$P(S, R) := \mathbb{P}\left[X_{t+1} = R \,|\, X_t = S\right],$$

for all $S, R \in \Omega$.

We next present two well-known chains that we will use throughout this thesis. Both of them are by construction reversible with respect to $p(\cdot) \propto$

$\exp(F(\cdot))$, that is, they satisfy the detailed balance conditions

$$p(S)P(S,R) = p(R)P(R,S),$$

for all $S, R \in \Omega$. It follows that they asymptotically converge to the unique stationary distribution p.

Gibbs sampler. One of the most commonly used chains is the (single-site) Gibbs sampler, also known as the Glauber dynamics, which adds or removes a single element at a time. It first selects uniformly at random an element $i \in V$; subsequently, it adds or removes i to the current state X_t according to the probability of the resulting state, as shown in Algorithm 2.2. More concretely, we define an adjacency relation $S \sim R$ on the elements of the state space, which denotes that S and R differ by exactly one element, i.e., $\big||R| - |S|\big| = 1$. It follows that each $S \in \Omega$ has exactly n neighbors. We also define

$$p_{S \to R} = \frac{\exp(F(R))}{\exp(F(R)) + \exp(F(S))}.$$

Then, the transition matrix P^{G} of the Gibbs sampler is

$$P^{\mathrm{G}}(S,R) = \begin{cases} \frac{1}{n} p_{S \to R}, & \text{if } R \sim S \\ 1 - \sum_{T \sim S} \frac{1}{n} p_{S \to T}, & \text{if } R = S \\ 0, & \text{otherwise} \end{cases}.$$

In Figure 2.3 we illustrate one step of the Gibbs sampler on a small ground set $V = \{1, 2, 3\}$. Assuming that the current state is $X_t = \{2\}$, there are three potential new next states, namely $\varnothing$, $\{1, 2\}$, or $\{2, 3\}$, which are the three neighbors of X_t in the state space. The Gibbs sampler first selects one of the neighbors uniformly at random, and then either stays at X_t or moves to that neighbor according to the corresponding conditional probability.

It is important to note here that the computed conditional probabilities do not depend on the partition function Z, thus the chain can be simulated efficiently, even though Z is unknown and hard to compute. Moreover, it is easy to see that

$$\Delta_F(i \mid X_t) = [\![i \notin X_t]\!] F(i \mid X_t) + [\![i \in X_t]\!] F(i \mid X_t \setminus \{i\}).$$

Therefore, the Gibbs sampler only requires a black box for the marginal gains of F, which are often faster to compute in practice than the values of F.

Algorithm 2.2: The Gibbs sampler for prob. submodular models.

Input: Ground set V, distribution $p(\cdot) \propto \exp(F(\cdot))$

$X_0 \leftarrow$ random subset of V
for $t = 0$ **to** M **do**
  Draw $i \sim \textsc{Unif}(V)$
  $p_{\text{add}} \leftarrow \exp(\Delta_F(i \mid X_t))/(1 + \exp(\Delta_F(i \mid X_t)))$
  Draw $z \sim \textsc{Unif}([0, 1])$
  if $z \leq p_{\text{add}}$ **then**
    $X_{t+1} \leftarrow X_t \cup \{i\}$
  else
    $X_{t+1} \leftarrow X_t \setminus \{i\}$

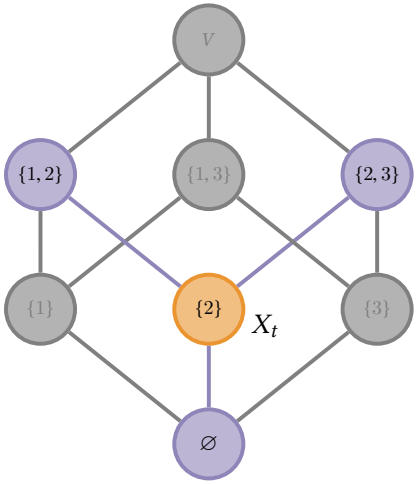


Figure 2.3: An illustration of a single Gibbs step on ground set $V = \{1, 2, 3\}$, and current state $X_t = \{2\}$. We first choose one of the neighbors of X_t uniformly at random, and then either stay at X_t or move to that neighbor according to the corresponding conditional probability.

Algorithm 2.3: The Metropolis sampler for prob. sub. models.

Input: Ground set V, distribution $p(\cdot) \propto \exp(F(\cdot))$, proposal q

$X_0 \leftarrow$ random subset of V
for $t = 0$ **to** M **do**
  Draw $S \sim q(\cdot \mid X_t)$
  $p_{\text{acc}} \leftarrow \min\left\{1, \frac{q(S \mid R)\,\exp(F(R))}{q(R \mid S)\,\exp(F(S))}\right\}$
  Draw $z \sim \textsc{Unif}([0,1])$
  if $z \leq p_{acc}$ **then**
    $X_{t+1} \leftarrow S$
  else
    $X_{t+1} \leftarrow X_t$

Metropolis sampler. Another well-studied chain is the Metropolis chain (Metropolis et al., 1953; Hastings, 1970), which, like the Gibbs sampler, also performs local moves between neighboring states, but does so following a somewhat different procedure. The Metropolis chain first draws a candidate next state according to a proposal distribution $q(\cdot \mid X_t)$; then, it either accepts the proposed state or not according to the probability ratio of the two states corrected by the proposal ratio, as shown in Algorithm 2.3.

More concretely, if we define the acceptance probability

$$p_{\text{acc}}(S,R) := \min\left\{1, \frac{q(S \mid R)\,p(R)}{q(R \mid s)\,p(S)}\right\} = \min\left\{1, \frac{q(S \mid R)\,\exp(F(R))}{q(R \mid s)\,\exp(F(S))}\right\},$$

then the transition matrix of the Metropolis chain can be defined as follows,

$$P^{\text{M}}(S,R) = \begin{cases} q(R \mid S)\,p_{\text{acc}}(S,R), & \text{if } R \neq S \\ 1 - \sum_{T \neq S} q(T \mid S)\,p_{\text{acc}}(S,T), & \text{otherwise} \end{cases}.$$

As with the Gibbs sampler, the computed acceptance probabilities do not depend on the partition function Z, therefore the chain can be simulated efficiently.

Approximating expectations. Approximating quantities of interest using MCMC methods is largely based on using time averages to estimate expectations over the desired distribution. In particular, we estimate the expected value of function $g : \Omega \to \mathbb{R}$ by

$$\mathbb{E}_p[g(X)] \approx \frac{1}{M} \sum_{r=1}^{M} g(X_{s+r}). \tag{2.1}$$

For example, to estimate the marginal probability $\mathbb{P}(i \in S)$, for some $i \in V$, we can define $g(S) = [\![i \in S]\!]$, for all $S \in \Omega$. The point in time $s \in \mathbb{N}$, after which we start taking samples into account, is often referred to as the burn-in time of the chain. Our goal is to perform marginal inference for the distributions described above. Concretely, for some fixed $A \subseteq B \subseteq V$, we would like to compute the probability of sets S that contain all elements of A, but no elements outside of B, that is, $p(A \subseteq S \subseteq B)$.

Approximating the partition function. There are two straightforward methods for estimating the partition function Z using sampling. The first, importance sampling (IS) (Neal, 2001), assumes that we have a tractable distribution $\pi : 2^V \to \mathbb{R}$, from which we draw M samples $\{x_{s+1}, \ldots, x_{s+M}\}$. Then, we can estimate the partition function of $p(\cdot) \propto \exp(F(\cdot))$ by

$$Z_{\text{IS}} := \frac{1}{M} \sum_{r=1}^{M} \frac{\exp(F(x_{s+r}))}{\pi(x_{s+r})}. \tag{2.2}$$

Although this is known to be an unbiased estimator of Z, it often has high variance, and tends to underestimate the true value of the partition function.

The second method, reverse important sampling (RIS) (Gelfand & Dey, 1994), works in the opposite direction, by first sampling M samples $\{x_{s+1}, \ldots, x_{s+M}\}$ from the target distribution p, and then estimating the partition function by

$$Z_{\text{RIS}} := \left(\frac{1}{M} \sum_{r=1}^{M} \frac{\pi(x_{s+r})}{\exp(F(x_{s+r}))} \right)^{-1}. \tag{2.3}$$

Similarly to the IS estimator, the RIS estimator also often has high variance, but, in contrast to IS, it tends to overestimate the true value of the partition function. Taking the average $0.5Z_{\text{IS}} + 0.5Z_{\text{RIS}}$ is a natural way to obtain an improved estimate, while other more involved related methods have also been proposed (Burda et al., 2015; Liu et al., 2015).

Mixing time. The choice of burn-in time s and number of samples M in (2.1)–(2.3) presents a tradeoff between computational efficiency and approximation accuracy. It turns out that the effect of both s and M is largely dependent on a fundamental quantity of the chain called *mixing time* (Levin et al., 2008b).

The mixing time of a chain quantifies the number of iterations t required for the distribution of X_t to get close to the stationary distribution p. More formally, it is defined as

$$t_{\text{mix}}(\epsilon) := \min \{t \mid d(t) \leq \epsilon\},$$

where $d(t)$ denotes the worst-case (over the starting state X_0 of the chain) total variation distance between the distribution of X_t and π, that is,

$$d(t) := \max_{X_0 \in \Omega} \|P^t(X_0, \cdot) - p\|_{\text{TV}}.$$

A generalization of Chebyshev's inequality to correlated Markov chain samples (see Theorem 12.19, Levin et al., 2008b) shows that upper bounding the mixing time is sufficient to guarantee efficient approximate sampling-based marginal inference.

3 Gibbs Sampling in Prob. Submodular Models

The majority of the content of this chapter has already been published in conference proceedings (Gotovos et al., 2015).

3.1 Introduction

In this chapter, we consider one of the simplest and most commonly used sampling procedures, namely the (single-site) Gibbs sampler, which is also known as the Glauber chain. While there has been extensive work on the properties of the Gibbs sampler on low-order models, for example, Ising models (Levin et al., 2008b, Ch. 15), not much is known about its behavior on higher-order models, except that, in general, we cannot hope for subexponential mixing times (Jerrum & Sinclair, 1993). In fact, we show that even for probabilistic submodular models defined by monotone submodular functions, there are simple model families with exponential lower bounds on mixing time.

Our goal is to establish theoretical conditions that guarantee rapid mixing of the Gibbs sampler in probabilistic submodular models, and at the same time, investigate in what way the properties of sub- and supermodularity affect the resulting conditions.

We focus on distributions of the form

$$p(S) = \frac{\exp(\beta F(S))}{Z}, \tag{3.1}$$

for all $S \subseteq V$, where F is submodular or supermodular. For now we assume that F is already learned or given, and omit the parameter vector $\boldsymbol{\theta}$ from the notation. Furthermore, we have introduced a scaling parameter $\beta \geq 0$, which is referred to as inverse temperature, and will be useful for our subsequent theoretical analysis. Intuitively, β controls the concentration of p around the high-value sets of F. When $\beta = 0$, p is the uniform distribution over 2^V; when $\beta \to \infty$, the mass of p fully concentrates around the maximizers of F.

3.2 Hardness of Inference

Performing exact inference in probabilistic submodular models is, in general, computationally infeasible. Only for very few exceptions, such as determinantal point processes, is exact inference possible in polynomial time (Kulesza & Taskar, 2012). As we mentioned before, even approximating the partition function of general Ising models—a subclass of probabilistic submodular models—is a hard problem; in particular, there is no FPRAS for this problem, unless RP = NP (Jerrum & Sinclair, 1993). This implies that the mixing time of any Markov chain with such a stationary distribution will, in general, be exponential in the size n of the ground set.

3.2.1 Example: Log-submodular Grid

To further highlight the hardness of inference in the general models we consider, we show that even for distributions defined through a seemingly benign subclass of submodular functions, mixing times can be exponential in n.

For the purposes of the following proposition, we will use a Metropolis chain (see Section 2.3), rather than a Gibbs chain, to simplify the exposition. While the two chains are not identical, they share the same principle of making local moves by considering ratios of probabilities of neighboring states.

Proposition 3.1. *There is a family of monotone submodular functions $(F_n)_n$, such that, for the corresponding log-submodular family of distributions $(p_n)_n$, the Metropolis chain has mixing time*

$$t_{\text{mix}} = \Omega(2^{n/2}),$$

for any value of β.

Proof. The functions used to prove the above lemma are based on the following construction. For any even $n \geq 2$, let $V_n = \{1, \ldots, n\}$, $R_n = \{1, \ldots, n/2\}$, and $C_n = \{n/2 + 1, \ldots, n\}$. To define function $F_n : 2^{V_n} \to \mathbb{R}$, we conceptually use a $n/2 \times n/2$ square grid, whose rows are indexed by R_n and columns by C_n. Each cell (i, j) of the grid is considered to be covered, if either row $i \in R$ or column $j \in C$ is selected. Formally, we define F_n by

$$F_n(S) = \frac{4}{n^2} \Big| \{(i, j) \in R \times C \mid i \in S \vee j \in S\} \Big|,$$

for any $S \subseteq V_n$, which results in $F_n(V_n) = 1$. Figure 3.1 shows an example of such a grid construction.

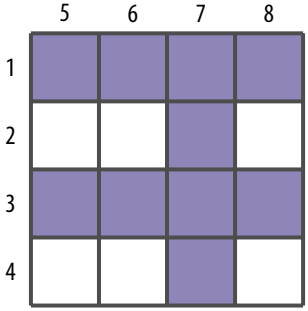


Figure 3.1: Example grid for $n = 8$ with the cells that correspond to $F_8(\{1, 3, 7\}) = 10/16$ shown shaded.

Furthermore, if we define $\mathcal{R}_n = \{S \subseteq V \mid R \subseteq S\}$, $\mathcal{C}_n = \{S \mid C \subseteq S \subseteq V\}$, and $\mathcal{K}_n = 2^V \setminus (\mathcal{R}_n \cup \mathcal{C}_n)$, then the following properties hold.

$$|\mathcal{R}_n| = |\mathcal{C}_n| = 2^{n/2} \tag{3.2}$$

$$\mathcal{R}_n \cap \mathcal{C}_n = \{V\} \tag{3.3}$$

$$\forall S \in \mathcal{R}_n \cup \mathcal{C}_n,\ f(S) = 1 \tag{3.4}$$

$$\forall S \in \mathcal{K}_n,\ f(S) \leq 1 - 4/n^2. \tag{3.5}$$

Assume a Metropolis chain with transition matrix P, and stationary distribution $p_n(S) \propto \exp(\beta F_n(S))$. To prove a lower bound on the mixing time of this chain, we are going to upper bound the bottleneck ratio (Levin et al., 2008a, Ch. 7) of set $\mathcal{T}_n = \mathcal{R}_n \setminus \{V\}$, defined as

$$\Phi(\mathcal{T}_n) = \frac{Q(\mathcal{T}_n, \mathcal{T}_n^c)}{\pi(\mathcal{T}_n)} = \frac{Q(\mathcal{T}_n, \mathcal{C}_n) + Q(\mathcal{T}_n, \mathcal{K}_n)}{\pi(\mathcal{T}_n)},$$

where $\mathcal{T}_n^c = 2^V \setminus \mathcal{T}_n$ is the complement of $\mathcal{T}_n$. We now compute or bound each of the terms $\pi(\mathcal{T}_n)$, $Q(\mathcal{T}_n, \mathcal{C}_n)$, and $Q(\mathcal{T}_n, \mathcal{K}_n)$.

- Computing $\pi(\mathcal{T}_n)$:

$$\begin{aligned} \pi(\mathcal{T}_n) &= |\mathcal{T}_n| \frac{e^\beta}{Z} && \text{by (3.4)} \\ &= (2^{n/2} - 1) \frac{e^\beta}{Z}. && \text{by (3.2)} \end{aligned}$$

Note that, by an analogous derivation, we get $\pi(C_n \setminus \{V\}) = \pi(\mathcal{T}_n)$ and, by (3.3),

$$\begin{aligned} &\pi(\mathcal{T}_n) + \pi(C_n \setminus \{V\}) < 1 \\ \Rightarrow\ &\pi(\mathcal{T}_n) < 0.5. \end{aligned}$$

- Computing $Q(\mathcal{T}_n, C_n)$:

$$\begin{aligned} Q(\mathcal{T}_n, C_n) &= \sum_{x \in \mathcal{T}_n} Q(x, C_n) \\ &= \sum_{x \in \mathcal{T}_n} Q(x, \{V\}) && \text{by (3.3)} \\ &= \sum_{x \in \mathcal{T}_n} \frac{1}{2n} \frac{e^{\beta}}{Z} && \text{by (3.4)} \\ &= n \frac{1}{2n} \frac{e^{\beta}}{Z} = \frac{e^{\beta}}{2Z}. \end{aligned}$$

- Bounding $Q(\mathcal{T}_n, \mathcal{K}_n)$:

$$\begin{aligned} Q(\mathcal{T}_n, \mathcal{K}_n) &= \sum_{x \in \mathcal{T}_n} Q(x, \mathcal{K}_n) \\ &\leq \sum_{x \in \mathcal{T}_n} \frac{1}{2n} \frac{e^{\beta - 4\beta/n^2}}{Z} && \text{by (3.5)} \\ &= \frac{2^{n/2} - 1}{2n} \frac{e^{\beta - 4\beta/n^2}}{Z}. && \text{by (3.2)} \end{aligned}$$

- Bounding $\Phi(\mathcal{T}_n)$:

$$\begin{aligned} \Phi(\mathcal{T}_n) &\leq \frac{1}{2^{n/2} - 1} \left(\frac{1}{2} + \frac{(2^{n/2} - 1) e^{-4\beta/n^2}}{2n} \right) \\ &= \frac{1}{2(2^{n/2} - 1)} + \frac{e^{-4\beta/n^2}}{2n}. \end{aligned}$$

Using Theorem 7.3 (Levin et al., 2008b), it follows that

$$t_{\text{mix}}(1/4) \geq \frac{1}{4\Phi(\mathcal{T}_n)} = \Omega(2^{n/2}).$$

□

3.3 Polynomial-time Mixing

Our first result provides conditions that guarantee polynomial mixing times in the size n of the ground set. As we will see, the conditions depend crucially on the following quantity, which is defined for any set function $F : 2^V \to \mathbb{R}$,

$$\zeta_F := \max_{A,B \subseteq V} |F(A) + F(B) - F(A \cup B) - F(A \cap B)| .$$

Intuitively, ζ_F quantifies a notion of distance to modularity. For submodular and supermodular functions, ζ_F represents the worst-case amount by which F violates the submodular inequality, and $\zeta_F = 0$ if and only if F is modular (cf. Section 2.1).

It is also important to note that, for submodular and supermodular functions, ζ_F depends only on the monotone part of F; if we decompose F according to Definition 2.7, then it is easy to see that $\zeta_F = \zeta_f$. A trivial upper bound on ζ_F, therefore, is $\zeta_F \leq f(V)$. Another quantity that has been used in the past to quantify the deviation of a submodular function from modularity is the curvature (Conforti & Cornuejols, 1984), defined as $\kappa_F := 1 - \min_{i \in V} (F(i \mid V \setminus \{i\})/F(i))$. Although of similar intuitive meaning, the multiplicative nature of its definition makes it significantly different from ζ_F, which is defined additively.

3.3.1 Examples

Concave over modular. As an example of a function class with ζ_F independent of n, assume a ground set $V = \bigcup_{\ell=1}^{L} V_\ell$, and consider functions of the form

$$F(S) = \sum_{\ell=1}^{L} \phi(|S \cap V_\ell|),$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is a bounded concave function, e.g, $\phi(x) = \min\{\phi_{\max}, x\}$. Functions of this form are submodular, and have been used in applications such as document summarization to encourage diversity (Lin & Bilmes, 2011). It is easy to see that $\zeta_F \leq L\phi_{\max}$, which shows that ζ_F is independent of n.

FLiD. For the FLiD model (see Example 2.13), we have $f(S) = \sum_{j=1}^{L} \max_{i \in S} w_{ij}$, therefore we get $\zeta_F \leq f(V) = \sum_{j=1}^{L} w_j^{\max}$, where $w_j^{\max} = \max_{i \in V} w_{ij}$. Since the values of $\boldsymbol{w}$ depend primarily on the number of repulsive groups, rather than the size of the ground set, we expect ζ_F to grow much slower than n in most practical applications.

FLDC. For the FLDC model (see Example 2.14), although F is neither submodular nor supermodular in general, we can still write it as

$$F(S) = m(S) + g(S) + h(S),$$

where m is a modular function, $g(S) := \sum_{j=1}^{L} \max_{i \in S} w_{ij}$ is submodular, and $h(S) := -\sum_{j=1}^{K} \max_{i \in S} v_{ij}$ is supermodular. Using the triangle inequality in the definition of ζ_F, we get that $\zeta_F \leq g(V) + h(V) = \sum_{j=1}^{L} w_j^{\max} + \sum_{j=1}^{K} v_j^{\max}$, where $w_j^{\max} = \max_{i \in V} w_{ij}$, and $v_j^{\max} = \max_{i \in V} v_{ij}$.

3.3.2 Mixing Time Bound

The following theorem establishes a bound on the mixing time of the Gibbs sampler run on models of the form (3.1). The bound is exponential in ζ_F, but polynomial in n.

Theorem 3.2. *For any function $F : 2^V \to \mathbb{R}$, the mixing time of the Gibbs sampler is bounded by*

$$t_{\text{mix}}(\epsilon) \leq 2n^2 \exp(2\beta\zeta_F) \log\left(\frac{1}{\epsilon p_{\min}}\right),$$

where $p_{\min} := \min_{S \in \Omega} p(S)$. If F is submodular or supermodular, then the bound is improved to

$$t_{\text{mix}}(\epsilon) \leq 2n^2 \exp(\beta\zeta_f) \log\left(\frac{1}{\epsilon p_{\min}}\right).$$

Note that, since the factor of two that constitutes the difference between the two statements of the theorem lies in the exponent, it can have a significant impact on the above bounds. The dependence on $p_{\min}$ is related to the (worst-case) starting state of the chain, and can be eliminated if we have a way to guarantee a high-probability starting state. If F is submodular or supermodular, this is usually straightforward to accomplish by using one of the standard constant-factor optimization algorithms (see Section 2.1) as a preliminary step. More generally, if F is bounded by $0 \leq F(S) \leq F_{\max}$, for all $S \subseteq V$, then $\log(1/p_{\min}) = O(n\beta F_{\max})$.

3.3.3 Proof of Theorem 3.2

Our proof of Theorem 3.2 is based on the method of *canonical paths* (Jerrum, 2003; Sinclair, 1992; Jerrum & Sinclair, 1989; Diaconis & Stroock, 1991). The

high-level idea of this method is to view the state space as a graph, and try to construct a path between each pair of states, which carries a certain amount of flow specified by the stationary distribution under consideration. Depending on the choice of these paths and the resulting load on the edges of the graph we can derive bounds on the mixing time of the Markov chain.

More concretely, let us assume that for some set function F and corresponding distribution p as in (3.1), we construct the Gibbs chain on state space $\Omega = 2^V$ with transition matrix P. We can view the state space as a directed graph that has vertex set Ω, and for any $A, B \in \Omega$, contains edge (S, S') if and only if $S \sim S'$, that is, if and only if S and S' differ by exactly one element. Now, for any pair of states $A, B \in \Omega$, we define a canonical path

$$\gamma_{AB} := (A = S_0, S_1, \ldots, S_\ell = B),$$

such that all (S_i, S_{i+1}) are edges in the above graph. We denote the length of path γ_{AB} by $|\gamma_{AB}|$, and define $Q(S, S') := p(S)P(S, S')$. We also denote the set of all pairs of states whose canonical path goes through (S, S') by

$$C_{SS'} := \{(A, B) \in \Omega \times \Omega \mid (S, S') \in \gamma_{AB}\}.$$

The following quantity, referred to as the congestion of an edge, uses a collection of canonical paths to quantify to what amount that edge is overloaded:

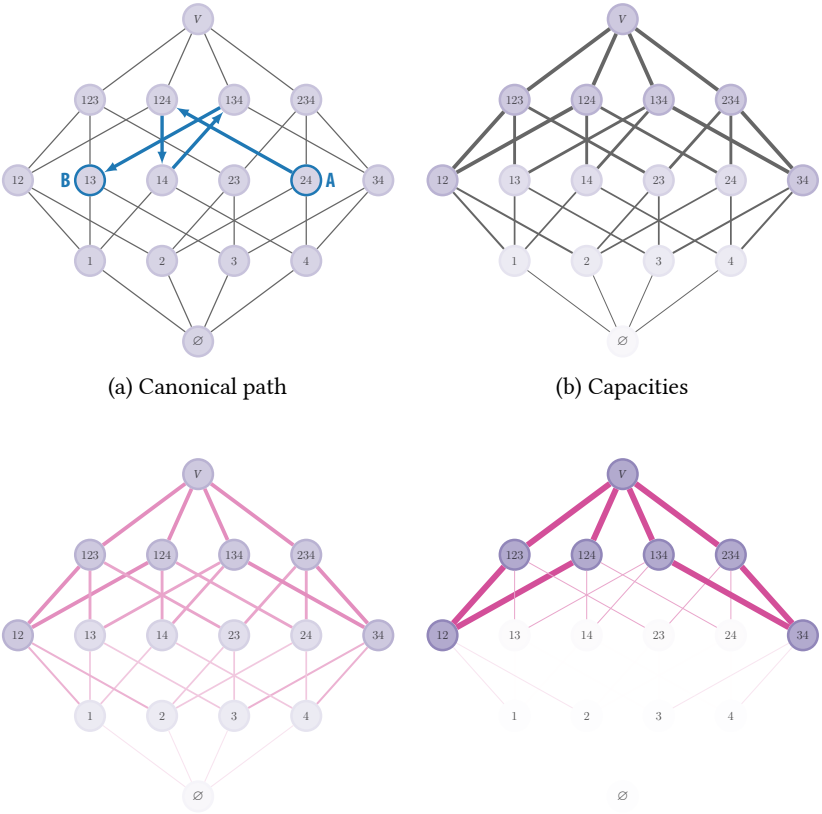
$$\rho(S, S') := \frac{1}{Q(S, S')} \sum_{(A,B)\in C_{SS'}} p(A)p(B)|\gamma_{AB}|. \tag{3.6}$$

The denominator $Q(S, S')$ quantifies the capacity of edge (S, S'), while the sum represents the total flow through that edge according to the choice of canonical paths. The congestion of the whole graph is then defined as $\rho := \max_{S\sim S'} \rho(S, S')$. Low congestion implies that there are no bottlenecks in the state space, and the chain can move around fast, which results in rapid mixing. The following theorem makes this statement more concrete.

Theorem 3.3 (Sinclair, 1992; Jerrum, 2003). *For any collection of canonical paths with congestion ρ, the mixing time of the chain is bounded by*

$$t_{\text{mix}}(\epsilon) \leq \rho \log\left(\frac{1}{\epsilon p_{\min}}\right),$$

where $p_{\min} := \min_{S\in\Omega} p(S)$.



(a) Canonical path

(b) Capacities

(c) Low congestion

(d) High congestion

Figure 3.2: (a) The state space for ground set $V = \{1, 2, 3, 4\}$, and an illustration of a canonical path from $A = \{2, 4\}$ to $B = \{1, 3\}$. (b) For an example distribution p, the width of each edge denotes the corresponding capacity $Q(S, S')$. (c) The color of each edge denotes the corresponding congestion $\rho(S, S')$; darker edges indicate higher congestion. (d) Similar to (c), but for a different distribution that has almost all of its mass concentrated on seven states. It can been seen that it has notably higher congestion ρ, and contains a significant bottleneck at $S = \{V\}$.

To apply Theorem 3.3 to our class of distributions, we need to construct a set of canonical paths in the corresponding state space 2^V, and upper bound the resulting congestion. First, note that, to transition from state $A \in \Omega$ to state $B \in \Omega$, in our case, it is enough to remove the elements of $A \setminus B$ and add the elements of $B \setminus A$. Each removal and addition corresponds to an edge in the state space graph, and the order of these operations identify a canonical path in this graph that connects A to B. For our analysis, we assume a fixed order on V (e.g., the natural order of the elements themselves), and perform the operations according to this order. Figure 3.2a shows an example of such a canonical path for a small state space, and Figures 3.2b–3.2d illustrate the capacities $Q(S, S')$, and congestions $\rho(S, S')$ for two example distributions.

Having defined the set of canonical paths, we proceed to bounding the congestion $\rho(S, S')$ for any edge (S, S'). The main difficulty in bounding $\rho(S, S')$ is due to the sum in (3.6) over all pairs in $\mathcal{C}_{SS'}$. To simplify this sum, we construct for each edge (S, S') an injective map $\eta_{SS'} : \mathcal{C}_{SS'} \to \Omega$; this is a combinatorial encoding technique that has been previously used in similar proofs to ours (Jerrum, 2003). The following lemma details this construction, where for sets A, B, we denote $A \oplus B := (A \setminus B) \cup (B \setminus A)$.

Lemma 3.4. *Define the maps $\eta_{SS'} : \mathcal{C}_{SS'} \to \Omega$, for each pair $(S, S') \in \Omega \times \Omega$ with $S \sim S'$, as follows:*

$$\eta_{SS'}(A, B) = \begin{cases} A \oplus B \oplus S, & \text{if } F(S') \geq F(S) \\ A \oplus B \oplus S', & \text{otherwise} \end{cases}.$$

Then, each map $\eta_{SS'}$ is injective.

Proof. Assume that $F(S') \geq F(S)$, and $S' = S \cup \{r\}$, for some $r \in V$. Assume that we are given $C := A \oplus B \oplus S$, and we want to recover A and B. We will denote by $\prec$ the natural ordering of the ground set V. First, we define

$$K^- := \{i \in C \oplus S \mid i \prec r\}$$
$$K^+ := \{i \in C \oplus S \mid i \succ r\}.$$

Then, we can recover A and V as follows:

$$A = S \oplus K^-$$
$$B = S' \oplus K^+.$$

The case $S' = S \setminus \{r\}$, as well as the two cases for $F(S') < F(S)$ are completely analogous. Note that the distinction based on the value of the function has

no effect on the proof here, but is technically needed for the next lemma. The only thing that changes between the cases is whether the element r that gets added or removed in the transition (S, S') belongs to A or B, which is always straightforward to determine from the type of the transition (for additions it belongs to B, and for removals to A). □

We then prove the following key lemma about the maps constructed above.

Lemma 3.5. *For any $S \sim S'$, and any $A, B \in \Omega$, it holds that*

$$p(A)p(B) \leq 2n \exp(2\beta\zeta_F) Q(S, S') p(\eta_{SS'}(A, B)).$$

If F is submodular or supermodular, then the bound is improved to

$$p(A)p(B) \leq 2n \exp(\beta\zeta_f) Q(S, S') p(\eta_{SS'}(A, B)).$$

Proof. We will consider the case $S' = S \cup \{r\}$, for some $r \in V$, with $F(S') \geq F(S)$. Again, the other three cases are completely analogous by using $\eta_{SS'}$ as defined in Lemma 3.4.

We first compute

$$\begin{aligned}
Q(S, S') &= p(S)P(S, S') \\
&= \frac{1}{n} \frac{p(S)p(S')}{p(S) + p(S')} && \text{by definition of the Gibbs sampler} \\
&= \frac{1}{nZ} \frac{\exp(\beta F(S)) \exp(\beta F(S'))}{\exp(\beta F(S)) + \exp(\beta F(S'))} && \text{by definition of our models} \\
&\geq \frac{1}{nZ} \frac{\exp(\beta F(S)) \exp(\beta F(S'))}{2 \exp(\beta F(S'))} && \text{by } F(S') \geq F(S) \\
&= \frac{\exp(\beta F(S))}{2nZ}.
\end{aligned}$$

As a result, we get

$$\frac{p(A)p(B)}{Q(S, S')} \leq \frac{2n}{Z} \exp(\beta(F(A) + F(B) - F(S))). \tag{3.7}$$

Let us denote

$$\zeta_F(A, B) := F(A) + F(B) - F(A \cup B) - F(A \cap B),$$

for any $A, B \subseteq V$, so that $\zeta_F = \max_{A,B \subseteq V} |\zeta_F(A, B)|$. Then, if we denote

$$C := \eta_{SS'}(A, B) = A \oplus B \oplus S,$$

we have

$$\begin{aligned}
&F(A) + F(B) - F(S) \\
&= (F(A) + F(B) - F(A \cup B) - F(A \cap B)) - \\
&\quad (F(S) + F(C) - F(A \cup B) - F(A \cap B)) + F(C) \\
&= (F(A) + F(B) - F(A \cup B) - F(A \cap B)) - \\
&\quad (F(S) + F(C) - F(S \cup C) - F(S \cap C)) + F(C) \\
&= \zeta_F(A, B) - \zeta_F(S, C) + F(C) \\
&\leq 2\zeta_F + F(C).
\end{aligned}$$

If F is submodular, then $\zeta_F(A, B)$ and $\zeta_F(S, C)$ are both non-negative, therefore $\zeta_F(A, B) - \zeta_F(S, C) + F(C) \leq \zeta_F + F(C) = \zeta_f + F(C)$. Similarly, if F is supermodular, then $\zeta_F(A, B)$ and $\zeta_F(S, C)$ are both non-positive, therefore $\zeta_F(A, B) - \zeta_F(S, C) + F(C) \leq \zeta_F + F(C) = \zeta_f + F(C)$. Substituting these bounds in (3.7) gives us the result of the lemma. □

Since $\eta_{SS'}$ is injective, it follows that $\sum_{(A,B)\in C_{SS'}} p(\eta_{SS'}(A, B)) \leq 1$. Furthermore, it is clear that each canonical path γ_{AB} has length $|\gamma_{AB}| \leq n$, since we need to add and/or remove at most n elements to get from state A to state B. Combining these two facts with the above lemma, we get

$$\rho(S, S') \leq 2n^2 \exp(2\beta\zeta_F),$$

for any set function F, and

$$\rho(S, S') \leq 2n^2 \exp(2\beta\zeta_f),$$

if F is sub- or supermodular.

3.4 Fast Mixing

We now proceed to show that, under some stronger conditions, we are able to establish even faster—$O(n \log n)$—mixing. For any function F, we denote

$$\Delta_F(i \mid S) := F(S \cup \{i\}) - F(S \setminus \{i\}),$$

and define the following quantity,

$$\gamma_{F,\beta} := \max_{\substack{S\subseteq V \\ r\in V}} \sum_{i\in V} \tanh\left(\frac{\beta}{2}\left|\Delta_F(i \mid S) - \Delta_F(i \mid S \cup \{r\})\right|\right),$$

which quantifies the (maximum) total influence of an element $r \in V$ on the values of F. For example, if the inclusion of r makes no difference with respect to other elements of the ground set, we will have $\gamma_{F,\beta} = 0$. The following theorem establishes conditions for fast mixing of the Gibbs sampler when run on models of the form (3.1).

Theorem 3.6. *For any set function $F : 2^V \to \mathbb{R}$, if $\gamma_{F,\beta} < 1$, then the mixing time of the Gibbs sampler is bounded by*

$$t_{\text{mix}}(\epsilon) \leq \frac{1}{1 - \gamma_{F,\beta}} n \left(\log n + \log \frac{1}{\epsilon} \right).$$

If F is additionally submodular or supermodular, and is decomposed according to Definition 2.7, then

$$t_{\text{mix}}(\epsilon) \leq \frac{1}{1 - \gamma_{f,\beta}} n \left(\log n + \log \frac{1}{\epsilon} \right).$$

Note that, in the second part of the theorem, $\gamma_{f,\beta}$ depends only on the monotone part of F.

3.4.1 Proof of Theorem 3.6

Our proof of Theorem 3.6 is based on the coupling technique (Aldous, 1983); more specifically, we use the *path coupling* method (Bubley & Dyer, 1997; Levin et al., 2008a; Jerrum, 2003). Given a Markov chain (Z_t) on state space Ω with transition matrix P, a coupling for (Z_t) is a new Markov chain (X_t, Y_t) on state space $\Omega \times \Omega$, such that both (X_t) and (Y_t) are by themselves Markov chains with transition matrix P. The idea is to construct the coupling in such a way that, even when the starting points X_0 and Y_0 are different, the chains (X_t) and (Y_t) tend to coalesce. Then, it can be shown that the coupling time $t_{\text{couple}} := \min \{t \geq 0 \mid X_t = Y_t\}$ is closely related to the mixing time of the original chain (Z_t) (Levin et al., 2008a).

The main difficulty in applying the coupling approach lies in the construction of the coupling itself, for which one needs to consider any possible pair of states (X_t, Y_t). The path coupling technique makes this construction easier by utilizing the same state-space graph that we used to define canonical paths in Section 3.3. The core idea is to first define a coupling only over adjacent states, and then extend it for any pair of states by using a metric on the graph. More concretely, let us denote by $d : \Omega \times \Omega \to \mathbb{R}$ the path metric on state space Ω; that is, for any $x, y \in \Omega$, $d(x, y)$ is the minimum length

of any path from x to y in the state space graph. The following theorem establishes fast mixing using this metric, as well as the diameter of the state space, $\mathrm{diam}(\Omega) := \max_{x,y\in\Omega} d(x,y)$.

Theorem 3.7 (Bubley & Dyer, 1997; Levin et al., 2008a). *For any Markov chain (Z_t), let (X_t, Y_t) be a coupling, such that, for some $a \geq 0$, and any $x, y \in \Omega$ with $x \sim y$, it holds that*

$$\mathbb{E}[d(X_{t+1}, Y_{t+1}) \,|\, X_t = x, Y_t = y] \leq e^{-\alpha} d(x,y).$$

Then, the mixing time of the original chain (Z_t) is bounded by

$$t_{\mathrm{mix}}(\epsilon) \leq \frac{1}{\alpha}\left(\log(\mathrm{diam}(\Omega)) + \log\frac{1}{\epsilon}\right).$$

In our case, the path metric d is the Hamming distance between the binary vectors representing the states (equivalently, the number of elements by which two sets differ). We need to construct a suitable coupling (X_t, Y_t) for any pair of states $x \sim y$. Consider the two corresponding sets $S, R \subseteq V$ that differ by exactly one element, and assume that $R = S \cup \{r\}$, for some $r \in V$. (The case $S = R \cup \{s\}$ for some $s \in V$ is completely analogous.) Remember that the Gibbs sampler first chooses an element $i \in V$ uniformly at random, and then adds or removes it according to the conditional probabilities. Our goal is to make the same updates happen to both S and R as frequently as possible. As a first step, we couple the candidate element for update $i \in V$ to always be the same in both chains. Then, we have to distinguish between the following cases.

If $i = r$, then the conditionals for both chains are identical, and we can couple both chains to add r with probability

$$p_{\mathrm{add}} := \frac{p(S \cup \{r\})}{p(S) + p(S \cup \{r\})},$$

which will result in new sets $S' = R' = S \cup \{r\}$, or remove r with probability $1 - p_{\mathrm{add}}$, which will result in new sets $S' = R' = S$. Either way, we will have $d(S', R') = 0$.

If $i \neq r$, we cannot always couple the updates of the chains, because the conditional probabilities of the updates are different. In fact, we are forced to have different updates (one chain adding i, the other chain removing i) with probability equal to the difference of the corresponding conditionals, which we denote here by $p_{\mathrm{dif}}(v)$, defined as follows,

$$p_{\mathrm{dif}}(i) := \left| \frac{p(S \cup \{i\})}{p(S \cup \{i\}) + p(S \setminus \{i\})} - \frac{p(R \cup \{i\})}{p(R \cup \{i\}) + p(R \setminus \{i\})} \right|.$$

In the case of different updates, we have $d(S', R') = 2$, otherwise the chains make the same update and still differ only by element r, that is, $d(S', R') = 1$.

Putting together the three possible cases for the value of $d(S', R')$ described above, we get the following expected distance after one step,

$$\mathbb{E}[d(S', R')] = 1 - \frac{1}{n} + \frac{1}{n} \sum_{i \neq r} p_{\text{dif}}(i).$$

We then prove the following lemma to bound the sum of p_{dif}.

Lemma 3.8. *For any $S, R \subseteq V$ with $R = S \cup \{r\}$,*

$$\sum_{i \neq r} p_{\text{dif}}(i) \leq \gamma_{F,\beta}.$$

Proof. For any $i \neq r$, we have

$$\begin{aligned}
p_{\text{dif}}(i) &= \left| \frac{\exp(\beta F(S \cup \{i\}))}{\exp(\beta F(S \cup \{i\})) + \exp(\beta F(S \setminus \{i\}))} - \right. \\
&\qquad \left. \frac{\exp(\beta F(R \cup \{i\}))}{\exp(\beta F(R \cup \{i\})) + \exp(\beta F(R \setminus \{i\}))} \right| \\
&= \left| \frac{\exp(\beta \Delta_F(i \mid S))}{1 + \exp(\beta \Delta_F(i \mid S))} - \frac{\exp(\beta \Delta_F(i \mid R))}{1 + \exp(\beta \Delta_F(i \mid R))} \right| \\
&= \left| \frac{\exp(\beta \Delta_F(i \mid S)) - \exp(\beta \Delta_F(i \mid R))}{(1 + \exp(\beta \Delta_F(i \mid S)))(1 + \exp(\beta \Delta_F(i \mid R)))} \right| \\
&\leq \left| \frac{\exp(\beta \Delta_F(i \mid S)) - \exp(\beta \Delta_F(i \mid R))}{\exp(\beta \Delta_F(i \mid S)) + \exp(\beta \Delta_F(i \mid R))} \right| \\
&= \left| \frac{\exp(\beta(\Delta_F(i \mid S) - \Delta_F(i \mid R))) - 1}{\exp(\beta(\Delta_F(i \mid S) - \Delta_F(i \mid R))) + 1} \right| \\
&= \tanh\left(\frac{\beta}{2} \left| (\Delta_F(i \mid S) - \Delta_F(i \mid R)) \right| \right).
\end{aligned}$$

The lemma follows by definition of $\gamma_{F,\beta}$, and the fact that $R = S \cup \{r\}$. □

Applying this lemma, we get

$$\mathbb{E}[d(S', R')] = 1 - \frac{1}{n} + \frac{1}{n} \sum_{i \neq r} p_{\text{dif}}(i)$$

$$\leq 1 - \frac{1}{n}(1 - \gamma_{F,\beta})$$
$$\leq \exp\left(-\frac{1 - \gamma_{F,\beta}}{n}\right),$$

and the result of Theorem 3.6 follows from applying Theorem 3.7 with $\alpha = \gamma_{F,\beta}/n$, and noting that $\operatorname{diam}(\Omega) = n$.

The specialization of Theorem 3.6 to sub- or supermodular functions is based on the following lemma.

Lemma 3.9. *If F is submodular or supermodular, and decomposed according to Definition 2.7, then*

$$\gamma_{F,\beta} = \gamma_{f,\beta}.$$

Proof. For any $S, R \subseteq V$ with $R = S \cup \{r\}$, and any $i \in V$, we have

$$\begin{aligned}
&\Delta_F(i \mid S) - \Delta_F(i \mid R) \\
&\quad = F(S \cup \{i\}) - F(S \setminus \{i\}) - F(R \cup \{i\}) + F(R \setminus \{i\}) \\
&\quad = f(S \cup \{i\}) - f(S \setminus \{i\}) - f(R \cup \{i\}) + f(R \setminus \{i\}) \\
&\quad = \Delta_f(i \mid S) - \Delta_f(i \mid R).
\end{aligned}$$

□

3.4.2 Additively Decomposable Functions

Some commonly used models, such as the Ising model and FLiD, can be written as a sum of simpler supermodular (resp. submodular) functions F_j,

$$F(S) = \sum_{j \in [L]} F_j(S). \tag{3.8}$$

We prove the following corollary that provides an easy to check condition for fast mixing of the Gibbs sampler when F can be additively decomposed as above.

Corollary 3.10. *For any submodular function F that can be written in the form of (3.8), with f being its monotone (also additively decomposable) part according to Definition 2.7, if we define*

$$\theta_f := \max_{i \in V} \sum_{j \in [L]} \sqrt{f_j(\{i\})} \quad \text{and} \quad \lambda_f := \max_{j \in [L]} \sum_{i \in V} \sqrt{f_j(\{i\})},$$

then it holds that

$$\gamma_{f,\beta} \leq \frac{\beta}{2}\theta_f \lambda_f.$$

Proof. For any $S, R \subseteq V$ with $R = S \cup \{r\}$, we have

$$\begin{aligned}
&\sum_{i \neq r} \tanh\left(\frac{\beta}{2}\big|(\Delta_f(i \mid S) - \Delta_f(i \mid R))\big|\right) \\
&\leq \sum_{i \neq r} \frac{\beta}{2}\big|(\Delta_f(i \mid S) - \Delta_f(i \mid R))\big| && \text{by } \tanh(x) \leq x, \text{ for all } x \geq 0 \\
&\leq \sum_{i \neq r} \frac{\beta}{2}(\Delta_f(i \mid S) - \Delta_f(i \mid R)) && \text{by submodularity of } f \\
&= \frac{\beta}{2}\sum_{i \neq r}(f(S \cup \{i\}) - f(S \setminus \{i\}) - f(S \cup \{r\} \cup \{i\}) + f(S \cup \{r\} \setminus \{i\})) \\
&= \frac{\beta}{2}\sum_{i \neq r}\sum_{j \in [L]}(f_j(S \cup \{i\}) - f_j(S \setminus \{i\}) - f_j(S \cup \{r\} \cup \{i\}) + f_j(S \cup \{r\} \setminus \{i\})) \\
&\leq \frac{\beta}{2}\sum_{i \neq r}\sum_{j \in [L]} \min\left\{f_j(S \cup \{i\}) - f_j(S \setminus \{i\}), f_j(S \cup \{r\} \setminus \{i\}) - f_j(S \setminus \{i\})\right\} \\
& && \text{by monotonicity of } f_j \\
&\leq \frac{\beta}{2}\sum_{i \neq r}\sum_{j \in [L]} \min\left\{f_j(i), f_j(r)\right\} && \text{by submodularity of } f_j \\
&\leq \frac{\beta}{2}\sum_{i \neq r}\sum_{j \in [L]} \sqrt{f_j(i) f_j(r)} \\
&= \frac{\beta}{2}\sum_{j \in [L]} \sqrt{f_j(r)} \sum_{i \neq r} \sqrt{f_j(i)}.
\end{aligned}$$

The result follows by maximizing both sides over S and r. □

Example. Applying the above corollary to the FLiD model, we get

$$\theta_f = \max_{i \in V} \sum_{j \in [L]} \sqrt{w_{ij}},$$

and

$$\lambda_f = \max_{j \in [L]} \sum_{i \in V} \sqrt{w_{ij}},$$

and we obtain fast mixing if $\theta_f \lambda_f \leq 2/\beta$. As a special case, if we consider the class of set cover functions ($w_{ij} \in \{0, 1\}$), such that each $i \in V$ covers at most δ sets, and each set indicated by $j \in [L]$ is covered by at most δ elements, then $\theta_f, \lambda_f \leq \delta$, and we obtain fast mixing if $\delta^2 \leq 2/\beta$. Note, that the corollary can be trivially applied to any submodular function by taking $L = 1$, but may, in general, result in a loose bound if used that way.

3.5 Experiments

In the following two experiments, we compare the Gibbs sampler against the variational approach proposed by Djolonga & Krause (2014) for performing inference in probabilistic submodular models. In particular, the authors propose two variational approximations, denoted in the following by "upper" and "lower", which are obtained from factorized distributions associated with modular upper and lower bounds respectively.

Estimating the log-partition function. We start with approximating the normalizers $\log(Z)$ for a family of (log-submodular) FLiD models on ground set sizes ranging from $n = 10$ to $n = 100$. These FLiD models are learned from synthetic data that we describe in Section 5.4.2. In short, each model represents a single approximately mutually exclusive group of three genes together with five frequently and independently occurring genes, as well as a number of random noise genes.

We obtain estimates for $\log(Z)$ via a Gibbs-based reverse importance sampling procedure (see Section 2.3), using 200, 1000, and 5000 samples. For each model we repeat the sampling procedure 100 times to get standard error estimates. Since estimating the exact value of $\log(Z)$ is infeasible for $n > 20$, we obtain an accurate estimate by computing the averaged importance sampling and reverse importance sampling estimates when run with $2 \cdot 10^6$ samples. Figure 3.3 shows the estimation errors with respect to this approximately true value; error bars depict two standard errors. As is natural, more Gibbs samples result in more accurate estimates, and we can also observe that reverse importance sampling tends to produce overestimates of the log-partition function. We also see that the two variational approaches, which guarantee upper and lower bounds respectively, are considerably less accurate.

Estimating marginals. We now repeat the experiments performed by Djolonga & Krause (2014) to estimate marginals, and use the same three models

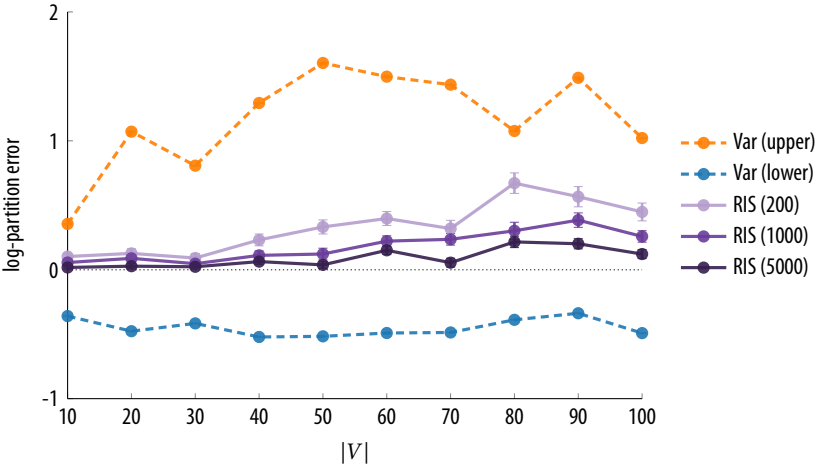


Figure 3.3: The error in estimating the log-partition function when using Gibbs-based reverse importance sampling compared to the variational approximations by Djolonga & Krause (2014).

that they used.

The first is a FLiD model, in which a manually added modular term penalizes the number of selected elements, that is, $p(S) \propto \exp(f(S) - 2|S|)$, where f is a submodular facility location function. The model is constructed from randomly subsampling real data from a problem of sensor placement in a water distribution network (Krause et al., 2008). In the experiments, we iteratively condition on random observations for each variable in the ground set.

The second is a log-supermodular pairwise Markov random field, constructed by first randomly sampling points from a two-dimensional two-cluster Gaussian mixture model, and then introducing a pairwise potential for each pair of points with exponentially-decreasing weight in the distance of the pair. In the experiments, we iteratively condition on pairs of observations, one from each cluster.

The third is a log-supermodular higher-order Markov random field, which is constructed by first generating a random Watts-Strogatz graph, and then creating one higher-order potential per node, which contains that node and all of its neighbors in the graph. The strength of the potentials is controlled

by a parameter μ, which is closely related to the curvature of the functions that define them. In the experiments, we vary this parameter from 0 (modular model) to 1 ("strongly" supermodular model).

For all three models, we constrain the size of the ground set to $n = 20$, so that we are able to compute, and compare against, the exact marginals. Furthermore, we run multiple repetitions for each model to account for the randomness of the model instance, and the random initialization of the Gibbs sampler. The marginals we compute are of the form $p(i \in S \,|\, C \subseteq S \subseteq D)$, for all $i \in V$. As before, we run the Gibbs sampler for 200, 1000, and 5000 iterations on each problem instance.

Figure 3.4 compares the average absolute error of the approximate marginals with respect to the exact ones. The averaging is performed over $i \in V$, and over the different repetitions of each experiment; error bars depict two standard errors. We notice a similar trend on all three models. For the regimes that correspond to less "peaked" posterior distributions (small number of conditioned variables, small μ), even a few thousand Gibbs iterations outperform both variational approximations. On the other hand, the variational methods gain an advantage when the posterior is concentrated around only a few states, which happens after having conditioned on almost all variables in the first two models, or for μ close to 1 in the third model.

3.6 Conclusion

In this chapter, we presented two conditions that guarantee upper bounds on the mixing time of the Gibbs sampler in probabilistic submodular models. Furthermore, we demonstrated that, in practice, the Gibbs sampler compares favorably to previously proposed variational approximations, particularly in regimes of high uncertainty.

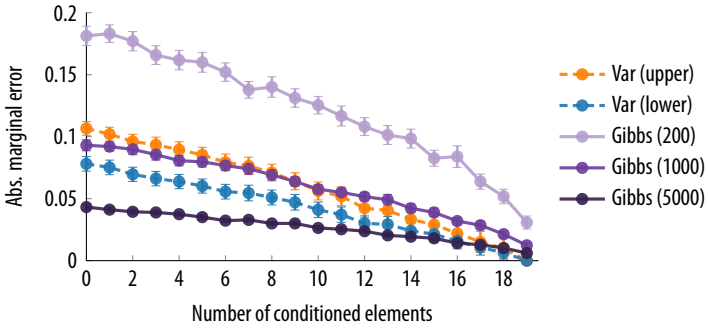
Further related work. In contemporary work to ours, Rebeschini and Karbasi (Rebeschini & Karbasi, 2015) analyzed the mixing times of log-submodular models. Using a method based on matrix norms, which was previously introduced by Dyer et al. (2009), and is closely related to path coupling, they arrived at a similar, though not directly comparable, condition to that of Theorem 3.6.

Li et al. (2016) extended our polynomial-time mixing result of Theorem 3.2 to the problem of sampling from a distribution under specific constraints. In particular, they used a similar to ours canonical path argument involving an analogous quantity to our ζ_F to prove mixing bounds under uniform and

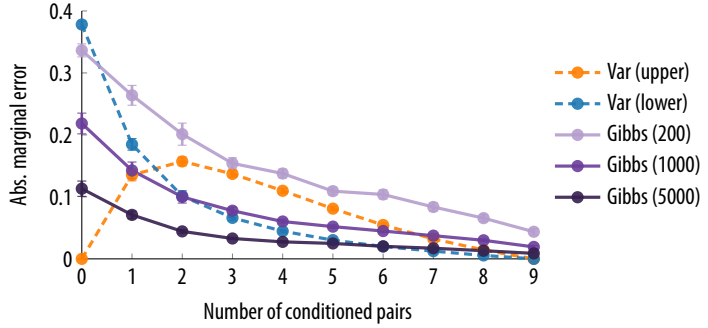
partition matroid constraints.

The canonical path method for bounding mixing times has been previously used in a number of theoretical results, such as approximating the partition function of ferromagnetic Ising models (Jerrum & Sinclair, 1993), approximating matrix permanents (Jerrum & Sinclair, 1989; Jerrum et al., 2004a), and counting matchings in graphs (Jerrum, 2003).

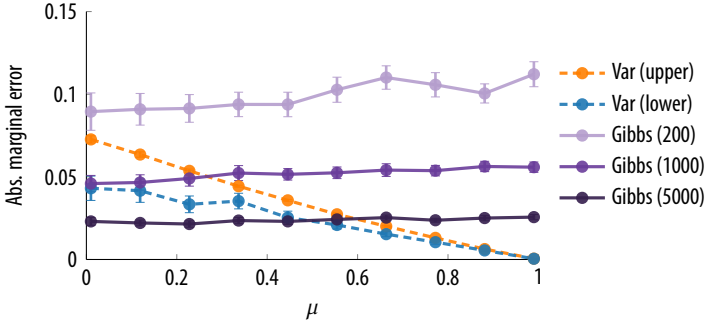
Coupling-based methods have been most prominently used for counting k-colorings in low-degree graphs (Jerrum, 1995; Bubley et al., 1998; Jerrum, 2003). Other applications of coupling include counting independent sets in graphs (Dyer & Greenhill, 2000), and approximating the partition function of various subclasses of Ising models at high temperatures (Levin et al., 2008b).



(a) Facility location



(b) Pairwise MRF



(c) Higher-order MRF

Figure 3.4: Absolute error of the marginals computed by the Gibbs sampler compared to the variational approximations by Djolonga & Krause (2014).

4 Improved Mixing using Semigradients

The majority of the content of this chapter has already been published in conference proceedings (Gotovos et al., 2018).

4.1 Introduction

The conditions derived in the previous chapter gave us some insight into the factors that determine the mixing rate of the Gibbs sampler in probabilistic submodular models. Unfortunately, oftentimes in practice these conditions do not hold, and the Gibbs sampler mixes prohibitively slowly. A fundamental reason for this slow mixing behavior is the existence of bottlenecks in the state space of the Markov chain. Conceptually, one can think about the state-space graph containing several isolated components that are poorly connected to each other, thus making it hard for the Gibbs sampler to move between them.

In this chapter, we propose a novel sampling strategy that allows for global moves in the state space, thereby avoiding bottlenecks, and, thus, accelerating mixing. Our sampler is based on using a proposal distribution that approximates the target p by a mixture of product distributions. We further propose an algorithm for constructing such a mixture using discrete semigradient information of the associated function F. This idea takes a step towards bridging optimization and sampling, a theme that has been successful in continuous spaces. Our sampler is readily combined with other existing samplers, and we show provable theoretical, as well as empirical examples of speedups.

Mixing time and spectral gap. As a reminder, the mixing time of a Markov chain $(X_t)_t$ denotes the minimum number of iterations required to get ϵ-close to stationarity, $t_{\text{mix}}(\epsilon) := \min\{t \mid d(t) \leq \epsilon\}$. The distance to stationarity, $d(t) := \max_{X_0 \in \Omega} d_{\text{TV}}\left(P^t(X_0, \cdot), p\right)$, is the maximum total variation distance, over all starting states, between X_t and the target distribution p (see Section 2.3).

A common way to obtain an upper bound on the mixing time of a chain is by lower bounding its spectral gap, defined as $\gamma := 1 - \lambda_2$, where λ_2 is the second largest eigenvalue of the corresponding transition matrix P. The following well-known theorem connects the spectral gap to mixing time.

Theorem 4.1 (cf. Theorems 12.3, 12.4 in (Levin et al., 2008b)). *Let P be the transition matrix of a lazy, irreducible, and reversible Markov chain, and let γ be its spectral gap, and $p_{min} := \min_{S \in \Omega} p(S)$. Then,*

$$\left(\frac{1}{\gamma} - 1\right) \log\left(\frac{1}{2\epsilon}\right) \leq t_{\text{mix}}(\epsilon) \leq \frac{1}{\gamma} \log\left(\frac{1}{\epsilon p_{min}}\right).$$

4.2 The Mixture Chain

Despite the simplicity and computational efficiency of the Gibbs sampler, the fact that it is constrained to performing local moves makes it susceptible to state-space bottlenecks, which hinder the movement of the chain around the state space. Intuitively, the state space may contain several high-probability regions arranged in such a way that moving from one to another using only single-element additions and deletions requires passing through states of very low probability. As a result, the Gibbs sampler may mix extremely slowly on the whole state space, despite the fact that it can move sufficiently fast within each of the high-probability regions.

To alleviate this shortcoming, it is natural to ask whether it is possible to bypass such bottlenecks by using a chain that performs larger moves. In this paper, we introduce a novel approach that uses a Metropolis chain based on a specific mixture of log-modular distributions, which we call the M^3 chain, to perform global moves in the state space. Concretely, we define a proposal distribution

$$\begin{aligned} q(S, R) = q(R) &= \frac{1}{Z_q} \sum_{i=1}^{r} \exp\left(F_i(R)\right) \\ &= \frac{1}{Z_q} \sum_{i=1}^{r} w_i \exp\left(m_i(R)\right), \end{aligned} \tag{4.1}$$

where each $F_i(R) = c_i + \sum_{v \in R} m_{iv}$ is a modular function, each $m_i(R) = \sum_{v \in R} m_{iv}$ is a normalized modular function ($m_i(\varnothing) = 0$), and $w_i = \exp(c_i) > 0$. If we denote by Z_i the normalizer of m_i, then the normalizer of the mixture

can be written in closed form as

$$\begin{aligned} Z_q = \sum_{R\in\Omega} q(R) &= \sum_{R\in\Omega}\sum_{i=1}^{r} w_i \exp\left(m_i(R)\right) \\ &= \sum_{i=1}^{r} w_i \sum_{R\in\Omega} \exp\left(m_i(R)\right) \\ &= \sum_{i=1}^{r} w_i Z_i. \end{aligned}$$

We define the M^3 chain as a Metropolis chain using q as a proposal distribution. Its transition matrix $P^{\mathrm{M}} : \Omega \times \Omega \to \mathbb{R}$ is given by

$$P^{\mathrm{M}}(S,R) = \begin{cases} q(R)p_a(S,R), & \text{if } R \neq S \\ 1 - \sum_{T\neq S} q(T)p_a(S,T), & \text{otherwise} \end{cases},$$

where

$$p_a(S,R) := \min\left\{1, \frac{p(R)q(S)}{p(S)q(R)}\right\}.$$

Note that, contrary to usual practice, the proposal q only depends on the proposed state, but not on the current state of the chain. As a result, the chain is not constrained to local moves, but rather can potentially jump to any part of the state space. In practice, M^3 sampling proceeds in two steps: first, a candidate set R is sampled according to q; then, the move to R is accepted with probability p_a. Sampling from q can be done in $O(n)$ time—first, sample a log-modular component, then sample a set from that component. Computing p_a requires $O(r)$ time for the sum in (4.1), and it can be straightforwardly improved by parallelizing this computation. All in all, the total time for one step of M^3 is $O(n + r)$.

As is always the case with Metropolis chains, the mixing time of the M^3 sampler will depend on how well the proposal q approximates the target distribution p. The following observation shows that, in theory, we can approximate any distribution of the form $p(S) \propto \exp(F(S))$ by a mixture of the form (4.1).

Proposition 4.2. *For any distribution $p(S) \propto \exp(F(S))$ on Ω, and any $\epsilon > 0$, there are positive constants $w_i = w_i(\epsilon) > 0$, and normalized modular functions $m_i = m_i(\epsilon)$, such that, if we define $q(S) := \sum_{i=1}^{r} w_i \exp(m_i(S))$, then $d_{TV}(p,q) \leq \epsilon$.*

Proof. Let $r = |\Omega|$, and let $(S_i)_{i=1}^r$ be an enumeration of all sets in Ω. For any $i \in \{1, \dots, r\}$, and any $v \in V$, we define

$$m_{iv} = \begin{cases} \beta_i\,, & \text{if } v \in S_i \\ -\beta_i\,, & \text{otherwise} \end{cases},$$

and $m_i(S) = \sum_{v \in S} m_{iv}$, for all $S \in \Omega$. We also define

$$w_i = \frac{p(S_i)}{Z_i} = \frac{p(S_i)}{\left(1 + e^{\beta_i}\right)^{|S_i|} \left(1 + e^{-\beta_i}\right)^{|V \setminus S_i|}}.$$

Then, for all $i \in \{1, \dots, r\}$, we have

$$\begin{aligned} d_i(\beta_1, \dots, \beta_r) &:= |p(S_i) - q(S_i)| \\ &= \left| p(S_i) - \sum_{j=1}^{r} p(S_j) \frac{e^{\beta_j |S_j|}}{\left(1 + e^{\beta_j |S_j|}\right)\left(1 + e^{-\beta_j |V \setminus S_j|}\right)} \right| \\ &\leq p(S_i) \left(1 - \frac{e^{\beta_i |S_i|}}{\left(1 + e^{\beta_i |S_i|}\right)\left(1 + e^{-\beta_i |V \setminus S_i|}\right)}\right) + \\ &\quad \sum_{j : S_j \neq S_i} p(S_j) \frac{e^{\beta_j |S_i|}}{\left(1 + e^{\beta_j |S_j|}\right)\left(1 + e^{-\beta_j |V \setminus S_j|}\right)}. \end{aligned}$$

Note that both terms vanish if we let all $\beta_j \to \infty$. Therefore, for any $\delta > 0$, there are $\beta_{ij} = \beta_{ij}(\delta)$, for all $j \in \{1, \dots, r\}$, such that $d_i(\beta_{i1}, \dots, \beta_{ir}) \leq \delta$. Finally, choosing $\hat{\beta}_j := \max_{i \in \{1, \dots, r\}} \beta_{ij}$, for all $j \in \{1, \dots, r\}$, we get

$$d_{\mathrm{TV}}(p, q) = \frac{1}{2} \sum_{i=0}^{r} d_i(\hat{\beta}_1, \dots, \hat{\beta}_r) \leq 2^{n-1} \delta.$$

The result follows by choosing $\delta = \epsilon / 2^{n-1}$. □

Conceptually, the proof relies on having one log-modular term per set in Ω. Therefore, while the above result shows that mixtures of log-modulars are expressive enough, the constructed mixture of exponential size in n is not useful for practical purposes. On the other hand, it is not necessary for us to have q be an accurate approximation of p everywhere, as long as the corresponding M^3 chain is able to bypass state-space bottlenecks. With this in mind, we suggest combining the M^3 and Gibbs chains, so that each of them serve complementary purposes in the final chain; the role of M^3 is to make global moves and avoid bottlenecks, while the role of Gibbs is to move fast

within well-connected regions of the state space. To make this happen, we define the transition matrix $P^{\mathrm{C}} : \Omega \times \Omega \to \mathbb{R}$ of the combined chain as

$$P^{\mathrm{C}}(S, R) = \delta P^{\mathrm{G}}(S, R) + (1 - \delta) P^{\mathrm{M}}(S, R), \tag{4.2}$$

where $0 < \delta < 1$. It is easy to see that P^{C} is reversible, and has stationary distribution p.

We next illustrate how combining the two chains works on an example model class, in which a mixture of only a few log-modular distributions can dramatically improve the mixing time compared to running the vanilla Gibbs chain.

4.3 Ising Model on the Complete Graph

We consider the Ising model on a finite complete graph (Levin et al., 2008a), also known as the Curie-Weiss model in statistical physics, which is family of log-supermodular distributions that can be written as follows,

$$p(S) = \frac{1}{Z(\beta)} \exp\left(-\frac{2\beta}{n}|S|(n - |S|)\right). \tag{Ising$_\beta$}$$

In particular, we focus on the case where $\beta = \ln(n)$, that is,

$$p(S) = \frac{1}{Z} \exp\left(-\frac{2\ln(n)}{n}|S|(n - |S|)\right). \tag{Ising}$$

In this case, if we define $d_n := 2\ln(n)/n$, then $F(S) = -d_n|S|(n - |S|)$.

The Gibbs sampler is known to experience poor mixing in this model; the following is an immediate corollary of Theorem 15.3 in (Levin et al., 2008b).

Corollary 4.3. *For $n \geq 3$, the Gibbs sampler on Ising has spectral gap $\gamma^{\mathrm{G}} = O\left(e^{-cn}\right)$, where $c > 0$ is a constant.*

From Theorem 4.1 it follows that the mixing time of Gibbs is

$$t_{\mathrm{mix}}(\epsilon) = \Omega\left((e^{cn} - 1)\log\left(\frac{1}{2\epsilon}\right)\right).$$

Yet, it has been shown that the only reason for this is a single bottleneck in the state space (Levin et al., 2008a). To make this statement more formal, let us define a decomposition of Ω into two disjoint sets (Jerrum et al., 2004b),

$$\Omega_0 := \{S \in \Omega \mid |S| < n/2\},$$

$$\Omega_1 := \{S \in \Omega \mid |S| > n/2\}.$$

To keep things simple, we will assume for the remainder of this section that n is odd; the analysis when n is even follows from the same arguments with only a minor technical adjustment.

The projection and restriction chains. Our goal is to separately examine two characteristics of the sampler: (i) its movement between the two sets Ω_0, Ω_1, and (ii) its movement when restricted to stay within each of these sets. For analyzing the "between-sets" behavior, we define the projection $\bar{p} : \{0,1\} \to \mathbb{R}$ of p as

$$\bar{p}(i) := \sum_{S \in \Omega_i} p(S),$$

and, for any reversible chain P, we define its projection chain $\bar{P} : \{0,1\} \times \{0,1\} \to \mathbb{R}$ as

$$\bar{P}(i,j) := \frac{1}{\bar{p}(i)} \sum_{S \in \Omega_i, R \in \Omega_j} p(S)P(S,R).$$

It is easy to see that $\bar{P}$ is also reversible and has stationary distribution $\bar{p}$. For analyzing the "within-set" behavior, we define the restrictions $p_i : \Omega_i \to \mathbb{R}$ of p as

$$p_i(S) := \frac{p_i(S)}{\bar{p}(i)},$$

and the two restriction chains $P_i : \Omega_i \times \Omega_i \to \mathbb{R}$ of P as

$$P_i(S,R) := \begin{cases} P(S,R), & \text{if } S \neq R \\ 1 - \sum_{T \in \Omega_i : T \neq S} P(S,T), & \text{otherwise} \end{cases}.$$

Again, it is easy to see that each of the P_i is also reversible and has stationary distribution p_i.

In Figure 4.1, we depict the structure of our reasoning for the rest of this section, including the results that we prove or use to ultimately arrive at the upcoming Theorem 4.6.

Gibbs restrictions. Coming back to the Gibbs sampler, if we could show that it mixes fast within each of Ω_0 and Ω_1, then we could deduce that the only reason for the slow mixing on Ω is the bottleneck between these two sets. Indeed, the following corollary of a theorem by Ding et al. (2009) shows exactly that.

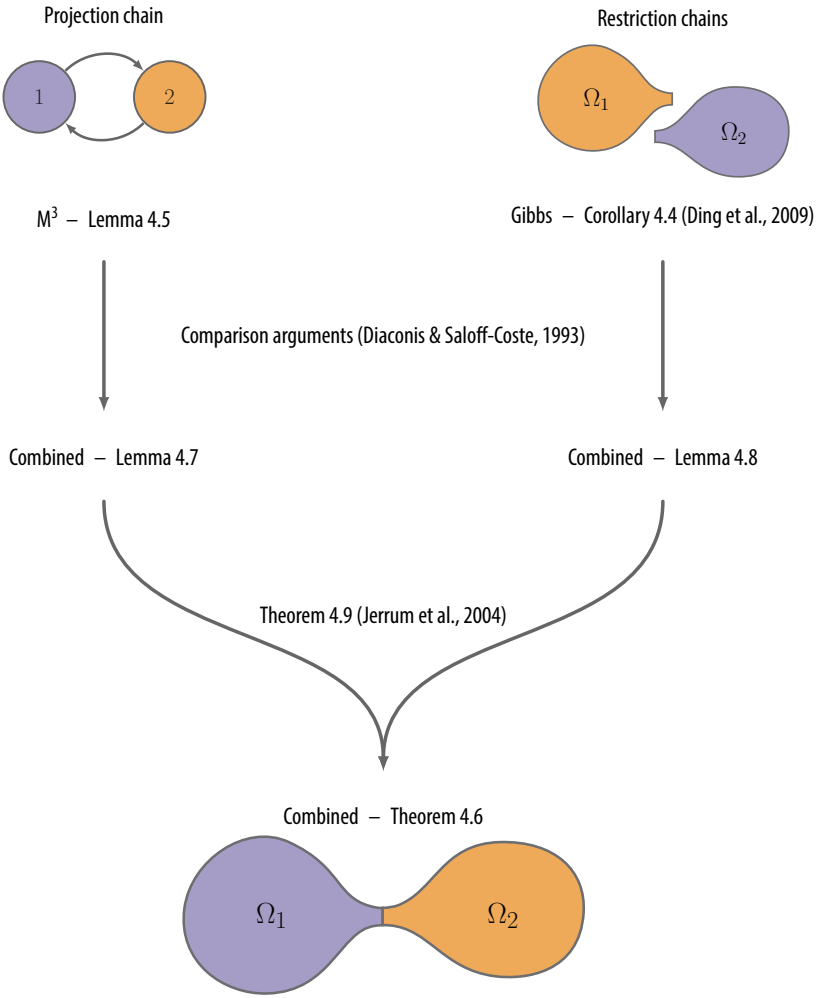


Figure 4.1: The structure of our reasoning to prove Theorem 4.6.

Corollary 4.4 (cf. Theorem 2, Ding et al., 2009). *For all $n \geq 3$, the restriction chains of the Gibbs sampler P_i^{G}, $i = 0, 1$, on Ising have spectral gap $\gamma_i^{\mathrm{G}} = \Theta\left(\frac{2\ln(n)-1}{n}\right)$.*

M^3 projection. To improve mixing we want to create an M^3 chain that is able to bypass the aforementioned bottleneck. For this purpose, we use a mixture of two log-modular distributions, the first of which puts most of its mass on Ω_0, and the second on Ω_1. We define the mixture of the form (4.1) by

$$m_1(S) = \sum_{v \in S} -d_n(n-1) = -d_n(n-1)|S|,$$
$$m_2(S) = \sum_{v \in S} d_n(n-1) = d_n(n-1)|S|.$$

We also use $w_1 = 1/Z_1$ and $w_2 = 1/Z_2$, where Z_1 and Z_2 are the normalizers of m_1 and m_2 respectively. The resulting proposal distribution can be written as follows,

$$q(S) = \frac{1}{2}\left(\frac{\exp(-d_n(n-1)|S|)}{Z_1} + \frac{\exp(d_n(n-1)|S|)}{Z_2}\right), \tag{4.3}$$

where $Z_1 = (1 + \exp(-d_n(n-1)))^n$, and $Z_2 = (1 + \exp(d_n(n-1)))^n$. It follows that $Z_q = 1/2$, and, furthermore, the mixture q is symmetric, that is, $q(S) = q(V \setminus S)$.

Since the proposal q is symmetric and state independent, we would expect the M^3 chain to jump between Ω_0 and Ω_1 without being hindered by the bottleneck described previously. We verify this intuition by proving the following lemma.

Lemma 4.5. *For all $n \geq 10$, the projection chain $\bar{P}^{\mathrm{M}}$ of the M^3 sampler on Ising has spectral gap $\bar{\gamma}^{\mathrm{M}} = \Omega(1)$.*

Proof. We define $p_k = \sum_{S \in \Omega, |S|=k} p(S)$, and $q_k = \sum_{S \in \Omega, |S|=k} q(S)$. We then proceed to upper bound p_k, and lower bound q_k.

Bounding p_k. By definition, we can write $p_k = \hat{p}_k / Z$, where $\hat{p}_0 = 1$, and for $k > 0$ we have

$$\hat{p}_k := \binom{n}{k} \exp\left(-\frac{2\ln(n)}{n} k(n-k)\right)$$
$$= \binom{n}{k} n^{-\frac{2k}{n}(n-k)}$$

$$\leq \left(\frac{en}{k}\right)^k n^{-\frac{2k}{n}(n-k)}$$
$$= \left(\frac{e}{k}\right)^k n^{-k+\frac{2k^2}{n}}.$$

It follows that

$$\ln(\hat{p}_k) \leq -k \ln\left(\frac{k}{e}\right) + \left(\frac{2k^2}{n} - k\right) \ln(n). \tag{4.4}$$

It is easy to verify that for all $n \geq 10$ and $3 \leq k \leq \lfloor n/2 \rfloor$, it holds that $(2k - n)\ln(n) \leq 0.5n \ln(k/e)$. Substituting this into (4.4), we get

$$\ln(\hat{p}_k) \leq -0.5k \ln\left(\frac{k}{e}\right)$$
$$\Rightarrow \quad \hat{p}_k \leq \exp(-0.5k \ln(k/e)).$$

Noting that, for all k, $\hat{p}_k \leq 1$, and using the fact that $\hat{p}_{n-k} = \hat{p}_k$, we get

$$\begin{aligned} Z &= \sum_{k=0}^{n} \hat{p}_k \\ &\leq 2 \sum_{k=0}^{\lfloor n/2 \rfloor} \hat{p}_k \\ &= 2(\hat{p}_0 + \hat{p}_1 + \hat{p}_2 + \sum_{k=3}^{\lfloor n/2 \rfloor} \hat{p}_k) \\ &\leq 3 + \sum_{k=3}^{\lfloor n/2 \rfloor} \exp(-0.5k \ln(k/e)) \\ &\leq c_1, \end{aligned} \tag{4.5}$$

where c_1 is a constant.

Bounding q_k. First, it is easy to see that, for all $n \geq 1$, $Z_1 \leq 3$.

$$\begin{aligned} q_k &= \sum_{S \in \Omega, |S|=k} q(S) \\ &\geq \sum_{S \in \Omega, |S|=k} \frac{1}{2} \frac{\exp(-d_n(n-1)|S|)}{Z_1} \qquad \text{(by (4.3))} \\ &\geq \frac{1}{6} \binom{n}{k} \exp(-d_n(n-1)|S|) \end{aligned}$$

Bounding the spectral gap. For the projection chain $\bar{P}^{\mathrm{M}}$, we have

$$\begin{aligned}
\bar{P}^{\mathrm{M}}(0,1) &= \frac{1}{\bar{p}(0)} \sum_{\substack{S \in \Omega_i \\ R \in \Omega_j}} p(S) P^{\mathrm{M}}(S,R) \\
&\geq 2 p_0 q_n && (\bar{p}(0) = 1/2 \text{ by symmetry of } p) \\
&= 2 p_0 q_0 && (\text{by symmetry of } q) \\
&\geq 2 \frac{\hat{p}_0}{Z} \frac{1}{6} && (q_0 \geq \tfrac{1}{6}) \\
&\geq 2 \frac{1}{c_1} \frac{1}{6} && (\hat{p}_0 = 1) \\
&= c \bar{p}(1),
\end{aligned}$$

where $c = (2/3)c_1$.

Finally, it is easy to show that the spectral gap of any reversible two-state chain P with stationary distribution p that satisfies $P(0,1) = c\,p(1)$ is c; for example, see Fact 6 by Anari et al. (2016). Applying this to the above projection chain shows that the spectral gap of $\bar{P}^{\mathrm{M}}$ is c. □

Combining the chains. Putting everything together, we show the following result about the combined chain P^{C}.

Theorem 4.6. *For all $n \geq 10$, the combined chain P^{C} on Ising has spectral gap*

$$\gamma^{\mathrm{C}} = \Omega\left(\frac{2\ln(n) - 1}{2n}\right).$$

The proof consists of two steps. In the first step, we make two comparison arguments (Diaconis & Saloff-Coste, 1993; Levin et al., 2008b) to show that the spectral gaps of the projection and restriction chains of the combined sampler are smaller by at most a constant factor in δ compared to those of Gibbs and M^3. In particular, we use the M^3 bound (Lemma 4.5) for the projection chain, and the Gibbs bound (Corollary 4.4) for the restriction chains. The following two lemmas make this more concrete.

Lemma 4.7. *For all $n \geq 10$, the projection chain $\bar{P}^{\mathrm{C}}$ of the combined chain on Ising has spectral gap $\bar{\gamma}^{\mathrm{C}} = \Omega(1)$.*

Proof. By definition, $\bar{P}^{\mathrm{C}}(S,R) \geq \delta \bar{P}^{\mathrm{M}}(S,R)$, therefore a simple comparison argument (e.g., Lemma 13.22 in (Levin et al., 2008b)) combined with the result of Lemma 4.5 gives us $\bar{\gamma}^{\mathrm{C}} \geq \delta \bar{\gamma}^{\mathrm{M}} = \Omega(1)$. □

Lemma 4.8. *For all $n \geq 3$, each of the restriction chains P_i^{C} of the combined chain on* Ising *has spectral gap $\gamma_i^{\mathrm{C}} = \Theta\left(\frac{2\ln(n)-1}{2n}\right)$.*

Proof. By definition, $P_i^{\mathrm{C}}(S,R) \geq \delta P_i^{\mathrm{G}}(S,R)$, therefore, using a comparison argument like above together with Corollary 4.4 gives us

$$\gamma_i^{\mathrm{C}} \geq \delta\gamma_i^{\mathrm{G}} = \Theta\left(\frac{2\ln(n)-1}{2n}\right).$$

□

The second step combines the projection and restriction bounds to establish a bound on the spectral gap of the combined chain. To accomplish this we use the following result by Jerrum et al. (2004b), which states that the spectral gap of the whole chain cannot be much smaller than the smallest of the projection and restriction spectral gaps.

Theorem 4.9 (Theorem 1, Jerrum et al., 2004b). *Given a reversible Markov chain P, if the spectral gap of its projection chain $\bar{P}$ is bounded below by $\bar{\gamma}$, and the spectral gaps of its restriction chains P_i are uniformly bounded below by γ_{min}, then the spectral gap of P is bounded below by*

$$\gamma = \min\left\{\frac{\bar{\gamma}}{3}, \frac{\bar{\gamma}\gamma_{min}}{3P_{max}+\bar{\gamma}}\right\},$$

where $p_{max} := \max_{i\in\{0,1\}} \max_{S\in\Omega_i} \sum_{R\in\Omega\setminus\Omega_i} P(S,R)$.

The result of Theorem 4.6 follows directly by combining the spectral gap bounds of Lemmas 4.7 and 4.8 in Theorem 4.9, and noting that $P_{\max} \leq 1$.

Finally, using Theorem 4.1, and noting that, in this case, $p_{\min} = O(e^{-n})$ (cf. proof of Lemma 4.5), we get a mixing time of $t_{\mathrm{mix}}(\epsilon) = O(n^2 \log(1/\epsilon))$ for the combined chain. This shows that the addition of the M^3 sampler results in an exponential improvement in mixing time over the Gibbs sampler by itself.

4.4 Constructing the Mixture

Having seen the positive effect of the M^3 sampler, we now turn to the issue of how to choose the proposal q. While a manual construction like the one we

Algorithm 4.1: Iterative semigradient-based mixture construction

Input: Set function F, mixture size r

for $i = 1$ **to** r **do**
  $\sigma \leftarrow \textsc{GreedyDifMax}(F, \{m_1, \ldots, m_{i-1}\})$
  $m_i \leftarrow \textsc{SemiGradient}(F, \sigma)$
return $\{m_1, \ldots, m_r\}$

just presented for the Ising model may be feasible in some cases, it is often more practical to have an automated way of obtaining the mixture.

Let us assume, as is usually the case, that we have access to a function oracle for F, and we want to create a mixture of size r. Ideally, we would like to construct a proposal q that is as close to p as possible, that is, minimize an objective such as the following,

$$\begin{aligned} E_1(q) &:= \min_q \|p - q\| \\ &= \min_q \left\| \frac{\exp(F(\cdot))}{Z} - \frac{1}{Z_q} \sum_{i=1}^{r} w_i \exp(m_i(\cdot)) \right\|, \end{aligned}$$

where $\|\cdot\|$ could be, for example, total variation distance or the maximum norm. Unfortunately, this problem is hard; both computing the partition function Z, and jointly optimizing over all w_i, m_i are infeasible in practice. To make the problem easier, we could try to get rid of the normalizers and weights w_i, and iteratively minimize over each m_i individually:

$$E_2^{(i)}(m_i) := \min_{m_i} \left\| \exp(F(\cdot)) - \sum_{j=1}^{i-1} \exp(m_i(\cdot)) \right\|,$$

for $i \in \{1, \ldots, r\}$. This problem is still hard, since optimizing $\|\exp(F(\cdot))\|$ is by itself infeasible in general.

To arrive at a practical algorithm, we approximate the above objective using the two-step procedure described in Algorithm 4.1. In the first step, we generate a permutation σ of the ground set V by running the greedy algorithm on function $D_i(S) := F(S) - \log \sum_{j=1}^{i-1} \exp(m_j(S))$, as shown in Algorithm 4.2 (cf. Algorithm 2.1). Intuitively, the sets that are formed by elements near the beginning of σ are those on which F and the current mixture disagree by the most. Therefore, in the second step, we would like to

Algorithm 4.2: Greedy difference maximization

Input: Set function F, modular functions $\{m_1, \ldots, m_{i-1}\}$

$D_i(S) \leftarrow F(S) - \log \sum_{j=1}^{i-1} \exp(m_j(S))$, for all $S \in \Omega$
$\sigma \leftarrow (1, \ldots, n)$
$A \leftarrow \varnothing$
for $i = 1$ **to** n **do**
  $v^* \leftarrow \operatorname{argmax}_{v \in V \setminus A} (D_i(A \cup \{v\}) - D_i(A))$
  $\sigma_i \leftarrow v^*$
  $A \leftarrow A \cup \{v^*\}$
return σ

add to the mixture a modular function m_i that is a good approximation for F on $\{\sigma_1, \ldots, \sigma_k\}$, for a choice of $1 \leq k \leq n$. To accomplish this, we propose using discrete *semigradients*.

Semigradients are modular functions that provide lower (subgradient) or upper (supergradient) approximations of a set function F (Fujishige, 2005; Iyer et al., 2013). More concretely, given a set $S \in \Omega$, a modular function m is a subgradient of F at S, if, for all $R \in \Omega$, $F(R) \geq F(S) + m(R) - m(S)$. Similarly, m is a supergradient if the inequality is reversed. Although, in general, a function is not guaranteed to have sub- or supergradients at each $S \in \Omega$, it has been shown that this is true when F is submodular or supermodular (Fujishige, 2005; Jegelka & Bilmes, 2011; Iyer & Bilmes, 2012).

Coming back to the second step of Algorithm 4.1, to create a subgradient of F given permutation σ we just need to define a modular function via marginal gains according to the permutation order (Iyer et al., 2013), as shown in Algorithm 4.3. Moreover, this is a subgradient of F at $\{\sigma_1, \ldots, \sigma_k\}$, for all $1 \leq k \leq n$. On the other hand, Algorithm 4.4 creates a supergradient of F at $\{\sigma_1, \ldots, \sigma_k\}$ for a randomly chosen k. (This type of supergradient is denoted by $\bar{g}_Y$ by Iyer et al. (2013).) In fact, the modular functions m_1, m_2 that we used in analyzing the Ising model in the previous section were supergradients of F at sets $S_1 = \varnothing$, and $S_2 = V$ respectively.

In practice, we can use Algorithm 4.1 regardless of whether F is sub- or supermodular. We have noticed that subgradients give better results when F is submodular, and vice versa for supergradients and supermodular F.

Algorithm 4.3: Subgradient computation

Input: Set function F, permutation σ

$A \leftarrow \emptyset$
$c \leftarrow F(\emptyset)$
for $v = 1$ **to** n **do**
  $m_v \leftarrow F(A \cup \{\sigma_v\}) - F(A)$
  $A \leftarrow A \cup \sigma_v$
return $m(S) := c + \sum_{v \in S} m_v$

Algorithm 4.4: Supergradient computation

Input: Set function F, permutation σ

Draw $k \sim \text{UNIF}(\{1, \ldots, n\})$
for $v = 1$ **to** k **do**
  $m_v \leftarrow F(V) - F(V \setminus \{v\})$
for $v = k + 1$ **to** n **do**
  $m_v \leftarrow F(\{v\})$
return $m(S) := \sum_{v \in S} m_v$

4.5 Experiments

We start by repeating the experiment of the previous chapter shown in Figure 3.3, which involved estimating the log-partition function using reverse importance sampling on a synthetic data set that contains a group of three mutually exclusive genes. Here we only focus on the ground set of size $|V| = 100$. Figure 4.2 shows the resulting (approximate) error in estimating $\log(Z)$ using the Gibbs sampler, compared to our proposed combined sampler using a mixture q constructed by Algorithm 4.1 (Combo-I). We also compare against a variation where we substitute the greedy procedure in line 2 of Algorithm 4.1 with picking a permutation σ of the ground set uniformly at random (Combo-R). Both variations use $r = 20$ subgradients, and we

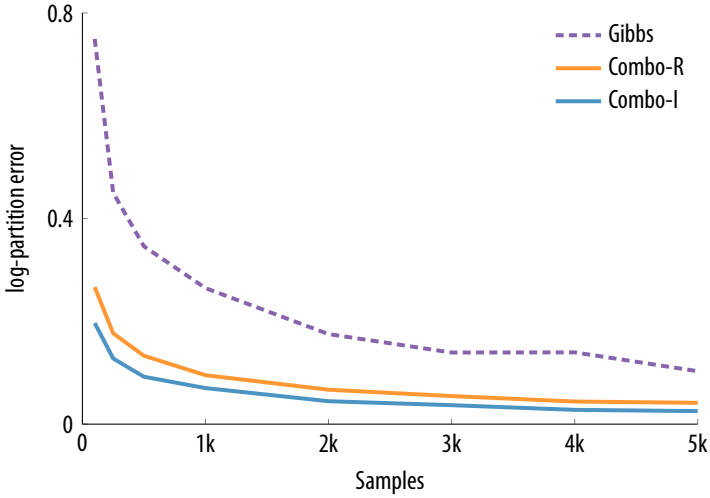


Figure 4.2: The error in estimating the log-partition function with the two versions of the combined sampler compared to the Gibbs sampler.

repeat the experiment 100 times. We can see that they clearly outperform the Gibbs sampler, while the difference between the two variations is not as significant.

Next we evaluate the marginal inference performance of our proposed sampler on the Ising model we analyzed earlier, as well as the following three models learned from real-world data sets.

Water. The same FLiD model that we used in the experimental section of last chapter (see Figure 3.4a), which was based on a problem of sensor placement in a water distribution network (Krause et al., 2008) In this case, we randomly subsample the original facility location matrix, so that $n = 50$, and $L = 500$.

Sensor. A determinantal point process (see Example 2.12), which was used in a problem of sensor placement for indoor temperature monitoring (Guestrin et al., 2005). The function F is of the form

$$F(S) = \log |K + \sigma^2 I|,$$

where K is a kernel matrix, and σ is a noise parameter. The size of the ground set is $n = 46$.

Game. A FLiD model that represents the characters that are chosen by players in the popular online game "Heroes of the Storm". We learned the model from an online data set of approximately $8,000$ teams of 5 characters[1] using noise-contrastive estimation, as described by Tschiatschek et al. (2016). The model has a ground set of size $n = 48$, and $L = 10$ latent dimensions.

In practice, we would only be interested in sampling sets of fixed size $\ell = 5$. The Gibbs sampler can be easily modified to sample under a cardinality constraint by using moves that swap an element in the current set X_t with an element in $V \setminus X_t$. Extending the M^3 chain to sample from cardinality-constrained models is also straightforward. In fact, the only additional ingredient required is a procedure to sample a set of size ℓ from a log-modular distribution, which can be easily done, as before, in $O(n)$ time.

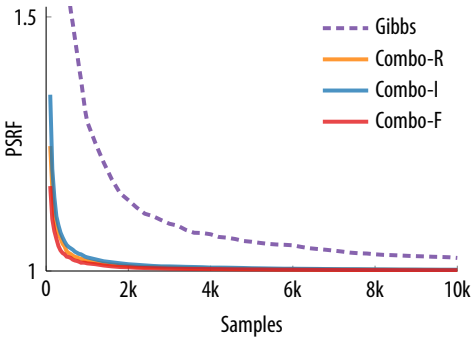
Results. To assess convergence, we use the potential scale reduction factor (PSRF) (Brooks et al., 2011) with 20 parallel chains. Intuitively, the PSRF compares the within-chain variance of some probabilistic quantity to the between-chain variance of that same quantity. As each of the chains converges to the stationary distribution, the PSRF is expected to converge to 0. In our experiments, we compute the PSRF using single-element marginal probabilities, and show the worst (highest PSRF) marginal averaged over 50 repetitions of each simulation.

In Figure 4.3 we show the results for the Ising model ($n = 6, 7, 8$). The additional Combo-F lines denote the combined sampler with two mixture components discussed in Section 4.3. The other two combined samplers use mixtures of size $r = 20$. Note that Gibbs mixes dramatically slower than the combined sampler, even for such small n, because of the significant bottleneck we described before.

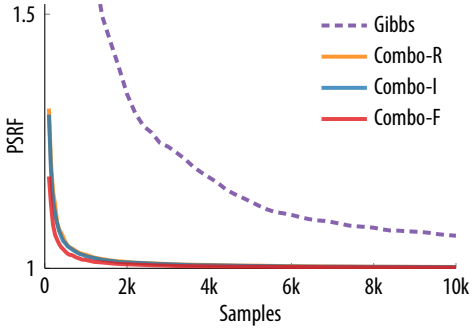
In Figure 4.4 we show the results on the three log-submodular models above using mixtures of size $r = 200$. We see again that even random permutations are enough to provide a significant improvement over the performance of Gibbs. Similar observations can be made with respect to computation time (see Figure A.1 in the appendix).

In Figure 4.5a we show how mixture size affects performance; as expected, adding more components to the mixture results in a proposal that approximates the target distribution better, and, therefore, mixes faster. Finally, in Figure 4.5b we illustrate the effect of the combination weight δ, while having the number of subgradients fixed to $r = 200$. We see that both Gibbs ($\delta = 1$) and M^3 ($\delta = 0$) perform poorly by themselves, but combining

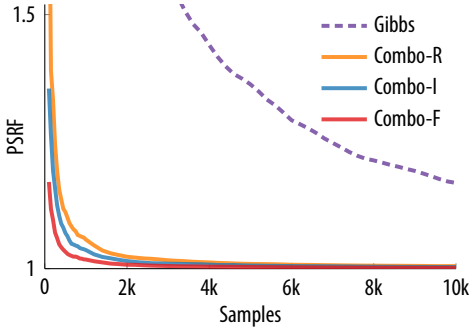
[1] https://www.hotslogs.com



(a) Ising ($n = 6$)

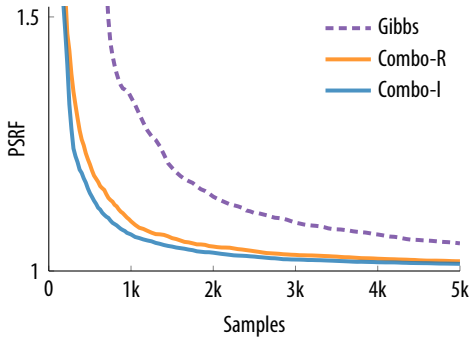


(b) Ising ($n = 7$)

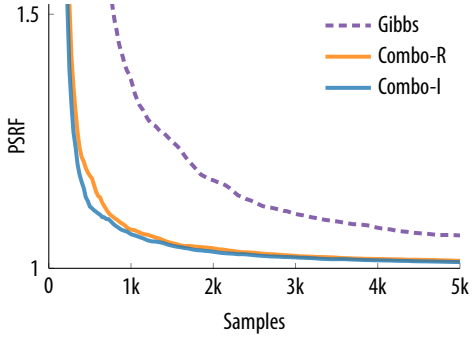


(c) Ising ($n = 8$)

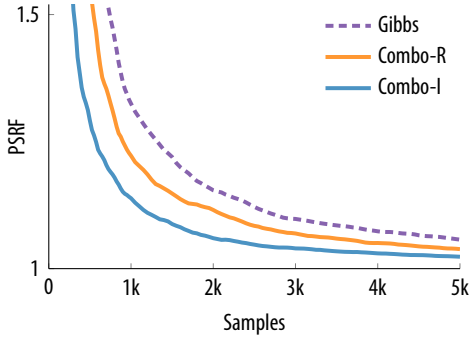
Figure 4.3: Ising model results for increasing n. Note how the previously discussed bottleneck significantly affects the Gibbs sampler's performance, while it has almost no effect on the combined chains.



(a) Water



(b) Sensor



(c) Game

Figure 4.4: Potential scale reduction factor (PSRF) as a function of sampling iterations. The combined chains have a clear advantage over Gibbs on all three models.

them results in much improved performance. This highlights again the complementary nature of the two chains (local vs. global moves) we discussed earlier.

4.6 Conclusion

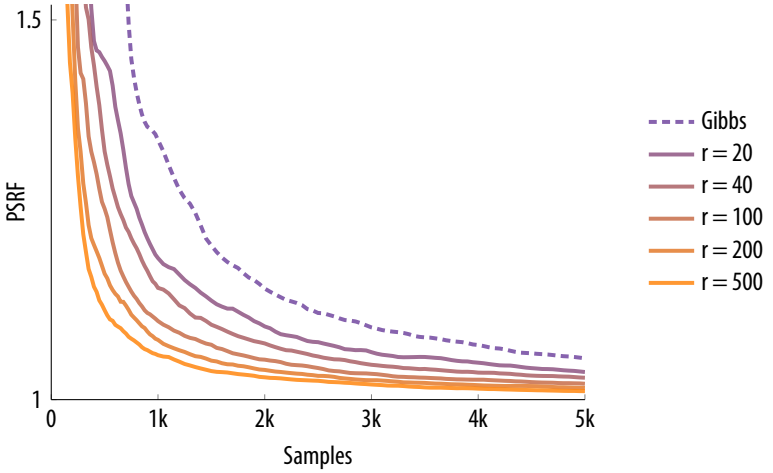
In this chapter, we presented the M^3 sampler that proposes global moves using a mixture of log-modular distributions. We theoretically analyzed the effect of combining our sampler with the Gibbs sampler on a class of Ising models, and proved an exponential improvement in mixing time. We also demonstrated notable improvements when combining the two samplers on three models of practical interest.

Further related work. There has been some recent work on mapping discrete inference to continuous domains (Zhang et al., 2012; Pakman & Paninski, 2013; Dinh et al., 2017; Nishimura et al., 2018) to enable the use of well-established continuous samplers, such as Hamiltonian Monte Carlo (Neal, 2012; Betancourt, 2017). It is worth pointing out that, while these methods usually outperform simple Gibbs or Metropolis samplers, they still tend to suffer from considerable slowdowns in multimodal distributions (Neal, 2012). Our work in this chapter is orthogonal to these methods, in the sense that our proposed sampler can be combined with any of the existing ones to provide a principled way for performing global moves that can lead to improved mixing.

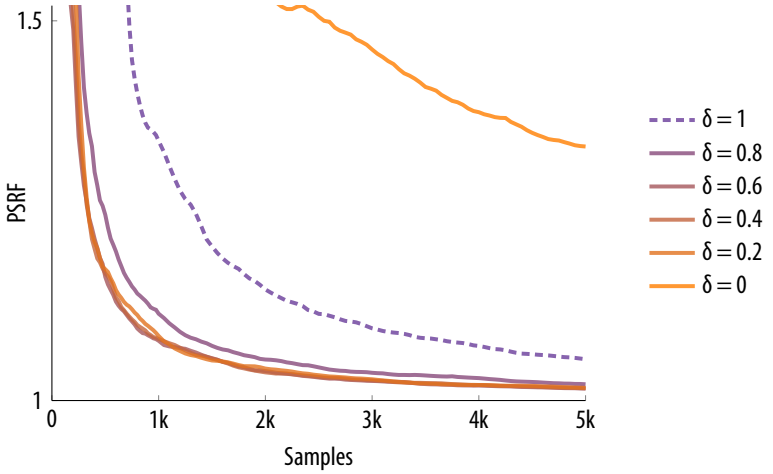
Both darting Monte Carlo (Sminchisescu & Welling, 2007; Ahn et al., 2013), and variational MCMC (de Freitas et al., 2001) share the high-level concept of combining two chains, one making global moves between high-probability regions, and another making local moves around those regions. However, their proposed global samplers for continuous spaces are generally not applicable to the class of discrete distributions we consider.

Other (non-MCMC) approaches to discrete sampling include Perturb-and-MAP (Papandreou & Yuille, 2011; Hazan et al., 2013), and random projections (Zhu & Ermon, 2015).

Semigradients of submodular set functions have recently been exploited for optimization (Iyer et al., 2013; Jegelka & Bilmes, 2011), and variational inference (Djolonga et al., 2016a), but, to our knowledge, no prior work has used them for sampling.



(a) Water – number of subgradients



(b) Water – combination weight

Figure 4.5: (a) Increasing the number of mixture components improves performance. (b) The combination of Gibbs and M^3 performs better than either of them does individually.

5 Learning Prob. Submodular Models

5.1 Introduction

As discussed before, learning probabilistic models from data is one of the main motivations of our work. The probabilistic framework we have considered throughout the thesis suggests a principled way to estimate model parameters given a data set, namely by maximizing the likelihood of those parameters under the given data.

Evaluating the likelihood or computing its gradient with respect to the model parameters boils down performing inference, in particular, computing expectations over the model distribution. This brings us back to the familiar setting of the previous chapters. We show how we can use the sampling procedures discussed before to approximate the likelihood gradients, and, thus, perform an approximate gradient ascent procedure. Unfortunately, the likelihood functions of probabilistic submodular models are generally non-convex, therefore the optimization is only guaranteed to find a local optimum of the likelihood.

The rest of this chapter is then focused on applying this learning procedure to the application of modeling the interactions between gene mutations in cancer patients that we discussed in the introduction of the thesis. We evaluate our proposed method on synthetic and real cancer data, visualize the results in several ways, and compare them to the state of the art.

5.2 Approximate Maximum Likelihood Learning

We reintroduce here the notation of explicitly stating the model parameters, and thus denote our distribution of interest by

$$p(S;\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\left(F(S;\boldsymbol{\theta})\right),$$

where $\boldsymbol{\theta} \in \mathbb{R}^d$ is a parameter vector to be learned. Given a data set of N sets, $\mathcal{D} := \{D_1, \ldots, D_N\}$, with $D_1, \ldots, D_N \subseteq V$, the log-likelihood of the above model can be written as

$$\begin{aligned}\ell(\boldsymbol{\theta}) &:= \sum_{i=1}^{N} \log p(D_i; \boldsymbol{\theta}) \\ &= \sum_{i=1}^{N} \left(F(D_i; \boldsymbol{\theta}) - \log Z(\boldsymbol{\theta})\right) \\ &= \sum_{i=1}^{N} F(D_i; \boldsymbol{\theta}) - N \log Z(\boldsymbol{\theta}).\end{aligned}$$

The gradient of the log-likelihood with respect to the parameters $\boldsymbol{\theta}$ is then

$$\begin{aligned}\boldsymbol{g}(\boldsymbol{\theta}) &:= \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}) \\ &= \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}} F(D_i; \boldsymbol{\theta}) - N \nabla_{\boldsymbol{\theta}} \log Z(\boldsymbol{\theta}) \\ &= \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}} F(D_i; \boldsymbol{\theta}) - N \frac{1}{Z(\boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} Z(\boldsymbol{\theta}) \\ &= \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}} F(D_i; \boldsymbol{\theta}) - N \frac{1}{Z(\boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} \sum_{S \subseteq V} \exp\left(F(S; \boldsymbol{\theta})\right) \\ &= \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}} F(D_i; \boldsymbol{\theta}) - N \sum_{S \subseteq V} \frac{\exp\left(F(S; \boldsymbol{\theta})\right)}{Z(\boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} F(S; \boldsymbol{\theta}) \\ &= \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}} F(D_i; \boldsymbol{\theta}) - N \sum_{S \subseteq V} p(S; \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} F(S; \boldsymbol{\theta}) \\ &= \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}} F(D_i; \boldsymbol{\theta}) - N \, \mathbb{E}_p \left[\nabla_{\boldsymbol{\theta}} F(S; \boldsymbol{\theta})\right] \\ &= \frac{1}{N} \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}} F(D_i; \boldsymbol{\theta}) - \mathbb{E}_p \left[\nabla_{\boldsymbol{\theta}} F(S; \boldsymbol{\theta})\right].\end{aligned}$$

This shows that the maximum likelihood parameters (when $\boldsymbol{g}(\boldsymbol{\theta}) = 0$) satisfy a generalized version of the well-known moment matching condition for exponential family models (Wainwright & Jordan, 2008; Koller & Friedman,

Algorithm 5.1: Approximate maximum likelihood maximization

Input: Data $\mathcal{D}$, iterations n_{iter}, samples M, step $(\gamma_i)_i$, grad. oracle $\nabla_{\boldsymbol{\theta}} F(S; \boldsymbol{\theta})$

Initialize θ
for $i = 1$ **to** n_{iter} **do**
  $\mathcal{S} \leftarrow$ sample M sets from $p(\cdot\,; \theta)$
  $\widetilde{\boldsymbol{g}}(\boldsymbol{\theta}) \leftarrow \frac{1}{N} \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}} F(D_i; \boldsymbol{\theta}) - \frac{1}{M} \sum_{i=1}^{M} \nabla_{\boldsymbol{\theta}} F(S_i; \boldsymbol{\theta})$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \gamma_i \, \widetilde{\boldsymbol{g}}(\boldsymbol{\theta})$
end for
return $\boldsymbol{\theta}$

2009). That is, at the maximum, the empirical mean of the function gradient over the data set matches the expected gradient over the model distribution.

While the expectation term in the log-likelihood gradient is, in general, infeasible to compute exactly, we can straightforwardly approximate it using the sampling methods discussed in the previous chapters. In particular, if we have drawn samples $\mathcal{S} = \{S_1, \ldots, S_M\}$, with $S_1, \ldots, S_M \subseteq V$, from distribution p, we can approximate the gradient $\boldsymbol{g}(\boldsymbol{\theta})$ by

$$\widetilde{\boldsymbol{g}}(\boldsymbol{\theta}) := \frac{1}{N} \sum_{i=1}^{N} \nabla_{\boldsymbol{\theta}} F(D_i; \boldsymbol{\theta}) - \frac{1}{M} \sum_{i=1}^{M} \nabla_{\boldsymbol{\theta}} F(S_i; \boldsymbol{\theta}).$$

We, therefore, propose learning the parameters $\boldsymbol{\theta}$ using an approximate gradient ascent procedure, which involves alternating between sampling from the current model to compute $\widetilde{\boldsymbol{g}}(\boldsymbol{\theta})$, and performing a gradient step towards the direction of $\widetilde{\boldsymbol{g}}(\boldsymbol{\theta})$, as shown in Algorithm 5.1.

Gradients of the FLDC model. Since we will be focusing on the FLDC model for the remainder of this chapter, we derive here the gradients of its potential function with respect to its parameters. For simplicity, we assume that we use an equal number of L dimensions for both the repulsive and the attractive matrices. As a reminder, the FLDC model is in that case defined via the following function (cf. Example 2.14),

$$F(S; \boldsymbol{u}, \boldsymbol{w}, \boldsymbol{v}) = \sum_{i \in S} u_i + \sum_{j=1}^{L} \left(\max_{i \in S} w_{ij} - \sum_{i \in S} w_{ij} - \max_{i \in S} v_{ij} + \sum_{i \in S} v_{ij} \right).$$

Due to the presence of the two "max" functions, F is differentiable only almost everywhere. For the points where it is not differentiable, we define subgradients that give equal contribution to all elements that belong to the corresponding set of maximizers. In particular, for all $i \in V, j \in [L]$, we have

$$\nabla_{u_i} F(S; \boldsymbol{u}, \boldsymbol{w}, \boldsymbol{v}) = [\![i \in S]\!]$$
$$\nabla_{w_{ij}} F(S; \boldsymbol{u}, \boldsymbol{w}, \boldsymbol{v}) = \frac{[\![i \in \operatorname{argmax}_{r \in S} w_{rj}]\!]}{|\operatorname{argmax}_{r \in S} w_{rj}|} - [\![i \in S]\!]$$
$$\nabla_{v_{ij}} F(S; \boldsymbol{u}, \boldsymbol{w}, \boldsymbol{v}) = -\frac{[\![i \in \operatorname{argmax}_{r \in S} v_{rj}]\!]}{|\operatorname{argmax}_{r \in S} v_{rj}|} + [\![i \in S]\!].$$

Tschiatschek et al. (2016) used an alternative set of subgradients, involving randomization over the choice of the "argmax" at each gradient step. We have noticed that our choice often results in slightly improved learning performance in practice.

Related work. Most of the previous work on learning discrete probabilistic models has focused on (pairwise) Markov random fields. Using a maximum likelihood approach to learn such models, or more generally, exponential family models, is fundamentally very similar to what we described above. As we mentioned, the moment matching condition requires approximating marginals, which has been accomplished in the past using a variety of sampling (Geyer, 1991), and variational (Wainwright & Jordan, 2008) methods. The method of contrastive divergence (Carreira-Perpiñán & Hinton, 2005) has been notable for making further approximations to the sampling procedure to speed up the learning process.

Alternative approaches seek to optimize different objectives than the likelihood to avoid performing inference altogether. Examples of other objectives include the pseudolikelihood (Besag, 1975), which involves easy to compute conditional probabilities; the noise-contrastive objective that aims to differentiate real data from noise Gutmann & Hyvärinen (2012); Tschiatschek et al. (2016); and the objective proposed by Domke (2013) that is directly based on marginal probabilities.

The problem of learning submodular functions has also been approached from non-probabilistic viewpoints. For example, Balcan & Harvey (2012) investigate learning such a function when given a black-box oracle of its value. Tschiatschek et al. (2014) learn mixtures of submodular functions, and Dolhansky & Bilmes (2016) learn submodular functions defined by deep architectures, both using large-margin approaches.

5.3 Modeling Interactions of Gene Mutations in Cancer

One of the goals of cancer genomics research is identifying so-called driver mutations, that is, somatic mutations that are responsible for various forms of cancer, and distinguishing them from randomly occurring passenger mutations. While sequencing data from large projects, such as The Cancer Genome Atlas (TCGA, 2008), has been available in increasing quantities, analyzing mutation interactions is a combinatorially daunting task.

Driver mutations often occur in a limited number of key biological pathways, and it has been observed that multiple mutations involved in the same pathway tend to not occur together in the same patient (Yeang et al., 2008). As a result, it is of interest to discover groups of gene alterations that are (approximately) mutually exclusive. Finding such a group is then an indication that the participating mutations are part of the same cancer-related pathway. Since most existing pathway databases lack in detail and accuracy, there has been particular interest in *de novo* methods, that is, methods that analyze the existing patient data without using any prior biological knowledge, and try to identify new potentially significant combinations of mutations. For a general review of the topic, we refer to Raphael et al. (2014).

Previous *de novo* methods have used different combinatorial or statistical scores to assess the degree of mutual exclusivity of a group of mutations. These are then paired with some discovery algorithm that exhaustively enumerates groups (Szczurek & Beerenwinkel, 2014; Yeang et al., 2008), progressively builds up larger groups from smaller ones (Babur et al., 2015; Miller et al., 2011; Ciriello et al., 2011; Constantinescu et al., 2015), or performs a randomized search in the group space (Vandin et al., 2011; Leiserson et al., 2013; 2015). As a result, these methods either scale poorly in the number of mutations at hand, or require prior assumptions on the exact or maximum size of the groups to be discovered. In the following sections, we compare our results against the CoMEt algorithm (Leiserson et al., 2015), which is a state-of-the-art method for discovering multiple groups of mutually exclusive mutations. While CoMEt requires prespecifying the number and size of groups to be searched for, it is able to ultimately put together a consensus of arbitrarily sized groups.

5.3.1 Our Approach

Assume that we are given a ground set $V = \{1, \ldots, n\}$ of possible gene mutations, and a data set of N patients, $\mathcal{D} := (D_1, \ldots, D_N)$, where $D_i \subseteq V$

is the combination of mutations that were present in patient i. The data is commonly represented in the literature using a binary alteration matrix, where each row represents a mutation, and each column a patient. Our goal is to discover groups of mutations $M_1, M_2, \ldots, M_i \subseteq V$, with the property that mutations that belong to the same group rarely occur together in the same patient (see Figure 1.1).

We propose using the patient data $\mathcal{D}$ to learn an FLDC model over the mutation space V. Based on the definition of this model, we expect the columns of the $\boldsymbol{w}$ and $\boldsymbol{v}$ matrices to encode groups of mutually exclusive and co-occurring mutations respectively. For the purposes of this thesis, we propose extracting potential groups by thresholding each matrix at a specified level; these groups can then be further assessed for mutual exclusivity or co-occurrence using some of the previously proposed statistical tests. More generally, one can perform inference in the learned model to compute various probabilistic quantities that may be useful in specific biological applications.

Our approach offers several advantages over previous work. First, it inherently uses higher-order potentials to directly capture mutation interactions of arbitrary size, without any need to specify the number of groups or size of each group in advance. Second, in addition to mutual exclusivity, it also models mutation co-occurrence, a property that may also provide useful information in cancer research (Yeang et al., 2008; Raphael et al., 2014). Finally, in terms of computational complexity, the only potentially super-linear component in our learning procedure is the number of samples required to get an accurate gradient approximation. This further justifies the pursuit of efficient sampling methods in the previous chapters. In practice, our algorithm only takes a few minutes to run on real cancer data sets containing hundreds of mutations.

5.3.2 Experimental Setup

We provide here some more details about each step of the procedure we use to discover mutually exclusive groups of mutations. The steps for discovering co-occurring groups are analogous.

Step 1: Learning the FLDC model. We use the approximate maximum likelihood method described in Section 5.2. By definition of the FLDC model, the elements of matrices $\boldsymbol{w}$ and $\boldsymbol{v}$ must be non-negative. To achieve this during learning, we project the entries of $\boldsymbol{w}$ and $\boldsymbol{v}$ to the positive orthant after each gradient step. Furthermore, we have found it beneficial in practice to

induce sparsity on these matrices, in order to reduce the effect of noisy data on the learned models, and obtain more interpretable solutions. To this end, we employ an L_1-regularization to both $\boldsymbol{w}$ and $\boldsymbol{v}$ by projecting each row and column of these matrices to the corresponding L_1-ball after each gradient step.

We initialize the entries of $\boldsymbol{u}$ to the maximum likelihood estimates of the respective product distribution, that is,

$$u_i = \log\left(\frac{f_i}{1 - f_i}\right),$$

where f_i is the frequency of element $i \in V$ in the data set $\mathcal{D}$. We randomly initialize the entries of $\boldsymbol{w}$ and $\boldsymbol{v}$ by drawing each of them from a uniform distribution $\mathcal{U}[0, 0.01]$. To avoid duplicate latent dimensions in the two matrices, for the first half of the iterations, we check the columns of $\boldsymbol{w}$ and $\boldsymbol{v}$ after each gradient step, and reinitialize a column when we detect that its L_1-distance to another column of the same matrix is smaller than a predefined threshold.

Unless otherwise stated, we use $n_{\text{iter}} = 2 \cdot 10^4$ gradient iterations, and $M = 200|V|$ samples per iteration. We use the combined sampler detailed in Chapter 4 with a mix of 100 random sub- and supergradients, and a combination weight of $\delta = 0.5$. Finally, we use a fixed step size ($\gamma = 5 \cdot 10^{-4}$) for the first half of the iterations, and a geometrically decreasing step size ($\gamma_i = \gamma r^i$ with $r = 10^{-3/n_{\text{iter}}}$) for the second half.

Step 2: Extracting proposed mutation sets. We start by thresholding each column of the learned $\boldsymbol{w}$ matrix at a fixed level $w_{\text{th}} = 1.5$. We then proceed to create a graph that contains one clique of nodes for each group extracted in the previous step. Our proposed mutation sets consist of all maximal cliques in this constructed graph. Creating the graph, rather than directly proposing the groups extracted from the matrix columns, can be useful for merging smaller groups of genes that have been encoded in separate columns of $\boldsymbol{w}$ during learning.

Step 3: Testing mutual exclusivity. We make use of two statistical tests for testing the degree of mutual exclusivity of a mutation group.

The first was proposed by Babur et al. (2015), and used as part of the Mutex algorithm. For each gene in a proposed mutation group, we run Fisher's one-tailed exact test on the contingency table that results from examining the occurrences of that gene in the data set versus the union of all other genes in the group. This results in one p-value per gene in the mutation

group, and the output of the test is the maximum of these p-values. We will call this the "one vs. all" test, and denote its output p-value by p_{ova}.

The second was proposed by Leiserson et al. (2015), and used as part of the CoMEt algorithm. It generalizes Fisher's exact test to higher-dimensional contingency tables. In particular, it consists of a null hypothesis of independent hypergeometric distributions, one for each mutation in the group, and uses as a test statistic the sum of patients in which exactly one mutation from the group occurs. We will call this the "generalized Fisher" test, and denote its output p-value by p_{gf}.

Step 4: FDR control. For the synthetic experiments, we will want to make a final decision of whether a proposed group is significantly mutually exclusive or not, in order to compare to the ground truth. For that purpose, we employ the one vs. all test discussed above, and correct for multiple testing by using an online FDR control procedure known as LORD++ (Ramdas et al., 2017; Javanmard & Montanari, 2018). In contrast to classic offline methods, such as the BH step-up procedure (Benjamini & Hochberg, 1995), online methods can be applied to settings where the hypotheses to be tested are not necessarily known in advance, and my arrive in an arbitrary order. This is useful in our case, because we want to output maximal mutually exclusive groups, which means that the decision of whether to test a group or not will depend on whether a supergroup has already been rejected or not. The LORD++ procedure takes as input the significance level α at which we are testing. For the procedure's "starting alpha-wealth" parameter we use $W_0 = 0.8\alpha$.

For the real data experiments, in the absence of ground truth, we take a more exploratory approach, and do not employ multiple testing. Rather, we illustrate and discuss the most significant discovered groups, as indicated by their p-values according to both statistical tests described above.

Co-occurrence tests. For assessing co-occurrence, we define a version of the "one vs. all" test that is completely analogous to the one described above, except that we use the opposite tail of the null distribution in Fisher's test compared to the mutually exclusive case. To define a "generalized Fisher" test for co-occurrence, we use as a test statistic the sum of patients in which all mutations from the corresponding group occur simultaneously.

5.4 Synthetic Data

We proceed to practically apply our learning algorithm to the problem of modeling gene interactions, starting with three synthetic data sets.

5.4.1 Learning

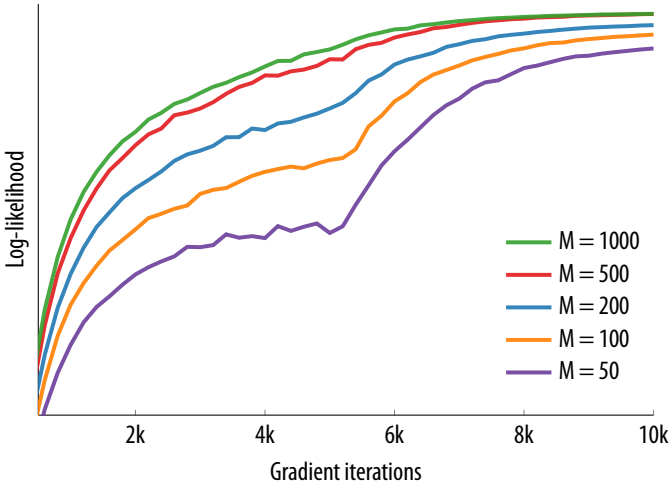
To begin with, we want to illustrate how the gradient approximation via sampling affects the learning algorithm. We create a reduced version of one of the real cancer data sets (see Section 5.5), so that we are able to compute the exact log-likelihood during learning. Starting with the AML data set detailed in the next section, we only keep the 17 gene mutations shown in Figure 5.8, thus creating a data set of 17 mutations and 200 patients. We then learn a FLDC model with $L = 10$ latent dimensions for $n_{\text{iter}} = 10^4$ gradient iterations.

Figure 5.1a shows the evolution of the log-likelihood for an increasing number of samples, while keeping the number of semigradients constant. As expected, using a larger number of samples leads to a more accurate gradient approximation, which results in faster learning. We also see that the benefit of increased samples plateaus after some point; for example, we see minimal benefit when increasing the samples from 500 to 1000.

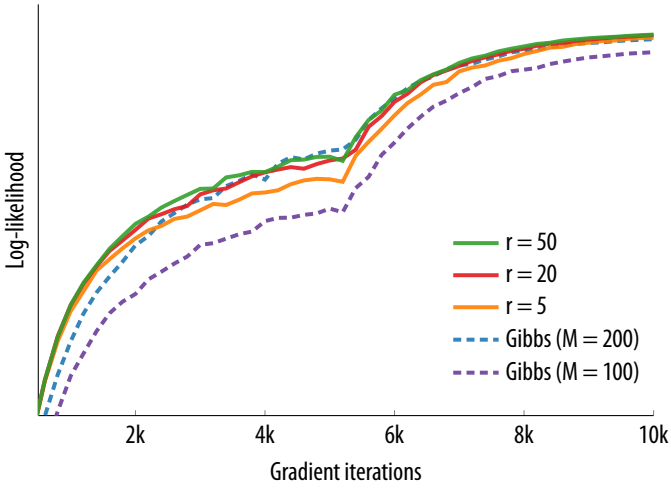
Similar conclusions can be drawn from the results Figure 5.1b about the effect of the number of semigradients used. We can see that for this example we get practically no benefit from adding more than 20 semigradients, but this number will likely need to be adjusted when learning from data with larger ground sets. We also see that the benefit obtained from 20 semigradients corresponds to about doubling the number of Gibbs samples from 100 to 200.

5.4.2 Single Mutually Exclusive Group

We focus now on extracting a single group of mutually exclusive mutations. We create synthetic data sets of 100 mutations and 500 patients, following the procedure outlined by Leiserson et al. (2015). First, we choose $k = 3$ mutations, which cover a fraction γ of the patients, and have them be completely mutually exclusive, that is, only one of the three mutations occurs in each of the 500γ patients. Furthermore, each of the three mutations appear in a fraction 0.5, 0.35, and 0.15 of the 500γ patients, respectively. Second, we choose 5 mutations, which occur frequently (with fractions 0.67, 0.49, 0.29, 0.29, 0.2 in the 500 patients), and completely independently of each



(a) Number of semigradients fixed to $r = 20$.



(b) Number of samples fixed to $M = 200$.

Figure 5.1: Learning curves on the reduced AML data set for (a) varying number of samples, and (b) varying number of semigradients.

other and of any other mutation, including the mutually exclusive ones. Finally, we add random noise by independently activating each mutation in each patient with probability 0.0028.

Figure 5.2 shows the learned $\boldsymbol{w}$ matrix of the FLDC model for such a synthetic data set with $\gamma = 0.5$. Note how the mutually exclusive group $S = \{16, 23, 68\}$ is distinctly encoded in the last column of the matrix.

To evaluate the ability of our method to recover the true mutually exclusive group, we computed the F-measure of the union of the resulting extracted groups compared to the true group. Figure 5.3 shows the results across different values of the fraction γ of patients covered by the mutually exclusive group, ranging from 0.1 to 1.0. For each value of γ, we repeat the experiment on 50 randomly generated data sets.

We show the results of our procedure for two different values of the considered hypothesis testing level α, which trades off between false negatives and false positives. For $\alpha = 0.01$, we have practically perfect recovery when $\gamma \geq 0.4$, but are not able to extract statistically significant groups below $\gamma = 0.3$. Using a much higher $\alpha = 0.3$ trades off some false positives to gain statistical power, and exceeds the performance of CoMEt for almost all values of γ. Either way, the results show that, in the majority of the cases, the learned FLDC model is able to encode the correct group, and propose it for further testing. It is important to emphasize that CoMEt takes the size $k = 3$ of the group as input, although it can still output groups of different size; our method, on the other hand, does not use any such information.

5.4.3 Multiple Mutually Exclusive Groups

We now move to the problem of extracting multiple groups of mutually exclusive mutations. Again, we create synthetic data sets following a procedure outlined by Leiserson et al. (2015). In this case, we start with $20,000$ mutations and 500 patients, and select t groups of k mutually exclusive mutations. The number of groups t range from 2 to 4, and the mutations per group k from 3 to 5. Each group covers a fraction of patients ranging from 0.4 to 0.7, and the mutations of each group cover equal number of patients. As before, we add 5 independently mutated genes with high frequencies, as well as random noise. Finally, we remove genes that are mutated in fewer than 5 patients, which results in a final ground set of average size $|V| \approx 275$. To assess the quality of the recovered groups against the true ones, we use the adjusted Rand index (Hubert, 1985), which is a measure that compares the similarity of two clusterings of a set of elements. Thus, an adjusted Rand index of 1 indicates that the recovered groups are identical to the true ones.

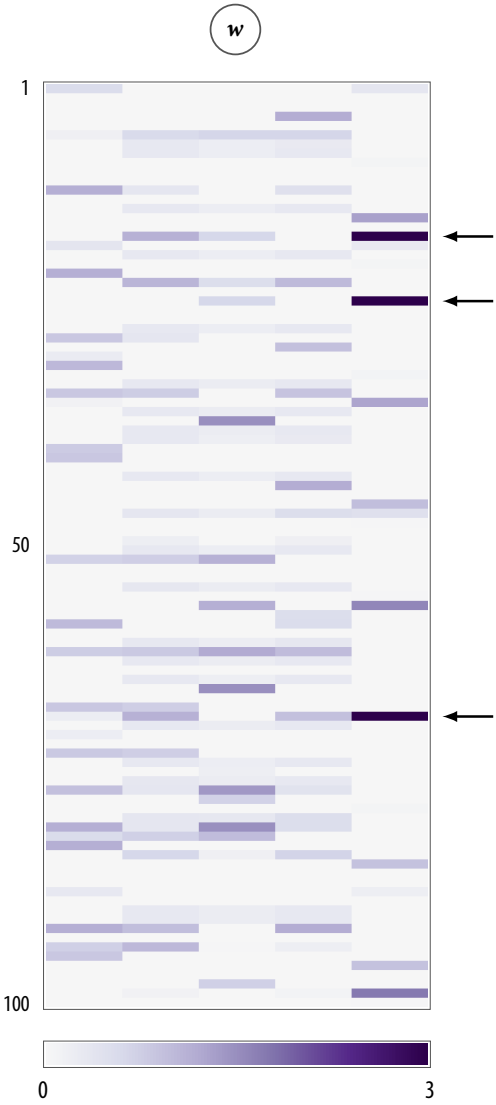


Figure 5.2: The $\boldsymbol{w}$ matrix of the learned FLDC model on a synthetic data set of 100 genes and 500 patients. The implanted mutually exclusive group of $k = 3$ mutations covers a fraction $\gamma = 0.5$ of the patients, and is distinctly encoded in the last column of the matrix.

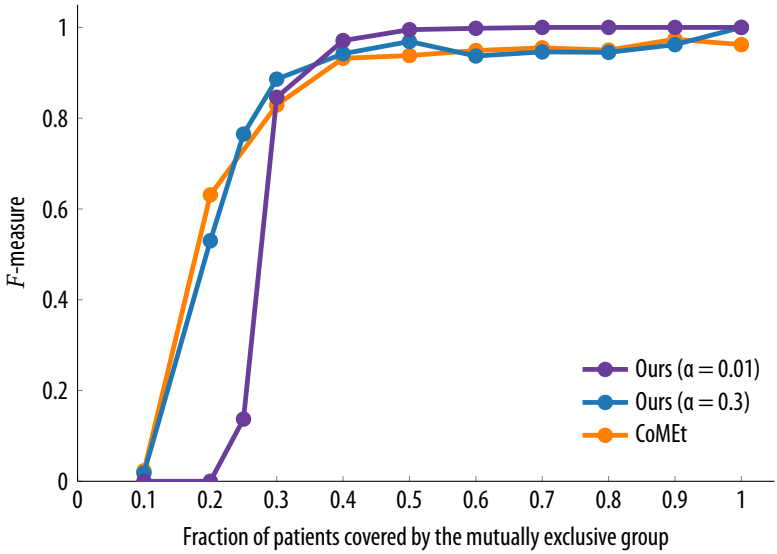


Figure 5.3: Results on recovering a single group of $k = 3$ mutually exclusive mutations for different values of the fraction γ of patients covered by that group. The level α at which we test trades off statistical power at low γ for false positives at high γ.

For each combination of t and k we repeated the experiment on 50 randomly generated data sets. We ran CoMEt with fixed values of $t = 3$ and $k = 4$, as was done in the original paper (Leiserson et al., 2015). Figure 5.4 shows the results, in which we see that our method significantly outperforms CoMEt, especially at the lower values of t and k. This showcases the problems encountered by CoMEt when the number and size of the groups to be found is misspecified. These problems are shared with several other methods proposed in the past. In contrast, we see that our method performs consistently well across all different values of t and k, without any knowledge of these parameters. We also see that, in this case, a higher level α degrades the quality of the final results, because of the frequent occurrence of false positives.

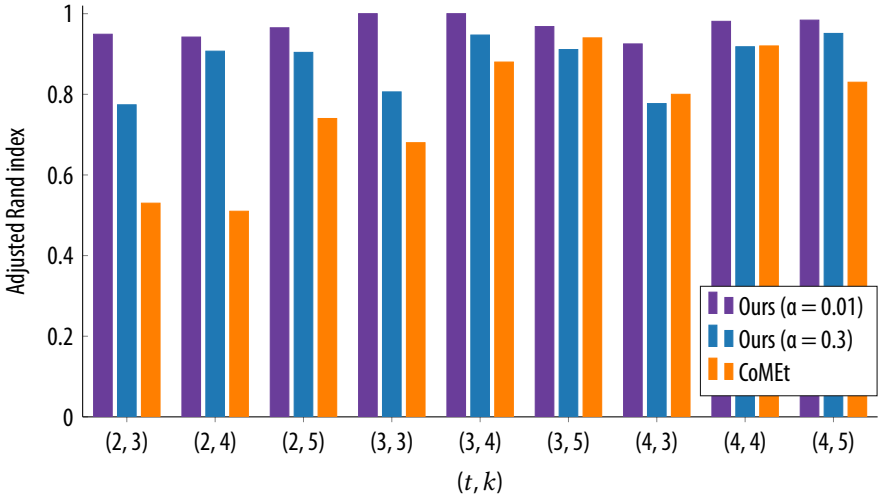


Figure 5.4: Results on recovering t groups of k mutually exclusive mutations. Our method performs consistently well across all different t and k, while CoMEt performs poorly when t and k are misspecified.

5.5 Real Cancer Data

We analyzed two real cancer data sets from TCGA, the first pertaining to acute myeloid leukemia (TCGA, 2013), and the second to breast cancer (TCGA, 2012). For both data sets, we used the preprocessed versions by Leiserson et al. (2015) available on GitHub[1].

5.5.1 Acute Myeloid Leukemia (AML)

The data set consists of 51 mutations and 200 patients. The learned FLDC model ($L = 10$) is shown in Figure 5.5. We have annotated on the two matrices the discovered mutually exclusive and co-occurring groups that have (uncorrected) p-values ≤ 0.01 for both p_{gf} and p_{ova}. (We have also decided to include group R_2, even though its p_{ova} is barely above 0.01.)

Figures 5.6 and 5.7 illustrate in detail each of the mutually exclusive

[1] https://github.com/raphael-group/comet

groups R_1–R_8 by showing the corresponding permuted data rows, as well as the computed p-values for each group. Figure 5.8 summarizes these eight groups into a graphical structure. Each node represents a mutation, with darker nodes corresponding to higher marginal frequency in the data set. The mutually exclusive groups are shown shaded, with darker groups corresponding to lower p_{ova}. Figure 5.9 shows the five discovered groups A_1–A_5 of co-occurring mutations.

It is notable that CoMEt detects groups R_1 (without the last mutation, RUNX1-RUNX1T1) and R_2, but fails to detect any of the other six groups, even though almost all of them have particularly low p_{gf}. We suspect that this is caused by the combination of CoMEt using as input a fixed number of groups and sizes thereof, as well as the fact that the algorithm's sampling-based searching procedure based works on a particularly slow-mixing landscape. As an example, although the authors are searching for a group of size 6, they never discover the group R_3 shown in Figure 5.6, because their sampler presumably gets stuck on group R_1, which has a few orders of magnitude lower p_{gf} compared to R_3. On the other hand, CoMEt detects two other groups (see Figure A.2 in the appendix), which have p_{ova} significantly above our cutoff, and furthermore, also have p_{gf} significantly higher than all our discovered groups except for R_8. For a comparison of the discovered mutation interactions, we also refer to (TCGA, 2013, Figure S8), which confirms many of our findings, although it is limited to pairwise interactions.

5.5.2 Breast Cancer (BRCA)

The data set consists of 375 mutations and 507 patients. We show the resulting learned FLDC model ($L = 15$) in Figure 5.10, the extracted mutually exclusive groups R_1–R_{11} in Figures 5.11 and 5.12, and a graphical summarization of all these groups in Figure 5.13. In Figure 5.14 we show the six extracted co-occurring groups with the highest coverage; groups A_7–A_{14} can be found in Figures A.3 and A.4 in the appendix.

The CoMEt results on BRCA take into account additional information about the classification of each patient into four different subtypes of breast cancer. While this means that we cannot directly compare our results to theirs, we still recover some of their findings (e.g., the first three mutations in R_1), but also extract significant groups that were not found by CoMEt, even when constrained to a specific subtype (e.g., R_3 for the "Luminal-A" subtype). Furthermore, we observe that the co-occurring groups A_1–A_6 are particularly significant, with p-values of order 10^{-5} or less.

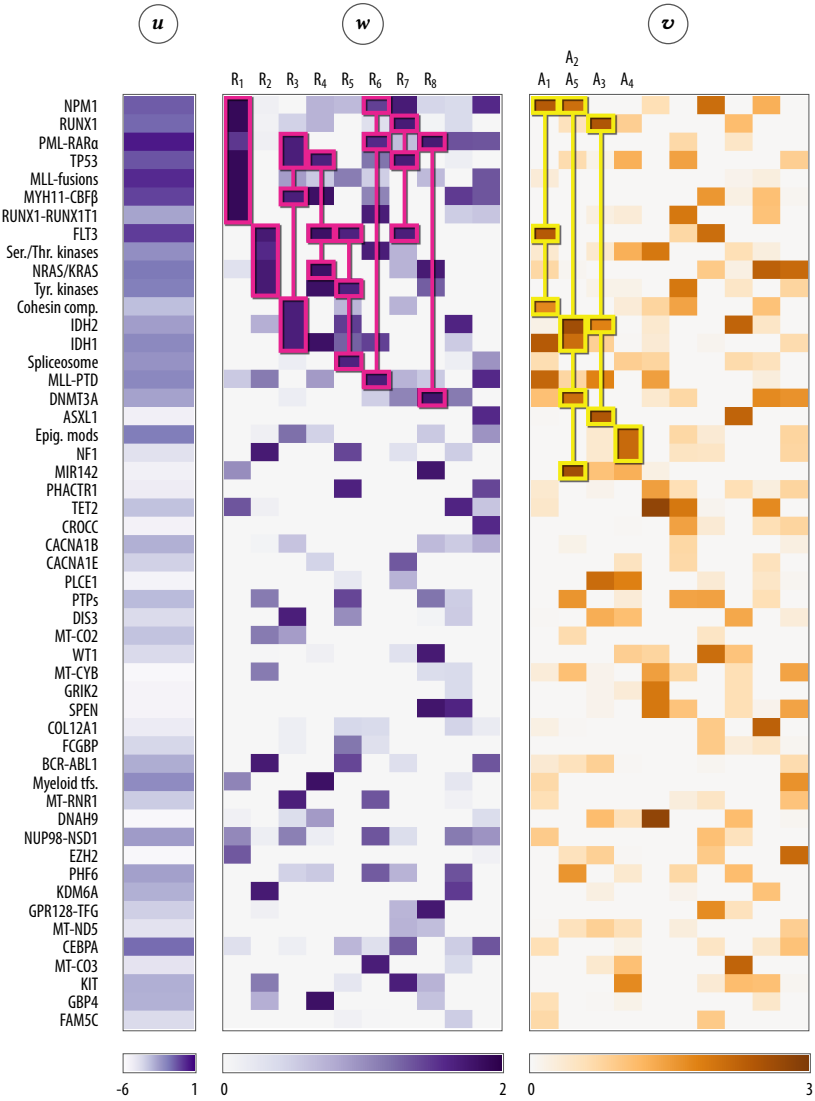


Figure 5.5: The learned utility ($\boldsymbol{u}$), repulsive ($\boldsymbol{w}$), and attractive ($\boldsymbol{v}$) matrices of the FLDC model ($L = 10$) on the TCGA AML data set. We have permuted the rows and columns to bring the discovered groups towards the upper left of the matrices. Groups R_1–R_8 are mutually exclusive, while groups A_1–A_5 are co-occurring.

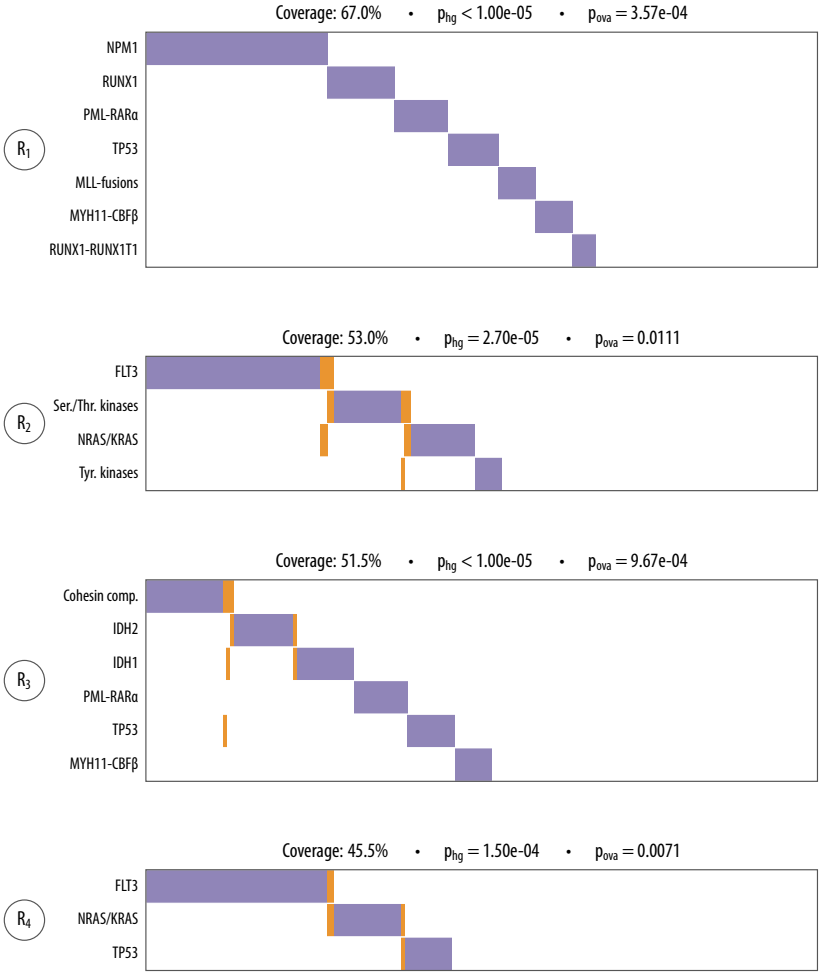


Figure 5.6: The first four mutually exclusive groups extracted from the TCGA AML data set. Each row corresponds to a mutation, and each column to a patient. The highlighted entries represent co-occurring mutations.

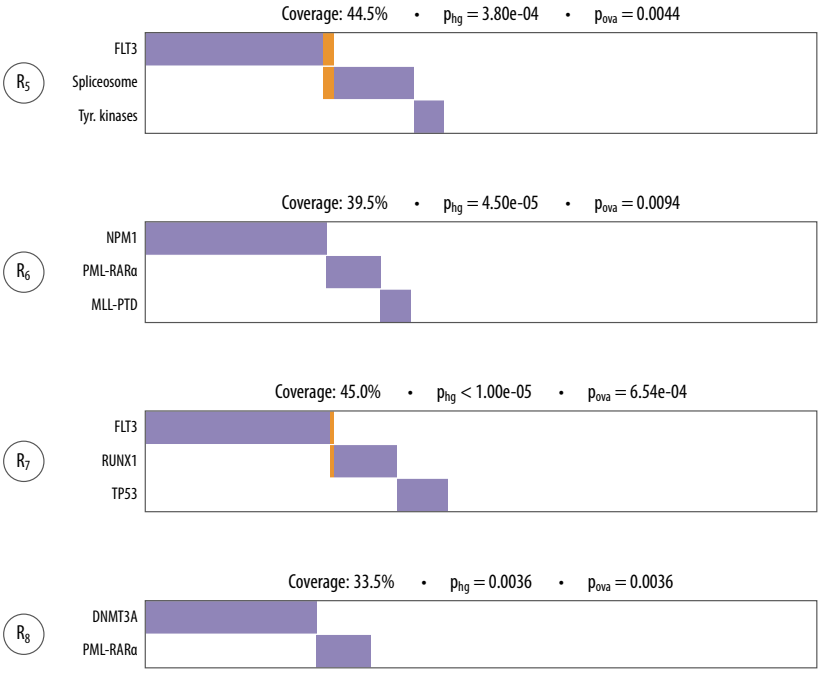


Figure 5.7: The next four mutually exclusive groups extracted from the TCGA AML data set. Each row corresponds to a mutation, and each column to a patient. The highlighted entries represent co-occurring mutations.

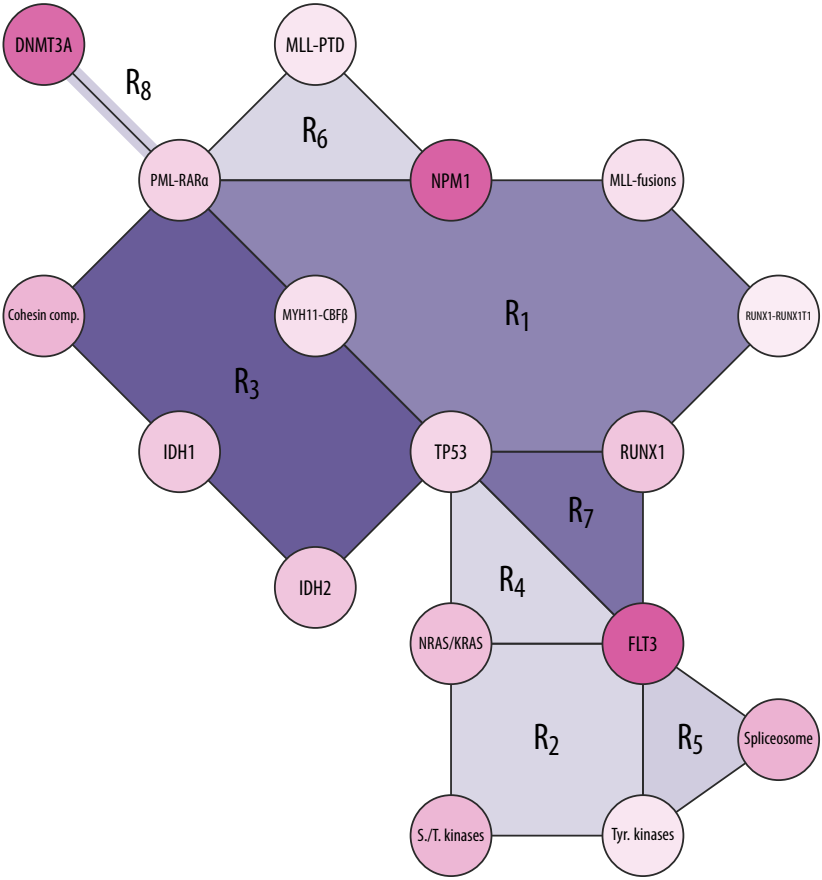


Figure 5.8: A graphical representation of the eight discovered mutually exclusive groups in the TCGA AML data set. Darker nodes correspond to more frequent mutations, and darker shaded polygons correspond to more significant (that is, lower p_{ova}) groups.

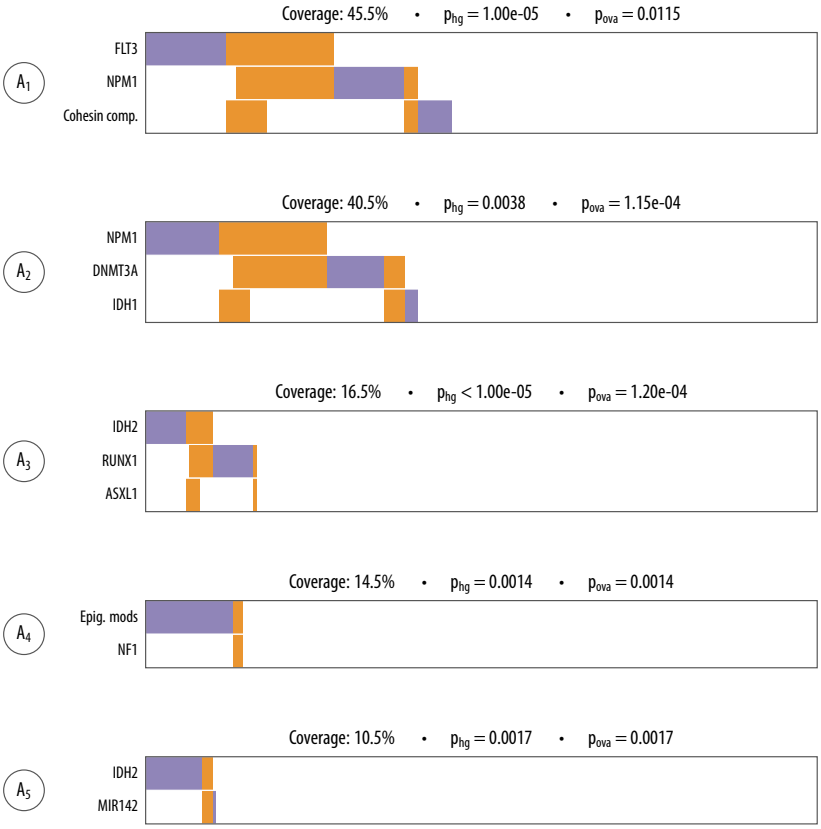


Figure 5.9: The five co-occurring groups extracted from the TCGA AML data set. Each row corresponds to a mutation, and each column to a patient. The highlighted entries represent co-occurring mutations.

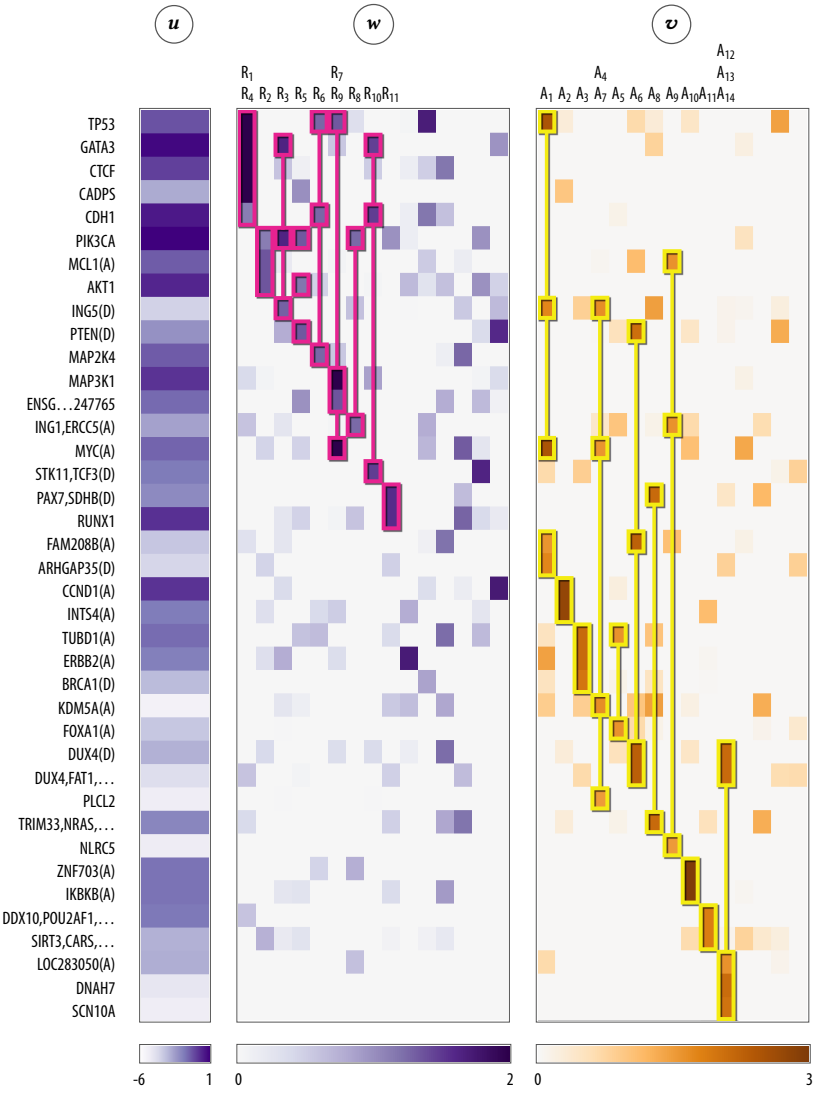


Figure 5.10: The learned utility ($\boldsymbol{u}$), repulsive ($\boldsymbol{w}$), and attractive ($\boldsymbol{v}$) matrices of the FLDC model ($L = 15$) on the TCGA BRCA data set. For illustration purposes, we only show the submatrices corresponding to the 39 mutations that participate in the extracted groups. We have also permuted the rows and columns to bring these groups towards the upper left of the matrices. Groups R_1–R_{11} are mutually exclusive, while groups A_1–A_{14} are co-occurring.

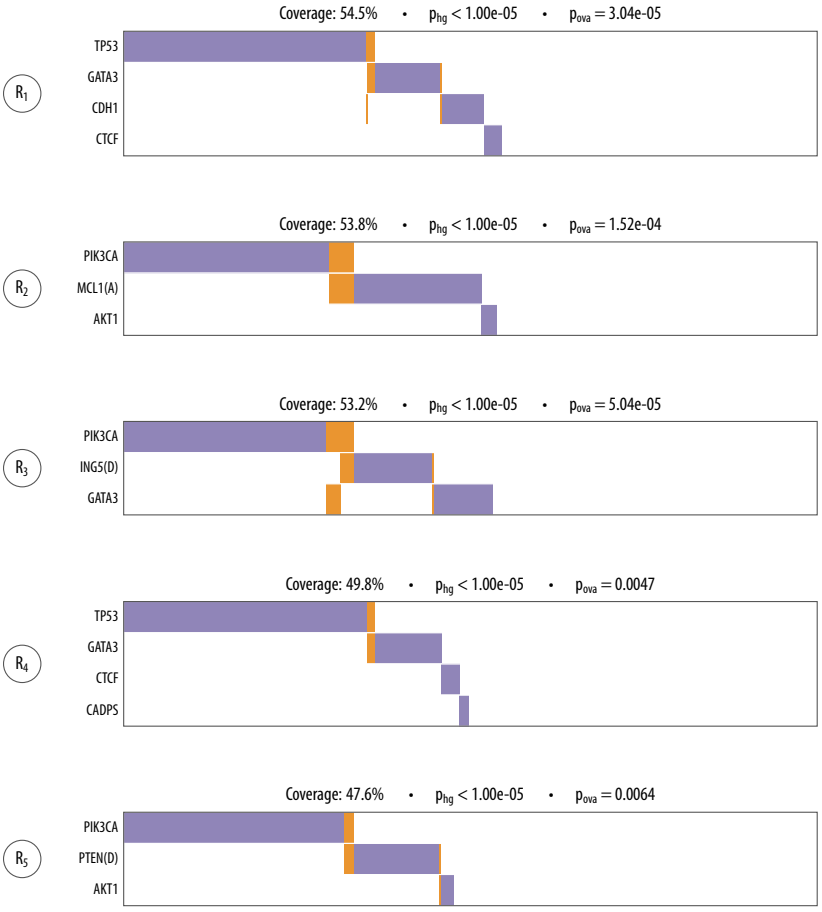


Figure 5.11: The first five mutually exclusive groups extracted from the TCGA BRCA data set. Each row corresponds to a mutation, and each column to a patient. The highlighted entries represent co-occurring mutations.

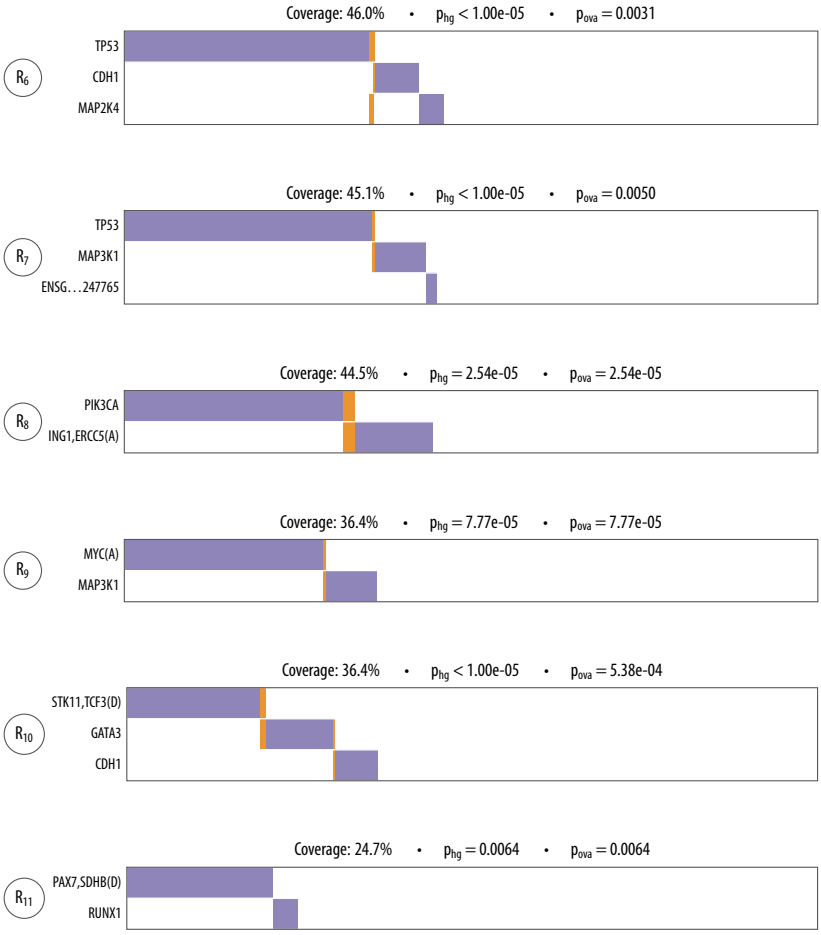


Figure 5.12: The next six mutually exclusive groups extracted from the TCGA BRCA data set. Each row corresponds to a mutation, and each column to a patient. The highlighted entries represent co-occurring mutations.

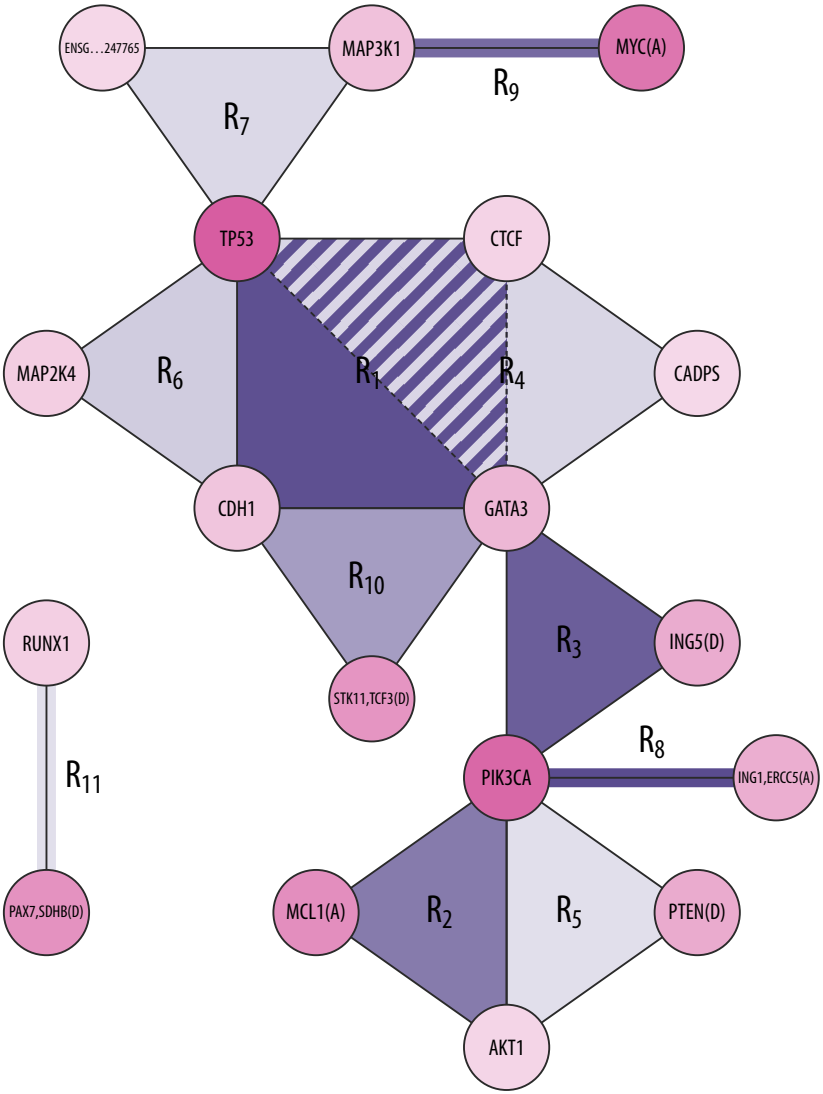


Figure 5.13: A graphical representation of the 11 discovered mutually exclusive groups in the TCGA BRCA data set. Darker nodes correspond to more frequent mutations, and darker shaded polygons correspond to more significant (that is, lower p_{ova}) groups.

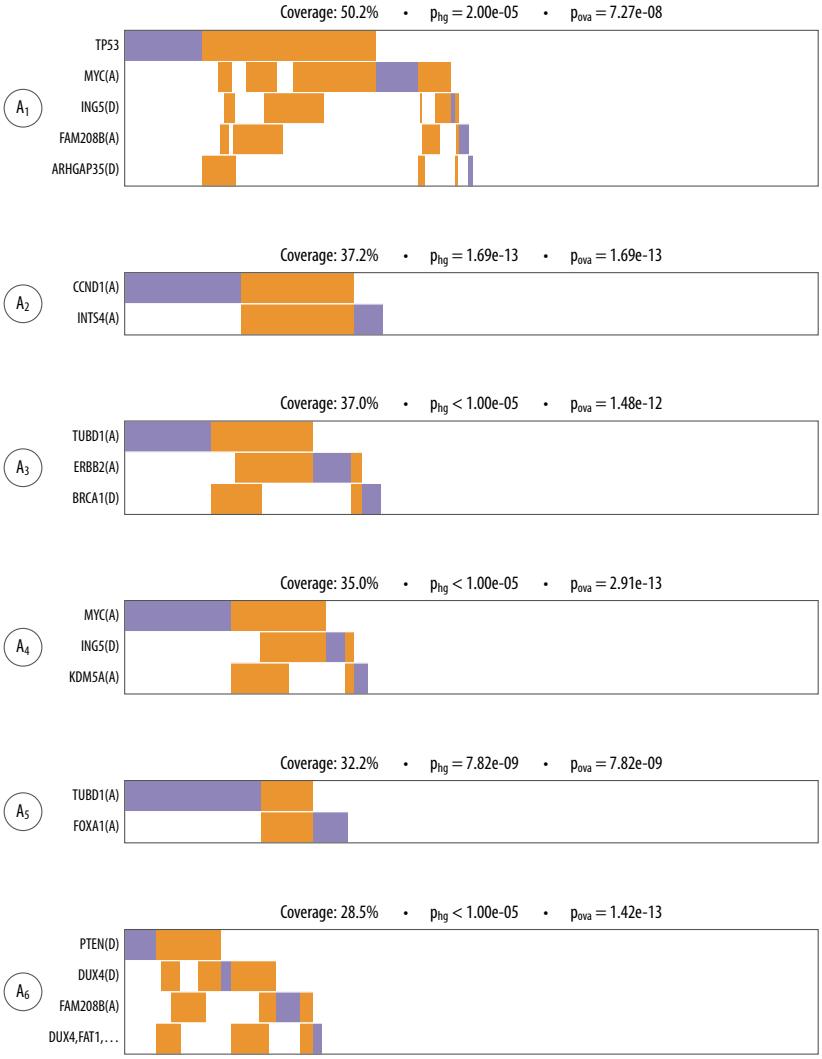


Figure 5.14: The first six co-occurring groups extracted from the TCGA BRCA data set. Each row corresponds to a mutation, and each column to a patient. The highlighted entries represent co-occurring mutations.

5.6 Conclusion

In this chapter, we have seen how sampling can be effectively used to obtain estimates of the gradient of a probabilistic submodular model with respect to its parameters. It, thus, facilitates applying an approximate maximum likelihood maximization procedure to learn such models from data. We have applied this learning procedure to the problem of modeling interactions of genetic mutations in cancer patients, with the particular goal of discovering groups of mutually exclusive and co-occurring mutations. We have shown that our method often outperforms the state of the art for this task by naturally capturing these higher-order interactions without the need to prespecify the number or size of groups to be found.

6 Conclusion

In this thesis, we focused on discrete probabilistic models defined via submodular or supermodular functions, and investigated the use of Markov chain Monte Carlo sampling techniques to perform inference in such models.

We analyzed the mixing behavior of the Gibbs sampler on probabilistic submodular models, and showed that under conditions that quantify the distance from modularity or the influence of element on the function values, we can guarantee polynomial-time or $O(n \log n)$ mixing, respectively. These conditions also showed how sub- or supermodularity can lead to improved mixing bounds.

We then proposed a novel sampling procedure that combines the Gibbs sampler with a Metropolis chain that performs global moves to avoid state-space bottlenecks. The construction of this chain involved creating a mixture of semigradient-based log-modular distributions, and illustrated how concepts from discrete optimization may be leveraged in probabilistic inference.

Finally, we showed how we can use sampling to approximate the likelihood gradients, and perform an approximate likelihood maximization using gradient ascent. We applied this learning procedure to the problem of modeling interactions of genetic mutations in cancer patients, in particular mutual exclusivity and co-occurrence. Our results illustrated that our probabilistic framework provides a flexible way to encode such interactions without the need to specify the number or size of the groups that are being searched for. Moreover, in real cancer data we discovered significant groups of mutations that previous state-of-the-art methods failed to find.

6.1 Future Work

We list here a few promising directions for future work related to this thesis.

Learning models with efficiency guarantees. Our approach in Chapter 3 was to provide conditions for efficient sampling in general probabilistic sub-

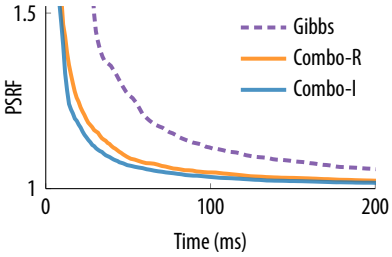
modular models. On the other hand, our learning procedure of Chapter 5 cannot guarantee that these conditions will hold in the resulting model. It is interesting to consider the problem of directly incorporating conditions for guaranteed inference efficiency into the learning process, in order to make sure that the learned models are amenable to inference. There has been little work in this direction for a limited model class (Domke, 2015).

Continuous sampling for discrete inference. Sampling methods for continuous domains, such as Hamiltonian Monte Carlo, have received increasing attention in the past few years, for their ability to combine gradient information with random momentum to perform more efficient moves in the state space. There has been some promising recent work on embedding discrete models into suitable continuous domains, then using a continuous sampler, and finally converting the samples back to the discrete domain (Zhang et al., 2012; Pakman & Paninski, 2013; Dinh et al., 2017; Nishimura et al., 2018). It is also interesting to investigate whether it is possible to directly define discrete counterparts of momentum and gradients, and, as a result, a discrete version of Hamiltonian Monte Carlo.

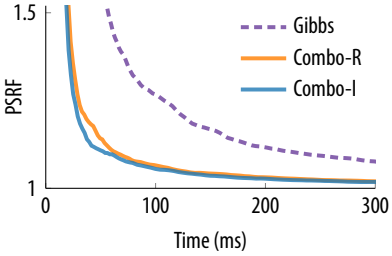
Strongly Rayleigh distributions. Strongly Rayleigh (Borcea et al., 2008) distributions capture a strong notion of negative dependence between elements, and have been shown to allow for efficient sampling (Anari et al., 2016; Li et al., 2016). Except for determinantal point processes, only very few interesting classes of parametric distributions are known to be strongly Rayleigh (Li et al., 2017). It is interesting to investigate under what conditions some well-known model classes are strongly Rayleigh, and to find efficient ways to represent and learn more general strongly Rayleigh models.

A Additional Experimental Results

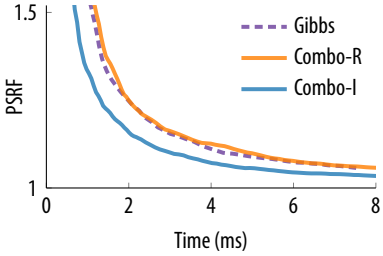
A.1 Results from Chapter 4



(a) Water



(b) Sensor



(c) Game

Figure A.1: Potential scale reduction factor as a function of wall-clock time.

A.2 Results from Chapter 5

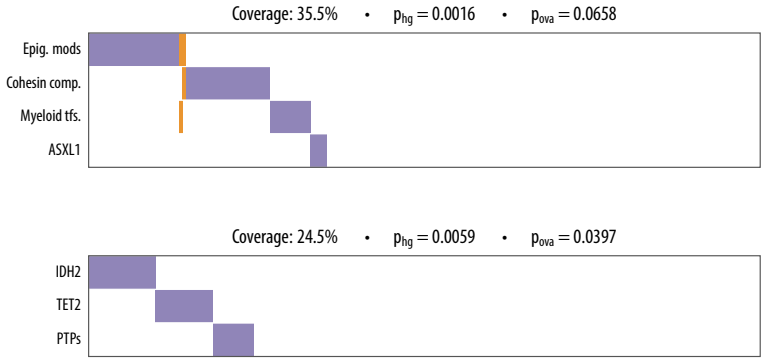


Figure A.2: Two more groups reported by CoMEt as mutually exclusive, which our method rejects due to the high p_{ova} values.

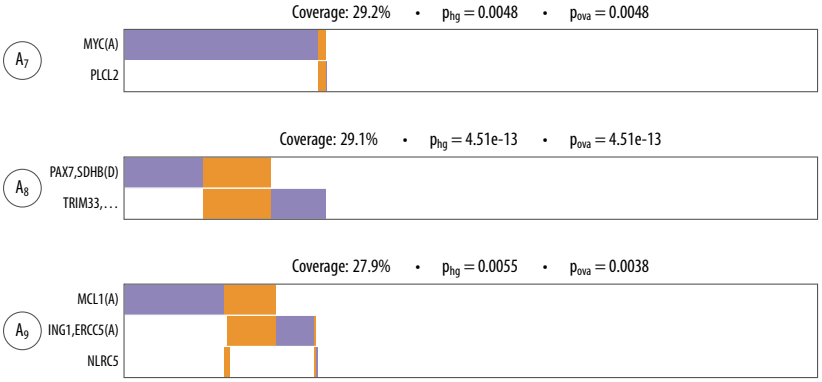


Figure A.3: The next three co-occurring groups extracted from the TCGA BRCA data set.

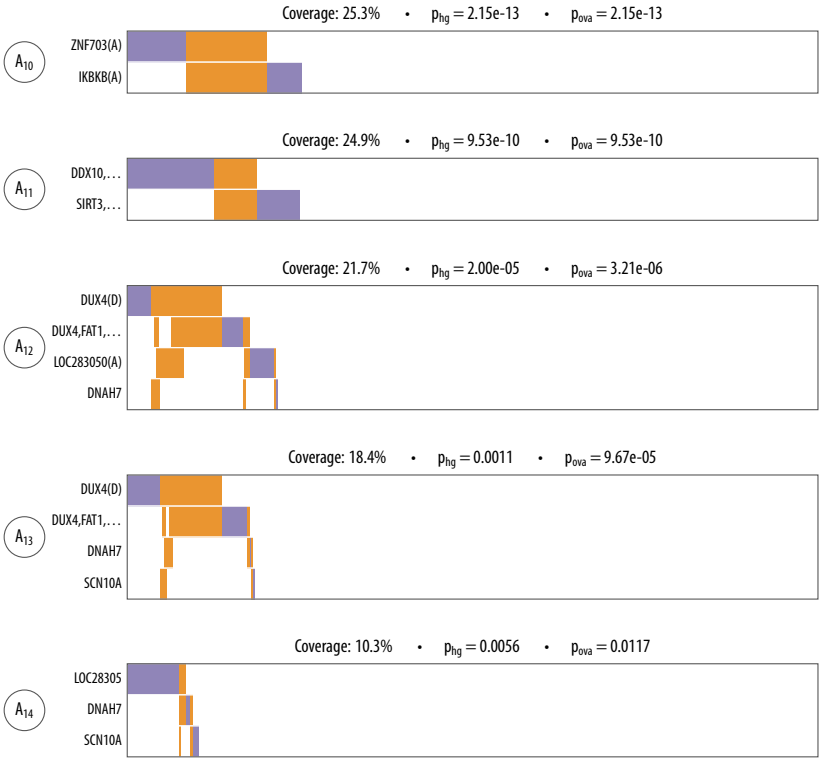


Figure A.4: The rest of the co-occurring groups extracted from the TCGA BRCA data set.

Bibliography

Ahn, S., Chen, Y., and Welling, M. Distributed and adaptive darting monte carlo through regenerations. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2013.

Aldous, D. Random walks on finite groups and rapidly mixing markov chains. In *Seminaire de Probabilites XVII*. Springer, 1983.

Anari, N., Gharan, S. O., and Rezaei, A. Monte Carlo Markov chain algorithms for sampling strongly Rayleigh distributions and determinantal point processes. In *Conference on Learning Theory (COLT)*, 2016.

Babur, Ö., Gönen, M., Aksoy, B. A., Schultz, N., Ciriello, G., Sander, C., and Demir, E. Systematic identification of cancer driving signaling pathways based on mutual exclusivity of genomic alterations. *Genome Biology*, 2015.

Balcan, M.-F. and Harvey, N. J. A. Learning submodular functions. In *Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, 2012.

Benjamini, Y. and Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*, 1995.

Besag, J. Statistical analysis of non-lattice data. *Journal of the Royal Statistical Society*, 1975.

Betancourt, M. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv*, 2017.

Borcea, J., Brändén, P., and Liggett, T. M. Negative dependence and the geometry of polynomials. *Journal of the American Mathematical Society*, 2008.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. In *Handbook of Markov Chain Monte Carlo*. CRC Press, 2011.

Bubley, R. and Dyer, M. Path coupling: A technique for proving rapid mixing in markov chains. In *Symposium on Foundations of Computer Science*, 1997.

Bubley, R., Dyer, M., and Greenhill, C. Beating the 2d bound for approximately counting colourings: A computer-assisted proof of rapid mixing. In *Symposium on Discrete Algorithms*, 1998.

Buchbinder, N., Feldman, M., Naor, J., and Schwartz, R. Submodular maximization with cardinality constraints. In *Symposium on Discrete Algorithms (SODA)*, 2014.

Burda, Y., Grosse, R. B., and Salakhutdinov, R. Accurate and conservative estimates of mrf log-likelihood using reverse annealing. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

Calinescu, G., Chekuri, C., Pál, M., and Vondrák, J. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 2011.

Carreira-Perpiñán, M. Á. and Hinton, G. E. On contrastive divergence learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2005.

Chekuri, C., Vondrak, J., and Zenklusen, R. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Symposium on Theory of Computing (STOC)*, 2011.

Ciriello, G., Cerami, E., Sandler, C., and Schultz, N. Mutual exclusivity analysis identifies oncogenic network modules. *Genome Research*, 2011.

Conforti, M. and Cornuejols, G. Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the rado-edmonds theorem. *Disc. App. Math.*, 1984.

Constantinescu, S., Szczurek, E., Mohammadi, P., Rahnenführer, J., and Beerenwinkel, N. TiMEx: a waiting time model for mutually exclusive cancer alterations. *Bioinformatics*, 2015.

de Freitas, N., Højen-Sørensen, P., Jordan, M. I., and Russell, S. Variational mcmc. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2001.

Diaconis, P. and Saloff-Coste, L. Comparison techniques for random walk on finite groups. *Annals of Probability*, 1993.

Diaconis, P. and Stroock, D. Geometric bounds for eigenvalues of markov chains. *The Annals of Applied Probability*, 1991.

Ding, J., Lubetzky, E., and Peres, Y. Censored Glauber dynamics for the mean field Ising model. *Journal of Statistical Physics*, 2009.

Dinh, V., Bilge, A., Zhang, C., and IV, F. A. M. Probabilistic path Hamiltonian Monte Carlo. In *International Conference on Machine Learning (ICML)*, 2017.

Djolonga, J. and Krause, A. From MAP to marginals: Variational inference in bayesian submodular models. In *Neural Information Processing Systems*, 2014.

Djolonga, J., Jegelka, S., Tschiatschek, S., and Krause, A. Cooperative graphical models. In *Neural Information Processing Systems (NIPS)*, 2016a.

Djolonga, J., Tschiatschek, S., and Krause, A. Variational inference in mixed probabilistic submodular models. In *Neural Information Processing Systems (NIPS)*, 2016b.

Dolhansky, B. W. and Bilmes, J. A. Deep submodular functions: Definitions and learning. In *Neural Information Processing Systems (NIPS)*, 2016.

Domke, J. Learning graphical model parameters with approximate marginal inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2013.

Domke, J. Maximum likelihood learning with arbitrary treewidth via fast-mixing parameter sets. In *Neural Information Processing Systems (NIPS)*, 2015.

Dyer, M. and Greenhill, C. On markov chains for independent sets. *J. of Algorithms*, 2000.

Dyer, M., Goldberg, L. A., and Jerrum, M. Matrix norms and rapid mixing for spin systems. *Annals of Applied Probability*, 2009.

Feige, U., Mirrokni, V. S., and Vondrák, J. Maximizing non-monotone submodular functions. *SIAM Journal on Computing*, 2011.

Fujishige, S. *Submodular Functions and Optimization*. Elsevier Science, 2005.

Gelfand, A. E. and Dey, D. Bayesian model choice: Asymptotics and exact calculations. *Journal of the Royal Statistical Society*, 1994.

Geyer, C. J. Markov chain monte carlo maximum likelihood. *Symposium on the Interface*, 1991.

Golovin, D. and Krause, A. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research (JAIR)*, 2011.

Gong, B., Chao, W.-L., Grauman, K., and Sha, F. Diverse sequential subset selection for supervised video summarization. In *Neural Information Processing Systems (NIPS)*, 2014.

Gotovos, A., Hassani, H. S., and Krause, A. Sampling from probabilistic submodular models. In *Neural Information Processing Systems (NIPS)*, 2015.

Gotovos, A., Hassani, H., Krause, A., and Jegelka, S. Discrete sampling using semigradient-based product mixtures. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

Guestrin, C., Krause, A., and Singh, A. P. Near-optimal sensor placements in gaussian processes. In *International Conference on Machine Learning (ICML)*, 2005.

Gutmann, M. U. and Hyvärinen, A. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 2012.

Hastings, W. K. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 1970.

Hazan, T., Maji, S., and Jaakkola, T. On sampling from the Gibbs distribution with random maximum a-posteriori perturbations. In *Neural Information Processing Systems (NIPS)*, 2013.

Hubert, L. Comparing partitions. *Journal of Classification*, 1985.

Ising, E. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 1925.

Iyer, R. and Bilmes, J. The submodular Bregman and Lovász-Bregman divergences with applications. In *Neural Information Processing Systems (NIPS)*, 2012.

Iyer, R. and Bilmes, J. Submodular point processes with applications in machine learning. In *International Conference on Artificial Intelligence and Statistics*, 2015.

Iyer, R., Jegelka, S., and Bilmes, J. Fast semidifferential-based submodular function optimization. In *International Conference on Machine Learning (ICML)*, 2013.

Javanmard, A. and Montanari, A. Online rules for control of false discovery rate and false discovery exceedance. *The Annals of Statistics*, 2018.

Jegelka, S. and Bilmes, J. Submodularity beyond submodular energies: Coupling edges in graph cuts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

Jerrum, M. A very simple algorithm for estimating the number of k-colorings of a low-degree graph. *Random Structures and Algorithms*, 1995.

Jerrum, M. *Counting, Sampling and Integrating: Algorithms and Complexity*. Birkhäuser, 2003.

Jerrum, M. and Sinclair, A. Approximating the permanent. *SIAM Journal on Computing*, 1989.

Jerrum, M. and Sinclair, A. Polynomial-time approximation algorithms for the Ising model. *SIAM Journal on Computing*, 1993.

Jerrum, M., Sinclair, A., and Vigoda, E. A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. *Journal of the ACM*, 2004a.

Jerrum, M., Son, J.-B., Tetali, P., and Vigoda, E. Elementary bounds on Poincaré and log-Sobolev constants for decomposable Markov chains. *Annals of Applied Probability*, 2004b.

Kempe, D., Kleinberg, J., and Tardos, E. Maximizing the spread of influence through a social network. In *Conference on Knowledge Discovery and Data Mining*, 2003.

Koller, D. and Friedman, N. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

Krause, A., Guestrin, C., Gupta, A., and Kleinberg, J. Near-optimal sensor placements: Maximizing information while minimizing communication cost. In *Information Processing in Sensor Networks*, 2006.

Krause, A., Leskovec, J., Guestrin, C., Vanbriesen, J., and Faloutsos, C. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management*, 2008.

Kulesza, A. and Taskar, B. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 2012.

Lee, J., Mirrokni, V. S., Nagarajan, V., and Sviridenko, M. Non-monotone submodular maximization under matroid and knapsack constraints. In *ACM symposium on Symposium on theory of computing (STOC)*, 2009.

Leiserson, M. D., Wu, H.-T., Vandin, F., and Raphael, B. J. CoMEt: A statistical approach to identify combinations of mutually exclusive alterations in cancer. *Genome Biology*, 2015.

Leiserson, M. D. M., Blokh, D., Sharan, R., and Raphael, B. J. Simultaneous identification of multiple driver pathways in cancer. *PLoS Computational Biology*, 2013.

Levin, D. A., Luczak, M. J., and Peres, Y. Glauber dynamics for the mean-field Ising model: cut-off, critical power law, and metastability. *Probability Theory and Related Fields*, 2008a.

Levin, D. A., Peres, Y., and Wilmer, E. L. *Markov Chains and Mixing Times*. American Mathematical Society, 2008b.

Li, C., Jegelka, S., and Sra, S. Fast mixing markov chains for strongly Rayleigh measures, DPPs, and constrained sampling. In *Neural Information Processing Systems (NIPS)*, 2016.

Li, C., Jegelka, S., and Sra, S. Polynomial time algorithms for dual volume sampling. In *Neural Information Processing Systems (NIPS)*, 2017.

Liggett, T. M. Negative correlations and particle systems. *Markov Processes and Related Fields*, 2002.

Lin, H. and Bilmes, J. A class of submodular functions for document summarization. In *Human Language Technologies*, 2011.

Liu, Q., Peng, J., Ihler, A., and III, J. F. Estimating the partition function by discriminance sampling. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015.

Lyons, R. Determinantal probability measures. *Publications mathématiques de l'IHÉS*, 2003.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 1953.

Miller, C. A., Settle, S. H., Sulman, E. P., Aldape, K. D., and Milosavljevic, A. Discovering functional modules by identifying recurrent and mutually exclusive mutational patterns in tumors. *BMC Medical Genomics*, 2011.

Neal, R. Annealed importance sampling. *Statistics and Computing*, 2001.

Neal, R. M. MCMC using Hamiltonian dynamics. *arXiv*, 2012.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 1978.

Nishimura, A., Dunson, D., and Lu, J. Discontinuous Hamiltonian Monte Carlo for models with discrete parameters and discontinuous likelihoods. *arXiv*, 2018.

Pakman, A. and Paninski, L. Auxiliary-variable exact Hamiltonian Monte Carlo samplers for binary distributions. In *Neural Information Processing Systems (NIPS)*, 2013.

Papandreou, G. and Yuille, A. L. Perturb-and-MAP random fields: Using discrete optimization to learn and sample from energy models. In *International Converence on Computer Vision (ICCV)*, 2011.

Pemantle, R. Towards a theory of negative dependence. *Journal of Mathematical Physics*, 2000.

Potts, R. B. Some generalized order-disorder transformations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 1952.

Ramdas, A., Yang, F., Wainwright, M. J., and Jordan, M. I. Online control of the false discovery rate with decaying memory. In *Neural Information Processing Systems (NIPS)*, 2017.

Raphael, B. J., Dobson, J. R., Oesper, L., and Vandin, F. Identifying driver mutations in sequenced cancer genomes: computational approaches to enable precision medicine. *Genome Medicine*, 2014.

Rebeschini, P. and Karbasi, A. Fast mixing for discrete point processes. In *Conference on Learning Theory*, 2015.

Sinclair, A. Improved bounds for mixing rates of markov chains and multicommodity flow. *Combinatorics, Probability and Computing*, 1992.

Sminchisescu, C. and Welling, M. Generalized darting monte carlo. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.

Szczurek, E. and Beerenwinkel, N. Modeling mutual exclusivity of cancer mutations. *PLoS Computational Biology*, 2014.

TCGA. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 2008.

TCGA. Comprehensive molecular portraits of human breast tumours. *Nature*, 2012.

TCGA. Genomic and epigenomic landscapes of adult de novo acute myeloid leukemia. *The New England Journal of Medicine*, 2013.

Tschiatschek, S., Iyer, R., Wei, H., and Bilmes, J. Learning mixtures of submodular functions for image collection summarization. In *Neural Information Processing Systems (NIPS)*, 2014.

Tschiatschek, S., Djolonga, J., and Krause, A. Learning probabilistic submodular diversity models via noise contrastive estimation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

Vandin, F., Upfal, E., and Raphael, B. J. De novo discovery of mutated driver pathways in cancer. *Genome Research*, 2011.

Wagner, D. G. Negatively correlated random variables and mason's conjecture for independent sets in matroids. *Annals of Combinatorics*, 2008.

Wainwright, M. J. and Jordan, M. I. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 2008.

Wang, C., Komodakis, N., and Paragios, N. Markov random field modeling, inference & learning in computer vision & image understanding: A survey. *Computer Vision and Image Understanding*, 2013.

Yeang, C.-H., McCormick, F., and Levine, A. Combinatorial patterns of somatic gene mutations in cancer. *The FASEB Journal*, 2008.

Zhang, Y., Sutton, C., Storkey, A., and Ghahramani, Z. Continuous relaxations for discrete Hamiltonian Monte Carlo. In *Neural Information Processing Systems (NIPS)*, 2012.

Zhu, M. H. and Ermon, S. A hybrid approach for probabilistic inference using random projections. In *International Conference on Machine Learning (ICML)*, 2015.