Probabilistic Graphical Models

David Sontag

New York University

Lecture 7, March 14, 2012

Approximate marginal inference

- Given the joint $p(x_1, \ldots, x_n)$ represented as a graphical model, how do we perform **marginal inference**, e.g. to compute $p(x_1 \mid e)$?
- We showed in Lecture 4 that doing this exactly is NP-hard
- Nearly all *approximate inference* algorithms are either:
  1. Monte-carlo methods (e.g., likelihood reweighting, MCMC)
  2. Variational algorithms (e.g., mean-field, TRW, loopy belief propagation)
- These next two lectures will be on variational methods

Variational methods

- **Goal**: Approximate difficult distribution $p(\mathbf{x} \mid \mathbf{e})$ with a new distribution $q(\mathbf{x})$ such that:
  1. $p(\mathbf{x} \mid \mathbf{e})$ and $q(\mathbf{x})$ are "close"
  2. Computation on $q(\mathbf{x})$ is easy
- How should we measure distance between distributions?
- The **Kullback-Leibler divergence** (KL-divergence) between two distributions p and q is defined as

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

  (measures the expected number of extra bits required to describe *samples from* $p(\mathbf{x})$ using a code based on q instead of p)
- $D(p \,\|\, q) \geq 0$ for all p, q, with equality if and only if $p = q$
- Notice that KL-divergence is **asymmetric**

KL-divergence *(see Section 8.5 of K&F)*

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

- Suppose p is the true distribution we wish to do inference with
- What is the difference between the solution to

  $$\arg\min_q D(p\|q)$$

  (called the *M-projection* of q onto p) and

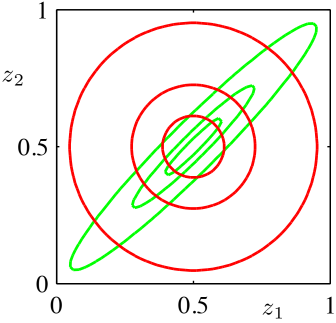
  $$\arg\min_q D(q\|p)$$

  (called the *I-projection*)?
- These two will differ only when q is minimized over a restricted set of probability distributions $Q = \{q_1, \dots\}$, and in particular when $p \notin Q$

KL-divergence – M-projection

$$q^* = \arg\min_{q \in Q} D(p \| q) = \sum_{\mathbf{x}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})}.$$

For example, suppose that $p(\mathbf{z})$ is a 2D Gaussian and Q is the set of all Gaussian distributions with diagonal covariance matrices:

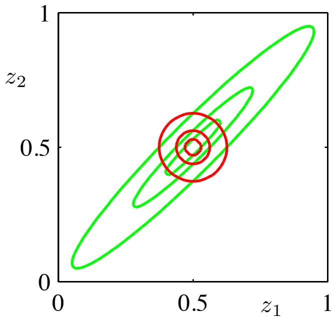


p=Green, q^*=Red

KL-divergence – I-projection

$$q^* = \arg\min_{q \in Q} D(q\|p) = \sum_{\mathbf{x}} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})}.$$

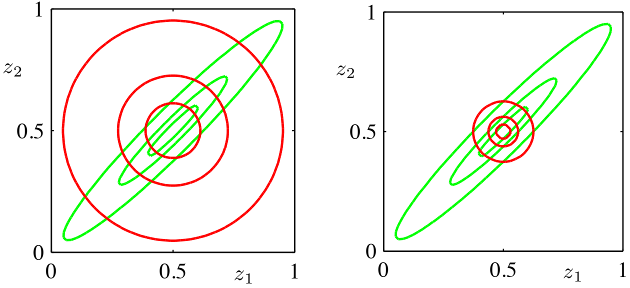
For example, suppose that $p(\mathbf{z})$ is a 2D Gaussian and Q is the set of all Gaussian distributions with diagonal covariance matrices:



p=Green, q^*=Red

KL-divergence (single Gaussian)

In this simple example, both the M-projection and I-projection find an approximate $q(\mathbf{x})$ that has the correct mean (i.e. $E_p[\mathbf{z}] = E_q[\mathbf{z}]$):

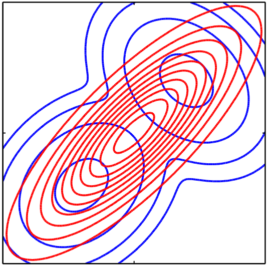


What if $p(\mathbf{x})$ is multi-modal?

KL-divergence – M-projection (mixture of Gaussians)

$$q^* = \arg\min_{q \in Q} D(p\|q) = \sum_{\mathbf{x}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})}.$$

Now suppose that $p(\mathbf{x})$ is mixture of two 2D Gaussians and Q is the set of all 2D Gaussian distributions (with arbitrary covariance matrices):

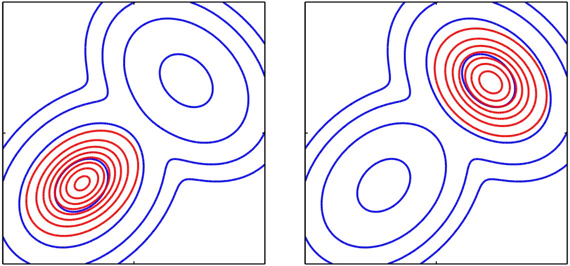


p=Blue, q^*=Red

M-projection yields distribution $q(\mathbf{x})$ with the correct mean and covariance.

KL-divergence – I-projection (mixture of Gaussians)

$$q^* = \arg\min_{q \in Q} D(q \| p) = \sum_{\mathbf{x}} q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})}.$$



p=Blue, q^*=Red (two equivalently good solutions!)

Unlike the M-projection, the I-projection does not necessarily yield the correct moments.

Mapping of distributions to/from moments

- Recall the definition of probability distributions in the exponential family:
$$q(\mathbf{x};\eta) = h(\mathbf{x})\exp\{\eta\cdot\mathbf{f}(\mathbf{x}) - \ln Z(\eta)\}$$

  $\mathbf{f}(\mathbf{x})$ are called the *sufficient statistics*
- In the exponential family, there is a one-to-one correspondance between distributions $q(\mathbf{x};\eta)$ and marginal vectors $E_q[\mathbf{f}(\mathbf{x})]$
- For example, when q is a Gaussian distribution,
$$q(\mathbf{x};\mu,\Sigma) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu)\right)$$

  then $\mathbf{f}(\mathbf{x}) = [x_1, x_2, \ldots, x_k, x_1^2, x_1x_2, x_1x_3, \ldots, x_2^2, x_2x_3, \ldots]$
- The expectation of $\mathbf{f}(\mathbf{x})$ gives the first and second-order (non-central) moments, from which one can solve for μ and Σ

Finding the M-projection is the same as exact inference

M-projection is:

$$q^* = \arg\min_{q \in Q} D(p \| q) = \sum_{\mathbf{x}} p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})}.$$

- Suppose that Q is an exponential family ($p(\mathbf{x})$ can be arbitrary) and that we could perform the M-projection, finding q^*
- It can be shown (see Thm 8.6) that the expected sufficient statistics, with respect to $q^*(\mathbf{x})$, are *exactly* the marginals of $p(\mathbf{x})$:

$$E_{q^*}[\mathbf{f}(\mathbf{x})] = E_p[\mathbf{f}(\mathbf{x})]$$

- Thus, solving for the M-projection is just as hard as the original inference problem

Most variational inference algorithms make use of the I-projection

Variational methods

- Suppose that we have an arbitrary graphical model:

$$p(\mathbf{x};\theta) = \frac{1}{Z(\theta)} \prod_{\mathbf{c}\in C} \phi_c(\mathbf{x_c}) = \exp\Big(\sum_{\mathbf{c}\in C} \theta_c(\mathbf{x_c}) - \ln Z(\theta) \Big)$$

- All of the approaches begin as follows:

$$\begin{aligned}
D(q\|p) &= \sum_{\mathbf{x}} q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{x})} \\
&= -\sum_{\mathbf{x}} q(\mathbf{x}) \ln p(\mathbf{x}) - \sum_{\mathbf{x}} q(\mathbf{x}) \ln \frac{1}{q(\mathbf{x})} \\
&= -\sum_{\mathbf{x}} q(\mathbf{x}) \big(\sum_{\mathbf{c}\in C} \theta_c(\mathbf{x_c}) - \ln Z(\theta) \big) - H(q(\mathbf{x})) \\
&= -\sum_{\mathbf{c}\in C} \sum_{\mathbf{x}} q(\mathbf{x}) \theta_c(\mathbf{x_c}) + \sum_{\mathbf{x}} q(\mathbf{x}) \ln Z(\theta) - H(q(\mathbf{x})) \\
&= -\sum_{\mathbf{c}\in C} E_q[\theta_c(\mathbf{x_c})] + \ln Z(\theta) - H(q(\mathbf{x})).
\end{aligned}$$

The log-partition function

- Since $D(q\|p) \geq 0$, we have

$$-\sum_{\mathbf{c}\in C} E_q[\theta_c(\mathbf{x_c})] + \ln Z(\theta) - H(q(\mathbf{x})) \geq 0,$$

which implies that

$$\ln Z(\theta) \geq \sum_{\mathbf{c}\in C} E_q[\theta_c(\mathbf{x_c})] + H(q(\mathbf{x})).$$

- Thus, *any* approximating distribution $q(\mathbf{x})$ gives a lower bound on the log-partition function (for a BN, this is the probability of the evidence)
- Recall that $D(q\|p) = 0$ if and only if $p = q$.Thus, if we allow ourselves to optimize over *all* distributions, we have:

$$\ln Z(\theta) = \max_q \sum_{\mathbf{c}\in C} E_q[\theta_c(\mathbf{x_c})] + H(q(\mathbf{x})).$$

Two types of variational algorithms: Mean-field and relaxation

$$\max_{q} \sum_{\mathbf{c} \in C} E_q[\theta_c(\mathbf{x_c})] + H(q(\mathbf{x})).$$

- Although this function is concave and thus in theory should be easy to optimize, we need some compact way of representing $q(\mathbf{x})$
- *Mean-field* algorithms assume a factored representation of the joint distribution:
  $$q(\mathbf{x}) = \prod_{i \in V} q_i(x_i)$$
  [topic of next week's lecture]
- *Relaxation* algorithms work directly with *pseudomarginals* which may not be consistent with any joint distribution
  [loopy sum-product BP is an example of this!]

Re-writing objective in terms of moments

$$\ln Z(\theta) = \max_q \sum_{\mathbf{c} \in C} E_q[\theta_c(\mathbf{x_c})] + H(q(\mathbf{x})).$$

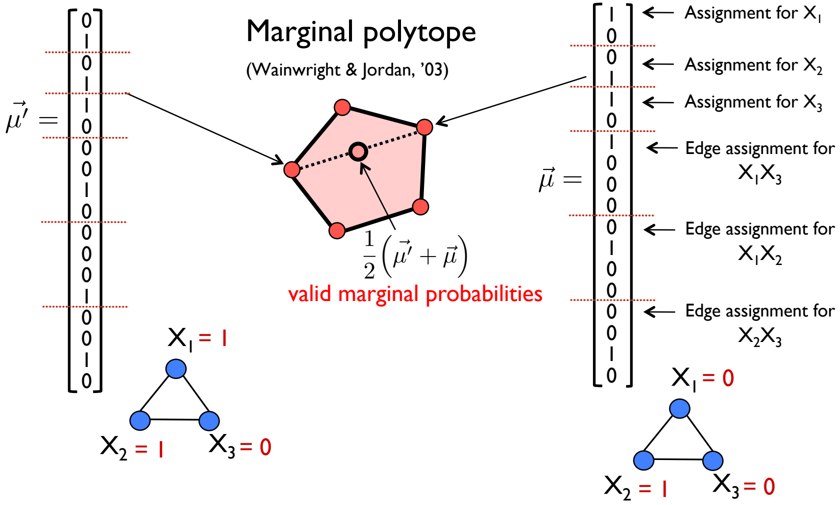
- Assume that $p(\mathbf{x})$ is in the exponential family, and let $\mathbf{f}(\mathbf{x})$ be its sufficient statistic vector
- Let Q be the exponential family with sufficient statistics $\mathbf{f}(\mathbf{x})$
- Define $\mu_q = E_q[\mathbf{f}(\mathbf{x})]$ be the *marginals* of $q(\mathbf{x})$
- We can re-write the objective as

$$\ln Z(\theta) = \max_q \sum_{\mathbf{c} \in C} \sum_{\mathbf{x_c}} \theta_c(\mathbf{x_c}) \mu_q^c(\mathbf{x_c}) + H(\mu_q),$$

  where we define $H(\mu_q)$ to be the entropy of the *maximum entropy distribution* with marginals μ_q
- Next, instead of optimizing over distributions $q(\mathbf{x})$, optimize over valid marginal vectors μ. We obtain:

$$\ln Z(\theta) = \max_{\mu \in M} \sum_{\mathbf{c} \in C} \sum_{\mathbf{x_c}} \theta_c(\mathbf{x_c}) \mu_c(\mathbf{x_c}) + H(\mu)$$

Marginal polytope (same as from Lecture 6)



Relaxation

$$\ln Z(\theta) = \max_{\mu \in M} \sum_{\mathbf{c} \in C} \sum_{\mathbf{x_c}} \theta_c(\mathbf{x_c}) \mu_c(\mathbf{x_c}) + H(\mu)$$

- We still haven't achieved anything, because:
  1. The marginal polytope M is complex to describe (in general, exponentially many vertices and facets)
  2. $H(\mu)$ is very difficult to compute or optimize over
- We now make two approximations:
  1. We replace M with a *relaxation* of the marginal polytope, e.g. the local consistency constraints M_L
  2. We replace $H(\mu)$ with a function $\tilde{H}(\mu)$ which approximates $H(\mu)$

Local consistency constraints (same as from Lecture 6)

- Force every "cluster" of variables to choose a local assignment:

$$
\begin{aligned}
\mu_i(x_i) &\geq 0 \quad \forall i \in V, x_i \\
\sum_{x_i} \mu_i(x_i) &= 1 \quad \forall i \in V \\
\mu_{ij}(x_i, x_j) &\geq 0 \quad \forall ij \in E, x_i, x_j \\
\sum_{x_i, x_j} \mu_{ij}(x_i, x_j) &= 1 \quad \forall ij \in E
\end{aligned}
$$

- Enforce that these local assignments are globally consistent:

$$
\begin{aligned}
\mu_i(x_i) &= \sum_{x_j} \mu_{ij}(x_i, x_j) \quad \forall ij \in E, x_i \\
\mu_j(x_j) &= \sum_{x_i} \mu_{ij}(x_i, x_j) \quad \forall ij \in E, x_j
\end{aligned}
$$

- The *local consistency polytope*, M_L is defined by these constraints

Entropy of a tree distribution

- Suppose that q is a tree-structured distribution, so that we are optimizing only over marginals $\mu_{ij}(x_i, x_j)$ for $ij \in T$
- The entropy of q as a function of its marginals can be shown to be

$$H(\vec{\mu}) = \sum_{i \in V} H(\mu_i) - \sum_{ij \in T} I(\mu_{ij})$$

where

$$\begin{aligned} H(\mu_i) &= -\sum_{x_i} \mu_i(x_i) \log \mu_i(x_i) \\ I(\mu_{ij}) &= \sum_{x_i, x_j} \mu_{ij}(x_i, x_j) \log \frac{\mu_{ij}(x_i, x_j)}{\mu_i(x_i)\mu_j(x_j)} \end{aligned}$$

- Can we use this for non-tree structured models?

Bethe-free energy approximation

- The Bethe entropy approximation is (for any graph)

$$H_{bethe}(\vec{\mu}) = \sum_{i \in V} H(\mu_i) - \sum_{ij \in E} I(\mu_{ij})$$

- This gives the following variational approximation:

$$\max_{\mu \in M_L} \sum_{\mathbf{c} \in C} \sum_{\mathbf{x_c}} \theta_c(\mathbf{x_c}) \mu_c(\mathbf{x_c}) + H_{bethe}(\vec{\mu})$$

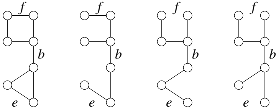
- For non tree-structured models this is not concave, and is hard to maximize
- Loopy belief propagation, if it converges, finds a saddle point!

Concave relaxation

- Let $\tilde{H}(\mu)$ be an *upper bound* on $H(\mu)$, i.e. $H(\mu) \leq \tilde{H}(\mu)$
- As a result, we obtain the following **upper bound** on the log-partition function:

$$\ln Z(\theta) \leq \max_{\mu \in M_L} \sum_{\mathbf{c} \in C} \sum_{\mathbf{x_c}} \theta_c(\mathbf{x_c}) \mu_c(\mathbf{x_c}) + \tilde{H}(\mu)$$

- An example of a **concave** entropy upper bound is the **tree-reweighted** approximation (Jaakkola, Wainwright, & Wilsky, '05), given by specifying a distribution over spanning trees of the graph



Letting $\{\rho_{ij}\}$ denote edge appearance probabilities, we have:

$$H_{TRW}(\vec{\mu}) = \sum_{i \in V} H(\mu_i) - \sum_{ij \in E} \rho_{ij} I(\mu_{ij})$$