Massachusetts Institute of Technology
Department of Electrical Engineering and Computer Science
6.438 Algorithms for Inference
Fall 2014

17 Variational Inference

Prompted by loopy graphs for which exact inference is computationally intractable, we tread into algorithms for approximate inference in search for efficient solutions that may have error. We began with loopy belief propagation, which meant applying the parallel Sum-Product algorithm to loopy graphs with at most pairwise potentials. Our analysis revealed that the problem could be viewed in terms of approximating the log partition function of the distribution of interest.

Today we extend this train of analysis, presenting a general approach that approximates distributions we care about through modifying the hard log partition optimization. Previously, we applied a Bethe approximation to the log partition optimization to recover an optimization problem intricately related to loopy BP. The Bethe approximation amounted to changing the space we're optimizing over to be a simpler set, which slammed down the computational complexity of inference. But we could have easily chosen a different space to optimize over such as some family of probability distributions, which would yield different approximating distributions! This general approach is called *variational inference*, and cleverly selecting which family of distributions to optimize over will offer us lower and upper bounds on the log partition function.

We'll first review the log partition optimization and Bethe approximation before plunging into how to bound the log partition function for distributions with at most pairwise potentials. Each bound will directly have an algorithm that gives an approximating distribution. But variational inference works for distributions with potentials on larger cliques as well! We'll save this for the end, when we'll also briefly inject variational inference with an information-theoretic interpretation.

17.1 Log partition optimization and Bethe approximation

We blaze through some previously stated results while defining a few new variables. Unless otherwise stated, we take distribution $\mathbf{x} \in \mathcal{X}^N$ defined over graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to have factorization

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_{i \in \mathcal{V}} \phi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j) \right), \tag{1}$$

where Z is the partition function.

Denoting $H(\mu) \triangleq -\mathbb{E}_\mu[\log(\mu(\mathbf{x}))] = -\sum_x \mu(x) \log(\mu(x))$ to be the entropy of distribution μ, the hard log partition variational problem is:

$$\log Z = \sup_{\mu \in \mathcal{M}} \left\{ \mathbb{E}_\mu \left[\sum_{i \in \mathcal{V}} \phi_i(\mathsf{x}_i) + \sum_{(i,j) \in \mathcal{E}} \psi_{ij}(\mathsf{x}_i, \mathsf{x}_j) \right] + H(\mu) \right\}, \tag{2}$$

where $\mathcal{M}$ is the set of all distributions over $\mathcal{X}^N$.

Applying the Bethe approximation amounted to saying that rather than optimizing over $\mu \in \mathcal{M}$, we'll optimize over μ with factorization

$$\mu(\mathbf{x}) = \prod_{i \in \mathcal{V}} \mu_i(x_i) \prod_{(i,j) \in \mathcal{E}} \frac{\mu_{ij}(x_i, x_j)}{\mu_i(x_i)\mu_j(x_j)}, \tag{3}$$

where μ_i's and μ_{ij}'s are constrained to behave like node and edge marginals. Extremely important is the fact that $\mathcal{E}$ in the factorization is precisely the edge set in the graph for $p_{\mathbf{x}}$, which may be loopy! Adding the above factorization constraint on μ to the log partition variational problem and scrawling a few lines of algebra, we're left with the Bethe variational problem below.

$$\log Z_{\text{bethe}} = \sup_{\mu} \left\{ \sum_{i \in \mathcal{V}} \mathbb{E}_{\mu_i}[\phi_i(x_i)] + \sum_{(i,j) \in \mathcal{E}} \mathbb{E}_{\mu_{ij}}[\psi_{ij}(x_i, x_j)] + \sum_{i \in \mathcal{V}} H(\mu_i) + \sum_{(i,j) \in \mathcal{E}} H(\mu_{ij}). \right\} \tag{4}$$

subject to:

$$\begin{aligned}
\mu_i(x_i) &\geq 0 && \text{for all } i \in \mathcal{V}, x_i \in \mathcal{X} \\
\sum_{x_i \in \mathcal{X}} \mu_i(x_i) &= 1 && \text{for all } i \in \mathcal{V} \\
\mu_{ij}(x_i, x_j) &\geq 0 && \text{for all } (i,j) \in \mathcal{E}, x_i, x_j \in \mathcal{X} \\
\sum_{x_j \in \mathcal{X}} \mu_{ij}(x_i, x_j) &= \mu_i(x_i) && \text{for all } (i,j) \in \mathcal{E}, x_i \in \mathcal{X} \\
\sum_{x_i \in \mathcal{X}} \mu_{ij}(x_i, x_j) &= \mu_j(x_j) && \text{for all } (i,j) \in \mathcal{E}, x_j \in \mathcal{X}
\end{aligned}$$

We get out approximate node marginals μ_i's and approximate edge marginals μ_{ij}'s. Also, we previously argued that if $p_{\mathbf{x}}$ has a tree graph, then $\log Z = \log Z_{\text{bethe}}$. However, if $p_{\mathbf{x}}$ has a loopy graph, then $\log Z$ and $\log Z_{\text{bethe}}$ may differ, and how much these differ is messy but can be described by loop series expansions.??? Unfortunately, it's unclear whether $\log Z_{\text{bethe}}$ is a lower or upper bound for $\log Z$, as we'll discuss later.

But why is bounding the log partition function interesting? Recall that the marginal over a subset $\mathcal{S} \subset \mathcal{V}$ can be computed by

$$p_{\mathbf{x}_\mathcal{S}}(\mathbf{x}_\mathcal{S}) = \frac{\sum_{\mathbf{x}_{\setminus \mathcal{S}}} \left(\sum_{i \in \mathcal{V}} \phi_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \psi_{ij}(x_i, x_j) \right)}{Z} = \frac{Z(\mathbf{x}_\mathcal{S} = \mathbf{x}_\mathcal{S})}{Z},$$

where $\mathbf{x}_{\setminus \mathcal{S}}$ denotes all x_i variables not in $\mathbf{x}_\mathcal{S}$, and $Z(\mathbf{x}_\mathcal{S} = \mathbf{x}_\mathcal{S})$ is the partition function evaluated when $\mathbf{x}_\mathcal{S}$ is fixed to have value $\mathbf{x}_\mathcal{S}$. If we can bound log partition functions, then we can bound partition functions. Thus, finding a lower bound on $Z(\mathbf{x}_\mathcal{S} = \mathbf{x}_\mathcal{S})$ and an upper bound on Z would give us a lower bound for $p_{\mathbf{x}_\mathcal{S}}(\mathbf{x}_\mathcal{S})$. Meanwhile, finding an upper bound on $Z(\mathbf{x}_\mathcal{S} = \mathbf{x}_\mathcal{S})$ and a lower bound on Z would give an upper bound for $p_{\mathbf{x}_\mathcal{S}}(\mathbf{x}_\mathcal{S})$. So it's possible to sandwich $p_{\mathbf{x}_\mathcal{S}}(\mathbf{x}_\mathcal{S})$ into an interval!

17.2 Lower bound using mean field

Imagine if we're asked to optimize over 10 items, but we're too lazy or it's too expensive to check all 10. So we stop after checking five of them and return the optimal solution so far. Surely our solution provides a lower bound for what the best solution is since if we checked the rest of the items, our solution will either stay the same or improve. This idea is precisely what we'll use to secure a lower bound on $\log Z$: rather than optimizing over all distributions in $\mathcal{M}$, we'll optimize over a subset of $\mathcal{M}$. Our solution will thus yield a lower bound on the log partition function.

The simplest family of distributions that is guaranteed to be a subset of $\mathcal{M}$ is the set of distributions that fully factorize as singleton factors:

$$\mu(\mathbf{x}) = \prod_{i \in \mathcal{V}} \mu_i(x_i).$$

This factorization is called the *mean field factorization*, and the family of distributions with such a factorization will be denoted $\mathcal{M}_{\text{MF}}$. At a first glance, mean field might seem too simplistic as there is no neighbor structure; all μ_i's are independent! But as we'll see, by optimizing over $\mu \in \mathcal{M}_{\text{MF}}$, the solution will involve looking at neighbors in the original graph. Furthermore, in literature, the mean field factorization is far and away the most popular way to do variational inference.

By looking at the original log partition variational problem and plugging in the mean field factorization constraint on μ, a few lines of algebra show that the mean field variational inference problem is as follows.

$$\log Z_{\text{MF}} = \max_{\mu} \left\{ \sum_{i \in \mathcal{V}} \mathbb{E}_{\mu_i}[\phi_i(x_i)] + \sum_{(i,j) \in \mathcal{E}} \mathbb{E}_{\mu_i \mu_j}[\psi_{ij}(x_i, x_j)] + \sum_{i \in \mathcal{V}} H(\mu_i) \right\} \tag{5}$$

subject to:

$$\begin{aligned} \mu_i(x_i) &\geq 0 && \text{for all } i \in \mathcal{V}, x_i \in \mathcal{X} \\ \sum_{x_i \in \mathcal{X}} \mu_i(x_i) &= 1 && \text{for all } i \in \mathcal{V} \end{aligned}$$

The μ_i's can be viewed as approximate node marginals of $p_{\mathbf{x}}$. As we've already justified, we are guaranteed that $\log Z_{\text{MF}} \leq \log Z$.

Let's actually optimize for μ by slapping in some Lagrange multipliers and setting derivatives to 0. As with the Bethe variational problem, we need not introduce Lagrange multipliers for the nonnegativity constraints; we'll find that without explicitly enforcing nonnegativity, our solution will have nonnegative $\mu_i(x_i)$'s anyways. Thus, for each i, we introduce Lagrange multiplier λ_i for constraint $\sum_{x_i \in \mathcal{X}} \mu_i(x_i) = 1$. The

Lagrangian is given by:

$$\begin{aligned}\mathcal{L}(\mu,\lambda) &= \sum_{i\in\mathcal{V}} \mathbb{E}_{\mu_i}[\phi_i(\mathsf{x}_i)] + \sum_{(i,j)\in\mathcal{E}} \mathbb{E}_{\mu_i\mu_j}[\psi_{ij}(x_i,x_j)] + \sum_{i\in\mathcal{V}} H(\mu_i) + \sum_{i\in\mathcal{V}} \lambda_i \left(\sum_{x_i\in\mathcal{X}} \mu_i(x_i) - 1\right) \\ &= \sum_{i\in\mathcal{V}} \sum_{x_i\in\mathcal{X}} \mu_i(x_i)\phi_i(\mathsf{x}_i) + \sum_{(i,j)\in\mathcal{E}} \sum_{x_i,x_j\in\mathcal{X}} \mu_i(x_i)\mu_j(x_j)\psi_{ij}(x_i,x_j) \\ &\quad - \sum_{i\in\mathcal{V}} \sum_{x_i\in\mathcal{X}} \mu_i(x_i)\log\mu_i(x_i) + \sum_{i\in\mathcal{V}} \lambda_i \left(\sum_{x_i\in\mathcal{X}} \mu_i(x_i) - 1\right).\end{aligned}$$

Hence,

$$\frac{\partial\mathcal{L}(\mu,\lambda)}{\partial\mu_i(x_i)} = \phi_i(x_i) + \sum_{j\in N(i)} \mu_j(x_j)\psi_{ij}(x_i,x_j) - (\log\mu_i(x_i)+1) + \lambda_i.$$

Setting this to zero, we obtain

$$\mu_i(x_i) \propto \exp\left\{\phi_i(x_i) + \sum_{j\in N(i)} \mu_j(x_j)\psi(x_i,x_j)\right\}.$$

As advertised earlier, the solution at node i involves its neighbors. To actually compute the approximate marginals μ_i's now, we typically will need to do an iterative update, such as:

$$\mu_i^{t+1}(x_i) \propto \exp\left\{\phi_i(x_i) + \sum_{j\in N(i)} \mu_j^t(x_j)\psi(x_i,x_j)\right\},$$

where t indexes iteration numbers. If we manage to procure an optimal μ, then plugging it back into the objective function yields $\log Z_{\mathrm{MF}}$. Analyzing convergence in general requires work, similar to analyzing loopy BP convergence. For example, if $p_{\mathbf{x}}$ has structure that makes the mean field variation problem (5) a convex optimization problem, then our iterative updates will converge to the globally optimal solution.

17.3 Other lower bounds

Because $\mathcal{M}_{\mathrm{MF}} \subset \mathcal{M}$, restricting the log partition variational problem to optimize over $\mathcal{M}_{\mathrm{MF}}$ instead of $\mathcal{M}$ results in a lower bound for $\log Z$. Of course, optimizing over any subset of $\mathcal{M}$ yields a lower bound. For example, in principle we could optimize over $\mathcal{M}_{\mathrm{tree}}$, the space of all tree distributions over nodes $\mathcal{V}$. This would be overkill as there are N^{N-2} such trees. We could instead restrict our attention to one specific tree τ that connects all of $\mathcal{V}$. Then we could optimize over $\mathcal{M}_{\mathrm{tree}(\tau)}$, defined to be the family

of distributions with factorization (3) except where the edges are from tree τ. Unlike Bethe approximation, since τ is a tree, we are never optimizing over a loopy graph. A hierarchical diagram relating these families is in Figure 1.

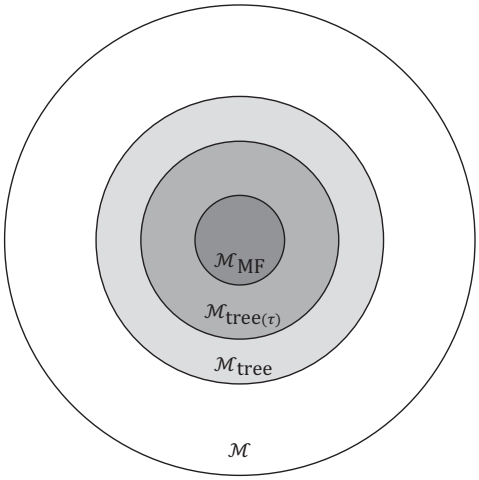


Figure 1: Hierarchy of a few probability family distributions.

Denoting $\log Z_{\text{tree}}$ and $\log Z_{\text{tree}(\tau)}$ to be the outputs of the log partition variational problem restricted to $\mathcal{M}_{\text{tree}}$ and $\mathcal{M}_{\text{tree}(\tau)}$ respectively, then our earlier argument and the hierarchy diagram indicate that:

$$\log Z_{\text{MF}} \leq \log Z_{\text{tree}(\tau)} \leq \log Z_{\text{tree}} \leq \log Z.$$

Of course, other subsets of $\mathcal{M}$ can be chosen; we're just giving the above families as examples as they're easy to describe.

Lastly, we mention that the family of "distributions" $\mathcal{M}_{\text{bethe}(\mathcal{G})}$ corresponding to "distributions" with factorization in (3), where specifically the nodes and edges are from graph $\mathcal{G}$, is *not necessarily a subset of* $\mathcal{M}$ and therefore may include members that, strictly speaking, don't correspond to any valid distribution over $\mathcal{X}^N$. To see this, consider a 3-node fully-connected graph $\mathcal{G}$. Then members of $\mathcal{M}_{\text{bethe}(\mathcal{G})}$ have factorization

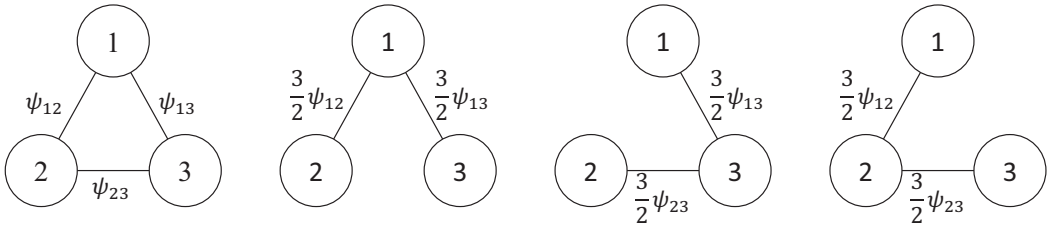
$$\begin{aligned}\mu(\mathbf{x}) &= \mu_1(x_1)\mu_2(x_2)\mu_3(x_3)\frac{\mu_{12}(x_1,x_2)}{\mu_1(x_1)\mu_2(x_2)}\frac{\mu_{23}(x_2,x_3)}{\mu_2(x_2)\mu_3(x_3)}\frac{\mu_{13}(x_1,x_3)}{\mu_1(x_1)\mu_3(x_3)} \\ &= \frac{\mu_{12}(x_1,x_2)}{\mu_1(x_1)}\frac{\mu_{23}(x_2,x_3)}{\mu_2(x_2)}\frac{\mu_{13}(x_1,x_3)}{\mu_3(x_3)},\end{aligned}$$

which, invoking the definition of conditional probability, would mean that μ has a cyclic directed graphical model! So even though we're guaranteed that μ_i's and μ_{ij}'s are valid distributions, a solution to the Bethe variational problem may have the joint "distribution" μ not correspond to any consistent distribution over $\mathcal{X}^N$. This explains why $\log Z_{\text{bethe}}$ may not be a lower bound for $\log Z$.

17.4 Upper bound using tree-reweighted belief propagation

Providing an upper bound for the log partition function turns out to be less straightforward. One way to do this is via what's called tree-reweighted belief propagation, which looks at convex combinations[1] of trees. We'll only sketch the method here, deferring details to Wainwright, Jaakkola, and Willsky's paper.[2]

Suppose we want an upper bound on the log partition function for $p_{\mathbf{x}}$ defined over the graph in Figure 2a with only edge potentials. We'll instead look at its spanning trees where we cleverly assign new edge potentials as shown in Figures 2b, 2c, and 2d. Here, each tree is given a weight of 1/3 in the convex combination.



(a) Original graph (b) Tree τ_1 (c) Tree τ_2 (d) Tree τ_3

Figure 2: A loopy graph and all its spanning trees τ_1, τ_2, and τ_3. Edge potentials are shown next to each edge.

Where do the edge potentials in the spanning trees come from? We'll illustrate the basic idea by explaining how the potentials for edge (1,2) across the spanning trees are defined. The same idea works for defining potentials on the other edges of the spanning trees. Observe that

$$\psi_{12} = \underbrace{\frac{1}{3}}_{\substack{\text{weight of} \\ \text{tree } \tau_1}} \underbrace{\left(\frac{3}{2}\psi_{12}\right)}_{\substack{\text{edge (1,2)'s} \\ \text{potential in} \\ \text{tree } \tau_1}} + \underbrace{\frac{1}{3}}_{\substack{\text{weight of} \\ \text{tree } \tau_2}} \underbrace{(0 \cdot \psi_{12})}_{\substack{\text{edge (1,2)'s} \\ \text{potential in} \\ \text{tree } \tau_2}} + \underbrace{\frac{1}{3}}_{\substack{\text{weight of} \\ \text{tree } \tau_3}} \underbrace{\left(\frac{3}{2}\psi_{12}\right)}_{\substack{\text{edge (1,2)'s} \\ \text{potential in} \\ \text{tree } \tau_3}}.$$

The next critical observation will involve optimizing over the family of distributions $\mathcal{M}_{\text{tree}(\tau_k)}$ defined in the previous section, where $\tau_k = (\mathcal{V}, \mathcal{E}_k)$ is now one of our spanning trees. We shall denote ψ_{ij}^k to be the potential for edge $(i,j) \in \mathcal{E}$ in tree τ_k, where if the edge isn't present in tree τ_k, then the potential is just the 0 function. As a reminder, we are assuming that we only have edge potentials, which is fine since if we had singleton potentials, these could always be folded into edge potentials. We

[1] A convex combination of n items $x_1, \ldots, x_n$ is like a linear combination except that the weights must be nonnegative and sum to 1, e.g., $\sum_{i=1}^n \alpha_i x_i$ is a convex combination of $x_1, \ldots, x_n$ if all α_i's are nonnegative and $\sum_{i=1}^n \alpha_i = 1$.

[2] See M.J. Wainwright, T.S. Jaakkola, and A.S. Willsky's "A New Class of Upper Bounds on the Log Partition Function" (2005).

arrive at the following pivotal equation:

$$\begin{aligned}
\mathbb{E}_\mu\left[\sum_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j)\right] &= \mathbb{E}_\mu\left[\sum_{(i,j)\in\mathcal{E}} \frac{1}{3}\sum_{k=1}^{3} \psi_{ij}^k(x_i, x_j)\right] \\
&= \frac{1}{3}\sum_{k=1}^{3} \mathbb{E}_\mu\left[\sum_{(i,j)\in\mathcal{E}} \psi_{ij}^k(x_i, x_j)\right] \\
&= \frac{1}{3}\sum_{k=1}^{3} \mathbb{E}_\mu\left[\sum_{(i,j)\in\mathcal{E}_k} \psi_{ij}^k(x_i, x_j)\right]. \qquad (6)
\end{aligned}$$

We are now ready to upper-bound the partition function:

$$\begin{aligned}
\log Z &= \sup_{\mu\in\mathcal{M}}\left\{\mathbb{E}_\mu\left[\sum_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j)\right] + H(\mu)\right\} \\
&= \sup_{\mu\in\mathcal{M}}\left\{\frac{1}{3}\sum_{k=1}^{3} \mathbb{E}_\mu\left[\sum_{(i,j)\in\mathcal{E}_k} \psi_{ij}^k(x_i, x_j)\right] + H(\mu)\right\} \qquad \text{(using equation (6))} \\
&= \sup_{\mu\in\mathcal{M}}\left\{\frac{1}{3}\sum_{k=1}^{3}\left\{\mathbb{E}_\mu\left[\sum_{(i,j)\in\mathcal{E}_k} \psi_{ij}^k(x_i, x_j)\right] + H(\mu)\right\}\right\} \\
&\leq \frac{1}{3}\sum_{k=1}^{3} \sup_{\mu\in\mathcal{M}}\left\{\mathbb{E}_\mu\left[\sum_{(i,j)\in\mathcal{E}_k} \psi_{ij}^k(x_i, x_j)\right] + H(\mu)\right\},
\end{aligned}$$

where the last step uses the fact that for any functions f_1, f_2 defined over the same domain and range, we have:

$$\sup_x \{f_1(x) + f_2(x)\} \leq \left(\sup_x f_1(x)\right) + \left(\sup_x f_2(x)\right).$$

Note that for each spanning tree τ_k,

$$\sup_{\mu\in\mathcal{M}}\left\{\mathbb{E}_\mu\left[\sum_{(i,j)\in\mathcal{E}_k} \psi_{ij}^k(x_i, x_j)\right] + H(\mu)\right\} = \sup_{\mu\in\mathcal{M}_{\text{tree}(\tau_k)}}\left\{\mathbb{E}_\mu\left[\sum_{(i,j)\in\mathcal{E}_k} \psi_{ij}^k(x_i, x_j)\right] + H(\mu)\right\},$$

since the underlying distribution is actually just tree τ_k so it suffices to optimize over family $\mathcal{M}_{\text{tree}(\tau_k)}$. This optimization is a specific case of when the Bethe variational problem is exact, so we can solve it using belief propagation!

Of course, we've only worked out a simple example for which we could have chosen different weights for our convex combination, not just one where the weights

are all equal. Furthermore, we didn't give an algorithm that actually produces an approximating distribution! Worse yet, for larger graphs still defined over pairwise potentials, the number of spanning trees explodes and optimizing the choice of weights to get the tightest upper bound requires care. Luckily, all of these loose ends are resolved in the paper by Wainwright *et al.* We encourage those who are interested to peruse the paper for the gory details.

17.5 Larger cliques and information-theoretic interpretation

We now consider if $p_{\mathbf{x}}$ still has graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ but now has factorization

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C) = \frac{1}{Z} \exp \left\{ \sum_{C \in \mathcal{C}} \log \psi_C(x_C) \right\}, \tag{7}$$

where $\mathcal{C}$ is the set of maximal cliques, which may be larger than just pairwise cliques. The log partition variational problem in this case is

$$\log Z = \sup_{\mu \in \mathcal{M}} \left\{ \mathbb{E}_\mu \left[\sum_{C \in \mathcal{C}} \log \psi_C(\mathsf{x}_C) \right] + H(\mu) \right\}. \tag{8}$$

As a sanity check, we repeat a calculation from the loopy BP lecture to ensure that the right-hand side optimization does yield $\log Z$. First note that

$$\sum_{C \in \mathcal{C}} \log \psi_C(x_C) = \log Z + \log p_{\mathbf{x}}(\mathbf{x}).$$

Then

$$\begin{aligned} \mathbb{E}_\mu \left[\sum_{C \in \mathcal{C}} \log \psi_C(\mathsf{x}_C) \right] + H(\mu) &= \mathbb{E}_\mu \left[\sum_{C \in \mathcal{C}} \log \psi_C(\mathsf{x}_C) - \log \mu(\mathbf{x}) \right] \\ &= \mathbb{E}_\mu \left[\log Z + \log p_{\mathbf{x}}(\mathbf{x}) - \log \mu(\mathbf{x}) \right] \\ &= \log Z - \mathbb{E}_\mu \left[\log \frac{\mu(\mathbf{x})}{p_{\mathbf{x}}(\mathbf{x})} \right] \\ &= \log Z - D(\mu \parallel p_{\mathbf{x}}), \end{aligned} \tag{9}$$

where $D(p \parallel q)$ is the KL divergence between distributions p and q over the same alphabet. Since $D(p \parallel q) \geq 0$ and with equality if and only if $p \equiv q$, plugging this inequality into (9) gives

$$\mathbb{E}_\mu \left[\sum_{C \in \mathcal{C}} \log \psi_C(\mathsf{x}_C) \right] \leq \log Z,$$

where equality is achieved by setting $\mu \equiv p_{\mathbf{x}}$. Thus, indeed the right-hand side maximization of (8) yields $\log Z$. Note that our earlier discussion on lower-bounding the log partition function easily carries over to this more general case. However, the upper bound requires some heavier machinery, which we'll skip.

The last line of (9) says that maximizing the objective of (8) over $\mu \in \mathcal{M}$ is tantamount to minimizing $D(\mu \parallel p_{\mathbf{x}})$, so

$$\operatorname*{argmin}_{\mu \in \mathcal{M}} D(\mu \parallel p_{\mathbf{x}}) = \operatorname*{argmax}_{\mu \in \mathcal{M}} \left\{ \mathbb{E}_{\mu} \left[\sum_{C \in \mathcal{C}} \log \psi_C(\mathsf{x}_C) \right] + H(\mu) \right\}.$$

This has a nice information-theoretic implication: constraining which family of distributions we optimize over can be viewed in terms of KL divergence. For example, if we just want the approximating mean field distribution and don't care for the value of $\log Z_{\text{MF}}$, then we solve

$$\operatorname*{argmin}_{\mu \in \mathcal{M}_{\text{MF}}} D(\mu \parallel p_{\mathbf{x}}). \tag{10}$$

Thus, variational inference can be viewed as finding a member within a family of approximating distributions that is closest to original distribution $p_{\mathbf{x}}$ in KL divergence! This can be interpreted as projecting $p_{\mathbf{x}}$ onto a family of approximating distributions that we must pre-specify.

We end by solving for optima of mean field variational problem (10) using the more general clique factorization (7) for $p_{\mathbf{x}}$. The steps are nearly identical to the pairwise factorization case but involves a little more bookkeeping. As before, we introduce Lagrange multiplier λ_i for constraint $\sum_{x_i \in \mathcal{X}} \mu_i(x_i) = 1$. Then the Lagrangian is

$$\begin{aligned}
\mathcal{L}(\mu, \lambda) &= D(\mu \parallel p_{\mathbf{x}}) + \sum_{i \in \mathcal{V}} \lambda_i \left(\sum_{x_i \in \mathcal{X}} \mu_i(x_i) - 1 \right) \\
&= \mathbb{E}_{\mu} \left[\log \frac{\mu(\mathbf{x})}{p_{\mathbf{x}}(\mathbf{x})} \right] + \sum_{i \in \mathcal{V}} \lambda_i \left(\sum_{x_i \in \mathcal{X}} \mu_i(x_i) - 1 \right) \\
&= \mathbb{E}_{\mu}[\log \mu(\mathbf{x})] - \mathbb{E}_{\mu}[\log p_{\mathbf{x}}(\mathbf{x})] + \sum_{i \in \mathcal{V}} \lambda_i \left(\sum_{x_i \in \mathcal{X}} \mu_i(x_i) - 1 \right) \\
&= \mathbb{E}_{\mu} \left[\log \prod_{i \in \mathcal{V}} \mu_i(\mathsf{x}_i) \right] - \mathbb{E}_{\mu} \left[-\log Z + \sum_{C \in \mathcal{C}} \log \psi_C(\mathsf{x}_C) \right] + \sum_{i \in \mathcal{V}} \lambda_i \left(\sum_{x_i \in \mathcal{X}} \mu_i(x_i) - 1 \right) \\
&= \sum_{i \in \mathcal{V}} \mathbb{E}_{\mu}[\log \mu_i(\mathsf{x}_i)] - \sum_{C \in \mathcal{C}} \mathbb{E}_{\mu}[\log \psi_C(\mathsf{x}_C)] + \sum_{i \in \mathcal{V}} \lambda_i \left(\sum_{x_i \in \mathcal{X}} \mu_i(x_i) - 1 \right) + \log Z
\end{aligned}$$

$$= \sum_{i \in \mathcal{V}} \sum_{x_i \in \mathcal{X}} \mu_i(x_i) \log \mu_i(x_i) - \sum_{C \in \mathcal{C}} \sum_{x_C \in \mathcal{X}^{|C|}} \left(\prod_{j \in C} \mu_j(x_j) \right) \log \psi_C(x_C)$$
$$+ \sum_{i \in \mathcal{V}} \lambda_i \left(\sum_{x_i \in \mathcal{X}} \mu_i(x_i) - 1 \right) + \log Z.$$

Thus,

$$\frac{\partial \mathcal{L}(\mu, \lambda)}{\partial \mu_i(x_i)} = \log \mu_i(x_i) + 1 - \sum_{\substack{C \in \mathcal{C} \\ \text{s.t. } i \in C}} \sum_{x_C \setminus x_i} \log \psi_C(x_C) \prod_{\substack{j \in C \\ j \neq i}} \mu_j(x_j) + \lambda_i.$$

Setting this to 0 gives the mean field update equation

$$\mu_i(x_i) \propto \exp \left\{ \sum_{\substack{C \in \mathcal{C} \\ \text{s.t. } i \in C}} \sum_{x_C \setminus x_i} \log \psi_C(x_C) \prod_{\substack{j \in C \\ j \neq i}} \mu_j(x_j) \right\}.$$

17.6 Concluding remarks

We've presented variational inference as a way to approximate distributions. This approach has interpretations of approximating the log partition function or doing an information-theoretic projection. We've also given a flavor of the calculations involved for obtaining update rules (which describe an algorithm) that find an approximating distribution. In practice, simple approximating distributions such as mean field are typically used because more complicated distributions can have update rules that are hard to derive or, even if we do have formulas for them, computationally expensive to compute.

Unfortunately, simple approximating distributions may not characterize our original distribution well. A different way to characterize a distribution is through samples from it. Intuitively, if we have enough samples, then we have a good idea of what the distribution looks like. With this inspiration, our next stop will be on how to sample from a distribution without knowing its partition function, leading to a different class of approximate inference algorithms.

6.438 Algorithms for Inference

Fall 2014