Approximate Inference Methods

Amir Globerson

Exact inference in graphical models takes exponential time in the tree-width of the model, in the worst case. This is often not acceptable. For example, a 2D grid graph of size (n, n) has tree-width n. Here we will learn about methods for approximate inference. Some of these work by converting the discrete inference problem (namely a problem of summing or maximizing over a large, but discrete, set of possible solutions) into a continuous optimization problem. These methods are known as variational approaches. See [1] for an excellent introduction. The idea would be to follow the following steps:

- Cast the exact inference problem as a continuous optimization problem $\max_{\boldsymbol{\mu} \in S} f(\boldsymbol{\mu})$ where S is some set and f is some function. For reasons that will become clear this problem will not be efficiently solvable (as we should expect, if the original inference problem was hard). The problem is going to be with both the set S and the function $\boldsymbol{\mu}$.
- Replace both f and S with functions and constraints that will make optimization feasible.

In what follows we focus on pairwise MRFs, defined as:

$$p(\boldsymbol{x}) = \frac{1}{Z} \prod_{ij \in E} \phi_{ij}(x_i, x_j) \prod_i \phi_i(x_i) \tag{1}$$

It will be easier for us to introduce $\theta_{ij}(x_i, x_j) = \log \phi_{ij}(x_i, x_j)$ and $\theta_i(x_i) = \log \phi_i(x_i)$ and write:

$$p(\boldsymbol{x}) = \frac{1}{Z} e^{\sum_{ij} \theta_{ij}(x_i, x_j) + \sum_i \theta_i(x_i)} \tag{2}$$

1 Quick review of continuous optimization

To understand the variational approach, we need to first recall some facts about continuous optimization. The general form of an optimization problem is:

$$\begin{array}{ll} \max & f_0(\boldsymbol{x}) \\ \text{s.}t. & f_i(\boldsymbol{x}) \leq 0 \quad i = 1, \ldots, m \\ & h_i(\boldsymbol{x}) \leq 0 \quad i = 1, \ldots, p \end{array} \tag{3}$$

Key to optimization is the following Lagrangian function:

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = f_0(\boldsymbol{x}) + \sum_i \lambda_i f_i(x) + \sum_i \nu_i h_i(x) \tag{4}$$

And the dual function:

$$g(\boldsymbol{\lambda}, \boldsymbol{\nu}) = \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}) \tag{5}$$

Note that g is concave in $\boldsymbol{\lambda}, \boldsymbol{\nu}$. It's easy to see that for every feasible $\boldsymbol{x}$ and $\boldsymbol{\lambda} \geq 0$ it holds that $g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \leq f_0(\boldsymbol{x})$. Thus:

$$\max_{\boldsymbol{\lambda} \geq 0, \boldsymbol{\nu}} g(\boldsymbol{\lambda}, \boldsymbol{\nu}) \leq f_0(\boldsymbol{x}^*) = \min_{\boldsymbol{x} \in S} f_0(\boldsymbol{x}) \tag{6}$$

where $\boldsymbol{x}^*$ is the optimum of the original problem. It turns out that there is actually an equality in the above when all functions f are convex and all h are linear. And a feasible point exists where all inequalities are strict (Slater conditions).

1.1 Linear Programs

Of specific interest in this class is a set of optimization problems known as linear programs. They have the following form:

$$\begin{array}{ll} \max & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s}.t. & A\boldsymbol{x} \leq \boldsymbol{b} \end{array} \tag{7}$$

(equalities may also be added, but they can be converted to inequalities).

Let us assume that the constraint set is bounded (in which case it's called a polytope). A vertex of the polytope is defined as a point that cannot be obtained as a convex combination of any two other points in the polytope. It can be shown that:

- The optimum of the LP must be achieved at a vertex (although it might also be achieved at other points as well).
- The polytope is the convex hull of its vertices (this is sometimes referred to as the V-representation of the polytope as opposed to the H-representation via $A\boldsymbol{x} \leq \boldsymbol{b}$)

2 An LP Approach to the MAP Problem

The MAP problem for the model in Equation 2 is to maximize the following function $f(\boldsymbol{x}; \boldsymbol{\theta})$ over all possible assignments $\boldsymbol{x}$.

$$f(\boldsymbol{x}; \boldsymbol{\theta}) = \sum_{ij} \theta_{ij}(x_i, x_j) + \sum_i \theta_i(x_i) \tag{8}$$

Assuming we cannot do this exactly for computational reasons, we need to think of approximations. To arrive at an approximation, we rewrite the problem a bit differently.

$$\begin{aligned}\max_{\boldsymbol{x}} f(\boldsymbol{x};\boldsymbol{\theta}) &= \max_{p(\boldsymbol{x})} \sum_{\boldsymbol{x}} p(\boldsymbol{x}) f(\boldsymbol{x};\boldsymbol{\theta}) \\ &= \sum_{ij}\sum_{\boldsymbol{x}} p(\boldsymbol{x})\theta_{ij}(x_i,x_j) + \sum_i \sum_{\boldsymbol{x}} p(\boldsymbol{x})\theta_i(x_i) \\ &= \sum_{ij}\sum_{x_i,x_j} p(x_i,x_j)\theta_{ij}(x_i,x_j) + \sum_i \sum_{x_i} p(x_i)\theta_i(x_i)\end{aligned}$$

So, we find that the optimization only depends on the singleton and pairwise marginals of p. The problem is that these cannot be set independently. In other words we must only optimize over marginals that correspond to *some* distribution p. We are allowed to optimize over only singletons and pairwise marginals as long as we only consider those that come from some distribution p. We need a definition for this set.

The following set of points of the same dimension as $\boldsymbol{\theta}$ is called the marginal polytope.

$$\mathcal{M}(G) = \{\boldsymbol{\mu} \mid \exists p(\boldsymbol{x}) \in \Delta \text{ s.t. } p(x_i,x_j) = \mu_{ij}(x_i,x_j), p(x_i) = \mu_i(x_i)\}. \quad (9)$$

With this definition we can write:

$$\begin{aligned}\max_{\boldsymbol{x}} f(\boldsymbol{x};\boldsymbol{\theta}) &= \sum_{ij}\sum_{x_i,x_j} \mu_{ij}(x_i,x_j)\theta_{ij}(x_i,x_j) + \sum_i \sum_{x_i} \mu_i(x_i)\theta_i(x_i) \\ &= \max_{\boldsymbol{\mu}\in\mathcal{M}(G)} \boldsymbol{\mu}\cdot\boldsymbol{\theta}\end{aligned}$$

At the optimum, we will have a $\boldsymbol{\mu}$ whose elements are zero or one, and it corresponds to the maximizing assignment (assume for simplicity that there is just one maximizing assignment).

The dimension of $\mathcal{M}(G)$ is small (i.e., polynomial in the graph size). The problem is optimizing over it.

Let us note a few facts about the $\mathcal{M}(G)$:

- It is the convex hull of $O(2^n)$ points, one for each assignment to $\bar{\boldsymbol{x}}$. Define:

$$\begin{aligned}\mu_i^{\bar{\boldsymbol{x}}}(x_i) &= \mathcal{I}(x_i = \bar{x_i}) \\ \mu_{i,j}^{\bar{\boldsymbol{x}}}(x_i,x_j) &= \mathcal{I}(x_i = \bar{x_i}, x_j = \bar{x_j})\end{aligned}$$

  Then it's easy to see that any point in $\mathcal{M}(G)$ whose source distribution is $p(x)$ can be written as:

$$\boldsymbol{\mu} = \sum_{\boldsymbol{x}} p(\boldsymbol{x})\mu^{\boldsymbol{x}} \quad (10)$$

The convex hull of a finite set of points is called a polytope and it can always be described as the intersection of a finite set of linear inequalities. In other words, there exists a matrix $A(G)$ and a vector $\boldsymbol{b}(G)$ such that $\mathcal{M}$ corresponds to $\boldsymbol{\mu}$ such that $A(G)\boldsymbol{\mu} \leq \boldsymbol{b}(G)$. The problem is that the number of inequalities (number of rows in $A(G)$ can be exponentially large in n. However we can conclude that: **the MAP problem is equivalent to an LP, although one that's difficult to solve**.

- The points $\mu^{\boldsymbol{x}}$ are the vertices of the polytope. To see this consider $\mu^{\bar{\boldsymbol{x}}}$. Assume that it's a convex combination of two points $\boldsymbol{\mu}', \boldsymbol{\mu}''$ Then both points must have zeros at all points where $\mu^{\bar{\boldsymbol{x}}}$ has zeros. It's easy to see that the remaining coordinates must be one, which implies are points are equal.

So, what is the point in using this formulation? It turns out it serves as a good starting point for approximations that work well in theory and in practice.

2.1 MAP LP Approximations

It is pretty easy to come up with other polytopes that are outer bounds on the marginal polytope. Suppose we come up with a different polytope $\mathcal{S}$ such that $\mathcal{S} \supseteq \mathcal{M}(G)$, and consider the optimization problem:

$$\max_{\boldsymbol{\mu} \in \mathcal{S}} \boldsymbol{\mu} \cdot \boldsymbol{\theta} \tag{11}$$

We can observe two things:

- The optimum of the $\mathcal{S}$ problem always upper bounds the MAP value.
- Assume that the vertices of $\mathcal{M}(G)$ are also vertices of $\mathcal{S}$. If for some $\boldsymbol{\theta}$ the optimum of $\mathcal{S}$ is a vertex of $\mathcal{M}(G)$, then we have found the MAP.

Clearly, for this to be useful we need to be able to maximize efficiently over $\mathcal{S}$. One case in which we can do this is when $\mathcal{S}$ is defined via a small enough set of inequalities (i.e., polynomial in n).

Let us define one such set $\mathcal{S}$. One way to construct an outer bound is to consider properties that any point in $\mathcal{M}(G)$ needs to satisfy. Since the $\boldsymbol{\mu}$ comes from some distribution, the following facts always hold:

- All elements are non-negative: $\boldsymbol{\mu} \geq 0$.
- All distributions sum to one: $\sum_{x_i} \mu_i(x_i) = 1$ and $\sum_{x_i, x_j} \mu_{ij}(x_i, x_j) = 1$.
- The pairwise distributions agree with the singleton ones: $\sum_{x_i} \mu_{ij}(x_i, x_j) = \mu_i(x_i)$. This will also imply that pairwise distribution agree on the singletons in their overlap.

Define the *local marginal polytope* $\mathcal{M}_L(G)$ as the set of $\boldsymbol{\mu}$ vectors that satisfy the above. Clearly it is an outer bound on $\mathcal{M}(G)$. It's also easy to see that the vertices of $\mathcal{M}(G)$ are vertices of $\mathcal{M}_L(G)$ (proof is similar to showing they are vertices of $\mathcal{M}(G)$).

2.2 The tree case

When the graph G is a tree, inference is tractable via dynamic programming (e.g., max-product). What happens with the LP approximation in this case?

Proposition: When G is a tree $\mathcal{M}(G) = \mathcal{M}_L(G)$. **Proof:** We know that $\mathcal{M}(G) \subseteq \mathcal{M}_L(G)$. We therefore just need to show that $\mathcal{M}(G) \supseteq \mathcal{M}_L(G)$. Given a point $\boldsymbol{\mu} \in \mathcal{M}_L(G)$, we'd like to show that $\boldsymbol{\mu} \in \mathcal{M}(G)$, that is there exists a distribution $p(x)$ that has the marginals $\boldsymbol{\mu}$. To do this, we can use the junction tree theorem, which says that if μ_{ij}, μ_i are consistent then the following $p(x)$ is a distribution and has μ_i as marginals:

$$p(x) = \prod_i \mu_i(x_i) \prod_{ij \in E} \frac{\mu_{ij}(x_i, x_j)}{\mu_i(x_i)\mu_j(x_j)} = \prod_i \mu_i^{1-d_i}(x_i) \prod_{ij \in E} \mu_{ij}(x_i, x_j) \qquad (12)$$

In the general case, this inequality does not hold. For example, consider a complete graph on three vertices. Then $\mathcal{M}_L(G)$ has points that are outside the marginal polytope. Here is one:

$$\mu_{12}(x_1, x_2) = \mu_{23}(x_2, x_3) = \mu_{13}(x_1, x_3) = \begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix}$$

And $\mu_i = [0.5, 0.5]$. This point satisfies all local polytope inequalities. However, suppose that there exists $p(x)$ with these marginals. Since $\mu_{12}(0, 1) = 0.5$, we must have $p(0, 1, 0) + p(0, 1, 1) = 0.5$. However, both these assignments must have zero probability since the marginals $p_{13}(0, 0) = 0$ and $p_{23}(1, 1) = 0$. In fact, the above $\boldsymbol{\mu}$ is a vertex of $\mathcal{M}_L(G)$ in this case and can actually be obtained as a solution of the relaxed LP.

Note that generally, it will be easier to characterize subsets of the variables $\boldsymbol{\mu}$. For example, we have a simple characterization when G is a tree but not when it is a complete graph.

There are many extensions to the above approach and in some cases we can prove results:

- We can add marginals over larger sets e.g., $\mu_{1234}(x_1, x_2, x_3, x_4)$ and force those to agree with the pairwise distributions within them. The larger the sets we add, the better (tighter) our approximation will become.
- For planar graphs with binary interactions $\theta_{ij}(x_i, x_j) = J_{ij}\mathcal{I}(x_i = x_j)$ (and no field), if we add larger sets for all triplets in the graph, we get

an exact representation (note that we have shown how to solve such models earlier in the course via a matching algorithm).

- For binary attractive models the above LP relaxation yields an exact result (although the local polytope is not equal to the marginal polytope).

3 Variational approach to marginal calculation

Recall that in the marginal problem, we want to calculate the marginal $p(x_i)$ of the distribution:

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} e^{\sum_{ij}\theta_{ij}(x_i,x_j)+\sum_i \theta_i(x_i)} \tag{13}$$

As in the MAP case we'd like to pose this as an optimization problem over continuous variables and then approximate this optimization problem via something we can solve.

We start with an optimization problem that seems a bit silly: find the distribution $q(x)$ that is closet to $p(x;\boldsymbol{\theta})$. Clearly this is $p(x;\boldsymbol{\theta})$ itself. But this will help us.

$$\min_{qx)} D_{KL}[q(x)|p(x;\boldsymbol{\theta})] = 0 \tag{14}$$

Rewrite this as:

$$\min_{q(x)} -H\left[q(x)\right] + \log Z(\boldsymbol{\theta}) - \sum_{ij}\sum_{x_i,x_j} q(x_i,x_j)\theta_{ij}(x_i,x_j) - \sum_{i,x_i} q(x_i)\theta_i(x_i) = 0 \tag{15}$$

Or:

$$\log Z(\boldsymbol{\theta}) = \max_{q(x)} \sum_{ij}\sum_{x_i,x_j} q(x_i,x_j)\theta_{ij}(x_i,x_j) + \sum_{i,x_i} q(x_i)\theta_i(x_i) + H\left[q(x)\right] \tag{16}$$

The first two terms are exactly like what we had for the MAP case, i.e., they only depend on single and pairwise marginals. We can do the optimization in two steps. First optimize over all possible pairwise marginals and then over q with these pairwise marginals.

$$\begin{aligned}
\log Z(\boldsymbol{\theta}) &= \max_{\boldsymbol{\mu}\in\mathcal{M}(G)} \sum_{ij}\sum_{x_i,x_j} \mu_{ij}(x_i,x_j)\theta_{ij}(x_i,x_j) + \sum_{i,x_i}\mu_i(x_i)\theta_i(x_i) + \max_{q:q_{ij}=\mu_{ij}} H\left[q(x)\right] \\
&= \max_{\boldsymbol{\mu}\in\mathcal{M}(G)} \boldsymbol{\mu}\cdot\boldsymbol{\theta} + \max_{q:q_{ij}=\mu_{ij}} H\left[q(x)\right]
\end{aligned}$$

The solution to the above maximization problem is:

$$q(x;\hat{\theta}) = \frac{1}{Z(\hat{\theta})} e^{\sum_{ij}\hat{\theta}_{ij}(x_i,x_j)+\sum_i \hat{\theta}_i(x_i)} \tag{17}$$

Where $\hat{\theta}$ is chosen such that the marginals of $q(x;\hat{\theta})$ are equal to $\boldsymbol{\mu}$. This argument is not exact since if the vector $\boldsymbol{\mu}$ has zeros there is no finite parameter vector $\hat{\theta}$ that can yield these marginals. See a more precise argument in Wainwright and Jordan's book. We denote this distribution by $q(x;\hat{\theta}(\boldsymbol{\mu}))$ and its entropy by $H\left[\hat{\theta}(\boldsymbol{\mu})\right]$.

We thus have that:

$$\log Z(\boldsymbol{\theta}) = \max_{\boldsymbol{\mu}\in\mathcal{M}(G)} \boldsymbol{\mu}\cdot\boldsymbol{\theta} + H\left[\hat{\theta}(\boldsymbol{\mu})\right] \tag{18}$$

It can be shown that this is a concave function of $\boldsymbol{\mu}$ so there are no local optima problem.

This is exact! At the optimum, the maximizing $\boldsymbol{\mu}$ are exactly the marginals of the distribution $p(x;\boldsymbol{\theta})$. BUT, there are now *two* reasons why we can't solve this efficiently. The first is the usual marginal polytope problem. The second is that the function $H\left[\hat{\theta}(\boldsymbol{\mu})\right]$ is not efficiently computable. There can even be cases where we have an exact marginal polytope representation but cannot compute the other term (these are problems where maximization is tractable but summation is not). We thus must turn to approximations of both parts (constraints and objective).

3.1 The tree case

As in other graphical models problems, we draw inspiration from the tree graph case. When G is a tree we know that the local marginal polytope $\mathcal{M}_L(G)$ is exactly equal to the marginal polytope. What can we say about the function $H\left[\hat{\theta}(\boldsymbol{\mu})\right]$ in that case?

First, we recall that the distribution:

$$q(x) = \prod_i \mu_i(x_i) \prod_{ij} \frac{\mu_{ij}(x_i,x_j)}{\mu_i(x_i)\mu_j(x_j)} = \prod_i \mu_i^{1-d_i}(x_i) \prod_{ij\in E} \mu_{ij}(x_i,x_j) \tag{19}$$

Has the marginals $\mu_i(x_i), \mu_{ij}(x_i,x_j)$. Thus $\hat{\boldsymbol{\theta}}(\boldsymbol{\mu})$ is given by:

$$\begin{aligned}
\hat{\theta}_{ij}(x_i,x_j) &= \log \frac{\mu_{ij}(x_i,x_j)}{\mu_i(x_i)\mu_j(x_j)} \\
\hat{\theta}_i(x_i) &= \log \mu_i(x_i)
\end{aligned}$$

Here is the entropy of the distribution $q(x)$:

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

Thus, when the graph is a tree we have a simple form for the exact optimization problem:

$$\log Z(\boldsymbol{\theta}) = \max_{\boldsymbol{\mu} \in \mathcal{M}_L(G)} \boldsymbol{\mu} \cdot \boldsymbol{\theta} + \sum_i H(\mu_i(x_i)) - \sum_{ij} I(\mu_{ij}(x_i, x_j)) \tag{20}$$

3.2 The non-tree case

Now say G is not a tree. Then neither the local marginal polytope and the entropy in Equation 20 is exact. One approach to approximating is to keep the same expression we had for a tree and optimize:

$$\log Z(\boldsymbol{\theta}) = \max_{\boldsymbol{\mu} \in \mathcal{M}_L(G)} \boldsymbol{\mu} \cdot \boldsymbol{\theta} + \sum_i H(\mu_i(x_i)) - \sum_{ij \in E} I(\mu_{ij}(x_i, x_j)) \tag{21}$$

The sets of edges E no longer correspond to a tree so the entropy approximation may in fact be negative (i.e., it is not the entropy of any distribution). Furthermore, the result is neither an upper or a lower bound on the true $\log Z(\boldsymbol{\theta})$ (an upper bound may be obtained by upper bounding $H\left[\hat{\theta}(\boldsymbol{\mu})\right]$ which can be done i.e. by the tree reweighting method of Wainwright). The above objective is called the Bethe approximation.

3.3 Relation to belief propagation

One of the most popular algorithms for approximate inference is loopy belief propagation. It is similar to the above in that it is exact for tree structured graphs. The updates in BP are given by:

$$m'_{ji}(x_i) \propto \sum_{x_j} e^{\theta_{ij}(x_i, x_j) + \theta_j(x_j)} \prod_{k \in N(j) \setminus i} m_{kj}(x_j) \tag{22}$$

A major contribution to understanding this algorithm was a paper by J. Yedidia W. Freeman and Y. Weiss, which related its fixed points to the Bethe optimization problem. The relation is described in the following proposition:
Proposition: The fixed points of the belief propagation algorithm can be mapped to local optima of the Bethe optimization problem (i.e., there a function from m to μ).
Proof: Write the Lagrangian:

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\mu}, \lambda, \boldsymbol{\delta}) \quad = \quad & \boldsymbol{\mu} \cdot \boldsymbol{\theta} + H_B(\boldsymbol{\mu}) + \sum_i \delta_i \left[\sum_{x_i} \mu_i(x_i) - 1 \right] + \sum_{ij} \left[\sum_{x_i, x_j} \mu_{ij}(x_i, x_j) - 1 \right] \\
& - \sum_{ij} \lambda_{ij}(x_j) \left[\sum_{x_i} \mu_{ij}(x_i, x_j) - \mu_j(x_j) \right] - \sum_{ij} \lambda_{ji}(x_i) \left[\sum_{x_j} \mu_{ij}(x_i, x_j) - \mu_i(x_i) \right]
\end{aligned}$$

Take the derivative w.r.t. μ_i:

$$\mu_i(x_i) \propto e^{\theta_i(x_i)+\sum_{k\in N(i)}\lambda_{ki}(x_i)} \tag{23}$$

And for $\mu_{ij}(x_i, x_j)$:

$$\mu_{ij}(x_i, x_j) \propto e^{\theta_{ij}(x_i,x_j)-\lambda_{ij}(x_j)-\lambda_{ji}(x_i)} \sum_{x_i} \mu_{ij}(x_i, x_j) \sum_{x_i} \mu_{ij}(x_i, x_j) \tag{24}$$

To characterize the optimal λ we need to substitute the constraints. So if we set the marginalization constraints we obtain:

$$\mu_{ij}(x_i, x_j) \quad \propto \quad e^{\theta_{ij}(x_i,x_j)-\lambda_{ij}(x_j)-\lambda_{ji}(x_i)+\theta_i(x_i)+\theta_j(x_j)+\sum_{k\in N(i)}\lambda_{ki}(x_i)+\sum_{k\in N(j)}\lambda_{kj}(x_j)}$$

And setting $\sum_{x_j} \mu_{ij}(x_i, x_j) = \mu_i(x_i)$ we get:

$$e^{\lambda_{ji}(x_i)} \propto \sum_{x_j} e^{\theta_{ij}(x_i,x_j)+\theta_j(x_j)} \prod_{k\in N(j)\setminus i} e^{\lambda_{kj}(x_j)} \tag{25}$$

If we set $m_{ji}(x_i) = e^{\lambda_{ji}(x_i)}$ we get the belief propagation update. This means that if we have found an m such that the above holds (i.e, a fixed point of BP) then we can use it to obtain a local optimum of the Bethe optimization problem. Note that the above is useful only if BP has indeed reached a fixed point. This is not necessarily the case since BP may not converge for non-tree graphs.

3.4 The Mean Field Algorithm

Another approach to approximating the marginal polytope is to consider marginals that are generated by particular distributions $p(\boldsymbol{x})$. This will yield an inner bound on the polytope since these marginals do correspond to some distribution.

The simplest illustration of this approach results in the mean-field method. Consider distributions of the form;

$$p(x_1, \ldots, x_n) = \mu_i(x_i) \tag{26}$$

where $\mu_i(x_i)$ are a set of n arbitrary distributions over the single variables. Then the following is clear:

- These are indeed valid distributions (non-negative and normalize to one).
- Their singleton marginals are $p_i(x_i) = \mu_i(x_i)$ and pairwise marginals are $p_{ij}(x_i, x_j) = \mu_i(x_i)\mu_j(x_j)$.

- To complete the variational approximation we need to figure out what $H\left[\hat{\theta}(\boldsymbol{\mu})\right]$ is in Equation 17. This is easily done for the marginals we have chosen. The entropy of p is:

$$\begin{aligned} H(p) = -\sum_{\boldsymbol{x}} p(\boldsymbol{x}) \sum_i \log \mu_i(x_i) &= -\sum_i \sum_{x_i} \sum_{\bar{\boldsymbol{x}}:\bar{x}_i = x_i} p(\bar{\boldsymbol{x}}) \log \mu_i(x_i) \\ &= -\sum_i \sum_{x_i} \mu_i(x_i) \log \mu_i(x_i) \end{aligned}$$

  Given a set of singleton and pairwise marginals $\mu_i(x_i)$ and $\mu_i(x_i)\mu_j(x_j)$, the distribution $p(x_1, \ldots, x_n)$ is a pairwise MRF that has these marginals, and therefore we have that:

$$H\left[\hat{\theta}(\boldsymbol{\mu})\right] = -\sum_i \sum_{x_i} \mu_i(x_i) \log \mu_i(x_i) \tag{27}$$

We can use the above to generate a new variational approximation. Consider the set of singleton and pairwise marginals defined above. Denote these by $\mathcal{M}_F$. It is clear that $\mathcal{M}_F \subset \mathcal{M}$. We can now introduce an approximation to Equation 18:

$$\log Z(\boldsymbol{\theta}) = \max_{\boldsymbol{\mu} \in \mathcal{M}(G)} \boldsymbol{\mu} \cdot \boldsymbol{\theta} + H\left[\hat{\theta}(\boldsymbol{\mu})\right] \geq \max_{\boldsymbol{\mu} \in \mathcal{M}_F} \boldsymbol{\mu} \cdot \boldsymbol{\theta} + H\left[\hat{\theta}(\boldsymbol{\mu})\right] \tag{28}$$

And given the observation above, we have that:

$$\max_{\boldsymbol{\mu} \in \mathcal{M}_F} \boldsymbol{\mu} \cdot \boldsymbol{\theta} + H\left[\hat{\theta}(\boldsymbol{\mu})\right] = \max_{\mu_i(x_i)} -\sum_{ij} \sum_{x_i, x_j} \mu_i(x_i)\mu_j(x_j)\theta_{ij}(x_i, x_j) - \sum_i \sum_{x_i} \mu_i(x_i)\theta_i(x_i) + \sum_i \sum_{x_i} \mu_i(x_i) \log \tag{29}$$

We note that this problem is equivalent to finding the best approximation of $p(\boldsymbol{x}; \boldsymbol{\theta})$ in terms of a distribution with independent variables. Namely:

$$\min_{\mu} D_{KL}[\prod_i \mu_i(x_i) | p(\boldsymbol{x}; \boldsymbol{\theta})] \tag{30}$$

The key thing to observe again is that we now have a constrained minimization problem over a function that is easy to evaluate. There is one caveat though: the function $F(\boldsymbol{\mu}, \boldsymbol{\theta})$ is not convex in its variables, and generally we can only find its local optima (in fact you can convince yourself, as an exercise, that if you could solve this problem, you would have been able to calculate MAP efficiently, which we know cannot be done).

What can be done is to find local optima of $F(\boldsymbol{\mu}, \theta)$? One simple approach is to change only a subset of the $\boldsymbol{\mu}$ variables at each iteration while keeping the others fixed. Here's one way of doing this. At each iteration:

- Pick some k.

- Fix all values in $\boldsymbol{\mu}$ except $\mu_k(x_k)$ (for all x_k). Now seek the values of $\mu_k(x_k)$ that minimize $F(\boldsymbol{\mu}, \theta)$.

This scheme is known as block coordinate descent or block coordinate minimization.

What is particularly nice is that the above optimal value of $\mu_k(x_k)$ can be found in closed form. We are now viewing F as a function only of μ_k (with the other being fixed). So we can write:

$$F(\mu_k) = -\sum_{j \in N(k)} \sum_{x_j, x_k} \mu_j(x_j)\mu_k(x_k)\theta_{kj}(x_k, x_j) - \sum_{x_k} \mu_k(x_k)\theta_k(x_k) + \sum_{x_k} \mu_k(x_k) \log \mu_k(x_k) \tag{31}$$

We would like to minimize this subject to non-negativity and normalization of μ_k. Lets forget about non-negativity for now. So the Lagrangian is:

$$\mathcal{L}(\mu_k, \lambda) = F(\mu_k) + \lambda \left(\sum_{x_k} \mu_k(x_k) - 1 \right) \tag{32}$$

Deriving wrt $\mu_k(x_k)$ (i.e., for a particular value of x_k. e..g, $x_k = 1$) we have:

$$\frac{\partial \mathcal{L}(\mu_k, \lambda)}{\partial \mu_k(x_k)} = -\sum_{j \in N(k)} \sum_{x_j} \mu_j(x_j)\theta_{kj}(x_k, x_j) - \theta_k(x_k) + 1 + \log \mu_k(x_k) + \lambda \tag{33}$$

Yielding:

$$\mu_k(x_k) = e^{\theta_k(x_k) + \sum_{j \in N(k)} \sum_{x_j} \mu_j(x_j)\theta_{kj}(x_k, x_j) - \lambda - 1} \tag{34}$$

To solve for λ we need to introduce the normalization constraint. But no need to solve explicitly, since at the end λ will just be a normalizing constant, and we will get that:

$$\mu_k(x_k) \propto e^{\theta_k(x_k) + \sum_{j \in N(k)} \sum_{x_j} \mu_j(x_j)\theta_{kj}(x_k, x_j)} \tag{35}$$

This is the mean field update, and it is quite intuitive. What it says is that the effect of neighbor j of k on k is the expected value of $\theta_{kj}(x_k, x_j)$ where expectation is taken with respect to the estimated marginal.

The mean field approximation tends to work well when the distribution we are approximating is close to uni-modal. One of its advantages over BP is that number of variables is smaller (one per node in mean-field as opposed to one per edge in belief propagation) and the overall cost of an iteration is lower.

References

[1] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.