Week 7: Graphs with Cycles

Constantine Caramanis*

November 6, 2002

This paper is a (partial) summary of chapters 2,5 and 7 of Martin Wainwright's Ph.D. Thesis. For a copy of this, as well as relevant papers, see his web page:

http://ssg.mit.edu/ mjwain

In addition to the results, all the figures appearing in this paper are taken from the Ph.D. Thesis, courtesy of Martin Wainwright.

1 Introduction

The problem around which this paper focuses is that of computing the marginals of a given distribution $p(\boldsymbol{x})$. This is a common problem that arises in many fields. One of the areas which has drawn much attention recently is that of error correcting codes in information theory. Here, so-called message passing techniques known under the name of belief propagation (BP) are used to quickly decode codes on graphs.

The Thesis, and thus this paper, uses convex analysis in a fundamental way, and takes a variational approach to the problem of estimation on loopy graphs, while maintaining a strong connection to the underlying geometry.

To appreciate the difficulty behind a problem that in theory has a simple solution, consider the naive approach: integration. Even for $\{0, 1\}$ random variables, marginalizing an N-variate distribution requires $O(2^N)$ steps, which quite quickly becomes intractable.

By developing some of the Riemannian geometry of exponential families of distributions, we are able to show that existing algorithms, in particular Belief Propagation, can be seen as successive projections onto a sequence of smooth manifolds. This allows a deeper analysis of the algorithms, the output of these algorithms, and their behavior.

This paper is roughly structured in linear correspondence with the chapters covered. Thus, Section 3 below corresponds to chapter 2, Section 4 to chapter 5, and Section 5 to chapter 7. Section 3 below primarily serves to introduce the geometry and the convex analysis and duality that will be the engine driving the theoretical results presented here. Using exponential families of distributions, distribution spaces are endowed with a Riemannian geometric structure. Then, exploiting the duality theory of convex analysis, further structure of these manifolds is revealed and explored. Section 4 provides an exposition of the main algorithmic framework of tree based reparameterization. Here, an alternative method for inference on graphs is presented, and its parallels with BP highlighted. We also give in this section some higher order generalizations, corresponding to generalized belief propagation, and also provide a discussion on the approximation qualities and behavior of the tree reparameterization. Finally, at the end of section 4, we motivate the problem of computing upper bounds to the so-called log partition function. We turn to this problem in section 5. Section 6 provides a conclusion and summary of the paper.

*cmcaram@mit.edu

2 Recap of Mean Field and BP

For the sake of a point of reference, we give a (very) quick recap of mean field, and also belief propagation, as in the Yedidia paper that we read for last week's presentation, and in general, the papers ([4], [5], [2], [6]).

Belief propagation is an algorithm that efficiently computes marginal probabilities exactly, on distributions whose underlying graphical structure is singly connected, i.e. a tree (see the next section for explicit definitions). The BP algorithm is most often phrased in terms of "message passing" between neighboring nodes in the graph, although we show in this paper that it can be considered in a message-independent manner.

In addition, however, Yedidia et al. show that Belief propagation may be regarded as minimizing a particular function of the distribution. This is a variational approach to the problem. The main idea is to define some function, in this case a free energy, such that the minimizing argument must be exactly the marginals we seek. Not surprisingly, these functionals are in general intractable, i.e. minimizing them exactly is no easier than solving the original problem. From there, one recourse is to attempt to minimize a functional that approximates the original functional, thus obtaining a (we hope) useful approximation to the desired marginals.

Yedidia et al. have shown belief propagation attempts to minimize the so-called Bethe free energy, which, as we saw last week, is an approximation of the Gibbs free energy, using only single node and edge marginals. This approximation is built up by appoximating the entropy,

$$H_{Bethe}(\boldsymbol{T}) = \sum_{s \in V} H_s(T_s) + \sum_{(s,t)\in E} I_{st}(T_{st}),$$

and then approximating the Gibbs free energy,

$$G(p(\boldsymbol{x})) = \sum_{\boldsymbol{x}} p(\boldsymbol{x})E(\boldsymbol{x}) + \sum_{\boldsymbol{x}} p(\boldsymbol{x}) \log p(\boldsymbol{x}),$$

by the tree–decomposed

$$\min_{\boldsymbol{T}\in \mathrm{TREE}(G)} \left\{ -H_{Bethe}(\boldsymbol{T}) - \sum_{s\in V}\sum_{x_s} T_s(x_s)\log \psi_s(x_s) - \sum_{(s,t)\in E}\sum_{(x_s,x_t)} T_{st}(x_s,x_t)\log \psi_{st}(x_s,x_t) \right\}.$$

The above expression is in notation that is again defined and explained in section 4.

As section 3 below discusses in detail, such approximations are exact for trees, but not in general for graphs with loops. The method's exactness on trees is, roughly speaking, the variational justification for the Bethe approach.

Mean Field techniques may also be viewed in this light. In mean field, the variational problem we solve is:

$$\min_{p\in \mathcal{F}} D(p||p^*),$$

where $\mathcal{F}$ is some family of distributions, and p^* is the target distribution. Performing a global optimization would yield the true target distribution, p^*, but in general this is intractable. Therefore the problem is made tractable by performing a constrained minimization over a smaller family of distributions, $\mathcal{F}$. Typically, $\mathcal{F}$ is taken to be the family of fully factorized distributions. Higher complexity families are used as well, and in this case the method is known as structured mean field. We note that while it is possible for a mean field approach to perform the optimization over higher order families of distributions (e.g. tree structured distributions) nevertheless there are important differences from BP, and the tree reparameterization approach explained in the sequel. These differences are brought out rather blatantly, by a closer look at the underlying geometry.

3 Geometry and Convex Analysis

In this section we develop the main tools, namely, geometry and convex analysis, which we use throughout the paper. We try to give a summary of each subsection, to make for easier reading.

3.1 Basics of Factorizations

First, we give a quick recap of the meaning of graphical models, specifically, what it is that they model. Graphical models model the Markov structure of the dependence structure of some collection of random variables. In a given graph, the nodes represent the individual random variables. The edges in the graph represent the Markov structure of the graph. If our variables are $\{X_i\}_{i=1}^N$, then a subset $\{X_i\}_{i \in A}$ of the random variables, is independent of another (disjoint) subset $\{X_j\}_{j \in B}$ when conditioned on a third (disjoint) subset $\{X_k\}_{k \in C}$, if and only if there is no path in the graph from a node in A to a node in B, without passing through a node in C. We can see that this is a generalization of the situation on a Markov chain. We denote the underlying graph G, its nodes V, and its edges E. The **Hammersley-Clifford** theorem is a result about the possible factorizations of a distribution with a particular graph structure. It tells us that a distribution on a graph with set of cliques $\mathcal{C}$, may be factored according to the variables in those cliques:

$$p(\boldsymbol{x}) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(\boldsymbol{x}_C),$$

where Z is a normalization constant. The functions ψ_C are called compatibility functions. Note that in general, these functions are not the marginals of the subvector $\boldsymbol{x}_C$, i.e. it may not be possible to write a distribution as a product of its marginals, in such a factorized form. This brings up one of the fundamental properties of distributions whose underlying graph has a tree structure. If a distribution $p(\boldsymbol{x})$ has a tree graph structure (in particular, if its underlying graph has no cycles) then we can factor it as follows:

$$p(\boldsymbol{x}) = \prod_{i \in V} p_s(x_s) \prod_{(s,t) \in E} \frac{p_{st}(x_s, x_t)}{p_s(x_s) p_t(x_t)}.$$

This is merely the symmetrized version of the familiar conditional factorization. Thus the factorization, in this case, immediately reveals the marginal probabilities. We remark on a particular feature of such a factorization. Suppose we are given numbers

$$\{P_{s,j}, s \in V, P_{st,jk}, (s,t) \in E\},$$

that satisfy the following local consistency constraints (marginalization constraints):

$$\begin{aligned} &\sum_j P_{s,j} = 1, \quad \forall s, \\ &\sum_k P_{st,jk} = P_{s,j}, \quad \forall t, \\ &P_{s,j}, P_{st,jk} \geq 0. \end{aligned}$$

Then, the factorization above implies that there exists a distribution on $|V|$ variables, with single and pairwise marginals that agree with the values P_s and P_{st} specified. Essentially, this follows because, as the factorization reveals, the single and pairwise marginals specify the entire distribution, i.e. local constistency is enough to imply global consistency. This is clearly not the case for general distributions.

3.2 Exponential Families

In this section we introduce exponential families of random variables, and their Riemannian geometry.

The entropy maximization problem (as in, e.g. Cover and Thomas ([1])) reveals that maximum entropy distributions have what is known as an exponential, or Gibbs, form. In general however, any distribution may be expressed in an exponential form. Indeed, we can always write,

$$\begin{aligned} p(\boldsymbol{x}) &= \frac{1}{Z}\prod_{C\in\mathcal{C}}\psi_C(\boldsymbol{x}_C) \\ &= \exp\{\sum_{C\in\mathcal{C}}\phi_C(\boldsymbol{x}_C) - \log Z\}, \end{aligned}$$

where $\phi_C = \log\psi_C$.

Given some family of log compatibility functions, called potential functions $\{\phi_\alpha \,|\, \alpha \in \mathcal{A}\}$, we can describe a parametric family of distributions,

$$\begin{aligned} p(\boldsymbol{x};\theta) &= \exp\{\sum_\alpha \theta_\alpha\phi_\alpha(\boldsymbol{x}) - \Phi(\theta)\}, \\ \Phi(\theta) &= \log\left(\sum_{\boldsymbol{x}}\exp\{\sum_\alpha\theta_\alpha\phi_\alpha(\boldsymbol{x})\}\right). \end{aligned}$$

The normalization constant is now also parameterized by θ, and is called the **log partition function**. While this quantity may seem rather innocuous, in fact it is central to many difficult problems in this area, as well as many of the techniques developed here. It is in general very difficult to compute. Obtaining upper bounds to the log partition function is the subject of section 5. The parameter θ takes values in the set where this normalization constant is finite:

$$\Theta := \{\theta \in \mathbb{R}^{|\mathcal{A}|} \mid \Phi(\theta) < \infty\}.$$

If the potential functions $\{\phi_\alpha\}$ are linearly independent, we obtain a so-called minimal representation of the exponential family. For example, for a univariate Gaussian family, the functions $\{x, x^2\}$ constitute an independent and spanning family. Similarly, we can see that the set of functions

$$\left\{\phi_\alpha(\boldsymbol{x}) = \prod_{i\in\alpha} x_i, \alpha \subseteq \{1,\ldots,N\}\right\} = \left\{\{x_s\}_{s=1}^N \cup \{x_s x_t\}_{s<t} \cup \cdots \cup \{x_1\cdots x_N\}\right\},$$

are independent, and span the space of all $\mathbb{R}$-valued functions on $\{0,1\}^N$. We denote the dimension of the exponential family by $d(\theta)$. In the examples above, we have $d(\theta) = 2$ for the Gaussian, and $d(\theta) = 2^N - 1$ for the N $\{0,1\}$-variables. Note, however, that dimension $d(\theta)$, as the notation would suggest, is a property of the parameterization, and is only an invariant quantity among the minimal representations.

While the minimal representation is by definition the most compact, other, overcomplete representations are possible, and as we will see, quite useful.

Consider a minimal representation of some parametric family. Then as $p(\boldsymbol{x};\theta) > 0$ for all $\theta \in \Theta$, and all $\boldsymbol{x}$, the function

$$\theta \mapsto \log p(\boldsymbol{x};\theta),$$

is well defined. Under appropriate regularity conditions, the set $\mathcal{M} = \{\log p(\boldsymbol{x};\theta) | \theta \in \Theta\}$ is a $d(\theta)$-dimensional Riemannian manifold, with coordinate function as given above: $\theta \mapsto \log p(\boldsymbol{x};\theta)$. It is a straightforward exercise to see that the Riemannian structure of this manifold is given by the familiar Fisher information matrix,

$$G(\theta) = (g_{\alpha\beta}(\theta)) := (\mathrm{cov}_\theta\{\phi_\alpha, \phi_\beta\}).$$

Consider the tangent vector,

$$\boldsymbol{t}_\alpha = \frac{\partial}{\partial\theta_\alpha}\log p(\boldsymbol{x};\theta).$$

Then a direct computation yields the desired result:

$$\langle \boldsymbol{t}_\alpha, \boldsymbol{t}_\beta\rangle_\theta = \mathbb{E}_\theta\left[\frac{\partial}{\partial\theta_\alpha}\log p(\boldsymbol{x};\theta)\frac{\partial}{\partial\theta_\beta}\log p(\boldsymbol{x};\theta)\right] = \mathrm{cov}_\theta\{\phi_\alpha, \phi_\beta\}.$$

3.3 Legendre Transform and Dual Coordinates

This section describes one of the most important tools exploited throughout the Thesis, and thus this paper. Using Legendre duality, and exploiting the convexity of the log partition function, we link the exponential parameters $\{\theta\}$ to another set of parameters called the mean parameters.

Recall the log partition function is given by

$$\Phi(\theta) = \log\left(\sum_{\boldsymbol{x}} \exp\{\sum_{\alpha} \theta_\alpha \phi_\alpha(\boldsymbol{x})\}\right).$$

Differentiating, we find,

$$\begin{aligned}\frac{\partial \Phi}{\partial \theta_\alpha}(\theta) &= \mathbb{E}_\theta\left[\phi_\alpha\right] \\ \frac{\partial^2 \Phi}{\partial \theta_\alpha \partial \theta_\beta}(\theta) &= \mathrm{cov}_\theta\{\phi_\alpha, \phi_\beta\} = \mathbb{E}_\theta\left[(\phi_\alpha - \mathbb{E}_\theta\left[\phi_\alpha\right])(\phi_\beta - \mathbb{E}_\theta\left[\phi_\beta\right])\right].\end{aligned}$$

In particular, this yields the following important result:

Lemma 1 *The log partition function $\Phi(\theta)$ is convex as a function of θ, and strictly convex when the representation is minimal.*

Next, we use this convexity to use Legendre duality to couple the parameter θ with a dual parameter. The Legendre (or Fenchel-Legendre) dual is given by:

$$\Psi(\eta) := \sup_\theta\{\eta^T\theta - \Phi(\theta)\}.$$

By the strict convexity of the log partition function Φ, for some given η, the optimal value in the optimization above is attained at some unique value $\hat{\theta}$, which is linked to the dual parameter η by the usual stationarity conditions,

$$\eta_\alpha := \eta_\alpha(\hat{\theta}) = \mathbb{E}_{\hat{\theta}}\left[\phi_\alpha\right].$$

The dual parameters η are called mean parameters, because of the relation above. The Legendre dual of $\Phi(\theta)$ now takes the form:

$$\begin{aligned}\Psi(\eta(\hat{\theta})) &= \sum_\alpha \hat{\theta} \mathbb{E}_{\hat{\theta}}\left[\phi_\alpha\right] - \Phi(\hat{\theta}) \\ &= \mathbb{E}_{\hat{\theta}}\left[\sum_\alpha \hat{\theta}_\alpha \phi_\alpha(\boldsymbol{x}) - \Phi(\hat{\theta})\right] \\ &= \mathbb{E}_{\hat{\theta}}\left[\log p(\boldsymbol{x};\hat{\theta})\right] \\ &= -H(p),\end{aligned}$$

where H is the usual entropy function. The Legendre transform thus gives a map:

$$[\Lambda(\theta)]_\alpha = \frac{\partial \Phi(\theta)}{\partial \theta_\alpha} = \mathbb{E}_\theta\left[\phi_\alpha\right].$$

If Φ is strictly convext, then as mentioned above, the map is injective (1 to 1) and thus invertible on its image. Since negative entropy is convex, we can apply the Legendre transform a second time, this time to $\Psi(\eta)$, to obtain the inverse map, exactly as we did for the forward direction:

$$[\Lambda^{-1}(\eta)]_\alpha = \frac{\partial \Psi(\eta)}{\partial \eta_\alpha}.$$

Example:
Consider the exponential family (known as the Ising model, in physics) of distributions on $\{0,1\}^N$, given by the representation,

$$p(\boldsymbol{x};\theta) = \exp\{\sum_{s \in V} \theta_s x_s + \sum_{(s,t)\in E} \theta_{st} x_s x_t - \Phi(\theta)\}.$$

Then the potential functions are,

$$\phi_\alpha(\boldsymbol{x}) = \begin{cases} x_s & \alpha = s \\ x_s x_t & \alpha = (s,t) \in E, \end{cases}$$

and the dual variables are given by:

$$\begin{aligned} \eta_s &= \mathbb{E}_\theta[x_s] = p(x_s = 1; \theta) \\ \eta_{st} &= \mathbb{E}_\theta[x_s x_t] = p(x_s = 1, x_t = 1; \theta). \end{aligned}$$

I-Projections onto Flat Manifolds

We now have two sets of parameters that both parameterize the same manifold of distributions. In general, these two sets of parameters are linked by the non-linear map Λ. Due to this nonlinearity, a submanifold parameterized by an affine, or flat, subspace of the θ-parameters, will not correspond to a flat parameterization in the η-parameters, and vice versa. Therefore we have two types of "flat" manifolds: manifolds flat in the θ parameters, and manifolds flat in the η parameters.

Definition 1 *The image of an affine subset of Θ is called an e-flat manifold, denoted $\mathcal{M}_e$, while the image of an affine subset of the η parameter is called an m-flat manifold, denoted $\mathcal{M}_m$.*

Projection onto these flat manifolds forms two of the canonical optimization problems of information geometry, and is called I-projection. To give an idea of what such a projection might mean, consider again a collection of $\{0,1\}$ random variables. As given above, the minimal representation is exponential in dimension (2^N). We might want to approximate the true distribution using a distribution with, say, only pairwise interactions. Obtaining a "best" such approximation, is nothing more than performing a "projection" (for some metric) onto the e-flat manifold:

$$\mathcal{M}_e = \{p(\boldsymbol{x}; \theta) \,|\, \theta_\alpha = 0, |\alpha| \geq 3\}.$$

The metric used for I-projection is the Kullback–Leibler divergence,

$$D(p||q) = \sum_{\boldsymbol{x}} p(\boldsymbol{x})[\log p(\boldsymbol{x}) - \log q(\boldsymbol{x})],$$

which, while not symmetric in its arguments, has the important property of being nonnegative, and equal to zero only when $p = q$. Using the notation and duality established above, we can compute two expressions for the KL divergence, in terms of our exponential and mean parameters. These highlight some features of the KL divergence, in particular, its convexity in the second argument. We have:

$$\begin{aligned} D(\theta||\theta^*) &= \eta^T(\theta - \theta^*) + \Phi(\theta^*) - \Phi(\theta) \\ &= (\theta^*)^T(\eta^* - \eta) + \Psi(\eta) - \Psi(\eta^*). \end{aligned}$$

Projection onto an e-flat manifold, then, is given by the optimization,

$$\begin{cases} \min_\theta D(\theta^*||\theta) \\ \text{s.t. } \theta \in \mathcal{M}_e. \end{cases}$$

This is a convex objective function, subject to linear constraints. Computing the gradient, $\nabla_\theta D(\theta^*||\theta) = \eta - \eta^*$, we see that the defining condition for the global minimum $\hat{\theta}$ is thus

$$[\eta^* - \hat{\eta}]^T[\theta - \hat{\theta}] = 0.$$

We note that while mean field, as introduced in section 2 above, involves a "projection" onto an e-flat manifold, the optimization is over the first argument rather than the second, and hence is not an I-projection. Suppose $\hat{\theta}$ is the result of a projection of θ^* onto an e-flat manifold. Then using the fact that the Jacobian of the inverse mapping Λ^{-1} is the inverse Fisher matrix, we can see that the m-geodesic joining the two points, must be orthogonal to the e-flat manifold. We see this as follows. Consider an m-geodesic joining the points θ^* and $\hat{\theta}$:

$$\theta(t) = \Lambda^{-1}(\hat{\eta} + t[\eta^* - \hat{\eta}]).$$

We compute the tangent of this geodesic (curved in the exponential parameters because of the nonlinearity of the mapping Λ) to be $G^{-1}(\hat{\eta})[\eta^* - \hat{\eta}]$. Thus taking an inner product on the manifold, using the tangent inner product computed above, we find

$$\langle [\theta - \hat{\theta}], G^{-1}(\hat{\eta})[\eta^* - \hat{\eta}] \rangle_{G(\hat{\theta})} = [\eta^* - \hat{\eta}]^T [\theta - \hat{\theta}].$$

But by the gradient condition above, we see that this vanishes for all $\theta \in \mathcal{M}_e$. In other words, I-Projection onto an e-flat manifold can be accomplished by following an m-flat geodesic, and vice versa. This is important in the sequel.

4 Tree Reparameterization

As discussed briefly in the introduction, and more extensively in last week's papers and presentation, belief propagation (BP) is a technique for computing marginals efficiently and accurately for tree structured distributions, and it works fairly well as a heuristic algorithm on loopy graphs. This success has been explained (see Yedidia et al., [4],[5],[6]) by connecting BP to minimization of the Bethe free energy, an approximation to the Gibbs free energy, the minimization of which yields the desired marginals.

In this section we give the tree reparameterization algorithm, which is strongly related to the BP algorithm. In particular, the tree reparameterization allows us to see BP in a message free way, and understand its behavior through an analysis using the tools developed in the previous sections.

First we consider distributions whose graphs contain only pairwise cliques. From the Hammersley-Clifford theorem, we know that any such distribution may be expressed in the form:

$$p(\boldsymbol{x}) = \frac{1}{Z} \prod_{s \in V} \psi_s(x_s) \prod_{(s,t) \in E} \psi_{st}(x_s, x_t),$$

in other words we have only single node, and edge compatibility functions. After introducing the main ideas, algorithms, and analysis, we consider distributions with higher order cliques, and thus higher order compatibility functions. We introduce an extension of the ideas presented, that not only can handle the higher order structure, but in fact seeks to exploit it.

4.1 The Reparameterization

Consider a distribution with an underlying tree graph structure, as in figure one.

By virtue of the tree factorization, we know that this distribution has another factorization that explicitly reveals the marginals we are after:

$$\frac{1}{Z} \prod_{s \in V} \psi_s(x_s) \prod_{(s,t) \in E} \psi_{st}(x_s, x_t) = \prod_{s \in V} P_s(x_s) \prod_{(s,t) \in E} \frac{P_{st}(x_s, x_t)}{P_s(x_s) P_t(x_t)}.$$

One of the key points of this section, and the idea behind tree reparameterization, is that the factorization that reveals the marginals is just that: a reparameterization, rather than any distribution–altering operation.

We cannot appeal to such a convenient factorization for a general graph with cycles. Yet in the usual spirit of the BP-type algorithms, we seek to extend methods that are exact for tree structures, to distributions on loopy graphs. The idea is, given a distribution on a graph $G = (V, E)$, at each iteration, to consider only the nodes, edges, and associated potential functions, on some imbedded tree $\mathcal{T} \subset E$. Then by the tree factorization, it is possible to find a reparameterization that is locally consistent for the tree structure, and reveals the pseudo-marginals of the tree-distribution with the

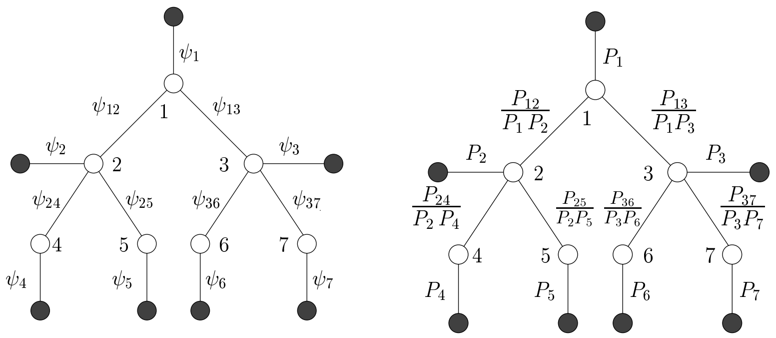

Figure 1: A tree structured distribution. In (a) we see the compatibility functions, and in (b) we have the reparameterized form.

same potential functions. An example best serves to illustrate this procedure. The six variable distribution given in figure 2, initially has the form:

$$p(\boldsymbol{x}) = \frac{1}{Z} \prod_{s=1^6} \psi_s(x_s) \prod_{(s,t)\in E} \psi_{st}(x_s, x_t).$$

Now consider the spanning tree $\mathcal{T} \subset G$ made up of all the nodes, and all the edges except for edges $\{(4,5),(5,6)\}$, as illustrated in the second panel of figure 2. Then we have:

$$\begin{aligned} p(\boldsymbol{x}) &= \frac{1}{Z}\left(\prod_{s=1^6} \psi_s(x_s) \prod_{(s,t)\in\mathcal{T}} \psi_{st}(x_s,x_t)\right) \psi_{45}(x_4,x_5)\psi_{56}(x_5,x_6) \\ &= \frac{1}{Z} p^{\mathcal{T}}(\boldsymbol{x}) r^{\mathcal{T}}(\boldsymbol{x}), \end{aligned}$$

where $p^{\mathcal{T}}(\boldsymbol{x})$ is the product of compatibility functions appearing in the tree $\mathcal{T}$, and $r^{\mathcal{T}}(\boldsymbol{x})$ a product of the residual terms: compatibility functions not in $\mathcal{T}$. Now we can use the tree factorization, to write

$$\begin{aligned} p^{\mathcal{T}}(\boldsymbol{x}) &\propto \prod_{s=1}^{6} \psi_s(x_s) \prod_{(s,t)\in\mathcal{T}} \psi_{st}(x_s,x_t) \\ &\propto \prod_{s=1}^{6} T_s \prod_{(s,t)\in\mathcal{T}} \frac{T_{st}}{T_s T_t}, \end{aligned}$$

where $\{T_s, T_{st}\}$ are the pseudo-marginals of the tree distribution, and correspond to the "belief" at any given step of the iteration, of the true marginals of the full distribution. We call these pseudo-marginals because while they are consistent locally (and hence globally on any tree substructure) they may not correspond to the marginals of any distribution on the full graph with cycles. Then we have, for possibly some different partition function Z, that the original distribution may be written:

$$p(\boldsymbol{x}) = \frac{1}{Z}\left(\prod_{s=1}^{6} T_s \prod_{(s,t)\in\mathcal{T}} \frac{T_{st}}{T_s T_t}\right) \psi_{45}\psi_{56}.$$

Again we stress that the underlying distribution has not been altered, we have merely reparameterized it. Choosing a different tree structured subgraph of G (one not contained in the previous choice $\mathcal{T}$) we repeat. This algorithm terminates if for every embedded tree graph of G, the current

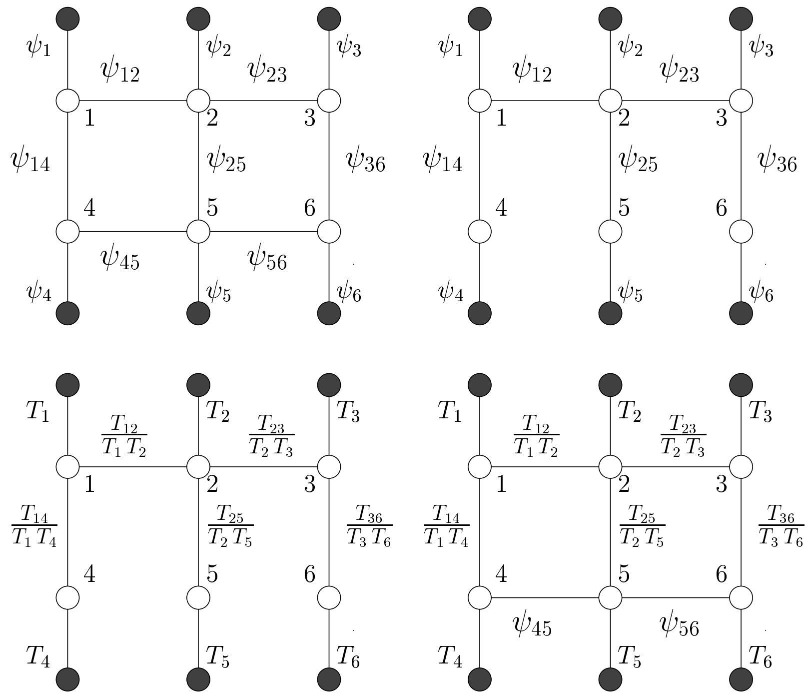

Figure 2: Here we have an example of a graph with loops. Removing two of the edges we are left with a tree structure, and on that we can perform the tree reparameterization. At the end of the iteration, note that the distribution is unaltered.

parameterization is locally consistent. Two of the results of this section are that (i) there always exists a reparameterization that is locally consistent for every embedded tree structure, and (ii) the fixed points of this algorithm correspond to a minimization of a Bethe free energy. In fact it is possible to show that the BP algorithm may be viewed as a tree reparameterization algorithm, not over spanning trees, but over two node trees, i.e. over edges.

We can translate this idea of reparameterization into a general procedure on our exponential families, as defined in the previous section. The machinery developed there, enables an enlightening look at what is going on behind these parameterizations, and in addition, will facilitate a deeper analysis of the success and failure of this algorithm, and along with it, the BP algorithm. Recall we have:

$$\begin{aligned}
p(\boldsymbol{x};\theta) &= \exp\{\sum_{\alpha}\theta_\alpha\phi_\alpha(\boldsymbol{x}) - \Phi(\theta)\}, \\
\Phi(\theta) &= \log\left(\sum_{\boldsymbol{x}}\exp\{\sum_{\alpha}\theta_\alpha\phi_\alpha(\boldsymbol{x})\}\right).
\end{aligned}$$

Here, we choose a minimal reference family of independent functions, but rather than use and work with the minimal family of independent functions, we choose in addition, an overcomplete family. This will allow us to express the reparameterization of the algorithm in terms of the exponential and mean parameters. Recall the key point of the reparameterization: the operation is nothing more than a reparameterization, and thus does not change the overall distribution, only its representation. Here, we suppose that all random variables are m-valued (finite valued). We define a minimal family in an analogous way as we did for binary valued random variables in the previous section:

$$\begin{aligned}
\mathcal{R}(s) &:= \{x_s^a | a = 1,\ldots,m-1\}, \;\; s \in V, \\
\mathcal{R}(s,t) &:= \{x_s^a x_t^b | a,b = 1,\ldots,m-1\}, \;\; (s,t) \in E.
\end{aligned}$$

Now for the overcomplete family, let $s, t \in V$ run over indices of the nodes of the graph, and j, k run over the m possible values of the random variables. Let

$$\mathcal{A} := \{(s;j), (st;jk) \,|\, s,t \in V, (s,t) \in E, \quad 1 \le j,k \le m\}.$$

Then we can define our overcomplete family,

$$\begin{aligned} \phi_\alpha(\boldsymbol{x}) &= \delta(x_s = j), \text{ for } \alpha = (s;j) \\ \phi_\alpha(\boldsymbol{x}) &= \delta(x_s = j)\delta(x_t = k), \text{ for } \alpha = (st;jk). \end{aligned}$$

For a hint behind the motivation of such an overcomplete family, recall that the dual parameters are given as expectations of the potential functions. Thus for such an exponential family, the dual parameters $\{\eta_\alpha\}$ are given by

$$\begin{aligned} \eta_{s;j} &= P_{s;j} = \mathbb{E}_\theta[\delta(x_s = j)], \\ \eta_{st;jk} &= P_{st;jk} = \mathbb{E}_\theta[\delta(x_s = j)\delta(x_t = k)]. \end{aligned}$$

In other words, the dual parameters are the true marginal probabilities we wish to compute:

$$\boldsymbol{P} = \Lambda(\theta).$$

The computation of $\Lambda(\theta)$ is, in general, difficult. The reparameterizations can be seen as approximations of $\Lambda(\theta)$.

We can express each function of our overcomplete representation as a linear combination of the functions in the minimal family, for example,

$$\delta(x_s = j) = \prod_{k \ne j} \frac{(k - x_s)}{(k - j)},$$

and similarly for the edge potentials. Now consider some particular distribution in this exponential family. Denote the vector of coefficients in the minimal representation by γ. By definition of the minimal representation, this γ must be unique. Then because any element of the overcomplete representation is a linear combination of functions in the minimal representation, we have:

$$\begin{aligned} \mathcal{M}_\gamma &= \{\theta \,|\, p(\boldsymbol{x};\theta) = p(\boldsymbol{x};\gamma)\} \\ &= \{\theta \,|\, A\theta = \gamma\}, \end{aligned}$$

for some matrix A. In other words, the set of vectors θ that correspond to a particular distribution, forms an e-flat manifold in $\mathbb{R}^{d(\theta)}$, of dimension $(d(\theta) - d(\gamma))$.

The point is that expressing tree reparameterization in terms of the parameters θ corresponding to the overcomplete representation given above, we can regard the reparameterization as an operation in the e-flat manifold that corresponds to the distribution whose marginals we wish to determine. This underscores yet again the important fact that reparameterization leaves the distribution unchanged at each iteration.

4.2 Geometry of TRP updates

We have seen already that reparameterization does not change the actual distribution, and therefore may be seen as moving within the e-flat manifold of parameter vectors θ for which the corresponding distribution matches the given distribution. Furthermore, we have seen that computing the exact marginals vector, $\boldsymbol{P}$, amounts to computing the exact dual parameters $\Lambda(\theta)$.

We now want to explicitly describe the reparameterization algorithm in terms of the e-flat manifold. We know from the previous section, that the fixed points of the reparameterization algorithm must be locally consistent, in the sense of node and edge marginals. Thus they must belong to the set:

$$\text{TREE}(G) := \{\boldsymbol{T} \,|\, \boldsymbol{T} \in (0,1)^{d(\theta)}, \sum_j T_{s;j} = 1 \;\; \forall s \in V; \; \sum_k T_{st;jk} = T_{s;j}, \;\; \forall (s,t) \in E\}.$$

For a tree with cycles, as implied by the tree factorization, any element, $\boldsymbol{T} \in \mathrm{TREE}(G)$ corresponds (uniquely) to a full distribution. This is not the case for loopy graphs. Thus the elements of $\mathrm{TREE}(G)$ are called, as above, pseudo-marginal vectors. Next we define a map from pseudo-marginals to exponential parameters:

$$[\Theta(\boldsymbol{T})]_\alpha = \begin{cases} \log T_{s;j} & \text{if } \alpha = (s;j) \in \mathcal{A} \\ \log \left[T_{st;jk} / \left(\sum_j T_{st;jk}\right) \left(\sum_k T_{st;jk}\right)\right] & \text{if } \alpha = (st;jk) \in \mathcal{A}. \end{cases}$$

We define the analogous set $\mathrm{TREE}(G)$ for the exponential parameters:

$$\mathrm{EXPTREE}(G) := \{\theta \,|\, \theta = \Theta(\boldsymbol{T}) \text{ for some } \boldsymbol{T} \in \mathrm{TREE}(G)\}.$$

Note that $\mathrm{EXPTREE}(G)$ is defined by linear constraints on the mean (or dual) parameters, and therefore it is an m-flat manifold. This will be important in the sequel. Now, if G is a tree, then for any $\boldsymbol{T} \in \mathrm{TREE}(G)$, $\boldsymbol{T}$ corresponds to actual marginals $\boldsymbol{P}$, and thus

$$\Lambda(\Theta(\boldsymbol{P})) = \boldsymbol{P}.$$

For a loopy graph G, this equality no longer holds. In other words, the operator $(\Lambda \circ \Theta)$ is the identity operator on the marginal probability vectors for trees, but not for general graphs. Consider the operator

$$\mathcal{R}(\theta) = (\Theta \circ \Lambda)(\theta).$$

This mapping sends exponential parameters to exponential parameters. The important property is that in our overcomplete parameter space, the e-flat manifold is mapped to itself when the underlying graph is a tree. In other words, if θ is the exponential parameter vector of a tree distribution, then

$$\mathcal{R}(\theta) = \hat{\theta} \in \mathcal{M}(\theta).$$

We are now ready to express tree reparameterization as an operation on the e-flat manifold of exponential parameters. First, fix a set of spanning trees: $\{\mathcal{T}^0, \ldots, \mathcal{T}^{L-1}\}$. Let E^i denote the edge set of tree i. The update according to tree i, will involve only the elements of θ corresponding to the partial index set $\mathcal{A}^i = \{(s;j), (st;jk) | s \in V, (s,t) \in E^i\}$. Similarly, we define constraint sets $\mathrm{TREE}^i(G)$ and $\mathrm{EXPTREE}^i(G)$ that only impose marginalization constraints one the edges belonging to tree i. Note of course that

$$\begin{aligned} \bigcap_i \mathrm{TREE}^i(G) &= \mathrm{TREE}(G) \\ \bigcap_i \mathrm{EXPTREE}^i(G) &= \mathrm{EXPTREE}(G). \end{aligned}$$

We define operators $\Lambda^i, \Theta^i, \mathcal{R}^i$ as above, but defined for the tree $\mathcal{T}^i$. In addition, we define the projection and injection operators, associated to the i^{th} tree

$$\begin{aligned} \Pi^i(\theta) &= \{\theta_\alpha | \alpha \in \mathcal{A}^i\} \\ \mathcal{I}^i(\Pi^i(\theta)) &= \begin{cases} \theta_\alpha & \text{if } \alpha \in \mathcal{A}^i \\ 0 & \text{if } \alpha \notin \mathcal{A}^i. \end{cases} \end{aligned}$$

Given a particular tree $\mathcal{T}^i$, the reparameterization operator takes the form (here I is the identity operator):

$$\mathcal{Q}^i(\theta) = \mathcal{I}^i(\mathcal{R}^i(\Pi^i(\theta))) + [I - \mathcal{I}^i \circ \Pi^i](\theta).$$

This looks more complicated than it truly is. Recall that given a particular spanning tree $\mathcal{T}$, we wrote the reparameterization operation as the identity operator on the compatibility functions **not** corresponding to a node or edge in the tree, and a tree factorization on the compatibility functions corresponding to nodes and edges in the tree:

$$
\begin{aligned}
p(\boldsymbol{x}) &\propto \prod_{s \in V} \psi_s(x_s) \prod_{(s,t)\in E} \psi_{st}(x_s, x_t) \\
&\propto \left(\prod_{s \in V} \psi_s(x_s) \prod_{(s,t)\in \mathcal{T}} \psi_{st}(x_s, x_t) \right) \left(\prod_{(s,t)\in E-\mathcal{T}} \psi_{st}(x_s, x_t) \right) \\
&\propto \left(\prod_{s \in V} T_s(x_s) \prod_{(s,t)\in \mathcal{T}} \frac{T_{st}(x_s, x_t)}{T_s(x_s)T_t(x_t)} \right) \left(\prod_{(s,t)\in E-\mathcal{T}} \psi_{st}(x_s, x_t) \right).
\end{aligned}
$$

This is then precisely what we have in the operator equation above. Initializing the sequence using the original graph functions $\{\psi_s\}$ and $\{\psi_{st}\}$, we have

$$
\theta^0_\alpha = \begin{cases} \log \psi_{s;j} & \text{if } \alpha = (s;j), \\ \log \psi_{st;jk} & \text{if } \alpha = (st;jk), \end{cases}
$$

and then

$$
\theta^{n+1} = \mathcal{Q}^{i(n)}(\theta^n).
$$

We also consider a relaxation involving a stepsize $\lambda^n \in (0,1]$:

$$
\theta^{n+1} = \lambda^n \mathcal{Q}^{i(n)}(\theta^n) + (1-\lambda^n)\theta^n.
$$

Note that as the relaxed update is a convex combination of the original point and $\mathcal{Q}^{i(n)}(\theta^n)$, then for any λ^n, again the point θ^{n+1} will be contained in the e-flat manifold. By definition of the operator $\mathcal{Q}^i$ and the set $\mathrm{EXPTREE}^i(G)$, we have that

$$
\mathcal{Q}^i(\theta) \in \mathrm{EXPTREE}^i(G),
$$

and the goal of the reparameterization is to obtain a point in the intersection $\bigcap_i \mathrm{EXPTREE}^i(G)$ of all the tree consistency constraints. As illustrated in the figure, the update moves within the e-flat manifold, towards an m-flat manifold.

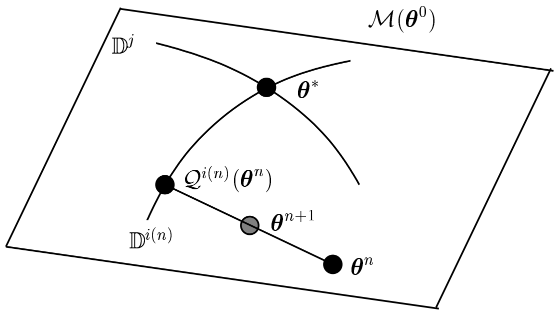

Figure 3: This figure shows the geometry of the tree reparameterization in the exponential parameter coordinate system. Here $\mathcal{M}(\theta^0)$ is the e-flat manifold, and the curved lines denoted $\mathbb{D}^i$, represent the m-flat manifolds EXPTREE(G).

Projection onto an m-Flat Manifold

Recall that belief propagation can be cast as one particular method of minimizing the so-called Bethe free energy – essentially a tree based approximation to the Gibbs free energy. We now show that in the same sense, the tree reparameterization algorithm given above, attempts to minimize a particular approximation to the Kullback–Leibler divergence. This approximation is exact on trees. We define the functions

$$\begin{aligned} \mathcal{G}_s(T_s;\theta_s) &= \sum_j T_{s;j}[\log T_{s;j} - \theta_{s;j}] \\ \mathcal{G}_{st}(T_{st};\theta_{st}) &= \sum_{j,k} T_{st;jk}\left\{\log[T_{st;jk}/(\sum_j T_{st;jk})(\sum_k T_{st;jk})] - \theta_{st;jk}\right\}, \end{aligned}$$

and the cost functions

$$\mathcal{G}(\boldsymbol{T};\theta) := \sum_{s\in V}\mathcal{G}_s(T_s;\theta_s) + \sum_{(s,t)\in E}\mathcal{G}_{st}(T_{st};\theta_{st}) = \sum_{\alpha\in\mathcal{A}} T_\alpha[\Theta(\boldsymbol{T}) - \theta]_\alpha.$$

and

$$\mathcal{G}^i(\Pi^i(\boldsymbol{T});\Pi^i(\theta)) := \sum_{s\in V}\mathcal{G}_s(T_s;\theta_s) + \sum_{(s,t)\in E^i}\mathcal{G}_{st}(T_{st};\theta_{st}).$$

When $\boldsymbol{T} \in \text{TREE}(G)$, then this cost function $\mathcal{G}$ is equivalent to Bethe free energy. The functions $\mathcal{G}^i$ are related to the KL divergence by

$$D(\Theta^i(\Pi^i(\boldsymbol{T}))||\Pi^i(\theta)) = \mathcal{G}^i(\Pi^i(\boldsymbol{T});\Pi^i(\theta)) + \Phi(\Pi^i(\theta)).$$

Minimizing the actual KL divergence on the left hand side of the equation above, yields the actual marginals on the spanning tree $\mathcal{T}^i$. This is because for a tree, we have

$$\boldsymbol{P}^i = \mathcal{T}^i(\Lambda^i(\Pi^i(\theta))) \in \text{TREE}(G).$$

By the equation above relating KL divergence and $\mathcal{G}^i$, we see that the same holds when minimizing $\mathcal{G}^i$. This argumentation breaks down, however, when we consider the original graph, precisely because Λ and Θ are no longer inverses. Nevertheless, we show that reparameterization with respect to spanning tree $\mathcal{T}^i$, is equivalent to a projection onto the tree constraint set $\text{TREE}^i(G)$, where the projection is given by minimizing the cost function $\mathcal{G}$. This will show that the tree reparameterization is a successive projection technique, where the tree reparameterization with respect to tree $\mathcal{T}^i$ corresponds to a projection onto constraint set $\text{TREE}^i(G)$, an m-flat manifold, defined by the constrained minimization of $\mathcal{G}^i$.

Theorem 1 (Pythagorean Relation) *If the relaxed reparameterization sequence $\{\theta^n\}$ is bounded, then for any i, and any $\boldsymbol{U} \in TREE^i(G)$,*

$$\mathcal{G}(\boldsymbol{U};\theta^{n+1}) = \mathcal{G}(\boldsymbol{U};\theta^{n+1}) + \lambda^n\mathcal{G}(\mathcal{T}^i(\Lambda^i(\Pi^i(\theta)));\theta^n).$$

In particular, for the unrelaxed tree reparameterization update,

$$\mathcal{G}(\boldsymbol{U};\theta^{n+1}) = \mathcal{G}(\boldsymbol{U};\theta^{n+1}) + \mathcal{G}(\mathcal{T}^i(\Lambda^i(\Pi^i(\theta)));\theta^n).$$

We can interpret this theorem geometrically in the dual parameter coordinates, by considering the pseudo-marginal vector $\boldsymbol{T}^n$ corresponding to θ^n, by the figure below. This figure is the view in the dual coordinates, analogous to figure 3, which shows the reparameterization operation in the exponential parameter coordinates.

Recall that reparameterization seeks ultimately to find a point in the intersection of all the tree consistency constraint sets $\text{TREE}^i(G)$. There are various techniques that take this form of successive projection. The Pythagorean relation established in the Theorem above, is important, as

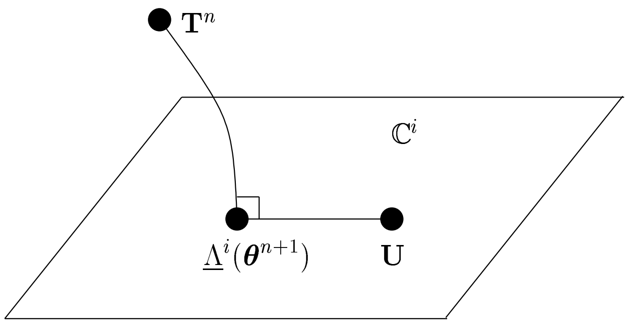

Figure 4: This picture, analogously to the previous figure, shows the geometry of tree reparameterization in the mean, or dual, parameters. Here, $\mathbb{C}^i$ denotes the m-flat manifold of the i^{th} tree constraint.

it helps show that even though the function $\mathcal{G}$ is not a so-called Bregman distance as, for instance, it is not nonnegative, the tree reparameterization procedure has fixed points that satisfy the necessary conditions to be a local minimum of $\mathcal{G}$ subject to the full set of tree consistency constraints TREE(G). The following Theorem states that fixed points of the tree reparameterization algorithm always exist, and then goes on to say that each fixed point θ^* satisfies the stationarity (necessary) constraints to be a local minimum of $\mathcal{G}$, and that in addition, that each fixed point corresponds to a unique pseudomarginal vector $\boldsymbol{T}^* \in$ TREE(G).

Theorem 2 *If the tree reparameterization update algorithm is performed with stepsize of size at least $\varepsilon > 0$, then,*

(i) *Fixed points exist, and coincide with the fixed points of BP.*

(ii) *If θ^* is a fixed point generated by a sequence of iterations, then it is fixed under all tree operators: $\theta^* = \mathcal{Q}^i(\theta^*)$ for all i, and it is associated to some (unique) pseudomarginal vector $\boldsymbol{T}^* \in$ TREE(G).*

(iii) *For a fixed point θ^* and corresponding pseudo-marginal vector $\boldsymbol{T}^*$ as in the item above, $\boldsymbol{T}^*$ satisfies the necessary conditions to be a minimum of $\mathcal{G}$ over TREE(G):*

$$\sum_{\alpha \in \mathcal{A}} \frac{\partial \mathcal{G}}{\partial T_\alpha}(\boldsymbol{T}^*; \theta^0)[\boldsymbol{U} - \boldsymbol{T}^*]_\alpha = 0.$$

It is important to note that while these results are phrased specifically as pertaining to the fixed points of the tree reparameterization, in fact they all apply to any fixed points of the Bethe free energy, whether obtained by tree reparameterization, message passing, or some other algorithm (that may, for instance, not stay strictly within the e-flat manifold).

4.3 Extensions to Hypergraphs

Up to this point, we have restricted our attention to graphs whose largest cliques are size two (single edges). In this section, we consider two extensions: first, we consider how to deal with graphs which have higher order cliques, and which have, therefore, potential functions corresponding to three or more nodes. In addition, we consider how explicitly grouping, or clustering groups of nodes together, may be used to obtain higher order approximations, naturally generalizing the tree reparameterization ideas (and the essentially equivalent belief propagation ideas) to higher order models.

Indeed, the main ideas used here amount to simply obtaining the appropriate extensions of notions used in the development thus far. Instead of trees, we use the analogous concept of hypertrees, to define local consistency, reparameterization, and most importantly, tractable approximations to entropy and energy terms. Thus, for example, rather than the Bethe approximation to entropy,

$$H_{Bethe}(\boldsymbol{T}) = \sum_{s \in V} H_s(T_s) + \sum_{(s,t) \in E} I_{st}(T_{st}),$$

which (as seen in section 2) is used for an approximation to the Gibbs free energy which as discussed in section 4, is equivalent in a precise sense to the $\mathcal{G}$–variational problem of tree reparameterization, we will have higher order entropy approximations. Thus in this section, we work to develop the appropriate generalizations, formulate the generalized problem, and then leave the remaining details to the reader.

The important generalizing notion here is that of a hyperedge, and a hypertree. A graph G consists of vertices, and edge sets. An edge set may be thought of as a collection of pairs of nodes. A hyperedge h is a subset of the node set V of any cardinality. A hypergraph $G_{\mathrm{HYP}} = (V, E)$, is a collection of nodes V, and hyperedges E. Hyperedges form a partially ordered set (poset) where inclusion is the partial order (i.e. $h \leq g$ if and only if $h \subseteq g$ as a subset of nodes). We use this partial order to represent the hypergraph graphically, with what is known in the poset literature (see, e.g. R. Stanley's book [3]) as a Hasse diagram. Here, the nodes are all the hyperedges, and there is a directed edge between nodes corresponding to hyperedges h, g if and only if $h < g$, and $h \leq f \leq g$ implies $f = h$ or $f = g$.

While distinct edges of an ordinary graph may overlap at most at a single endpoint, distinct hyperedges may have more substantial intersection. Therefore the analogous concepts of cyclic and acyclic hypergraphs are slightly more difficult to define.

Definition 2 *A tree decomposition of a hypergraph G_{HYP} is a tree structured graph whose nodes are all the maximal hyperedges of the hypergraph, and the edges are in one to one correspondence with the separator hyperedges, that are in the intersection of two maximal hyperedges.*

Note that any hypergraph G_{HYP} has a tree decomposition, in fact it has a chain decomposition. However remember that our aim is for a tree decomposition to correspond to a factorization of the distribution. As is well known from the junction–tree literature, this can only happen if the tree satisfies the running intersection property, defined below.

Definition 3 *A particular tree decomposition of a hypergraph G_{HYP} satisfies the running intersection property if for any two maximal nodes $h_{\max}, g_{\max}$, the unique path joining them contains the intersection $g_{\max} \cap h_{\max}$.*

Finally, we can define,

Definition 4 *A hypergraph is acyclic if it admits a tree decomposition satisfying the running intersection property.*

The motivation behind these definitions is to specify when the junction tree factorization, which is the natural generalization of the tree factorization we used heavily above, applies. Recall that the junction tree representation says that an acyclic hypergraph with maximal hyperedges $E_{\max}$, and hyperedges corresponding to separator sets in a tree decomposition E_{sep}, admits a factorization of the form

$$p(\boldsymbol{x}) = \frac{\prod_{h \in E_{\max}} P_h(\boldsymbol{x}_h)}{\prod_{g \in E_{sep}} [P_g(\boldsymbol{x})]^{d(g)-1}},$$

where $d(g)$ is the number of maximal hyperedges containing separator set g. But in fact this factorization may be further unraveled if we consider the subgraph where the hyperedges corresponding to separator sets are now taken to be the maximal hyperedges. Thus, we obtain an equivalent factorization, as follows:

Proposition 1 *If the hyperedge set of some hypertree contains all separator sets of some particular tree decomposition (with the running intersection property) then we have the following factorization:*

$$p(\boldsymbol{x}) = \prod_{h \in E} \phi_h(\boldsymbol{x}_h),$$

where we define,

$$\log \phi_h(\boldsymbol{x}_h) = \log P_h(\boldsymbol{x}_h) - \sum_{g \leq h} \log \phi_g(\boldsymbol{x}_g).$$

The Möbius inversion function provides a generalization to the well–known inclusion-exclusion formula (again, see Stanley [3] for a nice introduction and development). Applying this, we obtain

$$\log \phi_h(\boldsymbol{x}) = \sum_{g \leq h} \mu(g, h) \log P_g(\boldsymbol{x}_g),$$

which when substituted into our factorization above, yields a factorization in terms of marginals:

$$\begin{aligned}
\log p(\boldsymbol{x}) &= \sum_{h \in E} \log \phi_h(\boldsymbol{x}_h) \\
&= \sum_{h \in E} \sum_{g \leq h} \mu(g, h) \log P_g(\boldsymbol{x}_g) \\
&= \sum_{h \in E} c(h) \log P_h(\boldsymbol{x}_h),
\end{aligned}$$

where we have defined the *overcounting numbers* as

$$c(h) = \sum_{f \geq h} \mu(h, f).$$

One immediate consequence of our factorization, as in the case of the ordinary tree factorization, is that we have a decomposition of the entropy:

$$H_{hyper}(\boldsymbol{x}) = \sum_{h \in E} c(h) H_h(\boldsymbol{x}_h).$$

Such decompositions that are exact for hypertrees, will allow us to obtain higher order model approximations to the entropy of general hypergraphs (that may not be hypertrees).

Kikuchi Clustering

Given any hypergraph $\tilde{G}_{\mathrm{HYP}}$, we may wish to further cluster the nodes, forming a new hypergraph G_{HYP} with possibly higher order hyperedges. The condition governing what clusterings are allowed is the precise expression of the intuitively clear concept that each hyperedge h in the original hypergraph, must appear the correct number of times in the augmented hypergraph. This is the case if and only if

$$\sum_{g \geq h} c(g) = 1,$$

for every hyperedge g in the original hypergraph (that may or may not be included in the augmented hypergraph). This is known as the **single counting criterion**.

Assuming the given hypergraph is not a hypertree, the approach is exactly parallel to the case of graphs with cycles. We further cluster $\tilde{G}_{\mathrm{HYP}}$ to obtain an augmented hypergraph, G_{HYP}, and then approximating the entropy using only interactions to the level of the hypergraph:

$$H_{approx}(\boldsymbol{x}) = \sum_{h \in E} c(h) H_h(\boldsymbol{x}_h),$$

we define the variational problem, analogous to the variational problems defined for loopy graphs:

$$\min_{\boldsymbol{T}} \left\{ -H_{approx}(\boldsymbol{T}) - \sum_{\tilde{h} \in \tilde{E}} \sum_{\boldsymbol{x}_{\tilde{h}}} T_{\tilde{h}}(\boldsymbol{x}_{\tilde{h}}) \log \psi_{\tilde{h}}(\boldsymbol{x}_{\tilde{h}}) \right\},$$

where we require the marginal vector $\boldsymbol{T}$ to satisfy local consistency conditions, again exactly analogously to the situation for loopy graphs and tree consistency conditions denoted TREE(G):

$$\boldsymbol{T} \in \mathrm{HYP}(G_{\mathrm{HYP}}) = \left\{ \boldsymbol{T} \mid \sum_{\boldsymbol{x}'_h} T_h(\boldsymbol{x}'_h) = 1, \ \forall h \in E, \ \sum_{\{\boldsymbol{x}'_h : \boldsymbol{x}'_g = \boldsymbol{x}_g\}} T_h(\boldsymbol{x}'_h) = T_g(\boldsymbol{x}_g), \ \forall g < h \right\}.$$

At this point, we have obtained the generalization of the tree factorization, and the associated local consistency constraints that enable us to set up the variational problem naturally generalizing the variational problem whose stationary points give the fixed points of both the BP and the tree reparameterization algorithms. From here then, obtaining the explicit reparameterization expressions, fixed point characterizations, and error approximations, is a matter of notational care.

4.4 Analysis of Performance

In this section we consider the error of the approximation. We give some bounds on the approximation error on single node marginals. These bounds rely in turn on upper bounds on the log partition function, $\Phi(\theta)$, of the original graph with cycles (as computing the function exactly is as hard as the problem the BP and TRP approximations seek to avoid solving) which is the subject of the next section.

We wish to compute an upper bound on the error of the single node marginals: $|P_{s;j} - T^*_{s;j}|$. Note that in general, for any function $f(\boldsymbol{x})$, and for any distributions $p(\boldsymbol{x};\theta), p(\boldsymbol{x};\hat{\theta})$, we have the exact expression:

$$\mathbb{E}_{\hat{\theta}}[f(\boldsymbol{x})] = \mathbb{E}_{\theta}\left[\exp\{\sum_{\alpha} (\hat{\theta} - \theta)_\alpha \phi_\alpha(\boldsymbol{x}) + \Phi(\theta) - \Phi(\hat{\theta})\} f(\boldsymbol{x}) \right],$$

and thus taking $\hat{\theta} = \theta^*$ and $\theta = \Pi^i(\theta^*)$, and $f(\boldsymbol{x}) = \delta(x_s = j)$, and rearranging, we have

$$P_{s;j} - T^*_{s;j} = \mathbb{E}_{\Pi^i(\theta^*)}\left[(\exp\{ \sum_{\alpha \notin \mathcal{A}^i} \theta^*_\alpha \phi_\alpha(\boldsymbol{x}) - \Phi(\theta^*)\} - 1)\delta(x_s = j) \right].$$

This is an exact expression for the error of the marginal. Not surprisingly, computing this exactly is as difficult as the original inference problem. Instead, we approximate the error to the log marginals: $E_{s;j} = \log T^*_{s;j} - \log P_{s;j}$. We don't go into the derivation of this result, but rather refer the reader to Chapter 3 in [7]. We define the quantity

$$\Delta^i_{s;j} := \sum_{\alpha \in \mathcal{A} \ \mathcal{A}^i} \theta^*_\alpha \mathrm{cov}_{\Pi^i(\theta^*)}\{\delta(x_s = j), \phi_\alpha(\boldsymbol{x})\}.$$

Using this we have the following

Theorem 3 *Let θ^* be a fixed point of TRP/BP, with associated marginals $\boldsymbol{T}^*$, where $\boldsymbol{P}$ is the true marginal vector. Then, we have the following bounds,*

$$\begin{aligned} E_{s;j} &\leq D(\Pi^i(\theta^*) || \theta^*) - \frac{\Delta^i_{s;j}}{T^*_{s;j}} \\ E_{s;j} &\geq \log T^*_{s;j} - \log\left[1 - (1 - T^*_{s;j}) \exp\left\{ -D(\Pi^i(\theta^*) || \theta^*) - \frac{\Delta^i_{s;j}}{1 - T^*_{s;j}} \right\} \right]. \end{aligned}$$

The point we wish to make about this approximation is that it is tractable, with one exception: while computing the quantity $\Delta^i_{s;j}$ is tractable, computing the KL divergence term $D(\Pi^i(\theta^*)||\theta^*)$ involves computing the log partition function $\Phi(\theta^*)$ associated with the original loopy graph G. For both bounds, to obtain something useful, it is enough to obtain (interesting) upper bounds on the log partition function $\Phi(\theta^*)$. This is the subject of Chapter 7 of [7], and of the next section, to which we now turn.

5 Upper Bounds on the Partition Function

In this section we develop some techniques for obtaining upper bounds on the log partition function. We have already seen in the previous section how such bounds may be useful. For ease of notation, we assume that the random variables are all pairwise.

Again at the core of these upper bounds and their derivations, lies a property of the log partition function on which we have relied heavily before: $\Phi(\theta)$ is convex in its argument θ.

We want to obtain an approximation for $\Phi(\theta^*)$, the log partition function evaluated at some point θ^* (for instance, at a fixed point of the tree reparameterization). Let $\mathfrak{T}$ denote the set of all spanning trees in a graph G, and let

$$\boldsymbol{\theta} = \{\theta(\mathcal{T}) \,|\, \mathcal{T} \in \mathfrak{T}\},$$

be the collection of exponential parameter vectors corresponding to some spanning tree in the collection $\mathfrak{T}$. Let μ be some distribution on $\mathfrak{T}$, and for $e \in E$ some edge of the graph, let μ_e be the resulting edge probability. Given a distribution μ on a collection of Further define

$$\mathcal{A}(\theta^*) := \{(\boldsymbol{\theta}, \mu) \,|\, \mathbb{E}_\mu(\theta(\mathcal{T})) = \theta^*\}$$

By convexity of $\Phi(\theta)$, a simple application of Jensen's inequality yields the following bound:

$$\Phi(\theta^*) \le \mathbb{E}_\mu[\Phi(\theta(\mathcal{T}))] = \sum_{\mathcal{T}\in\mathfrak{T}} \mu(\mathcal{T})\Phi(\theta(\mathcal{T})).$$

These bounds depend both on the choice of spanning trees $\boldsymbol{\theta}$, and also on the distribution μ. Therefore we wish to optimize over both of these choices. First we consider optimizing over $\boldsymbol{\theta}$, and then over μ.

Fixing μ, therefore, we have the optimization problem:

$$\begin{cases} \min_{\boldsymbol{\theta}} \mathbb{E}_\mu[\Phi(\theta(\mathcal{T}))] \\ \text{s.t.} \quad \mathbb{E}_\mu[\boldsymbol{\theta}] = \theta^*. \end{cases}$$

The objective function is a convex combination of convex functions, and therefore is itself convex. The constraints are linear. Indeed it is not the inherent form of this optimization that poses difficulties: rather it is the problem's size. The size of the vector $\boldsymbol{\theta}$ may grow exponentially, quickly making the problem in this form intractable. Because of the problem's convexity, we know that it satisfies strong duality, and therefore we can consider, equivalently, the dual problem:

$$\begin{cases} \max_\lambda \mathcal{Q}(\lambda;\mu;\theta^*) = -\mathbb{E}_\mu[\Psi(\Pi^{\mathcal{T}}(\lambda))] + \sum_\alpha \lambda_\alpha \theta^*_\alpha \\ \text{s.t.} \quad \lambda \in \mathbb{L}(G), \end{cases}$$

where the constraint set on λ is given by

$$\mathbb{L}(G) = \{\lambda \,|\, 0 \le \lambda_{st} \le \lambda_s \le 1;\ \lambda_s + \lambda_t \le 1 + \lambda_{st}\}.$$

There are only $|V|+|E|$ variables in the resulting optimization, and thus it is tractable. Furthermore, since the objective function is the sum of a linear term and $(-\mathbb{E}_\mu[\Psi(\Pi^{\mathcal{T}}(\lambda))])$ it is strictly concave. Therefore as long as $||\theta^*|| < \infty$, the maximum is achieved in the interior of $\mathbb{L}(G)$, therefore allowing us to characterize the optima of the dual optimization with ordinary gradient conditions (i.e. we do not need to explicitly consider Lagrange multipliers associated to the constraints of the set $\mathbb{L}(G)$).

Consider again the entropy term that appears in the optimization: $\Psi(\Pi^{\mathcal{T}}(\lambda))$. This is an entropy over a tree structured distribution. Therefore, by the tree factorization, the entropy is only a function of the single node and edge marginals:

$$\Psi(\Pi^{\mathcal{T}}(\lambda)) = -\sum_{s \in V} H_s(\lambda) + \sum_{(s,t) \in E(\mathcal{T})} I_{st}(\lambda),$$

where H and I are the familiar entropy and mutual information functions. The expectation then becomes,

$$\begin{aligned}\mathbb{E}_\mu[\Psi(\Pi^{\mathcal{T}}(\lambda))] &= \sum_{\mathcal{T} \in \mathfrak{T}} \mu(\mathcal{T}) \left\{ -\sum_{s \in V} H_s(\lambda) + \sum_{(s,t) \in E(\mathcal{T})} I_{st}(\lambda) \right\} \\ &= -\sum_{s \in V} H_s(\lambda) + \sum_{(s,t) \in E} \mu_{st} I_{st}(\lambda).\end{aligned}$$

This allows us to efficiently perform the optimization over the distribution μ, which until now we had fixed to some nominal value, in order to analyze the optimization with respect to $\boldsymbol{\theta}$. We now have only $|E|$ values of μ over which we must perform the optimization. These are the marginal edge probabilities. As such, they are quite constrained. They must lie in the polytope:

$$\mathrm{MARG}(G) := \{\boldsymbol{\mu_e} \,|\, \mu_e = \mathbb{E}_\mu[\delta(e \in \mathcal{T})] \text{ for some } \mu; \ \forall e \in E\}.$$

This polytope is known as the spanning tree polytope in the literature, and has a well-known characterization in terms of linear inequalities. This we are able to perform optimization over the set, efficiently. Defining the function

$$\mathcal{F}(\lambda; \boldsymbol{\mu_e}; \theta^*) := -\sum_{s \in V} H_s(\lambda) + \sum_{(s,t) \in E} \mu_{st} I_{st}(\lambda) - \sum_\alpha \lambda_\alpha \theta^*_\alpha,$$

we have the following Theorem giving optimal upper bounds on the log partition function:

Theorem 4 *Define the function*

$$\mathcal{H}(\boldsymbol{\mu_e}; \theta^*) := \min_{\lambda \in \mathbb{L}(G)} \{\mathcal{F}(\lambda; \boldsymbol{\mu_e}; \theta^*)\}.$$

Then, the optimization over the spanning tree polytope gives a jointly optimal upper bound for the log partition function:

$$\Phi(\theta^*) \leq -\max_{\boldsymbol{\mu_e} \in MARG(G)} \mathcal{H}(\boldsymbol{\mu_e}; \theta^*).$$

6 Conclusion

In this paper, closely tracing the development in Chapters 2, 5 and 7 of [7], we have exploited geometric structure and convexity properties, to develop and analyze a tree reparameterization algorithm, or alternatively, a successive projection algorithm, for inference on graphs with cycles. The fundamental ideas used both in the original framework, as well as the extended higher order models, involved solving variational problems made tractable because of the equivalence between local and global consistency on tree structures. Furthermore, these variational problems can be seen to have a particularly nice geometric interpretation, in terms of successive orthogonal projections onto submanifolds parameterized by affine subspaces in two different coordinates. These coordinates were closely linked by the Legendre mapping obtained by exploiting the convexity of the log partition function. This convexity of the log partition function was used in a fundamental manner to obtain the dual parameterization and the relationship between the two parameters. In addition, the convexity was crucial in the final section, allowing us to obtain upper bounds to the partition function.

References

[1] Cover, T.; Thomas, J. "Elements of Information Theory," John Wiley and Sons, New York, 1991.

[2] Kschischang, F.R.; Frey, B.J.; Loeliger, H. "Factor Graphs and the Sum–Product Algorithm," IEEE Trans. Info. Theory, 47(2), February 2001.

[3] Stanley, R.P. "Enumerative Combinatorics," vol. 1, Cambridge University Press, Cambridge, UK, 1997.

[4] Yedidia, J.S.; Freeman, W.T.; Weiss, Y. "Generalized Belief Propagation," In NIPS 13, pages 689-695. MIT Press, 2001.

[5] Yedidia, J.S.; Freeman, W.T.; Weiss, Y. "Understanding Belief Propagation and its Generalizations," Technical Report TR2001-22, Mitsubishi Electric Research Labs, January 2002.

[6] Yedidia, J.S.; Freeman, W.T.; Weiss, Y. "Constructing Free Energy Approximations and Generalized Belief Propagation Algorithms," Information Theory Workshop at Mathematical Sciences Research Institute (MSRI), March 2002.

[7] Wainwright, M.J. "Stochastic Processes on Graphs with Cycles: Geometric and Variational Approaches," Ph.D. Thesis, MIT, January 2002.