10-708: Probabilistic Graphical Models 10-708, Spring 2014

Learning fully observed graphical models

Lecturer: Matthew Gormley *Scribes: Akash Bharadwaj, Devendra Chaplot, Sumeet Kumar*

1 Parameter estimation for fully observed directed graphical models

In the case of fully observed directed graphs, the product form of the joint distribution can be used to decompose the log-likelihood function into a sum of local terms, one per node:

$$\begin{aligned} l(\theta; D) &= log\ p(D|\theta) \\ &= log \prod_n \Big(\prod_i p(x_{n,i}|x_{\pi_i}, \theta_i) \Big) \\ &= \sum_n \Big(\sum_i log\ p(x_{n,i}|x_{\pi_i}, \theta_i) \Big) \\ &= \sum_i \Big(\sum_n log\ p(x_{n,i}|x_{\pi_i}, \theta_i) \Big) \end{aligned}$$

Since the joint probability decomposes into sum of local terms,the maximum likelihood problem decomposes into separate terms such that parameters θ_i appear in different terms, and thus each parameter can be estimated independently:

$$\begin{aligned} \theta^* &= \arg\max_\theta log\ p(D|\theta) \\ &= \arg\max_\theta log \prod_n \Big(\prod_i p(x_{n,i}|x_{\pi_i}, \theta_i) \Big) \\ &= \arg\max_\theta \sum_i \Big(\sum_n log\ p(x_{n,i}|x_{\pi_i}, \theta_i) \Big) \\ \theta_i^* &= \arg\max_{\theta_i} \sum_n log\ p(x_{n,i}|x_{\pi_i}, \theta_i) \end{aligned}$$

This is exactly like learning parameters of several separate small BNs, each of which consists of a node and its parents.

1.1 Example

Consider a bayesian network with 4 nodes as shown in Figure 1(a). It has the following joint probability distribution:

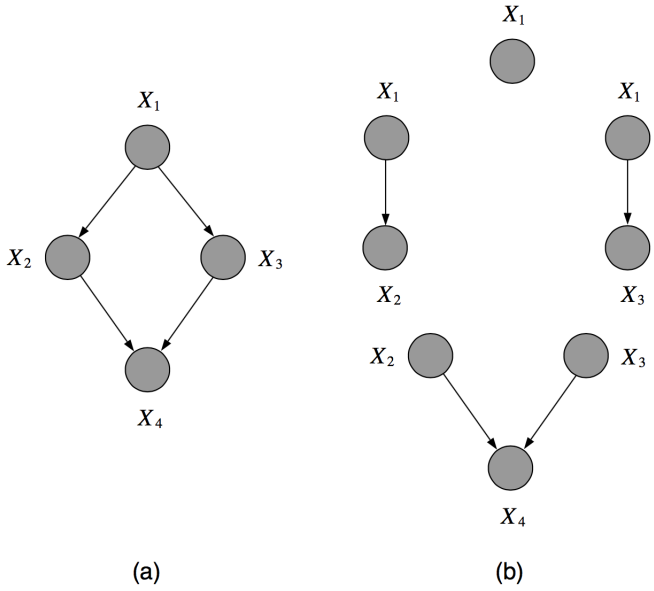

Figure 1: Maximum Likelihood parameter estimation in Bayesian Networks. (a) A Bayesian Network with 4 nodes. (b) Maximum likelihood problem in (a) can be broken down into separate maximum likelihood problem for each node conditioned on its parents

$$p(x|\theta) = p(x_1|\theta_1)p(x_2|x_1,\theta_1)p(x_3|x_1,\theta_1)p(x_4|x_2,x_3\theta_1)$$

Maximum Likelihood estimate of the parameters are calculated as follow:

$$\begin{aligned}\theta^* &= \arg\max_{\theta} log\ p(x_1,x_2,x_3,x_4) \\ &= \arg\max_{\theta} log\ p(x_1|\theta_1) + log\ p(x_2|x_1,\theta_2) + log\ p(x_3|x_1,\theta_3) + log\ p(x_4|x_2,x_3\theta_4)\end{aligned}$$

Once it is expressed as a sum, it is possible to estimate one parameter at a time.

$$\begin{aligned}\theta_1^* &= \arg\max_{\theta_1} log\ p(x_1|\theta_1) \\ \theta_2^* &= \arg\max_{\theta_2} log\ p(x_2|x_1,\theta_2) \\ \theta_3^* &= \arg\max_{\theta_3} log\ p(x_3|x_1,\theta_3) \\ \theta_4^* &= \arg\max_{\theta_4} log\ p(x_4|x_2,x_3,\theta_4)\end{aligned}$$

This decomposition is equivalent to splitting the bayesian network into four small bayesian network corresponding to a node and its parent as shown in Figure 1(b).

Note that marginal distributions such as $p(x_1|\theta_1)$ are often represented by exponential family distributions, while conditional distributions such as $p(x_2|x_1,\theta_2)$, $p(x_3|x_1,\theta_3)$ and $p(x_4|x_2,x_3,\theta_4)$ are conveniently represented using Generalized Linear Models.

2 Parameter estimation for fully observed Undirected Graphical Models

The previous section described how MLE estimates for parameters can be obtained for fully observed **directed** graphical models (bayes nets). In that case, we see that the log likelihood breaks down into separate terms for each set of local parameters (one per node) i.e. there is no parameter sharing between different terms in the log likelihood formulation. This however is not the case for even fully observed Undirected Graphical Models (UGMs). The source of our trouble is the partition function as usual, because of which we no longer get disparate terms in the log likelihood. This section presents two approaches to estimate parameters for undirected graphical models; one for a special sub-class of UGMs called decomposable UGMs and another approach called Iterative Proportional Fitting for arbitrary UGMs. We restrict this discussion to UGMs involving discrete random variables for simplicity. While these techniques can be adapted for continuous random variables, readers are recommended to refer to [3] for further details.

2.1 Notation

First we clarify some notation to be used in the rest of this section. $\mathcal{X}_\mathcal{V}$ indicates the random vector corresponding to the entire graph $\mathcal{G}$ associated with the UGM. Let $\mathcal{C}$ be the set of cliques in this graph. Then $\mathcal{X}_C$ for some C $\in$ $\mathcal{C}$ refers to the subset of random variables associated with the nodes in the clique C. x_C refers to a specific instantiaion (value assignment) of the random variables in $\mathcal{X}_C$. The UGM is parametrized by potential functions $\psi_C(x_C)$ associated with each clique C in the UGM. The joint probability of all the random variables in the graph then defined as:

$$p(x_\mathcal{V}|\theta) = \frac{1}{Z} * \prod_{C \in \mathcal{C}} (\psi_C(x_C)) \tag{1}$$

where $\theta = \{\psi_C(x_C) \; \forall \; C \in \mathcal{C}\}$.

Assuming each data sample is i.i.d, the n^{th} data sample is associated with its own replica of the UGM $\mathcal{G}_n$ with random variables $\mathcal{X}_{\mathcal{V},n}$. Parameters for each node are shared across replicas.

Since we are dealing with UGMs involving discrete random variables, we define the following marginal counts:

$$m(x_\mathcal{V}) = \sum_n \delta(x_\mathcal{V}, x_{\mathcal{V},n}) \qquad \text{(number of times } x_\mathcal{V} \text{ occurs in data set)} \tag{2}$$

$$m(x_C) = \sum_{x_{\mathcal{V}\setminus C}} m(x_\mathcal{V}) \qquad \text{(marginal count for a value assignment } x_C \text{ to clique C)} \tag{3}$$

$$N = \sum_{x_\mathcal{V}} m(x_\mathcal{V}) \qquad \text{(total number of samples in the data set)} \tag{4}$$

Having defined the notation we shall use, we proceed to formulate the log likelihood:

$$p(x_{\mathcal{V},n}|\theta) = \prod_{x_\mathcal{V}} p(x_\mathcal{V}|\theta)^{\delta(x_\mathcal{V}, x_{\mathcal{V},n})} \tag{5}$$

$$p(\mathcal{D}|\theta) = \prod_n p(x_{\mathcal{V},n}) = \prod_n \prod_{x_\mathcal{V}} p(x_\mathcal{V}|\theta)^{\delta(x_\mathcal{V}, x_{\mathcal{V},n})} \tag{6}$$

$$l(\mathcal{D}|\theta) = \log p(\mathcal{D}|\theta) = \sum_n \sum_{x_\mathcal{V}} (\delta(x_\mathcal{V}, x_{\mathcal{V},n}) * \log(p(x_\mathcal{V}|\theta))) \tag{7}$$

By rearranging the order of the summation, applying the summation over n and using eqn 1 and 2 and, we get:

$$l(\mathcal{D}|\theta) = \sum_{x_\mathcal{V}} m(x_\mathcal{V}) * \sum_{C \in \mathcal{C}} \log(\psi_C(x_C)) - \sum_{x_\mathcal{V}} m(x_\mathcal{V}) * log(Z) \tag{8}$$

Observe that $\sum_{x_\mathcal{V}} m(x_\mathcal{V}) = N = \sum_{x_C} m(x_C)$. By using this, we get:

$$l(\mathcal{D}|\theta) = \sum_{C \in \mathcal{C}} \sum_{x_C} m(x_C) * \log(\psi_C(x_C)) - N * log(Z) \tag{9}$$

2.2 MLE for UGMs

Using the formulation of log likelihood in eqn (9), we use the standard technique of finding the derivative of the log likelihood and setting it to 0 to find the MLE estimates. As we shall see shortly, this doesn't give us a closed form solution for the MLE parameters as we would have hoped, but rather gives us a condition involving the parameters, that must hold for them to be MLE estimates. We now proceed to obtain derivates with respect to each of our parameters in θ. Remember that $\theta = \{\psi_C(x_C) \; \forall \; C \; \in \; \mathcal{C}\}$.

$$\frac{\partial(m(x_C) * \log(\psi_C(x_C)))}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C(x_C)} \tag{10}$$

$$\frac{\partial \log Z}{\partial \psi_C(x_C)} = \frac{1}{Z} * \frac{\partial \left(\sum_{\tilde{x}} \prod_D \psi_D(\tilde{x}_D)\right)}{\partial \psi_C(x_C)} \qquad \text{(using definition of Z)} \tag{11}$$

By applying the differentiation, all terms where $\tilde{x}_C \neq x_C$ are eliminated. Consequently:

$$\frac{\partial \log Z}{\partial \psi_C(x_C)} = \frac{1}{Z} * \sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) * \frac{\partial}{\partial \psi_C(x_C)} \left(\prod_{D \in \mathcal{C}} \psi_D(\tilde{x}_D) \right) \tag{12}$$

$$\frac{\partial \log Z}{\partial \psi_C(x_C)} = \frac{1}{Z} * \sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) * \left(\prod_{D \neq C} \psi_D(\tilde{x}_D) \right) \tag{13}$$

$$\frac{\partial \log Z}{\partial \psi_C(x_C)} = \sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) * \frac{1}{\psi_C(\tilde{x}_C)} * \frac{1}{Z} * \left(\prod_{D} \psi_D(\tilde{x}_D) \right) \tag{14}$$

$$\frac{\partial \log Z}{\partial \psi_C(x_C)} = \frac{1}{\psi_C(x_C)} * \sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) * \frac{1}{Z} * \left(\prod_{D} \psi_D(\tilde{x}_D) \right) \tag{15}$$

Note that the model's marginal distribution of $\tilde{x}$ is defined as $p(\tilde{x}) = \frac{1}{Z} * (\prod_D \psi_D(\tilde{x}_D))$. Using this definition:

$$\frac{\partial \log Z}{\partial \psi_C(x_C)} = \frac{1}{\psi_C(x_C)} * \sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) * p(\tilde{x}) \qquad \text{(equivalent to marginalizing out all } X_{i \notin C}) \tag{16}$$

$$\frac{\partial \log Z}{\partial \psi_C(x_C)} = \frac{1}{\psi_C(x_C)} * p(x_C) \tag{17}$$

$$\Rightarrow \frac{\partial l}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C(x_C)} - N * \frac{p(x_C)}{\psi_C(x_C)} \tag{18}$$

Note that without loss of generality, we can assume the potential functions are positive valued, since negative score functions can always be exponentiated to ensure the potential value has strictly non-negative range (0 being an extremal case). Consequently, when the gradient is 0, likelihood is maximized as follows:

$$\frac{\partial l}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C(x_C)} - N * \frac{p(x_C)}{\psi_C(x_C)} = 0 \tag{19}$$

$$\Rightarrow p_{MLE}(x_C) = \frac{m(x_C)}{N} \tag{20}$$

By defining $\frac{m(x)}{N}$ as the empirical marginal distribution $\tilde{p}(x)$, we see that we have obtained a condition constraining the MLE model marginal distribution to be equal to the empirical distribution. However, as mentioned before, we have not obtained a closed form solution for each of the parameters themselves since each such constraint involves all the parameters in it. This impasse leads use to two approaches to obtaining MLE estimates for the parameters.

2.3 Decomposable Models

As seen in the previous section, the appearance of the partition function means that equating the derivative of the log likelihood to 0 doesn't give us MLE estimates for the parameters. This is mainly because the log likelihood doesn't decompose into disparate terms as was the case in Bayes Nets. However, for a special subset of UGMs, the likelihood does indeed factor out conveniently enough to enable us to obtain the MLE estimates by inspection. This special subset of UGMs is the set of decomposable UGMs. They are defined as follows:

2.3.1 Definition of Decomposable models

An undirected graphical model is said to be decomposable if it can be recursively sub-divided into three subsets of nodes A, S and B such that A,S,B are disjoint, S separates nodes in A from nodes in B and S is complete.
There are several alternate definitions of decomposable models as well [1]. They can be defined as:

1. Markov random fields whose underlying graph is chordal i.e. all cycles with 4 or more vertices have at least one edge (a chord) that is not a part of the cycle that connects two vertices on the cycle.
2. Bayes nets with no V-structures (common child).
3. Bayes nets with a Markov field perfect map.
4. Graphical models whose underlying (hyper)graph is a junction tree.

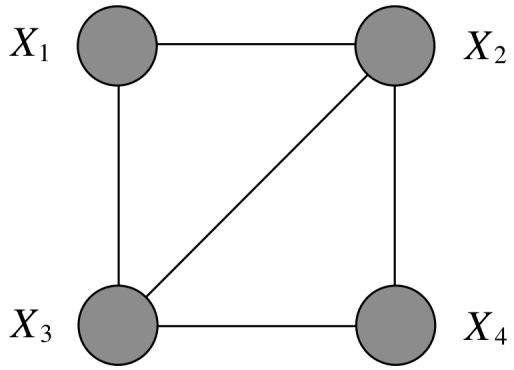

Figure 2: Decomposable graph with A = $\{X_1\}$, B = $\{X_4\}$ and S = $\{X_2, X_3\}$

Indeed, decomposable models are the intersection between directed graphical models and undirected graphical models. It is not surprising then that a simple technique exists, that can be used to obtain MLE estimates for parameters of such decomposable UGMs simply by inspection.

2.3.2 MLE for Decomposable models

MLE estimates can be easily obtained by parametrizing decomposable UGMs using potential functions associated with maximal cliques only. Let $\mathcal{C}$ be the set of maximal cliques in the UGM. Given this constraint, use the folloqing procedure to obtain MLE estimates for the potential functions:

1. For each clique $C \in \mathcal{C}$, set the clique parameter $\theta_C(x_C)$ $(= \psi_C(x_C))$ to be the empirical clique marginal $\tilde{p}(x_C) = \frac{m(x_C)}{N}$.
2. For each non-empty intersection between cliques, let the empirical marginal associated with that intersection be $\psi_S(x_S)$. Divide this potential into the parameter of one of the intersecting cliques involved (say $\theta_C(x_C)$) and set the parameter of that clique to the quotient i.e. $\theta_C(x_C) = \frac{\theta_C(x_C)}{\psi_S(x_S)}$.

As an example, consider the decomposable graph in figure 3. Its decomposition has been provided in the image caption. Applying the above procedure for MLE via inspection, we ge the following estimates for it:

$$p_{MLE}(x_1, x_2, x_3) = \tilde{p}(x_1, x_2, x_3) \tag{21}$$

$$p_{MLE}(x_2, x_3, x_4) = \frac{\tilde{p}(x_2, x_3, x_4)}{\tilde{p}(x_2, x_3)} \tag{22}$$

$$\Rightarrow p_{MLE}(x_1, x_2, x_3, x_4) = \frac{\tilde{p}(x_2, x_3, x_4) * \tilde{p}(x_1, x_2, x_3)}{\tilde{p}(x_2, x_3)} \tag{23}$$

2.4 Iterative Proportional Fitting for arbitrary UGMs

The simple procedure described in the previous sections for MLE by inspection works only for decomposable fully observed UGMs. To deal with arbitrary fully observed UGMs, we use eqn (19) along with a technique called fixed point iteration to develop an algorithm called Iterative Proportional Fitting which can be applied to arbitray UGMs to obtain MLE parameters. For decomposable UGMs, IPF converges in a single iteration (through all parameters) and ends up performing the same operations as the MLE by inspection method.

The procedure is as follows. We use eqn 19 and the definitions of empirical and model marginals, we get:

$$\frac{\tilde{p}(x_C)}{\psi_C(x_C)} = \frac{p(x_C)}{\psi_C(x_C)} \tag{24}$$

Fixed point iteration suggests that we hold the parameter $\psi_C(x_C)$ constant on the RHS (say $\psi_C^{(t)}(x_C)$) and solve for the free parameter (say $\psi_C^{(t+1)}(x_C)$) on the LHS. Thus we get:

$$\psi_C^{(t+1)}(x_C) = \psi_C^{(t)}(x_C) * \frac{\tilde{p}(x_C)}{p^{(t)}(x_C)} \tag{25}$$

Note that $\psi_C^{(t)}(x_C)$ appears in $p^{(t)}(x_C)$ internally. IPF performs this operation by iterating through parameters associated with all maximal cliques $C \in \mathcal{C}$ cyclically.

2.4.1 IPF as coordinate ascent

In general, fixed point iteration is not guaranteed to converge and is not guaranteed to be well behaved (monotonic). However, IPF both converges and is well behaved in the sense that log likelihood is guaranteed not to decrease at any step. This can be justified by showing that IPF is actually a form of coordinate ascent, where the coordinates are the potential functions.This shown by using eqn 13 and plugging it into the derivative of the log likelihood. This gives:

$$\frac{\partial l}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C(x_C)} - \frac{N}{Z} \sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) \prod_{D \neq C} \psi_D(\tilde{x}_D) \tag{26}$$

This can be viewed as a maximization of the parameter $\psi_C(x_C)$ while holding the rest of the parameters $\psi_{D \neq C}(x_D)$ constant. We annotate these constant parameters with a timestamp (t). This gives us:

$$\frac{\partial l}{\partial \psi_C(x_C)} = \frac{m(x_C)}{\psi_C(x_C)} - \frac{N}{Z} \sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) \prod_{D \neq C} \psi_D^{(t)}(\tilde{x}_D) \tag{27}$$

Now we make use of an insight provided in [2], were it is shown that by updating the free parameter $\psi_C(x_C)$ as per the IPF update equation (eqn 25), the value of the partition function doesn't change. Thus, $Z^{(t+1)} = Z^{(t)}$ when $\psi_C^{(t+1)}(x_C) = \psi_C^{(t)}(x_C) * \frac{\tilde{p}(x_C)}{p^{(t)}(x_C)}$. We make use of this property, update $\psi_C^{(t)}(x_C)$ and both multiply and divide eqn (27) by $\psi_C^{(t)}(x_C)$. We evaluate the new derivative at this new value $\psi_C^{(t+1)}(x_C)$:

$$\frac{\partial l}{\partial \psi_C^{(t+1)}(x_C)} = \frac{m(x_C)}{\psi_C^{(t+1)}(x_C)} - \frac{N}{\psi_C^{(t)}} \sum_{\tilde{x}} \delta(\tilde{x}_C, x_C) * \frac{1}{Z^{(t)}} * \prod_D \psi_D^{(t)}(\tilde{x}_D) \tag{28}$$

$$\frac{\partial l}{\partial \psi_C^{(t+1)}(x_C)} = \frac{m(x_C)}{\psi_C^{(t+1)}(x_C)} - \frac{N}{\psi_C^{(t)}} * \frac{\tilde{p}(x)_C}{p^{(t)}(x_C)} \tag{29}$$

By substituting the actual value of $\psi_C^{(t+1)}$ as per the IPF update rule in eqn (25), we see that the updated value forces the new gradient to be 0. In this sense, IPF is a coordinate ascent algorithm where the coordinates are the parameters associated with the maximal cliques in the graph.

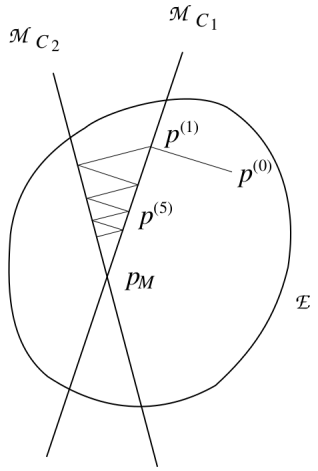

Figure 3: Visualization of IPF. Each step of IPF is a projection onto a manifold such that one of the cliques has the correct marginal. The point of convergence is where all such manifolds intersect.

3 Generalized Iterative Scaling (GIS)

GIS is one of the ways to estimate parameter of an Undirected Graphical Model (UGM) and is particularly useful for non decomposable models. GIS like IPF is an iterative model, but it can be broadly applied to exponential family potentials. As we saw in the previous section, IPF maximizes log likelihood by maximizing clique potential function by differentiation. Instead of optimizing the log liklihood directly, GIS uses the lower bound of log-likelihood to find the optima.

In a general case in which the clique potentials are parameterized by arbitrary collection of features, we could use a general exponential family model.

$$p(x|\theta) = \frac{1}{Z(\theta)} exp \sum_i \theta_i f_i(x) \tag{30}$$

The scaled likelihood function could be written as:

$$\tilde{l}(\theta|D) = \sum_x \tilde{p}(x) log p(x), \textit{where } \tilde{l}(\theta|D) = l(\theta|D)/N \textit{ and } \tilde{p}(x) \textit{is the empirical distribution.} \tag{31}$$

$$\tilde{l}(\theta|D) = \sum_x \tilde{p}(x) log \sum_i \theta_i f_i(x) - log Z(\theta) = \sum_i \theta_i \sum_x \tilde{p}(x) f_i(x) - log Z(\theta) \tag{32}$$

If the following two constraints are satisfied, GIS could be used to find the maximum likelihood parameter estimate.

$$f_i(x) \geq 0 \textit{ and } \sum_i f_i(x) = 1 \tag{33}$$

We use the convexity property to design another function which is a lower bound to log likelihood. Then we increase the lower bound to increase the log likelihood.

Using convexity property:

$$\begin{aligned} logz(\theta) &\leq \mu Z(\theta) - log(\mu) - 1, \texttt{ where } \mu = Z^{-1}(\theta^{(t)}) \\ => \tilde{l}(\theta|D) &\geq \sum_x \tilde{p}(x) \sum_i \theta_i f_i(x) - \frac{z(\theta)}{Z(\theta^{(t)})} - logZ(\theta^{(t)}) + 1 \end{aligned} \tag{34}$$

$$\textit{Lets define: } \Delta_i \theta_i^{(t)} = \theta_i - \theta_i^{(t)} \tag{35}$$

$$\tilde{l}(\theta|D) \geq \sum_x \tilde{p}(x) \sum_i \theta_i f_i(x) - \frac{1}{Z(\theta^{(t)})} \sum_x exp\Big(\sum_i \theta_i^{(t)} f_i(x)\Big) \sum_x exp\Big(\sum_i \Delta\theta_i^{(t)} f_i(x)\Big) - logZ(\theta^{(t)}) + 1 \tag{36}$$

Using Convexity and Jensen's inequality, we get:

$$exp\Big(\sum_i \pi_i x_i\Big) \leq \Big(\sum_i \pi_i exp(x_i)\Big) \textit{ for } \sum_i \pi_i = 1 \tag{37}$$

In the above equation, f's are positive and sum to one, so it can play the role of π_i*, that gives*

$$\tilde{l}(\theta|D) \geq \sum_i \theta_i \sum_x \tilde{p}(x) f_i(x) - \sum_x p(x|\theta^{(t)} \sum_i f_i(x) exp\Big(\Delta\theta_i^{(t)}\Big) - logZ(\theta^{(t)}) + 1 = \Lambda(\theta) \tag{38}$$

Note the above equation is a lower bound and parameters are decoupled. Taking the derivative of the above equation, wrt theta i and setting it to zero, gives:

$$exp\Delta\theta_i^{(t)} = \Big(\frac{\sum_x \tilde{p}(x) f_i(x)}{\sum_x p^{(t)}(x) f_i(x)} (Z(\theta^{(t)})\Big) \tag{39}$$

We have a relationship between the update function of theta and total distribution:

$$\begin{aligned} \theta_i^{(t+1)} &= \theta_i^{(t)} + \Delta\theta_i(t) \\ p^{(t+1)}(x) &= p^{(t)} \prod_i exp(\Delta\theta_i(t) f_i(x)) \end{aligned} \tag{40}$$

Using the above equations, the final update equation could be written as:

$$p^{(t+1)}(x) = \frac{p^{(t)} x}{Z(\theta^{(t)}} \prod_i \Big(\frac{\sum_x \tilde{p}(x) f_i(x)}{\sum_x p^{(t)}(x) f_i(x)} (Z(\theta^{(t)})\Big)^{f_i(x)} \tag{41}$$

$$= p^{(t)} x \prod_i \Big(\frac{\sum_x \tilde{p}(x) f_i(x)}{\sum_x p^{(t)}(x) f_i(x)}\Big)^{f_i(x)} \tag{42}$$

$$\theta_i^{(t+1)} = \theta_i^{(t)} + log\Big(\frac{\sum_x \tilde{p}(x) f_i(x)}{\sum_x p^{(t)}(x) f_i(x)}\Big) \tag{43}$$

As seen in the derivation above, the key idea is to first define a function that lower bonds the log-liklihood. Since the bound is tight means we can increase lower-bound by fixed-point iteration in order to increase log-likelihood. Please check [3] chapter 20 for details.

Comparison Between GIS and IPF:
GIS is a fully parallel algorithm, where as IPF is parallel at the level of a single clique. GIS is an iterative algorithm (like IPF), but more broadly applies to exponential family potentials

GIS have been largely surpassed by the gradient based methods which is discussed in the next section.

4 Gradient Based Methods (GBMs)

If potential function could be described as:

$$\psi_c(x_c) = \theta_c^T f_c(x_c) \tag{44}$$

Its log likelihood function could be written as:

$$l(\theta) = \sum_n \sum_k \theta_k f_k(x_n) - N log Z(\theta) \tag{45}$$

The derivative is:

$$\frac{\partial l}{\partial \theta_j} = \sum_n f_i(x_n) - N \frac{\partial}{\partial \theta_j} log Z \tag{46}$$

Which could further be resolved as:

$$\frac{\partial l}{\partial \theta_j} = \sum_n f_i(x_n) - N E[f_j(X)] \tag{47}$$

Any gradient-based optimization algorithm could be used for finding the global MLE by passing the above derivative.

Steps involved in GBMs are:

1. Design the objective function
2. Compute partial derivatives of the objective function
3. Feed the objective function and derivatives to an optimization algorithm. A number of optimization algorithms like Newton's Method, Quasi-Newton's methods or Stochastic gradient methods could be used.
4. Get back the optimized parameters from the optimization algorithm.

5 Summary

- Maximum Likelihood Parameter estimation in completely observed Bayesian Networks is easy thanks to decomposability.

- MLE estimation for fully observed UGMs is easy for decomposable UGMs and can be achieved by inspection.
- MLE estimation for arbitrary fully observed UGMs is possible using the IPF algorithm, which is a form of coordinate ascent that is guaranteed to converge and to be well behaved.
- GIS uses fixed point iteration over the derivative of a lower-bound of the likelihood objective to estimate maximum likelihood parameter.
- Gradient Based Methods uses simple algorithms like SGD, have usually a faster convergence than GIS and applies to arbitrary potentials.

Note: A lot of the materials in these scribe notes have been adapted from the citations below. More in-depth reading of these materials is highly recommended.

References

[1] Decomposable graphical models, triangulation and the Junction Tree, Marina Meila (Available at: `http://www.stat.washington.edu/courses/stat535/fall11/Handouts/l5-decomposable.pdf`)

[2] Chapter 9, Probabilistic Graphical Models, Michael I. Jordan, pg 17

[3] Chapter 19,20, Probabilistic Graphical Models, Michael I. Jordan