**10-708: Probabilistic Graphical Models 10-708, Spring 2015**

# 4: Parameter Estimation in Fully Observed BNs

*Lecturer: Eric P. Xing* *Scribes: Purvasha Chakravarti, Natalie Klein, Dipan Pal*

## 1 Learning Graphical Models

Learning a graphical model involves both learning the structure of the model and estimating the parameters involved in the structure of the graph. This lecture mostly focuses on estimating the parameters of a completely observed graphical model from the available data and performing inference on them. We also discuss a few structural learning algorithms for completely observed graphical models.

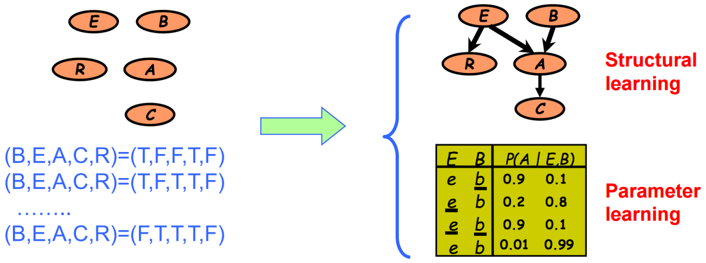

Figure 1: Given a set of variables (B, E, A, C and R) and independent samples on the left we want to learn a structure and the parameters involved (given on right) from the data.

The goal of learning a graphical model is find the best (or most likely) Bayesian Network (both DAG and CPDs) given a set of independent samples (or assignments of random variables). To succeed in finding the best model first, we either learn the structure of the model or assume a structure of the model given by an expert. Structural leaning has multiple limitations and hence there is not too much literature on it. The second step involves learning or estimating the parameters involved in the model.

While learning a model we come across different scenarios. We could have a directed or undirected completely observed graphical model or we could have a directed or undirected partially or unobserved graphical model. Learning an undirected partially observed graphical model is an open research topic.

We could also use different estimation principles in order to find the best Bayesian network that fits the data. We could use,

- Maximum likelihood estimation
- Bayesian estimation
- Maximum conditional likelihood (a lot of recent work has been done on this as it is more flexible)
- Maximal ”Margin” graphical model
- Maximum entropy graphical model

We generally use learning as a name for the process of estimating the parameters, and in some cases, the topology of the network, from data.

## 2 ML Structural Learning for completely observed GMs

In this section we discuss two "optimal" algorithms (vehicles to get to optimum) for learning the structure of completely observed GMs that guarantee to return a structure that maximizes the objective function (example, log likelihood function). That is, we want,

$$G' \equiv \arg\max_{G} L(D,G),$$

where $D$ is the data and $L$ is the objective function to be maximized. Many heuristics used to be popular , but most of them do not guarantee attainment of optimality, interpret-ability or even an explicit objective function. For example, structured Expectation-Maximization (EM), Module network, greedy structural search, deep learning via auto-encoders, gradient ascent method, samping method to MLE, etc do not provide these guarantees and hence are not "optimal".

We learn two classes of algorithms for guaranteed structural learning. These are likely to be the only known methods enjoying such guarantee, but they only apply to certain families of graphs. The two classes are,

- **The Chow-Liu algorithm** which holds only for trees, that is, for graphs where every node has only one parent. We discuss this algorithm in this lecture.
- **Covariance selection, neighborhood selection** which holds for continuous or discrete pairwise Markov Random Fields (MRFs).

Learning the structure of a graphical model is a very difficult task which is the reason there is not too much literature on it. To search for the best structure, let us first count how many structures are possible given n nodes. The number of graphs possible over n nodes is of the order $O\left(2^{n^2}\right)$. To see this we have to first realize that the best way to represent any graph over n nodes is to consider the corresponding adjacency matrix of order $n \times n$. Now each entry in the matrix could either be a 0 or a 1. Hence there are $2^{n^2}$ many options.

Since it is computationally difficult to consider so many graphs, its easier to focus only on trees. Now there are $O(n!)$ many trees possible over n nodes as one of the possibility is that every node has exactly one parent and one child. This can be done in $n!$ ways. So the first generation gets $n$ options, the second generation gets $n-1$ options and so on. Hence there are $O(n!)$ many trees possible.

It turns out that we can find an exact solution of an optimal tree (under MLE)! The first trick is to decompose the MLE score to edge-related elements. This is possible due to the product form of the likelihood function of a graphical model. As a result of this trick, now every change in an edge changes the likelihood function very systematically. The second trick is to realise that every node has only one parent and so we need to search for only one parent which increases the likelihood function the most. This is computationally easier than finding multiple parents. Applying these tricks we finally use the Chow-liu algorithm.

### 2.1 Information Theoretic Interpretation of ML

Considering the log likelihood function to be the objective function, for any graph $G$, parameters $\theta_G$ and data $D$, the objective functions is,

$$l(\theta_G) = \log p(D|\theta_G, G)$$

Let there be $M$ i.i.d. samples (assignments of random variables) and $K$ random variables or nodes in the data $D$, such that $D = \{x_1, ..., x_M\}$ where $x_n$ is a vector of $K$ values, one per node, $(x_{n1}, ..., x_{nK})$ for every $n = 1, 2, ..., M$. We denote $\pi_i(G)$ to be the parent node of the $i^{th}$ node in graph $G$. Hence $(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})})$ denotes the value at node i and its parents. Then the objective function becomes,

$$\begin{aligned}
l(\theta_G) &= \log p(D|\theta_G, G) \\
&= \log \prod_{n=1}^{M} \left( \prod_{i=1}^{K} p(x_{n,i}|\mathbf{x}_{\mathbf{n},\pi_\mathbf{i}(\mathbf{G})}, \theta_{i|\pi_i(G)}) \right) \quad \text{(by factorization rule)} \\
&= \sum_{i=1}^{K} \left( \sum_{n=1}^{M} \log p(x_{n,i}|\mathbf{x}_{\mathbf{n},\pi_\mathbf{i}(\mathbf{G})}, \theta_{i|\pi_i(G)}) \right) \quad \text{(taking log inside and interchanging sums)} \\
&= M \sum_{i=1}^{K} \left( \sum_{x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}} \frac{count(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})})}{M} \log p(x_i|\mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}, \theta_{i|\pi_i(G)}) \right) \quad \text{(counting configurations)} \\
&\qquad (count(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}) \text{ denotes no. of times } (x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}) \text{ appears in } M \text{ samples}) \\
&= M \sum_{i=1}^{K} \left( \sum_{x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}} \hat{p}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}) \log p(x_i|\mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}, \theta_{i|\pi_i(G)}) \right) \quad (\hat{p} \text{ is the empirical probability from } D) \\
&= M \sum_{i=1}^{K} \left( \sum_{x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}} \hat{p}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}) \log p(x_i|\mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}, \theta_{i|\pi_i(G)}) \right) \quad (\hat{p} \text{ is the empirical probability from } D)
\end{aligned}$$

Since we do not know the true $p$, we replace it by $\hat{p}$ the empirical probability. Therefore the objective function we now consider is,

$$\begin{aligned}
l(\theta_G) &= \log \hat{p}(D|\theta_G, G) \\
&= M \sum_{i=1}^{K} \left( \sum_{x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}} \hat{p}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}) \log \hat{p}(x_i|\mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}, \theta_{i|\pi_i(G)}) \right) \\
&= M \sum_{i=1}^{K} \left( \sum_{x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}} \hat{p}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}) \log \frac{\hat{p}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}, \theta_{i|\pi_i(G)})}{\hat{p}(\mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})})} \frac{\hat{p}(x_i)}{\hat{p}(x_i)} \right) \quad \text{(conditional probability definition)} \\
&= M \sum_{i=1}^{K} \left( \sum_{x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}} \hat{p}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}) \log \frac{\hat{p}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}, \theta_{i|\pi_i(G)})}{\hat{p}(\mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})})\hat{p}(x_i)} \right) - M \sum_{i=1}^{K} \left( \sum_{x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}} \hat{p}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}) \log \hat{p}(x_i) \right) \\
&= M \sum_{i=1}^{K} \left( \sum_{x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}} \hat{p}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}) \log \frac{\hat{p}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}, \theta_{i|\pi_i(G)})}{\hat{p}(\mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})})\hat{p}(x_i)} \right) - M \sum_{i=1}^{K} \left( \sum_{x_i} \hat{p}(x_i) \log \hat{p}(x_i) \right) \\
&= M \sum_{i=1}^{K} \hat{I}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}) - M \sum_{i=1}^{K} \hat{H}(x_i),
\end{aligned}$$

where $\hat{I}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}) = \sum_{x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}} \hat{p}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}) \log \frac{\hat{p}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}, \theta_{i|\pi_i(G)})}{\hat{p}(\mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})})\hat{p}(x_i)}$ is the decomposable score of mutual information of the node $i$ and its parents and $\hat{H}(x_i) = \sum_{x_i} \hat{p}(x_i) \log \hat{p}(x_i)$ is the entropy of node $i$.

## 2.2 Chow-Liu tree learning algorithm

As shown in the previous section, the objective function of structure learning for a graph can be written as,

$$\begin{aligned} l(\theta_G) &= \log \hat{p}(D|\theta_G, G) \\ &= M \sum_{i=1}^{K} \hat{I}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}) - M \sum_{i=1}^{K} \hat{H}(x_i). \end{aligned}$$

Since the second term does not depend on the graph and only the first term depends on the graph structure, the objective function to find the best graph structure becomes,

$$C(G) = M \sum_{i=1}^{K} \hat{I}(x_i, \mathbf{x}_{\pi_\mathbf{i}(\mathbf{G})}).$$

The Chow-Liu's algorithm can now be given in three steps,

1. For each pair of variable $x_i$ and $x_j$,
   - Compute the empirical distribution: $\hat{p}(x_i, x_j) = \frac{count(x_i, x_j)}{M}$, where $count(x_i, x_j)$ is the number of times $(x_i, x_j)$ occurs together in the $M$ samples.
   - Compute mutual information: $\hat{I}(x_i, x_j) = \sum_{x_i, x_j} \hat{p}(x_i, x_j) \log \frac{\hat{p}(x_i, x_j)}{\hat{p}(x_i)\hat{p}(x_j)}$. This is because in a tree every node $i$ just has one parent. So, $\pi_i(G) = \{j\}$ for some $j$. Hence the information can be broken into edge scores, $\hat{I}(x_i, x_j)$.
2. Define a graph with node $x_1, ..., x_K$.
   - Edge (i,j) gets weight $\hat{I}(x_i, x_j)$.
3. Find the optimum Bayesian net given the edge scores.
   - For undirected graphical model, compute maximum weight spanning tree.
   - For directed graphical model, after finding the maximum weight spanning tree, pick any node as root, then do a breadth-first-search to define directions. That is, for the root node any other node that shares an edge with it in the maximum weight spanning tree becomes its child and so on.

We notice that depending on the root node we choose at the last step, we could have some equivalent trees, ie, we notice I-equivalence between trees. An example of this can be seen in Figure 2

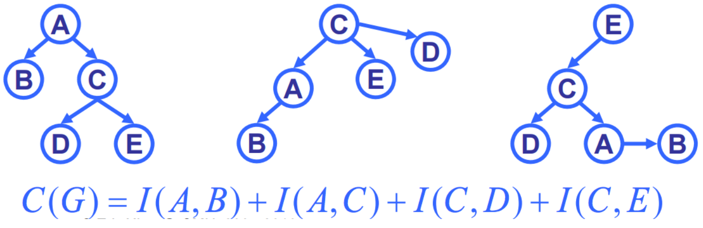

Figure 2: Given a set of variables (A, B, C, D and E) and the value of objective function $C(G)$ the following graphs are I-equivalent.

Therefore the structure of trees can be found using the Chow-Liu algorithm but unfortunately if there are more parents the problem of structure learning becomes very hard. The following theorem formalizes it further.

**Theorem:** The problem of learning a BN structure with at most $d$ parents is NP-hard for any (fixed) $d \geq 2$.

Most structure learning approaches use heuristics that exploit the score decomposition. Two heuristics that exploit the decomposition in different ways are greedy search through space of node -orders and local search of graph structures.

# 3 ML Parameter Estimation for completely observed GMs of given structure

We assume the structure of $G$ is known and fixed in order to do parameter estimation. The structure can be known from either an expert's design or an intermediate outcome of iterative structure learning.

The goal of parameter estimation is to estimate parameters from a data set of $M$ independent, identically distributed (iid) training cases $D = \{x_1, ..., x_M\}$. In general, each training case $x_n = (x_{n,1}, ..., x_{n,K})$ is a vector of $K$ values, one per node for each $n = 1, ..., M$. The model can be completely observed, i.e., every element in $x_n$ is known (no missing values, no hidden variables). Or the model can be partially observed, i.e., $\exists$ $i$, s.t. $Xn, i$ is not observed.

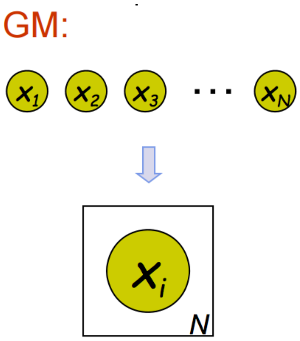

Figure 3: The graphical model representation of density estimation.

In this lecture we only consider estimating parameters for a BN given a structure and a completely observable model. In particular, we notice that density estimation can be viewed as a single node graphical model as seen in Figure 3. Also density estimation forms the building block of general GM. The next two sections deal with estimating the parameters of different distributions given the data. We discuss some instances of exponential family distributions. We also consider the MLE and Bayesian estimates of the parameters in order to estimate the density.

# 4 Discrete Distributions

We first assume a parametric distribution of the data in order to perform parameter estimation. Here we review common discrete distributions and examine both MLEs and Bayesian estimators for their parameters.

## 4.1 Definition of discrete distributions

- Bernoulli distribution: Ber(p)

  The random variable $X$ takes values in $\{0,1\}$, where $Pr(x=1)=p$ and $Pr(x=0)=1-p$.

  The probability mass function can be written:

  $$p(x) = p^x(1-p)^{1-x}$$

- Multinomial distribution (over indicators): Mult$(1, \theta)$

  Also called the categorical distribution, this is the generalization of the Bernoulli distribution to more than two outcomes. Let there be $K$ total outcomes. Suppose each outcome $k = 1, ..., K$ has probability $\theta_k$ of occurring in a single trial, where $\sum_k \theta_k = 1$. A convenient way to represent the outcome of a single trial is as a vector $X = [X_1, ..., X_K]$ where each $X_k \in \{0,1\}$ and $\sum_k X_k = 1$. In other words, element $k$ of the vector $(X_k)$ is 1 (with probability $\theta_k$) and the rest of the elements are 0.

  The probability mass function can be written:

  $$p(x) = \prod_k \theta_k^{x_k} = \theta^x$$

  A simple example is a single roll of a six-sided die; let $k = 1, ..., 6$ index the die face, where each face/outcome has underlying probability $\theta_k$. Then the observation $X = [1,0,0,0,0,0]$ corresponds to face 1 occurring after a single roll. If the die is fair, $\theta_k = 1/6$ for all $k = 1, ..., 6$.

- Multinomial distribution (over counts): Mult$(N, \theta)$

  The previous distribution could be thought of as the outcome of a single trial; we now generalize to $N$ trials. Now we think of a vector of outcomes $n = [n_1, ..., n_K]$ where each $n_k$ is the number of occurrences of outcome $k$, so $\sum_k n_k = N$ is the total number of trials.

  The probability mass function can be written:

  $$p(x) = \frac{N!}{n_1! n_2! \cdots n_K!} \prod_k \theta_k^{n_k} = \frac{N!}{n_1! n_2! \cdots n_K!} \theta^n$$

## 4.2 Parameter estimation in a multinomial model using MLE

Suppose the data is comprised of $N$ iid draws from $\text{Mult}(1,\theta)$, so a single observation $n$ is the vector $x_n = [x_{n,1}, ..., x_{n,K}]$ where $x_{n,k} \in \{0,1\}$ and $\sum_{k=1}^{K} x_{n,k} = 1$. Then the likelihood of one observation is:

$$L(\theta|x_n) = p(x_n|\theta) = \prod_{k=1}^{K} \theta_k^{x_{n,k}}$$

Using the fact that each draw is iid, the likelihood of the entire dataset $D = \{x_1, ..., x_N\}$ is

$$L(\theta|x_1, ..., x_N) = p(x_1, ..., x_N|\theta) = \prod_{n=1}^{N} p(x_n|\theta) = \prod_{n=1}^{N}\prod_{k=1}^{K} \theta_k^{x_{n,k}} = \prod_{k=1}^{K} \theta_k^{\sum_n x_{n,k}} = \prod_{k=1}^{K} \theta_k^{n_k}$$

where $n_k$ counts the number of occurrences of state $k$ across all trials.

The MLE is the estimate of $\theta$ which maximizes the likelihood (or log likelihood):

$$\ell(\theta|D) = \log L(\theta|x_1, ..., x_N) = \sum_k n_k \log \theta_k$$

The multinomial model is an exponential family distribution, but it is not full rank because the parameters must satisfy the linear constraint $\sum_k \theta_k = 1$. If we view $\ell(\theta|D)$ as an objective function, we can naturally incorporate the linear constraint using a Lagrange multiplier:

$$\tilde{\ell}(\theta|D) = \sum_k n_k \log \theta_k + \lambda \left(1 - \sum_k \theta_k\right)$$

Differentiating with respect to $\theta_k$ and setting equal to zero gives the following system of equations:

$$\frac{\partial \tilde{\ell}}{\partial \theta_k} = \frac{n_k}{\theta_k} - \lambda = 0$$

Then since $n_k = \lambda\theta_k$, we see $N = \sum_k n_k = \lambda \sum_k \theta_k = \lambda$, so the MLE is $\hat{\theta}_{k,MLE} = \frac{n_k}{N} = \frac{1}{N}\sum_n x_{n,k}$.

This corresponds to the notion that the counts $n_k$ are **sufficient statistics** for the data $D$; in other words, the distribution depends on the data only through the counts, and therefore we can estimate the MLE directly from the counts without need for the original data.

## 4.3 Parameter estimation in a multinomial model using Bayesian estimation

While MLE treats the parameter as a fixed, unknown constant that is estimated from the data, Bayesian estimation treats the parameter as random with a given prior distribution. The prior distribution along with the distribution of the data allows calculation of a posterior distribution, and the mean of the posterior distribution can be used as a point estimator

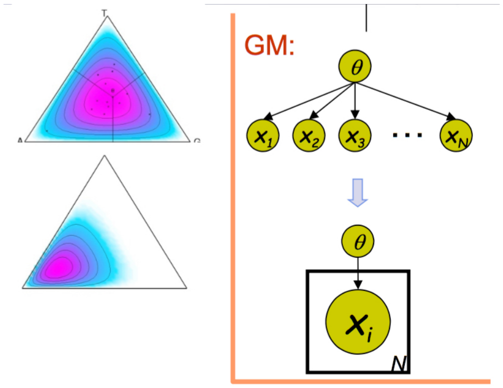

Figure 4: Depiction of the model and the simplex where $\theta$ is defined.

for the parameter. In practice, this estimator will generally be similar to the MLE for large sample sizes, but for smaller samples it will blend the influence of the prior with the influence of the data. For this reason, Bayesian estimators can be helpful with small sample sizes, particularly when certain outcomes are not actually observed and it is not desirable to set the probability of those outcomes to zero.

In general, if $p(\theta)$ is the prior distribution of the parameter, Bayes' rule states

$$p(\theta|x_1, ..., x_N) = \frac{p(x_1, ..., x_N|\theta)p(\theta)}{p(x_1, ..., x_N)}$$

where $p(\theta|x_1, ..., x_N)$ is the posterior distribution we wish to calculate. Arbitrary choices of prior will not be tractable mathematically, so it is convenient to use special conjugate priors, so that the prior and posterior distribution are easy to work with mathematically and take similar forms.

For example, if the data is multinomial (as in the last example), the parameter vector $\theta$ exists in a **simplex** since $\sum_k \theta_k = 1$ and $\theta_k \geq 0$ for all $k$. Instead of a point estimate we can create a distribution of $\theta$ over the simplex to inform our estimation; see examples in Figure 4.

### 4.3.1 The Dirichlet prior

A convenient prior is the Dirichlet distribution, $\mathrm{Dir}(\alpha = (\alpha_1, ..., \alpha_K))$:

$$p(\theta) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k \theta_k^{\alpha_k - 1} = C(\alpha) \prod_k \theta_k^{\alpha_k - 1}$$

where $\Gamma(\cdot)$ is the **gamma function**; for integers, $\Gamma(n+1) = n!$, but in general $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt$. Note the gamma function also has the convenient property that $\Gamma(t+1) = t\Gamma(t)$.

Now that the normalization constant $C(\alpha)$ is given by:

$$\frac{1}{C(\alpha)} = \int \cdots \int \theta_1^{\alpha_1 - 1} \cdots \theta_K^{\alpha_K - 1} d\theta_1 \cdots d\theta_K = \frac{\prod_k \Gamma(\alpha_k)}{\Gamma(\sum_k \alpha_k)}$$

where the integration can be done using integration by parts.

With the Dirichlet prior, the posterior distribution becomes (up to some proportionality constants):

$$p(\theta|x_1, ..., x_N) \propto \prod_k \theta_k^{n_k} \prod_k \theta_k^{\alpha_k - 1} = \prod_k \theta_k^{\alpha_k + n_k - 1}$$

So Dirichlet is a **conjugate prior** for multinomial because the posterior distribution takes a form similar to the prior. This form also explains why the Dirichlet parameters are referred to as psuedo-counts, because they behave similarly to how the counts derived from the data behave in the likelihood function.

To find the exact closed-form posterior distribution, first note that the marginal likelihood can be written:

$$p(x_1, ..., x_N|\alpha) = p(n|\alpha) = \int p(n|\theta)p(\theta|\alpha)d\theta = \frac{C(\alpha)}{C(\alpha + n)}$$

so the posterior is

$$p(\theta|x_1, ..., x_n, \alpha) = \frac{p(n|\theta)p(\theta|\alpha)}{p(n|\alpha)} = C(n + \alpha) \prod_k \theta_k^{\alpha_k + n_k - 1}$$

Now we recognize this is $\mathrm{Dir}(n + \alpha)$, again underscoring the conjugacy of the prior with the posterior.

Therefore if we observe $N'$ samples with sufficient statistics/counts vector $n'$, the posterior distribution is $p(\theta|\alpha, n') = \mathrm{Dir}(\alpha + n')$. If we then observe another $N''$ samples with statistics $n''$, the posterior becomes $p(\theta|\alpha, n', n'') = \mathrm{Dir}(\alpha + n' + n'')$, allowing for **sequential Bayesian updating** as we receive new information, which is similar to online learning.

To get a point estimate for $\theta$ we can take the mean of the posterior distribution:

$$\hat{\theta}_k = \int \theta_k p(\theta|D)d\theta = C \int \theta_k \prod_k \theta_k^{\alpha_k + n_k - 1} d\theta = \frac{n_k + \alpha_k}{N + |\alpha|}$$

Notice that this estimate is essentially a weighted combination of the actual observed frequencies and the psuedo-counts from the prior distribution. As the number of observations $N$ grows large, this estimate will be essentially the same as the MLE.

Another quantity of interest is the **posterior predictive rate**, which in essence predicts the outcome of the $X_{N+1}$ event given the first $N$ observations:

$$p(x_{N+1} = i|x_1, ..., x_N, \alpha) = \int C(n + \alpha) \prod_k \theta_k^{\alpha_k + n_k - 1} \theta_i^{\alpha_k + n_k} d\theta = \frac{C(n + \alpha)}{C(x_N + \alpha + n)} = \frac{n_i + \alpha_i}{|n| + |\alpha|}$$

### 4.3.2 Hierarchical Bayesian Models

When we specified a Dirichlet prior in the previous section, we needed to specify the parameters $\alpha = (\alpha_1, ..., \alpha_k)$, which are also called the psuedo-counts. **Hierarchical Bayesian models** put another prior on these parameters rather than specifying exact values.

So while we have parameters $\theta$ for the likelihood $p(x|\theta)$ and parameters $\alpha$ for the prior $p(\theta|\alpha)$, we could continue to add more 'layers' by putting a prior distribution on the $\alpha$ values with its own parameters known as **hyperparamters**. Adding layers could be done indefinitely, but typically adding more layers does not have too much influence on the final results, particularly if there is enough data.

While our choice of Dirichlet prior for $\theta$ was motivated by conjugacy with the multinomial distribution, how do we choose a prior for $\alpha$? One approach is to make an intelligent guess, or use a uniform or other **noninformative prior**, so that the prior should not have too much influence but still allows us to avoid making arbitrary parameter choices. Another approach is called **empirical Bayes** (or Type-II maximum likelihood). The idea rests on the following equation, which integrates $\theta$ out of the model to directly get a distribution of $n$ given $\alpha$:

$$p(n|\alpha) = \int p(n|\theta)p(\theta|\alpha)d\theta$$

In the case of the Dirichlet prior, this function will be a gamma function. Then we can select the following estimator for $\alpha$:

$$\hat{\alpha}_{MLE} = \arg\max_{\alpha} p(n|\alpha)$$

Typically this estimation is done using some data, then applied to further data; 'treat yesterday as a prior for today'. (Eric mentioned during class that he may post further notes for a more full treatment of this approach.)

### 4.3.3 The Logistic Normal Prior

While the Dirichlet prior is convenient, it has some drawbacks. In particular, it can only give rise to certain kinds of distributions of $\theta$ over the simplex, which are symmetric or concentrated in one corner, as shown in Figure 5.

An alternative prior is the **logistic Normal distribution** (or logit Normal distribution), which is more difficult to use because it is no longer conjugate (though we will talk more later in the class about how to get around this issue).

We say $\theta$ is logistic Normal if $\theta \sim LN_k(\mu, \Sigma)$. To define the distribution, first let $\gamma \sim N_{K-1}(\mu, \Sigma)$ with $\gamma_K = 0$. Then $\theta_i = \exp \gamma_i \log \left(1 + \sum_{i=1}^{K-1} e^{\gamma_i}\right)$. One difficulty with this distribution is the log partition function/normalization constant which takes the form $C(\gamma) = \log \left(1 + \sum_{i=1}^{K-1} e^{\gamma_i}\right)$.

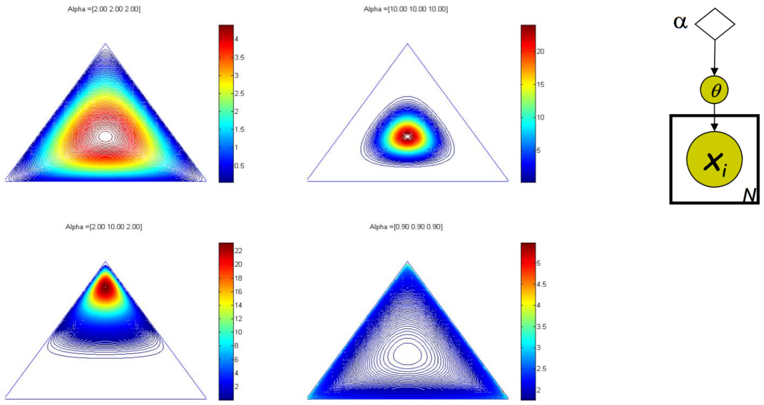

Figure 5: Examples of distributions of $\theta$ over the simplex with the Dirichlet prior.

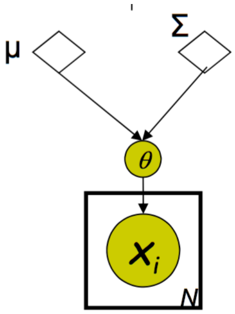

Figure 6: Plate diagram of logistic Normal prior.

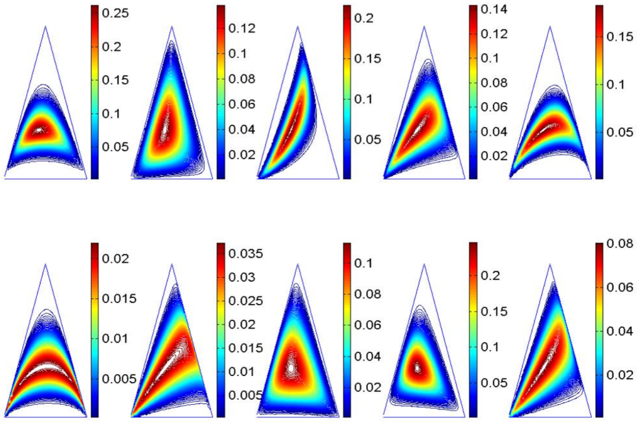

Figure 7: Examples of distributions of $\theta$ over the simplex with the logistic Normal prior.

One benefit of the logistic Normal prior is it involves a covariance structure that we can exploit. It also gives rise to different kinds of distributions on the simplex, as shown in Figure 7.

# 5 Continuous Distribution

Parametric distributions can also be of the continuous form. We review some continuous distributions below.

## 5.1 Some continuous distributions

1. **Uniform Distribution:** Uniform distributions are basically "flat" distributions. Parametrized as

$$p(x) = \frac{1}{(b-a)} \quad \forall a \leq x \leq b$$

and 0 elsewhere

2. **Normal Density Function:** The Normal distribution is parametrized as

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp -(x-\mu)^2/2\sigma^2$$

The distribution is symmetric, and has two moments characterized by the mean $\mu$ and the variance $\sigma$.

3. **Multivariate Gaussian:** Multivariate Gaussian distribution is simply a high dimensional Gaussian distribution. It is parametrized as

$$p(x) = \frac{1}{2\pi^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} \exp\{-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)\}$$

## 5.2 Bayesian estimation of parameters for the Gaussian

We look at the following cases:

1. **Known $\mu$ and unknown $\lambda = \frac{1}{\sigma^2}$:**
   The conjugate prior for $\lambda$ with shape $a$ and rate (inverse scale) $b$.

   $$p(\lambda \mid a, b) = \frac{1}{\Gamma(a)} b^a \lambda^{a-1} \exp -b\lambda$$

   The conjugate prior for inverse $\sigma$ is the Gamma-inverse

   $$IG(\sigma^2 \mid a, b) = \frac{1}{\Gamma(a)} b^a (\sigma^2)^{-(a+1)} \exp \frac{-b}{\sigma^2}$$

2. **Unknown $\mu$ and unknown $\sigma$:**
   The conjugate prior is

   $$P(\mu, \sigma^2) = P(\mu|\sigma^2)P(\sigma^2) = N(\mu \mid m, \sigma^2, V)IG(\sigma^2 \mid a, b)$$

3. **Multivariate case** :
   The conjugate prior is

   $$P(\mu, \Sigma) = P(\mu \mid \Sigma)P(\Sigma) = N(\mu \mid \mu_0, \frac{1}{\kappa_0}\Sigma)IW(\Sigma \mid \Sigma, \Lambda_0^{-1}, v_0)$$

## 5.3 Estimation of conditional densities

1. Estimation of individual conditional densities can be viewed as two-node graphical models. The parameters of the child are then estimated given a configuration of the parent.

2. The two-node models are the fundamental building blocks of general larger graphical models. However, for parameter estimation, they can be considered separately when all nodes are observed.

3. Parameter estimation can be carried out through Maximum Likelihood estimates or Bayesian estimation. Given enough data, MLE estimates are usually preferred. However, Bayesian estimation is useful in the case when few samples are available.

**Decomposability of the log-likelihood**

Under the global independence assumption, if all nodes are observed, then the log-likelihood of the network decomposes into a sum of local likelihoods.

$$l(\theta, D) = \log p(D \mid \theta) = \log \prod_n \left( \prod_i p(x_{n,i} \mid x_{n,\pi_i}, \theta_i) \right) = \sum_i \left( \sum_n \log p(x_{n,i} \mid x_{n,\pi_i}, \theta_i) \right)$$

This makes the parameter estimation much easier, tractable and parallelizable. The problem basically becomes, to individually estimate the CPDs for each node separately and then combine the resulting parameters together into one model.

We now illustrate this phenomenon. Consider the general form of the distribution represented by a directed acyclic graphical model.

$$p(x \mid\theta) = p(x_1 \mid \theta_1)p(x_2 \mid x_1, \theta_2)p(x_3 \mid x_2, x_3, \theta_3)p(x_4 \mid x_1, x_2, x_3, \theta_4)$$

As we saw in the equation above this, equivalently we have four independent graphical models each with just a single child and fully observed parents.

## 5.4 MLE for Bayesian Networks with tabular CPDs

In the case of tabular CPDs, *i.e.* the CPDs are in the form of a table (possibly multinomial), the MLE estimate is straight forward. The parameter we need to estimate is

$$\theta_{ijk} = P(X_i = j \mid X_{\pi_i} = k)$$

In the case of a single parent, the table is a two dimensional one. Higher dimensional tables result in the case of multiple parents. The difficulty in estimation increases as the number of parents goes up, since we need to be able to observe enough samples for each configuration of the parents in order to have a good estimate. The counts of family (joint configurations of parents and child) serve as sufficient statistics of the distribution.

$$n_{ijk} = \sum_n x_{n,i}^j x_{n,\pi_i}^k$$

Thus the log likelihood becomes

$$l(\theta, D) = \log \prod_{i,j,k} \theta_{ijk}^{n_{ijk}} = \sum_{i,j,k} n_{ijk} \log \theta_{ijk}$$

Maximizing under the condition $\sum_j \theta_{ijk} = 1$, we have

$$\theta_{ijk}^{MLE} = \frac{n_{ijk}}{\sum_{j'} n_{ij'k}}$$

Thus, the MLE is simply the fraction of counts of a particular value of a child upon the total number of values the child took for a particular configuration of values of its parents.

### 5.5 Defining a parameter prior

Recall that the joint density can be factorized as

$$P(X = x) = \prod_{i=1}^{M} p(x_i \mid x_{\pi_i})$$

Each of the terms $p(x_i \mid x_{\pi_i})$ is in fact a local distribution $p(x_i^k \mid x_{\pi_i}^j) = \theta_{x_i^k \mid x_{\pi_i}^j}$

Geiger and Heckerman state a set of assumptions under which the parameter priors can be defined for a large class of directed acyclic graphs. These are:

1. **Complete Model Equivalence:** Given a data distribution or data $X$, any two complete DAG model which describe $X$, describe the same set of joint probability distributions.

2. **Global Parameter Independence:** For every DAG *model*, we have

$$p(\theta_m \mid G) = \prod_{i=1}^{M} p(\theta_i \mid G)$$

   The equation basically shows that the priors of every node are independent.

3. **Local Parameter Independence:** For every DAG *node*, we have

$$p(\theta_i \mid G) = \prod_{j=1}^{q_i} p(\theta_{x_i^k \mid x_{\pi_i}^j} \mid G)$$

   The equation basically shows that, for every *configuration* of the parents of a child, the priors on the child's resultant distribution are independent.

4. **Likelihood and Prior Modularity:** For every two DAG models of $X$ with the same structure (same parents for a given node in both models), then the local distributions of every node and the prior distributions for every parameter are the same.

### 5.6 Parameter Sharing

To illustrate parameter sharing, we consider a stationary (time-invariant) first-order Markov model defined by two parameters. First, the initial state probability $\pi_k$, and second, the state transition probabilities parameterized by $A_{ij} = p(X_t^j = 1 \mid X_{t-1}^i = 1)$. The parameter $A$ is shared by all future states.

The joint distribution becomes $p(X \mid \theta) = p(x_1 \mid \pi) \prod_{t=2}^{T} P(X_t \mid X_{t-1})$

Whereas the log-likelihood is $l(\theta, D) = \sum_n \log p(x_{n,1} \mid \pi) + \sum_n \sum_{t=2}^{T} \log p(x_{n,t} \mid x_{n,t-1})$

With optimize each parameter separately due to local independence. $\pi$ is estimated easily using techniques described before, since it is simply a multinomial frequency vector. We now discuss the estimation of $A$. We have the constraint that $\sum_j A_{ij} = 1$, each row of $A$ is a multinomial distribution. Thus, the MLE of $A$ becomes

$$A_{ij}^{MLE} = \frac{\#(i \to j)}{\#(i \to \cdot)} = \frac{\sum_n \sum_{t=2}^{T} x_{n,t-1}^i x_{n,j}^j}{\sum_n \sum_{t=2}^{T} x_{n,t-1}^i}$$