Notes on Probabilistic Graphical Models

Shoichiro Yamanishi

May 1, 2020

1 Overview

This document briefly goes through some important points regarding the probabilistic graphical models. A graphical model epresses the factoring of a joint distribution over the random variables. Each node represent a random variable, except for a factor graph in which node is either a factor or a random variable. A model can be programmatically drawn using Python package Daft which works with matplotlib. See [2].

- Directed Acyclic Graph (Bayesian Network) : This expresses conditional dependencies among the random variables by directed edges.
- Undirected Graph (Markov Random Field) : This expresses factoring by maximal cliques.
- Factor Graph : This expresses the exact factoring. Any path on the graph visits factors and random variables alternatingly.

Chap. 8 of PRML [1] is a good source of information.

2 Directed Acyclic Graph (Bayesian Network)

2.1 Chain

Two interesting cases: One is for a chain of 1-of-K discrete variables and the other is for gaussian distribution. The random variables on a chain are usually latent variables.

2.1.1 Distrete Variables Chain

Please see Figure 1 for a chain of Direchlet distribution for latent variables $\boldsymbol{z}_i \in \mathcal{R}^K$.

$$\begin{aligned} p(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_3, \cdots, \boldsymbol{z}_{N-1}, \boldsymbol{z}_N|\boldsymbol{\alpha}) &= p(\boldsymbol{z}_N|\boldsymbol{z}_{N-1})\cdots p(\boldsymbol{z}_3|\boldsymbol{z}_2)p(\boldsymbol{z}_2|\boldsymbol{z}_1)p(\boldsymbol{z}_1|\boldsymbol{\alpha}) \\ &= \mathrm{Dir}(\boldsymbol{z}_N|\boldsymbol{z}_{N-1})\cdots \mathrm{Dir}(\boldsymbol{z}_3|\boldsymbol{z}_2)\mathrm{Dir}(\boldsymbol{z}_2|\boldsymbol{z}_1)\mathrm{Dir}(\boldsymbol{z}_1|\boldsymbol{\alpha}) \end{aligned} \quad (1)$$

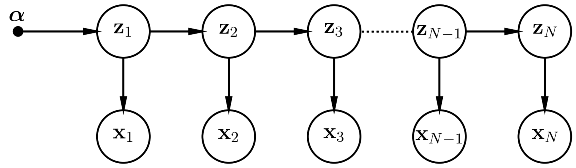

Figure 1: Chain of Direchlet Distribution

where $\mathrm{Dir}(\boldsymbol{z}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{j=1}^{K} \alpha_j)}{\prod_{j=1}^{K} \Gamma(\alpha_j)} \prod_{j=1}^{K} z_j^{\alpha_j - 1}$ The corresponding random variables can be one-of-K variable $\boldsymbol{x}_i \in \{0,1\}^K$ such that $\boldsymbol{x}_i \sim \mathrm{Mult}(\boldsymbol{x}_i|\boldsymbol{z}_i) = \prod_{j=1}^{K} z_{i_j}^{x_{i_j}}$.

2.1.2 Linear Gaussian Model

Please see Figure 2 for a chain of Gaussian distribution for latent variables $\boldsymbol{z}_i$. This is a conceptual model and not really useful. A useful example that includes inference to the covariance matrices is Kalman filter described in a separate document.

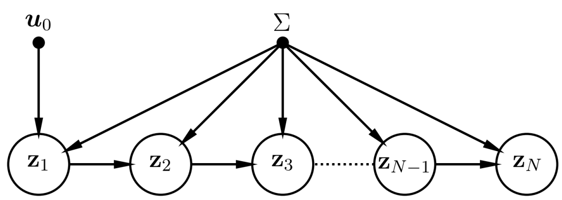

Figure 2: Chain of Gaussian Distribution

$$\begin{aligned} p(\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_{N-1}, \boldsymbol{z}_N|\boldsymbol{\alpha}) &= p(\boldsymbol{z}_N|\boldsymbol{z}_{N-1}) \cdots p(\boldsymbol{z}_2|\boldsymbol{z}_1) p(\boldsymbol{z}_1) \\ &= \mathcal{N}(\boldsymbol{z}_N; \boldsymbol{u}_{N-1}(\boldsymbol{z}_{N-1}), \Sigma) \cdots \mathcal{N}(\boldsymbol{z}_2; \boldsymbol{u}_1(\boldsymbol{z}_1), \Sigma) \mathcal{N}(\boldsymbol{z}_1; \boldsymbol{u}_0, \Sigma) \end{aligned} \tag{2}$$

2.2 Parameter Reduction

Consider a probabbility distribution $p(y|x_1, x_2, \cdots, x_N)$ where $\boldsymbol{x}_i \in \{0,1\}^K$, $\sum_{j=1}^{K} x_{i_j} = 1$, i.e., $\boldsymbol{x}_i$ is a 1-of-K variable. The possible combinations will add up to K^N, which could be unmanegeable. To reduce the complexity we can use a linear combination of $\boldsymbol{x}_i$ as follows. $p(y|x_1, x_2, \cdots, x_N, \boldsymbol{w}_0, W) = p(y|\boldsymbol{w}_0 + \sum_{i=1}^{N} W\boldsymbol{x}_i)$ where $\boldsymbol{w}_0 \in \mathcal{R}^K$, W is a $K \times N$ matrix. See figure 3.

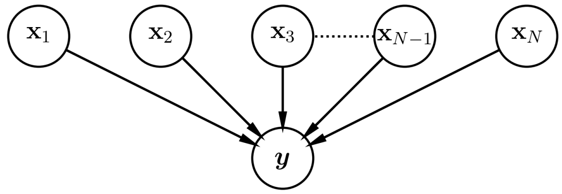

Figure 3: Parameter Reduction

2.3 I.I.D. and Naive Bayes

Consider the graphical models depicted in figure 4. This can manifest as *independent and identically distributed* assumption for N samples, or *Naive Bayes* model of a randam variable $\boldsymbol{x} \in \mathcal{R}^N$ with N features that depends on the class C, where each feature is assumed to be independent of each other conditioned on C.

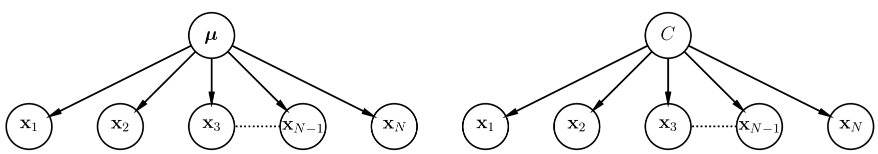

Figure 4: I.I.D. and Naive Bayes

2.4 Conditional Independence, D-Separaton, and Markov Blanket

2.4.1 Conditional Independence

BTW, the math symbols $\perp\!\!\!\perp$ and $\not\perp\!\!\!\perp$ for Text are *upmodels* and *nupmodels* in the package *MnSymbol*.

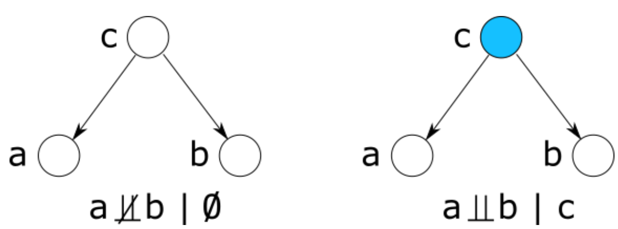

Figure 5: Conditional Independence Case 1

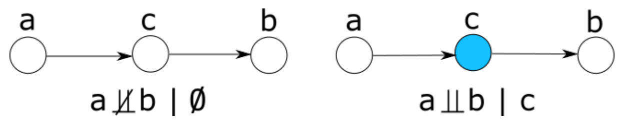

Figure 6: Conditional Independence Case 2

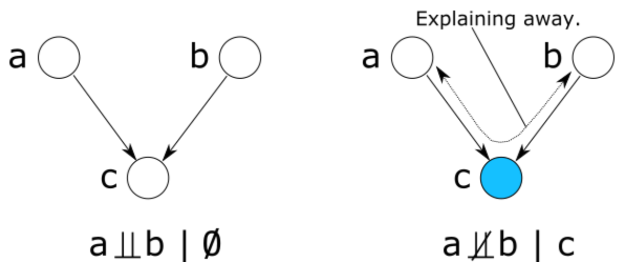

Figure 7: Conditional Independence Case 3

2.4.2 D-Separation

Let V_s be a set of random variables which you want to make conditionally independent from the reset. Given a set of observed random variables, how to determin if V_s are conditionally independent from the rest? D-Separation gives an answer. Let G_s be a subgraph induced by V_s. Also, Let $N^+(v)$ and $N^-(v)$ denote the in-neighbor and out-neighbor of v respectively. Then Let $N^+(G_s) = (\bigcup_{v \in G_s} N^+(v)) \setminus V(G_s)$ and $N^-(G_s) = (\bigcup_{v \in G_s} N^-(v)) \setminus V(G_s)$ where $V(G_s)$ denotes the vertices in the graph G_s. In other words, $N^+(G_s)$ is the immediate in-neighbor of G_s and $N^-(G_s)$ is the immediate out-neighbor of G_s. Let $D(G_s)$ be the subgraph induced by the nodes reachable from G_s. In other words, $D(G_s)$ is the desecendent subgraph of G_s. Then in order to make $N(G_s)$ conditionally independent, the following must hold.

- $N^+(G_s)$ must be observed.
- if any $v \in D(G_s)$ is observed, then $N^+(v) \setminus V(G_s)$ must be observed.

This is illustrated in figure 8 If the set of observed random variables meet the criteria above, then V_s are conditionally independent from the rest.

2.4.3 Markov Blanket

Again, let V_s be a set of random variables which you want to make conditionally independent from the reset. What other random variables must be observed to guarantee the conditional independence of V_s? Markov Blanket gives an anser. Please see figure 9. It's basically $N^+(G_s)$, $N^-(G_s)$, and its outneighbors.

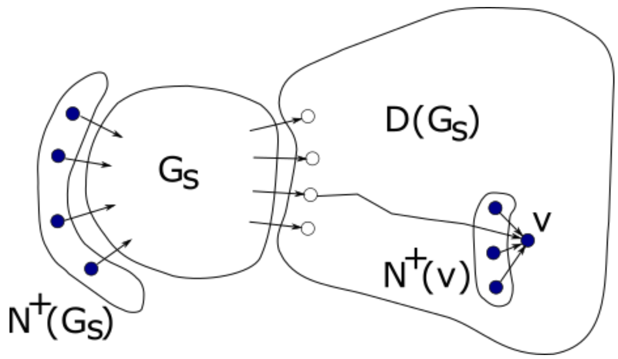

Figure 8: D-Separation

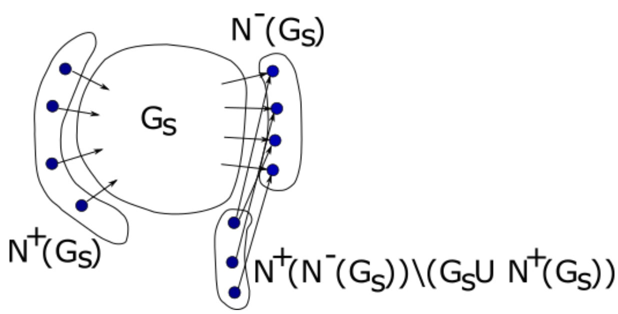

Figure 9: D-Separation

3 Undirected Graph (Markov Random Field) and Belief Propagation

We consider an exact inference called belief propagation in a tree. There are two types to consider. One is to find a marginal distribution of a particula random variable, and the other is find the values for all the random variables that together maximizes the joint probability as in finding a mode in a MAP estimage.

We first consider the simple case of a chain and expand the discussion to a tree.

3.1 Inference in a Chain : Simple Case

Consider the joint probability of discrete random variables $\boldsymbol{x}_i \in \{0,1\}^K$, $\sum_{j=1}^{K} x_{i_j} = 1$, which is factorized into a chain as in figure 10.

The joint probability is factorized into the following.

$$p(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{N-1}, \boldsymbol{x}_N) = \frac{1}{Z}\phi_{1,2}(\boldsymbol{x}_1, \boldsymbol{x}_2)\phi_{2,3}(\boldsymbol{x}_2, \boldsymbol{x}_3)\cdots\phi_{N-1,N}(\boldsymbol{x}_{N-1}, \boldsymbol{x}_N) \quad (3)$$

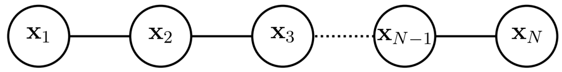

Figure 10: Undirected Chain

3.1.1 Marginal distribution $p(\boldsymbol{x}_N)$

Consider first the marginal distribution $p(\boldsymbol{x}_N)$.

$$\begin{aligned}
p(\boldsymbol{x}_N) &= \sum_{\boldsymbol{x}_1}\sum_{\boldsymbol{x}_2}\cdots\sum_{\boldsymbol{x}_{N-1}} \frac{1}{Z}\phi_{1,2}(\boldsymbol{x}_1,\boldsymbol{x}_2)\phi_{2,3}(\boldsymbol{x}_2,\boldsymbol{x}_3)\cdots\phi_{N-1,N}(\boldsymbol{x}_{N-1},\boldsymbol{x}_N) \\
&= \frac{1}{Z}\left[\sum_{\boldsymbol{x}_{N-1}}\phi_{N-1,N}(\boldsymbol{x}_{N-1},\boldsymbol{x}_N)\left[\sum_{\boldsymbol{x}_{N-2}}\phi_{N-2,N-1}(\boldsymbol{x}_{N-2},\boldsymbol{x}_{N-1})\left[\cdots\sum_{\boldsymbol{x}_2}\phi_{2,3}(\boldsymbol{x}_2,\boldsymbol{x}_3)\left[\sum_{\boldsymbol{x}_1}\phi_{1,2}(\boldsymbol{x}_1,\boldsymbol{x}_2)\right]\cdots\right]\right]\right]
\end{aligned} \tag{4}$$

On the first line, there are K^N terms. On the second, I abused the notation such that each square brackets pair indicates a function of one variable. We define the following recursive definition.

$$\phi'_i(\boldsymbol{x}_i) = \sum_{\boldsymbol{x}_{i-1}}\phi_{i-1,i}(\boldsymbol{x}_{i-1},\boldsymbol{x}_i)\phi'_{i-1}(\boldsymbol{x}_{i-1}) \tag{5}$$

Evaluation of each of K elements of $\phi'_i(\boldsymbol{x}_i)$ takes takes K evaluations of $\phi(\boldsymbol{x}_{i-1},\boldsymbol{x}_i)$, K multiplications, and $K-1$ additions. Hence the construction of one $\phi'_i(\boldsymbol{x}_i)$ is $O(K^2)$.

Then $p(\boldsymbol{x}_N)$ is constructed sequencially by evaluating from $\phi'_1(\boldsymbol{x}_2)$ up to $\phi'_1(\boldsymbol{x}_N)$ as follows.

$$\begin{aligned}
p(\boldsymbol{x}_N) &= \frac{1}{Z}\left[\sum_{\boldsymbol{x}_{N-1}}\phi_{N-1,N}(\boldsymbol{x}_{N-1},\boldsymbol{x}_N)\left[\sum_{\boldsymbol{x}_{N-2}}\phi_{N-2,N-1}(\boldsymbol{x}_{N-2},\boldsymbol{x}_{N-1})\left[\cdots\sum_{\boldsymbol{x}_2}\phi_{2,3}(\boldsymbol{x}_2,\boldsymbol{x}_3)\left[\sum_{\boldsymbol{x}_1}\phi_{1,2}(\boldsymbol{x}_1,\boldsymbol{x}_2)\right]\cdots\right]\right]\right] \\
&= \frac{1}{Z}\left[\sum_{\boldsymbol{x}_{N-1}}\phi_{N-1,N}(\boldsymbol{x}_{N-1},\boldsymbol{x}_N)\left[\sum_{\boldsymbol{x}_{N-2}}\phi_{N-2,N-1}(\boldsymbol{x}_{N-2},\boldsymbol{x}_{N-1})\left[\cdots\sum_{\boldsymbol{x}_2}\phi_{2,3}(\boldsymbol{x}_2,\boldsymbol{x}_3)\phi'_2(\boldsymbol{x}_2)\right]\right]\right] \\
&= \frac{1}{Z}\left[\sum_{\boldsymbol{x}_{N-1}}\phi_{N-1,N}(\boldsymbol{x}_{N-1},\boldsymbol{x}_N)\left[\sum_{\boldsymbol{x}_{N-2}}\phi_{N-2,N-1}(\boldsymbol{x}_{N-2},\boldsymbol{x}_{N-1})\left[\cdots\phi'_3(\boldsymbol{x}_3)\right]\right]\right] \\
&= \frac{1}{Z}\left[\sum_{\boldsymbol{x}_{N-1}}\phi_{N-1,N}(\boldsymbol{x}_{N-1},\boldsymbol{x}_N)\left[\sum_{\boldsymbol{x}_{N-2}}\phi_{N-2,N-1}(\boldsymbol{x}_{N-2},\boldsymbol{x}_{N-1})\phi'_{N-2}(\boldsymbol{x}_{N-2})\right]\right] \\
&= \frac{1}{Z}\left[\sum_{\boldsymbol{x}_{N-1}}\phi_{N-1,N}(\boldsymbol{x}_{N-1},\boldsymbol{x}_N)\phi'_{N-1}(\boldsymbol{x}_{N-1})\right] \\
&= \frac{1}{Z}\phi'_N(\boldsymbol{x}_N)
\end{aligned} \tag{6}$$

This derivation can be viewed as a message passing from $\boldsymbol{x}_2$ up to $\boldsymbol{x}_N$ with the message defined by $\phi'_i(\boldsymbol{x}_i)$. This is the simplest form of belief propagation.

Please note that $Z = \sum_{\boldsymbol{x}_N} \phi'_N(\boldsymbol{x}_N)$. Overall the complexity of finding $p(\boldsymbol{x}_N)$ is $O(NK^2)$.

3.1.2 Finding a Mode : $\text{argmax}\{p(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{N-1}, \boldsymbol{x}_N)\}$ for MAP etc

Now we want to find the values for each random variables that maximizes the joint probability. This occurs in a MAP estimate, where you want to get the highest mode of the probability distribution. Here we define an operation called *amax* to be a combined operation of max and argmax. I.e., it stores both the maximum value as well as the element that holds the maximum value.

$$\begin{aligned}(p^*, \boldsymbol{x}_1^*, \boldsymbol{x}_2^*, \cdots, \boldsymbol{x}_N^*) &= \underset{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N}{\text{amax}} \{p(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{N-1}, \boldsymbol{x}_N)\} \\ &= \underset{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N}{\text{amax}} \left\{\frac{1}{Z}\phi_{1,2}(\boldsymbol{x}_1, \boldsymbol{x}_2)\phi_{2,3}(\boldsymbol{x}_2, \boldsymbol{x}_3)\cdots\phi_{N-1,N}(\boldsymbol{x}_{N-1}, \boldsymbol{x}_N)\right\}\end{aligned} \tag{7}$$

We can exploit the chain structure to ditribute the *amax* operation as follows.

$$\begin{aligned}&\underset{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N}{\text{amax}} \left\{\frac{1}{Z}\phi_{1,2}(\boldsymbol{x}_1, \boldsymbol{x}_2)\phi_{2,3}(\boldsymbol{x}_2, \boldsymbol{x}_3)\cdots\phi_{N-1,N}(\boldsymbol{x}_{N-1}, \boldsymbol{x}_N)\right\} \\ &= \underset{\boldsymbol{x}_1}{\text{amax}}\left\{\underset{\boldsymbol{x}_2}{\text{amax}}\left\{\phi_{1,2}(\boldsymbol{x}_1, \boldsymbol{x}_2)\underset{\boldsymbol{x}_3}{\text{amax}}\left\{\phi_{2,3}(\boldsymbol{x}_2, \boldsymbol{x}_3)\cdots\underset{\boldsymbol{x}_N}{\text{amax}}\{\phi_{N-1,N}(\boldsymbol{x}_{N-1}, \boldsymbol{x}_N)\}\cdots\right\}\right\}\right\}\end{aligned} \tag{8}$$

where I abused the notation of $\underset{\boldsymbol{x}_{i-1}}{\text{amax}}\{\}$ be a function of $\boldsymbol{x}_i$.

We define the following recursive definition.

$$\phi'_{i-1}(\boldsymbol{x}_{i-1}) = \underset{\boldsymbol{x}_i}{\text{amax}}\{\phi_{i-1,i}(\boldsymbol{x}_{i-1}, \boldsymbol{x}_i)\phi'_{i-1}(\boldsymbol{x}_{i-1})\} \tag{9}$$

Evaluation of each of K elements of $\phi'_{i-1}(\boldsymbol{x}_{i-1})$ takes takes K evaluations of $\phi(\boldsymbol{x}_{i-1}, \boldsymbol{x}_i)$ and K comparisons. Hence the construction of one $\phi'_{i-1}(\boldsymbol{x}_{i-1})$ is $O(K^2)$. Then,

$$\begin{aligned}&\underset{\boldsymbol{x}_1}{\text{amax}}\left\{\underset{\boldsymbol{x}_2}{\text{amax}}\left\{\phi_{1,2}(\boldsymbol{x}_1, \boldsymbol{x}_2)\underset{\boldsymbol{x}_3}{\text{amax}}\left\{\phi_{2,3}(\boldsymbol{x}_2, \boldsymbol{x}_3)\cdots\underset{\boldsymbol{x}_N}{\text{amax}}\{\phi_{N-1,N}(\boldsymbol{x}_{N-1}, \boldsymbol{x}_N)\}\cdots\right\}\right\}\right\} \\ &= \underset{\boldsymbol{x}_1}{\text{amax}}\left\{\underset{\boldsymbol{x}_2}{\text{amax}}\left\{\phi_{1,2}(\boldsymbol{x}_1, \boldsymbol{x}_2)\underset{\boldsymbol{x}_3}{\text{amax}}\{\phi_{2,3}(\boldsymbol{x}_2, \boldsymbol{x}_3)\cdots\phi'_{N-1}(\boldsymbol{x}_{N-1})\cdots\}\right\}\right\} \\ &= \underset{\boldsymbol{x}_1}{\text{amax}}\left\{\underset{\boldsymbol{x}_2}{\text{amax}}\{\phi_{1,2}(\boldsymbol{x}_1, \boldsymbol{x}_2)\phi'_2(\boldsymbol{x}_2)\}\right\} \\ &= \underset{\boldsymbol{x}_1}{\text{amax}}\{\phi'_1(\boldsymbol{x}_1)\} \\ &= (p^*, \boldsymbol{x}_1^*)\end{aligned} \tag{10}$$

This derivation can be viewed as a message passing from $\boldsymbol{x}_{N-1}$ up to $\boldsymbol{x}_1$ with the message defined by $\phi'_i(\boldsymbol{x}_i)$. To obtain $(\boldsymbol{x}_2^*, \cdots, \boldsymbol{x}_N^*)$, we can back track the operation from $\boldsymbol{x}_1$ to $\boldsymbol{x}_{N-1}$. Overall the complexity of finding $p(\boldsymbol{x}_N)$ is $O(NK^2)$.

3.2 Inference in a Tree

We extend the idea described in the chain to trees. There are two types of extensions in the factor graph.

- One factor can take more than two random variables.
- One variable can be associated to more than two factors.

In order to accommodate those extensions, we define the recursive relation in two phases. Please see figure 11.

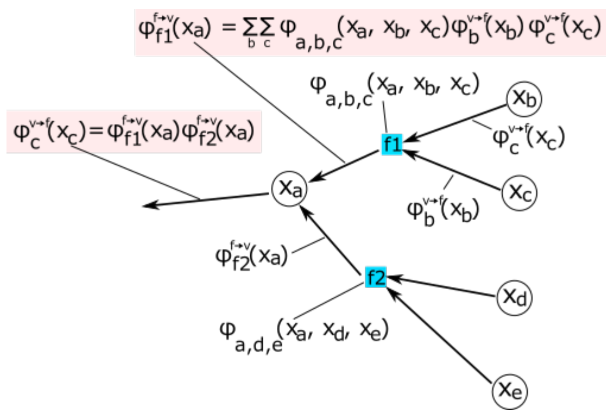

Figure 11: Propagation in a Tree

We have to nominate one random variable node $\boldsymbol{x}_{root}$ in the factor graph as the root, and form a rooted oriented tree by orienting the edges from the root toward leaves. Such a tree is unique up to the edge orientation. In case of marginalization, $\boldsymbol{x}_{root}$ is the random variable for which, the rest of the random variables are marginalized to obtain $p(\boldsymbol{x}_{root})$.

$\phi^{f\to v}_{\boldsymbol{f}}(\boldsymbol{x})$ is from a factor to a variable, and $\phi^{v\to f}_{\boldsymbol{x}}(\boldsymbol{x})$ is from a variable to a factor.

$\phi^{f\to v}_{\boldsymbol{f}_a}(\boldsymbol{x}_i)$ represent the subtree under $\boldsymbol{f}_a$ as a function of $\boldsymbol{x}_i$. $\phi^{v\to f}_{\boldsymbol{x}_i}(\boldsymbol{x}_i)$ represent the subtree under $\boldsymbol{x}_i$ as a function of $\boldsymbol{x}_i$, i.e.,

$$\phi^{v\to f}_{\boldsymbol{x}_i}(\boldsymbol{x}_i) = \prod_{\boldsymbol{f}_j \in N^-(\boldsymbol{x}_i)} \phi^{f\to v}_{\boldsymbol{f}_j}(\boldsymbol{x}_i)$$

Now, $\phi^{f\to v}_{\boldsymbol{f}}(\boldsymbol{x}_a)$ is defined as a marginalization or maximization of $\phi_{\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\cdots,\boldsymbol{x}_{iL}}(\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\cdots,\boldsymbol{x}_{iL})$ where $\boldsymbol{x}_a \in \{\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\cdots,\boldsymbol{x}_{iL}\}$. Then

$$\phi^{f\to v}_{\boldsymbol{x}_a}(\boldsymbol{x}_a) = \sum_{\{\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\cdots,\boldsymbol{x}_{iL}\}\setminus\boldsymbol{x}_a} \left(\phi_{\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\cdots,\boldsymbol{x}_{iL}}(\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\cdots,\boldsymbol{x}_{iL}) \prod_{\boldsymbol{x}_b}^{\{\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\cdots,\boldsymbol{x}_{iL}\}\setminus\boldsymbol{x}_a} \left(\phi^{v\to f}_{\boldsymbol{x}_b}(\boldsymbol{x}_b)\right)\right) \tag{11}$$

or,

$$\phi^{f\to v}_{\boldsymbol{x}_a}(\boldsymbol{x}_a) = \underset{\{\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\cdots,\boldsymbol{x}_{iL}\}\smallsetminus\boldsymbol{x}_a}{\text{amax}} \left\{\phi_{\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\cdots,\boldsymbol{x}_{iL}}(\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\cdots,\boldsymbol{x}_{iL}) \prod_{\boldsymbol{x}_b}^{\{\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\cdots,\boldsymbol{x}_{iL}\}\smallsetminus\boldsymbol{x}_a} \left(\phi^{v\to f}_{\boldsymbol{x}_b}(\boldsymbol{x}_b)\right)\right\} \tag{12}$$

3.3 Inference in a Grid

We can't use a belief propagation if the graph contains a cycle. If we force a belief propagation to such a graph and assume the propagation to converge, then it is called *loopy belief propagation*.

Assume the latent variables are discrete, i.e. $\boldsymbol{x}_i = \{0,1\}^K$, $\sum_{j=1}^K x_{i_j} = 1$ and the joint probability is represented as follows.

$$\begin{aligned}
p(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N,\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_N) &= \prod_{i=1}^N p(\boldsymbol{x}_i|\boldsymbol{z}_i)p(\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_N)\\
&= \prod_{i=1}^N p(\boldsymbol{x}_i|\boldsymbol{z}_i)\prod_{\{u,v\}\in\mathcal{C}} p(\boldsymbol{z}_u,\boldsymbol{z}_v)\\
&= \prod_{i=1}^N \text{Mult}(\boldsymbol{x}_i|\boldsymbol{z}_i)\prod_{\{u,v\}\in\mathcal{C}} \frac{1}{Z_{\{u,v\}}}\exp\left(-\phi\left(\boldsymbol{z}_u,\boldsymbol{z}_v\right)\right)\\
&= \prod_{i=1}^N\prod_{j=1}^K (z_{i_j}^{x_{i_j}})\frac{1}{Z_w}\exp\left(-\sum_{\{u,v\}\in\mathcal{C}}\phi\left(\boldsymbol{z}_u,\boldsymbol{z}_v\right)\right)\\
&= \frac{1}{Z}\exp\left(\sum_{i=1}^N\sum_{j=1}^K x_{i_j}\ln\!\left(z_{i_j}\right) - \sum_{\{u,v\}\in\mathcal{C}}\phi\left(\boldsymbol{z}_u,\boldsymbol{z}_v\right)\right)\\
&= \frac{1}{Z}\exp\left(\sum_{i=1}^N U_i(z_i) - \sum_{\{u,v\}\in\mathcal{C}} P_{u,v}\left(\boldsymbol{z}_u,\boldsymbol{z}_v\right)\right)
\end{aligned} \tag{13}$$

where $\mathcal{C}$ is the set of edges in the grid. Please see figure 12

After observing $(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N)$, the posterior will be:

$$\begin{aligned}
p(\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_N|\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N) &= \frac{p(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N,\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_N)}{p(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N)}\\
&\propto p(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N,\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_N)\\
&= \frac{1}{Z}\exp\left(\sum_{i=1}^N U_i(z_i) - \sum_{\{u,v\}\in\mathcal{C}} P_{u,v}\left(\boldsymbol{z}_u,\boldsymbol{z}_v\right)\right)
\end{aligned} \tag{14}$$

Then the MAP estimate will be:

$$\underset{\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_N}{\text{argmax}}\ \{p(\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N,\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_N)\} = \underset{\boldsymbol{z}_1,\boldsymbol{z}_2,\cdots,\boldsymbol{z}_N}{\text{argmax}}\left\{\sum_{i=1}^N U_i(z_i) - \sum_{\{u,v\}\in\mathcal{C}} P_{u,v}\left(\boldsymbol{z}_u,\boldsymbol{z}_v\right)\right\} \tag{15}$$

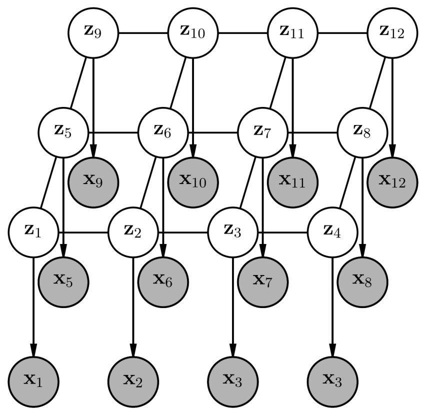

Figure 12: Generative Grid

If you want a MAP estimate, then we can use a max-flow min-cut paradigm, but it will be inpractical if K is large, as you have to form a graph whose number of nodes is $K + 1$ times the number of latent variables, and between two latent variables there are $2K + K^2$ edges as a acomplete bipartite graph. Please see section 12.2 and 12.3 of [3].

The same technique can be applied for a model in conditional random field as in figure 13. The probability distribution is formulated as follows.

$$p(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N, \boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_N) = \frac{1}{Z} \exp\left(- \sum_{\{u,v\} \in \mathcal{C}} \phi\left(\boldsymbol{z}_u, \boldsymbol{z}_v\right) - \sum_{i=1}^{N} \eta\left(\boldsymbol{x}_i, \boldsymbol{z}_i\right) \right) \tag{16}$$

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1 edition, 2007.

[2] David S. Fulford, Dan Foreman-Mackey, and David W. Hogg. Daft : Beautifully rendered probabilistic graphical models. https://docs.daft-pgm.org/en/latest. Accessed: 2020-03-30.

[3] Simon J. D. Prince. *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, 2012.

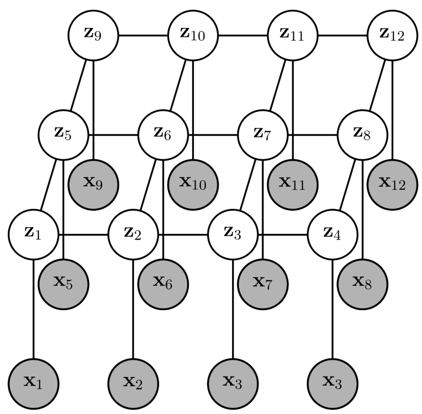

Figure 13: Conditional Random Field Grid