Notes on Markov Networks

Lili Mou
moull12@sei.pku.edu.cn

December, 2014

This note covers basic topics in Markov networks. We mainly talk about the formal definition, Gibbs sampling for inference, and maximum likelihood estimation.

This note is written by Lili Mou. The materials are derived from various sources, especially Daphne Koller's book and open courses.

1 Introduction

Joint probability and dependencies

Consider a set of random variables $\boldsymbol{X} = (X_1, \cdots, X_n)^T$. We assume the variables take finite discrete values. People will be happy if the **joint probability table** is known, e.g. God told you one night. See Table 1. The joint probability distribution is **complete** when modeling such random variables. However, two facts prevent it from being a real-world practice.

1. The number of parameters grows exponentially with respect to n, as long as there is no closed form representation of the joint probability. Let us say that there are n random variables, and each variable takes $|V|$ values. The table has $|V|^n$ entries, and $|V|^n - 1$ free parameters in general. (Note that $\sum_{\boldsymbol{x}} P(\boldsymbol{x}) = 1$.)
2. It is infeasible to estimate the parameters, as the number is exponential. Say we have 50 variables, maximum likelihood estimation (just counting the fraction in this scenario) would give most entries equal to 0, provided any reasonable sized training data samples.

Life will be easier if we have **independencies**. In the extreme case, where all variables are independent, we have

$$P(X_1, \cdots, X_n) = P(X_1) \cdot \ \cdots \ \cdot P(X_n) \tag{1}$$

The number of parameters grows linearly with respect to n. However, such model cannot represent most real-world data as the independence assumption is usually not true (and also not interesting).

Table 1: Joint probability table on $X_1, \cdots, X_n$

X_1	X_2	$\cdots$	$P(X_1, X_2, \cdots)$
0	0	$\cdots$	$\cdot$
1	0	$\cdots$	$\cdot$
0	1	$\cdots$	$\cdot$
1	1	$\cdots$	$\cdot$
	$\cdots$		$\cdots$

Bayesian Networks

Bayesian networks **decomposite** the joint distribution in a "**cause-and-effect**" fashion. For example, let us consider two variables

- R: Whether it rains
- W: Whether the road is wet

We know from physical rules that R is the cause of W. First, it rains or not as God likes with probability $P(r^1)$.[1] Second, if it rains, the road is wet with probability $P(w^1|r^1)$; if it does not rain, the road is wet with probability $P(w^1|r^0)$. Then, the joint distribution is

$$P(W, R) = P(R)P(W|R)$$

In general, for variables $X_1, \cdots, X_n$, with each variable X_i having its causes $\operatorname{Par}(X_i)$, the joint probability is

$$P(\boldsymbol{X}) = \prod_{i=1}^{n} P(X_i | \operatorname{Par}(X_i)) \tag{2}$$

Constructing the **Bayesian network** from the above conditional probability is easy. Each variable corresponds to a node in the graph. What we need to do then is to add a directed edge $X \to Y$ if X is the cause of Y. The Bayesian network for R, W is

$$\begin{tikzcd} (R) \arrow[r] & (W) \end{tikzcd}$$

Note that

1. $\emptyset \subseteq \operatorname{Par}(X_i) \subseteq \boldsymbol{X} \backslash \{X_i\}$, and at least one variable has no causes.
2. Two variables cannot be the causes for each other, considering the transitivity of the "cause-and-effect" relation. Formally speaking, the Bayesian network is a directed acyclic graph.

Many philosophy discussion can be added here, especially on what is cause and what is effect from a Bayesian network prospective. However, they are well beyond the scope of this note as we are talking about Markov networks.

Markov Networks

Even though Bayesian network is complete for modeling random variables, it is not successful in all applications. For example, in an image, two neighboring pixels are certainly correlated, but we can say little which pixel is the "cause" of the other. When modeling with a Bayesian network, the "causes" of a pixel will grow exponentially, and thus, little is gained from such decomposition in terms of computational cost.

Markov networks are yet another probability decomposition approach. As an **undirected graph**, a node X is connected to node Y if they are **somewhat related** regardless of other variables. Formal definitions of Markov networks will be given in the next section.

In short, both Bayesian networks and Markov networks model the joint probability among random variables by decomposition. The goal is to simplify the joint probability distribution, as well as preserve interesting dependencies.

[1] x^1 is a short notation for $X = 1$.

2 Markov Networks

A Toy Example

Take Daphne Koller's interesting example, where we have 4 random variables A, B, C, D, corresponding to whether Alice, Bob, Charles, Debbie have a correct understanding of some materials, say Bayesian networks. Alice and Bob study together often, as they may influence each other. So as Bob, Charles; Charles, Debbie; and Debbie, Alice.

How they effect each other is modeled by a set of factor tables (Table 2). As we see, Alice and Bob are good friends, they are extremely prone to have the same understanding (either correct or incorrect); Bob, Charles and Debbie, Alice also tend to have the same understanding, but they are not as strong as Alice and Bob; Charles and Debbie usually argue, and they are more likely to hold different opinions.

The **factor** here has many other terminologies, including potential, affinity, compatibility, soft constraint, etc. To best understand this concept, we quote Daphne Koller's explanation as "**local happiness for a certain assignment**." The word "local" emphasizes on one important fact that, the distribution of one variable not only depends on local factors, but also on many other variables, concretely, any variable appears in one factor, and those by transitivity of "relatedness. The exact meaning of factors is defined in the following part.

Formal definition

"Meaning is use," said Wittgenstein. The meaning of factors is defined exactly through how it is related to the joint probability.

Let $\phi_i(D_i), i = 1..k$ be a set of factors defined on random variables $X_1, \cdots, X_n$. $D_i \subseteq X_i$ is the scope of factor ϕ_i. The only constraint on factors is that the entries in the factors are non-negative. To be more friendly, they are often strictly positive. We define

- **Unnormalized measure**

$$\tilde{P}(\boldsymbol{X}) = \prod_{i=1}^{k} \phi_i(D_i) \tag{3}$$

- **Partition function**

$$Z = \sum_{\boldsymbol{x}} \tilde{P}(\boldsymbol{x}) = \sum_{\boldsymbol{x}} \prod_{i=1}^{k} \phi_i(D_i) \tag{4}$$

- **Probability**

$$P(\boldsymbol{X}) = \frac{1}{Z}\tilde{P}(X) = \frac{\prod_{i=1}^{k} \phi_i(D_i)}{\sum_{\boldsymbol{x}} \prod_{i=1}^{k} \phi_i(D_i)} \tag{5}$$

Do not be scared by the formula menagerie. In fact, the meaning of factors are pretty much the same as in Bayesian network—the probability is closely related to the product of all factors, analogous

Table 2: Factor tables for (A, B), (B, C), (C, D) and (D, A)

A	B	$\phi(A,B)$
0	0	1000
0	1	1
1	0	1
1	1	1000

B	C	$\phi(B,C)$
0	0	100
0	1	1
1	0	1
1	1	100

C	D	$\phi(C,D)$
0	0	1
0	1	10
1	0	10
1	1	1

D	A	$\phi(D,A)$
0	0	100
0	1	1
1	0	1
1	1	100

to the product of conditional probability. The only difference lies in that the products do not sum to 1 over $\boldsymbol{x}$'s. Thus, $\tilde{P}$ is called unnormalized measure in Equation 3. Then, to obtain a normalized probability, we divide $\tilde{P}$ by the sum over all possible x's (Equation 4 and 5). This model is often referred to as a **Markov Random Field** (MRF).

Beginners usually concern on how to obtain the parameters (ϕ's) in MRFs. It is typically not feasible for human experts to specify because people do not have a clear intuition about factors (except the fact that they are local happiness for a certain assignment). In practice, they are learned by maximum likelihood estimation, introduced in Section 4, or its variations.

Inducing a Markov Network

The Markov network, induced from the Markov random field, is defined as follows.

- Each node corresponds to a random variable.
- X_i is connected to X_j with an undirected edge if and only if there exits a factor, whose scope contains both X_i and X_j, i.e., $\exists D_k$, s.t. $X_i, X_j \in D_k$

It should also be noticed that, given a set of factors, the Markov network is unique; given a Markov network, we cannot read factorization from the network.

Toy Example Revisit

We are now equipped to compute the joint distribution in the toy example. Suppose we would like to know $P(a^0, b^1, c^0, d^0)$, we first compute

$$\begin{aligned}\tilde{P}(a^0, b^1, c^0, d^0) &= \phi_{A,B}(a^0, b^1)\phi_{BC}(b^1, c^0)\phi_{CD}(c^0, d^0)\phi_{DA}(d^0, a^0)\\ &= 1 \times 1 \times 1 \times 100\\ &= 100\end{aligned}$$

Then we compute $Z = \sum_{\boldsymbol{x}} \tilde{P}(\boldsymbol{x})$, which is a little bit difficult because we should sum over $4^4 = 256$ terms. Finally, the probability is obtained by $P(a^0, b^1, c^0, d^0) = \frac{1}{Z}\tilde{P}(a^0, b^1, c^0, d^0)$

As $D_1 = \{A, B\}, D_2 = \{B, C\}, D_3 = \{C, D\}, D_4 = \{D, A\}$, the induced Markov network is

$$\begin{tikzcd}
\mathrm{A} \arrow[r, dash] \arrow[d, dash] & \mathrm{B} \arrow[d, dash] \\
\mathrm{D} \arrow[r, dash] & \mathrm{C}
\end{tikzcd}$$

Notes on maximal cliques

In some text books, factors are defined on maximal cliques in an undirected graph. However, it is not essentially the case, and is somewhat misleading in that

1. We usually specify a set of factors and induce the Markov networks to help us understand the dependencies, but not vice versa.
2. Factors are not necessarily defined on *maximal* cliques. Even though any factor can be absorbed into maximal clique factors—that is, we can always replace a factor with a maximal clique factor, whose scope is a superset of the current one—the parametrization is not equivalent, especially during the learning process. In fact, it is a common practice to assign unigram factors along with other factors in sequential labeling in natural language processing.

Conditional Random Fields

Conditional Random Fields (CRFs) are essentially the same as MRFs except that we hold different point of views—we model the conditional probability instead of the joint probability.

Oftentimes, the set of random variables that we need to model can be split into two groups $\boldsymbol{X}$ and $\boldsymbol{Y}$. $\boldsymbol{X}$ is always given, so that we have special interest in $\boldsymbol{Y}$. (This is a major difference between supervised learning and unsupervised learning. Recall linear regression and principle component analysis.)

As $\boldsymbol{X}$ is always given, we may not model $P(\boldsymbol{X})$, and hence $P(\boldsymbol{X}, \boldsymbol{Y})$, which is sometimes uneconomic and infeasible. For example, in an image classification problem, the task is to tag a label (Y) to a given image ($\boldsymbol{X}$). That is, what we need to know is $P(Y|\boldsymbol{X})$ rather than $P(\boldsymbol{X}, Y)$. We call this a **discriminative model** if we do not model $P(\boldsymbol{X}, \boldsymbol{Y})$, in contrast to **generative models** modeling $P(\boldsymbol{X}, \boldsymbol{Y})$.

In the MRF setting, the conditional probability

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \frac{P(\boldsymbol{Y}, \boldsymbol{X})}{P(\boldsymbol{X})} = \frac{\tilde{P}(\boldsymbol{Y}, \boldsymbol{X})}{\tilde{P}(\boldsymbol{X})} = \frac{\tilde{P}(\boldsymbol{Y}, \boldsymbol{X})}{\sum_{\boldsymbol{y}} \tilde{P}(\boldsymbol{X}, \boldsymbol{y})}$$

Based on the above equation, we define unnormalized measure, partition function and probability for CRFs as follows.

- **Unnormalized measure**

$$\tilde{P}(\boldsymbol{X}, \boldsymbol{Y}) = \prod_{i=1}^{k} \phi_i(D_i) \tag{6}$$

- **Partition function**

$$Z(\boldsymbol{X}) = \sum_{\boldsymbol{y}} \tilde{P}(\boldsymbol{X}, \boldsymbol{y}) = \sum_{\boldsymbol{y}} \prod_{i=1}^{k} \phi_i(D_i) \tag{7}$$

- **Conditional probability**

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \frac{1}{Z(\boldsymbol{X})} \tilde{P}(\boldsymbol{X}, \boldsymbol{Y}) = \frac{\prod\limits_{i=1}^{k} \phi_i(D_i)}{\sum\limits_{\boldsymbol{y}} \prod\limits_{i=1}^{k} \phi_i(D_i)} \tag{8}$$

3 Gibbs Sampling

Sampling methods

A key problem in Markov network it to query the joint distribution, $P(\boldsymbol{x})$. As we have attempted to compute, but not completed in the previous section, we can always sum over all possible assignments $\boldsymbol{x}$. However, the computational cost is usually very high. Dynamic programming techniques can be applied, e.g. the clique tree calibration algorithm. In these cases, the complexity grows exponentially with respect to the tree width. Therefore, a densely connected graph is also not tractable for exact inference.

Sampling methods are a natural solution if we have a simulation system. For example, if we would like to know $P(\text{Head})$ of a coin, one approach is to go through all the physical and mathematical details, which does not seem to be a good idea. An alternative is to toss the coin for multiple times, which will give a fairly good estimation of $P(\text{Head})$.

Similarly, if we are able to get an unbiased sample of $\boldsymbol{X}$ in Markov networks, estimating $P(\boldsymbol{x})$ becomes just a matter of counting the fraction $\dfrac{\text{Occurrences of } \boldsymbol{x}}{\text{All samples}}$.

In the rest of this section, we will first introduce the Gibbs sampling algorithm, which is not difficult to understand. Then, we introduce Markov chain and a significant theorem. Finally, we prove that Gibbs sampling gives unbiased samples.

Gibbs Sampling Algorithm

The Gibbs sampling algorithm is simple, which continuously samples a variable $X_i, (i = 1 \cdots n)$ from its posterior distribution with all other variables temporally fixed. After a long while, the samples are guaranteed to be unbiased. Formally, the algorithm is presented in Algorithm 1.

The input is an initialization of variables $\boldsymbol{X}^{(0)}$, and the posterior distribution of X_i given all other variables, $\boldsymbol{X} \backslash \{X_i\}$, denoted as $\boldsymbol{X}_{-i}$. The initialization does not matter much in Gibbs sampling algorithm. Then, we continuously sample $\boldsymbol{X}^{(t)}$ by iterating over each variable X_i and sampling it from its posterior $X_i \sim P(X_i|\boldsymbol{X}_{-i})$. To illustrate this process, we consider three variables $\boldsymbol{X} = (X_1, X_2, X_3)^T$, each taking binary values $\{0, 1\}$. One run of Gibbs sampling is shown in Table 3. Note that the posterior distributions in the table and the outcomes of the random trials are as I like.

Computing posterior distributions in Markov networks

As the posterior distribution is the key of Gibbs sampling, it should be required that computing posteriors is tractable in Markov networks. Otherwise, we do not gain anything from Gibbs sampling, regardless of the long while before samples become unbiased.

In fact, the posterior distribution $X_i \sim P(X_i|\boldsymbol{X}_{-i})$ is only related to factors whose scopes contain X_i, which means the computation is "local," and hence oftentimes tractable. Noting

$$P_\Phi(X_i|\boldsymbol{X}_{-i}) = \frac{P(X_i, \boldsymbol{X}_{-i})}{P(\boldsymbol{X}_{-i})} = \frac{\frac{1}{Z}\tilde{P}(X_i, \boldsymbol{X}_{-i})}{\frac{1}{Z}\tilde{P}(\boldsymbol{X}_{-i})} = \frac{\prod\limits_{\phi\in\Phi} \phi}{\sum\limits_{x_i} \prod\limits_{\phi\in\Phi} \phi}$$

we pull out factors that do not contain x_i in the denominator, canceling out exactly those in the numerator. Therefore, we only need to multiply the factors that involve X_i, i.e.,

$$P_\Phi(X_i|\boldsymbol{X}_{-i}) \propto \prod_{j:X_i\in D_j} \phi_j(X_i, \boldsymbol{X}_{D_j - X_i}) \tag{9}$$

Markov chains

We are now able to implement Gibbs sampling efficiently for Markov networks. But we still need to verify that the **samples are unbiased**. To accomplish this goal, we now introduce Markov chains as a fundamental.

Algorithm 1: Gibbs Sampling

```
Input: Any initialization of $\boldsymbol{X}^{(0)} = (X_1^{(0)}, \cdots, X_n^{(0)})^T$,
       Posterior distributions $P_i(X_i|\boldsymbol{X}_{-i})$, $i = 1..n$
Output: Samples $\{X^{(t)}\}_{t=1}^{\infty}$, unbiased if $t \to \infty$
while $t = 1..\infty$ do
    for $i = 1..n$ do
        $X_i \sim P(X_i|\boldsymbol{X}_{-i})$
    end
    Obtain $\boldsymbol{X}^{(t)}$
end
```

Table 3: One run of Gibbs sampling.
Underlined variables are the one to be sampled in this iteration.

Samples	Variables	Sampling preocess
$\boldsymbol{X}^{(0)}$	$\underline{0}$, 1, 0	Random initialization
	1 , $\underline{1}$, 0	$X_1 \sim P(X_1 \mid X_2 X_3)$
	1, 1, $\underline{0}$	$X_2 \sim P(X_2 \mid X_1 X_3)$
$\boldsymbol{X}^{(1)}$	$\underline{1}$, 1, 0	$X_3 \sim P(X_3 \mid X_1 X_2)$
	0, $\underline{1}$, 0	$X_1 \sim P(X_1 \mid X_2 X_3)$
	0, 0, $\underline{0}$	$X_2 \sim P(X_2 \mid X_1 X_3)$
$\boldsymbol{X}^{(2)}$	0, 0, 1	$X_3 \sim P(X_3 \mid X_1 X_2)$
	⋮	

A **Markov chain** has two components

- States $s_1, \cdots, s_n$
- Transition probability $\mathcal{T}(s_i \to s_j) = P(s^{(t+1)} = s_j | s^{(t)} = s_i), \forall t$.

The system has discrete time clocks. As each time t, the system is in one (and only one) state. When the clock ticks, the system goes to a certain state with the transition probability. The **Markov assumption** assumes

1. The next state is only related to the current state, i.e., the transition probability is only conditioned on $s^{(t)}$, represented as $\mathcal{T}$.
2. For different time steps, the transition probability is exactly the same.

A famous example is the Markov chain of weather. Assume the weather can be {Sunny, Cloudy, Rainy}. Tomorrow's weather is related to and only to today's weather with some transition probability.

If the system's state at time t is unknown, we can model it with a probabilistic distribution $P(s^{(t)} = s_i)$. Then, the probability of the next time step $t+1$ is given by

$$P(s^{(t+1)} = s_j) = \sum_{i=1}^{n} p(s^{(t)} = i)\mathcal{T}(s_j \to s_i) \tag{10}$$

More compactly, let $\boldsymbol{p}^{(t)} \in \mathbb{R}^n$ be the distribution over states $s_1, \cdots, s_n$, and let $\mathcal{T} \in \mathbb{R}^{n\times n}$ be the transition matrix, with $\mathcal{T}_{i,j} = \mathcal{T}(s_i \to s_j)$.[2] We can rewrite the transition function as

$$\boldsymbol{p}^{(t+1)} = \mathcal{T}^T \boldsymbol{p}^{(t)} \tag{11}$$

Oftentimes, the Markov chain will reach a **stationary distribution** π after a long time, where the distribution does not change during transition. In such cases, the stationary distribution $\boldsymbol{\pi}$ satisfies

$$\boldsymbol{\pi} = \mathcal{T}^T \boldsymbol{\pi} \tag{12}$$

That is to say, $\boldsymbol{\pi}$ is the eigenvector of $\mathcal{T}^T$, corresponding to eigenvalue 1.

Theorem

[2]The transition matrix's row and column may be different from text book conventions. I am not sure about that.

A Markov chain as a unique stationary distribution $\boldsymbol{\pi}$

$$\Uparrow$$

The Markov chain is regular, i.e., $\exists k, \forall x, x', P(x \to x'\text{with exactly } k \text{ steps}) > 0$

$$\Uparrow$$

$$\forall i, j,\ \mathcal{T}_{i,j} > 0$$

■

The proof of this theorem is beyond the scope of this note. But here are some intuitive thoughts on the theorem.

- Google's PageRank algorithm takes advantage of Markov chains. No matter what web page you are visiting now, you will reach a unique distribution over web pages if you surf long enough on the Internet (provided that any two web pages are linked, usually accomplished by assigning a random jump with some low probability).
- Note that the transition is linear, which guarantees no chaos. Besides, each row in the transition matrix $\mathcal{T}$ sums to 1, which guarantees the system is conservative.

Gibbs sampling revisit

We now prove that Gibbs sampling will give unbiased samples by making use of Markov chains.
Proof Design a Markov chain as follows.

- Let the states be all possible assignments $\boldsymbol{x}$
- Let the transition matrix $\mathcal{T} = \prod_{i=1}^{n} \mathcal{T}_i$, where

$$\mathcal{T}_i\left((\boldsymbol{x}_{-i}, x_i') \to (\boldsymbol{x}_{-i}, x_i)\right) = P(x_i|\boldsymbol{x}_{-i}),\ \forall x_i' \tag{13}$$

  The transition process is as follows. We iterate over all variables x_i. For each variable, we sample x_i from its posterior, regardless of current value x_i'. Note that $\boldsymbol{x}_{-i}$ does not change when sampling x_i. As $\mathcal{T}_i$ is the transition probability when sampling x_i, the overall transition probability $\mathcal{T}$ is $\prod_{i=1}^{n} \mathcal{T}_i$.[3]

We have now designed a Markov chain, which conforms to Gibbs sampling process. What we are going to do is to prove the Markov chain has the unique stationary distribution $P(\boldsymbol{x})$. First, it is obvious that the Markov chain is regular. We now verify that

$$P(\boldsymbol{x}) = \sum_{\boldsymbol{x}'} \mathcal{T}_i(\boldsymbol{x}_i' \to \boldsymbol{x}_i) P(\boldsymbol{x}') \tag{14}$$

[3] $\boldsymbol{x}_{-i}$ means $\{x_1, \cdots, x_n\} \backslash \{x_i\}$

$$\begin{aligned} \text{LHS} &= P(\boldsymbol{x}) \\ \text{RHS} &= \sum_{x_i'} \mathcal{T}_i((\boldsymbol{x}_{-i}, x_i') \to (\boldsymbol{x}_{-i}, x_i))P(\boldsymbol{x}') && (15) \\ &= \sum_{x_i'} P(x_i|\boldsymbol{x}_{-i})P(\boldsymbol{x}') && (16) \\ &= \sum_{x_i'} P(x_i|\boldsymbol{x}_{-i})P(\boldsymbol{x}_{-i}, x_i') && (17) \\ &= P(\boldsymbol{x}_{-i})P(x_i|\boldsymbol{x}_{-i}) && (18) \\ &= P(x_i, \boldsymbol{x}_{-i}) && (19) \\ &= P(\boldsymbol{x}) && (20) \end{aligned}$$

Equation 15 holds by noticing that $\boldsymbol{x}_{-i}$ must remain the same. Equation 16 holds by the design of Markov chain transition probability (Equation 13). Equation 17 holds by rewriting $\boldsymbol{x}' = (x_i', \boldsymbol{x}_{-i})$. Equation 18 holds by pulling $P(x_i|\boldsymbol{x}_{-i})$ out of summation since x_i' do not appear in it. Then we marginalize out $\boldsymbol{x}_i'$. Equation 19 holds by chain rule. Finally, we rewrite $(x_i, \boldsymbol{x}_{-1}) = \boldsymbol{x}$.

■

4 Maximum Likelihood Estimation

Philosophy

Statistics is more about philosophy than mathematics. God knows the parameters, but we humans do not. Parameter estimation depends largely on people's philosophy or belief. **Maximum likelihood estimation** (MLE) assumes that more probable events happen first; less probable events happen later. Given a dataset, MLE would like the probability of the dataset to be maximum with respect to parameters. The general objective is

$$\underset{\Theta}{\text{maximize}} \mathscr{L}(\Theta, \mathcal{D}) \tag{21}$$

where $\mathscr{L}(\Theta, \mathscr{D}) = P(\mathscr{D}; \Theta)$. For convenience, we usually maximize the log likelihood $\ell(\Theta, \mathscr{D}) = \log \mathscr{L}(\Theta, \mathscr{D})$, which is exactly equivalent to maximizing likelihood.

Please verify yourself that Markov networks have no closed form solution for MLE. Fortunately, the objective (Equation 21) is concave for Markov networks. That is to say, we can reach global maximum by gradient ascent methods.

In the following part, we first re-parametrize MRF as a **log-linear model**. Then we derive the gradient for solving MLE.

Reparametrization

Let $f_i(\boldsymbol{x}), i = 1..k$, be a set of features defined on $\boldsymbol{x}$. The unnormalized measure of the log-linear model is defined as

$$\tilde{P}(\boldsymbol{x}) = \exp\left\{\sum_{i=1}^{k} \theta_i f_i(\boldsymbol{x})\right\} \tag{22}$$

The partition function and probability is defined exactly as before.

Any MRF can be re-parametrized by a log-linear model. For factor ϕ whose scope is $X_i, X_2, \cdots, X_d$, we define a bunch of features

$$f_{x_1, x_2, \cdots, x_n}(X_1, X_2, \cdots, X_d) = \begin{cases} 1, & \text{if } X_1 = x_1, X_2 = x_2, \cdots, X_d = x_d \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

Note that, if X_i takes $|V_i|$ values, the total number of features corresponding to the factor is $\prod_{i=1}^{d} |V_i|$.

As exponentiation transforms multiplications (in MRF) to summations (in the exponent in log-linear model), ϕ and θ have the following relationship.

$$\begin{aligned}
\phi_{X_1,\cdots,X_d}(x'_1,\cdots,x'_d) =& \exp\left\{\sum_{x'_1\cdots,x'_d} \theta x'_1,\cdots,x'_d f_{x'_1,\cdots,x'_d}(x_1,\cdots,x_d)\right\} & (24)\\
& \text{[Re-define } \phi] \\
=& \exp\left\{\theta_{x_1,\cdots,x_d}\right\} & (25)\\
& \text{[whenever } x'_i \neq x_i, f_{\boldsymbol{x}'} = 0\text{; otherwise } f_{\boldsymbol{x}'} = 1]
\end{aligned}$$

MLE for log-linear models

$$\begin{aligned}
\frac{1}{M}\ell(\Theta,\mathscr{D}) =& \frac{1}{M}\log P(\mathscr{D};\Theta)) \quad \text{[Definition of log-likelihood]}\\
=& \frac{1}{M}\log\prod_{j=1}^{M} P(X^{(j)};\Theta) \quad \text{[Data samples } \boldsymbol{x}^{(j)} \text{ iid]}\\
=& \frac{1}{M}\sum_{j=1}^{M}\log P(\boldsymbol{x}^{(j)};\Theta) \quad \text{[Pull the production out of logarithm]}\\
=& \frac{1}{M}\sum_{j=1}^{M}\log\frac{\exp\left\{\sum_{i=1}^{k}\theta_i f_i(\boldsymbol{x}^{(j)})\right\}}{\sum_{\boldsymbol{x}'}\exp\left\{\sum_{i=1}^{k}\theta_i f_i(\boldsymbol{x}')\right\}} \quad \text{[Definition of } P(\boldsymbol{x}) \text{ in log-linear models]}\\
=& \frac{1}{M}\sum_{j=1}^{M}\log\exp\left\{\sum_{i=1}^{k}\theta_i f_i(\boldsymbol{x}^{(j)})\right\} - \frac{1}{M}\sum_{j=1}^{M}\log\sum_{\boldsymbol{x}'}\exp\left\{\sum_{i=1}^{k}\theta_i f_i(\boldsymbol{x}')\right\} \quad \text{[Split the logarithm]}\\
=& \frac{1}{M}\sum_{j=1}^{M}\sum_{i=1}^{k}\theta_i f_i(\boldsymbol{x}^{(j)}) - \frac{1}{M}\sum_{j=1}^{M}\log\sum_{\boldsymbol{x}'}\exp\left\{\sum_{i=1}^{k}\theta_i f_i(\boldsymbol{x}')\right\} \quad \text{[log and exp cancel out]}
\end{aligned}$$

We compute the partial derivatives for each term separately.

$$\begin{aligned}
\frac{\partial}{\partial\theta_l}\frac{1}{M}\sum_{j=1}^{m}\sum_{i=1}^{k}\theta_i f_i(\boldsymbol{x}^{(j)}) =& \frac{1}{M}\sum_{j=1}^{m} f_l(\boldsymbol{x}^{(j)})\\
& \text{[Only those terms that contain } \theta_l \text{ have contribution}\\
& \text{to the partial derivative with respect to } \theta_l]\\
=& \mathbb{E}_{\boldsymbol{x}\sim\mathscr{D}}[f_i(\boldsymbol{x})] & (26)\\
& \text{[The partial derivative of the first term}\\
& \text{is the expectation of } f_i(\boldsymbol{x}) \text{ in data]}
\end{aligned}$$

$$
\begin{aligned}
\frac{\partial}{\partial\theta_l}\frac{1}{M}\sum_{j=1}^{M}\log\sum_{\boldsymbol{x}'}\exp\left\{\sum_{i=1}^{k}\theta_i f_i(\boldsymbol{x}')\right\} &= \frac{\partial}{\partial\theta_l}\log\sum_{\boldsymbol{x}'}\exp\left\{\sum_{i=1}^{k}\theta_i f_i(\boldsymbol{x}')\right\}\\
&\qquad [\frac{1}{M}\text{ cancels the summation}]\\
&= \frac{1}{\sum_{\boldsymbol{x}'}\exp\left\{\sum_{i=1}^{k}\theta_i f_i(\boldsymbol{x}')\right\}}\times\sum_{\boldsymbol{x}'}\left\{\exp\left\{\sum_{i=1}^{k}\theta_i f_i(\boldsymbol{x}')\right\}\times f_l(\boldsymbol{x}')\right\}\\
&\qquad [\text{Chain rule}]\\
&= \sum_{\boldsymbol{x}'}\frac{\exp\left\{\sum_{i=1}^{k}\theta_i f_i(\boldsymbol{x}')\right\}}{\sum_{\boldsymbol{x}''}\exp\left\{\sum_{i=1}^{k}\theta_i f_i(\boldsymbol{x}'')\right\}}f_l(\boldsymbol{x}')\\
&\qquad [\text{Pull the denominator into the summation.}\\
&\qquad \text{We can always do that, and we change the name a little bit.}]\\
&= \sum_{\boldsymbol{x}'}P(\boldsymbol{x}')f_l(\boldsymbol{x}')\\
&\qquad [\text{The fraction is exactly the definition of } P(\boldsymbol{x}')]\\
&= \mathbb{E}_{\boldsymbol{x}\sim\Theta}[f_l(\boldsymbol{x})] \qquad (27)\\
&\qquad [\text{The expectation of } f_i \text{ in model.}]
\end{aligned}
$$

Combining Equations 26 and 27, we obtain the beautiful equation

$$
\frac{\partial}{\partial\theta_l}\frac{1}{M}\ell(\Theta,\mathscr{D}) = \mathbb{E}_{\boldsymbol{x}\sim\mathscr{D}}[f_i(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim\Theta}[f_l(\boldsymbol{x})] \qquad (28)
$$