Probabilistic Graphical Models

Representation

1 Basic knowledges

2.1 Factors

Factor $\phi(X_1, X_2, \ldots, X_k)$ is a mapping $\phi : val(X_1, X_2, \ldots, X_k) \to \mathbb{R}$. $val(.)$ means all possible assignments for its scope $\{X_1, X_2, \ldots, X_k\}$.

2.2 Independencies in distributions

- For random variables X and Y, P satisfies X independent of Y, if any of the following formulas is satisfied:

$$P(X, Y) = P(X)P(Y)$$
$$P(X|Y) = P(X)$$
$$P(Y|X) = P(Y)$$

, written as $P \models X \perp Y$.

- For (sets of)random variables X , Y and Z, P satisfies X independent of Y given Z, if any of the following formulas is satisfied:

$$P(X, Y|Z) = P(X|Z)P(Y|Z)$$
$$P(X|Y, Z) = P(X|Z)$$
$$P(Y|X, Z) = P(Y|Z)$$
$$P(X, Y, Z) \propto \phi_1(X, Z)\phi_2(Y, Z)$$

, written as $P \models X \perp Y|Z$.

- For a distribution P, define $I(P)$ as all independencies in P, i.e.,

$$I(P) = \{X \perp Y|Z : P \models X \perp Y|Z\}$$

2 Three Types of Graphical Models

2.1 Bayesian Networks

Bayesian Network is a directed acyclic graph (DAG) G, whose nodes represent the random variables $X_1, X_2, \ldots, X_n$. For each node X_i, there is a CPD: $P(X_i|Par_G(X_i))$, where $Par_G(X_i)$ is the set of parents of X_i in G.

Bayesian Networks represent a joint distribution, via the chain rule for Bayesian Nets, i.e.,

$$P(X_1, X_2, \ldots, X_k) = \prod_i P(X_i|Par_G(X_i)).$$

And if $P(X_1, X_2, \ldots, X_k) = \prod_i P(X_i|Par_G(X_i))$, we say P factorizes over G.

2.2 Markov Networks

2.2.1 Gibbs Distribution

Having a set of factors $\Phi = \{\phi_1(\boldsymbol{D}_1), \phi_2(\boldsymbol{D}_2), \ldots, \phi_k(\boldsymbol{D}_k)\}$, Gibbs distribution is a distribution:

$$P_\Phi(X_1, X_2, \ldots, X_k) = \frac{1}{Z}\tilde{P}_\Phi(X_1, X_2, \ldots, X_k)$$

, where $\tilde{P}_\Phi(X_1, X_2, \ldots, X_k) = \prod_{i=1}^k \phi_i(\boldsymbol{D}_i)$ and Z is the normalizing factor.

2.2.2 Induced Markov Networks

Induced Markov Network is a undirected graph H_Φ, whose nodes represent the random variables $X_1, X_2, \ldots, X_n$. H_Φ has an edge $X_i - X_j$, whenever $\exists \phi_m \in \Phi, s.t. X_i, X_j \in \boldsymbol{D}_m$.

P factorizes over H, if $\exists \Phi = \{\phi_i(\boldsymbol{D}_i)\}_{i=1,2,\ldots,k}$, such that $P = P_\Phi$ and H is the induced graph for Φ.

NB:

- H_Φ can have edges $X_i - X_j$, even there does NOT exist $\phi \in \Phi, s.t. X_i, X_j \in \boldsymbol{D}_m$. Conversely, $\forall \boldsymbol{D}_m$and $\forall X_i, X_j \in \boldsymbol{D}_m$, there MUST exist edge $X_i - X_j$.
- Unlike BNs, MNs does NOT tell us a unique factorization, because for the same H there can be different ways for P to factorize it.

2.3 Conditional Random Fields

2.3.1 Definition

Instead of modeling a joint distribution $P(X_1, X_2, \ldots, X_k)$ as BNs and MNs, CRFs model a conditional distribution $P(Y|X)$. In CRFs, we are not trying to capture the distribution over X, thus the high correlations between X_i ignored.

The way that a CRF models a conditional distribution is just like the case of Markov Nets and Gibbs distribution:

Having a set of factors $\Phi = \{\phi_1(\boldsymbol{D}_1), \phi_2(\boldsymbol{D}_2), \ldots, \phi_k(\boldsymbol{D}_k)\}$, $\tilde{P_\Phi}(X, Y) = \prod_{i=1}^{k} \phi_i(\boldsymbol{D}_i)$, $Z_\Phi(X) = \sum_Y \tilde{P_\Phi}(X, Y)$, then

$$P_\Phi(Y|X) = \frac{1}{Z_\Phi(X)} \tilde{P_\Phi}(X, Y)$$

In this way, we only need factors $\phi(X_i, Y)$, hence successfully getting rid of $\phi(X_i)$.

2.3.2 CRFs and Logistic Models

The simplest CRF is logistic model.

For binary variables $X_1, X_2, \ldots, X_k, Y$, define $\phi_i(X_i, Y) = e^{w_i \mathcal{I}\{X_i=1, Y=1\}}$, i.e.,

$$\phi_i(X_i, Y = 1) = e^{w_i X_i}, \quad \phi_i(X_i, Y = 0) = 1$$

So $\tilde{P}_\Phi$ and $Z_\Phi(X)$ can be calculated as follow:

$$\begin{gathered}
\tilde{P}_\Phi(X_1, X_2, \ldots, X_k, Y = 1) = e^{\sum_i w_i X_i} \\
\tilde{P}_\Phi(X_1, X_2, \ldots, X_k, Y = 0) = 1 \\
Z_\Phi(X_1, X_2, \ldots, X_k) = 1 + e^{\sum_i w_i X_i}
\end{gathered}$$

Therefore the probability $Y = 1$ given $X_1, X_2, \ldots, X_k$ is just the form of logistic model:

$$P(Y = 1|X_1, X_2, \ldots, X_k) = \frac{e^{\sum_i w_i X_i}}{1 + e^{\sum_i w_i X_i}}$$

3 Independency and Factorization

We can view Graphical Models from two perspectives:

- A distribution over the random variables in the nodes (by definition);
- A set of independencies between the random variables in the nodes.

The first point of view is just the factorization as the definitions of BNs and MNs, we now discuss the independencies in BNs and MNs.

3.1 Independencies in Graphs

3.1.1 Bayesian Networks

- A trail $\{X_k\}$in Bayesian Network G is **active** given Z, if
  - $\forall$ v-structures $X_{i-1} \to X_i \leftarrow X_{i+1}$ in the trail, X_i or one of its descendants in Z;
  - no other X_k is in Z.
- If there is no active trail in G between X and Y given Z, we say X and Y are d-separated in Bayesian Networks G given Z, written as

$$d\text{-}sep_G(X, Y|Z)$$

- For a Bayesian Network, define $I(G)$ as all independencies in G, i.e.,

$$I(G) = \{X \perp Y|Z : d\text{-}sep_G(X, Y|Z)\}$$

3.1.2 Markov Networks

- A trail $\{X_k\}$ in Markov Network H is **active** given Z, if
  - no X_k is in Z.
- If there is no active trail in H between X and Y given Z, we say X and Y are separate in Markov Network H given Z, written as

$$sep_H(X \perp Y|Z)$$

- For a Bayesian Network, define $I(H)$ as all independencies in H, i.e.,

$$I(H) = \{X \perp Y|Z : sep_H(X, Y|Z)\}$$

3.1.3 I-map, I-equivalent and Perfect map

For a graph (BN or MN) G and a distribution P, we say G is

- **I-map**: $I(G) \subseteq I(P)$.
- **Minimal I-map**: I-map without redundant edges. But it does NOT indicate that $I(G) = I(P)$.
- **Perfect map**: $I(G) = I(P)$.

For 2 graphs G_1 and G_2, we say they are **I-equivalent** if $I(G_1) = I(G_2)$.

NB. Because of the existence of I-equivalent, perfect map of a distribution is NOT unique.

3.2 Independency and Factorization

- In Bayesian Networks G:

  P factorizes over $G \iff G$ is an I-map of P.

- In Markov Networks H:

  P factorizes over $H \Rightarrow H$ is an I-map of P;

  P is **positive** and H is an I-map of $P \Rightarrow P$ factorizes over H.

3.3 Converting between BNs and MNs

To convert between 2 types of nets, we should not add new independencies because otherwise the new network is not an I-map of the original distribution. But conversely, it is allowed to ignore some independencies when converting.

- BN to MN: lose independencies in v-structures;
- MN to BN: must add triangulating edges to loops.

Thus, perfect map does NOT always exist.

4 Local Structures of Graphical Models

4.1 Local Structures of Bayesian Networks

Tabular CPDs can be problematic, especially when there are too many parents. So we need **General CPDs** to characterize local structure of Bayesian Networks. A general CPD is a factor $\phi(X, Y_1, Y_2, \dots, Y_k)$ such that $\forall y_1, y_2, \dots, y_k,\ \sum_x \phi(x, y_1, y_2, \dots, y_k) = 1$. We can view a general CPD as a function taking values of $X, Y_1, Y_2, \dots, Y_k$ and outputting value $P \in [0, 1]$.

Here are some commonly used general CPDs:

4.1.1 Deterministic CPDs

4.1.1.1 Definition

X is a deterministic function of its parents $Y_1, Y_2, \dots, Y_k$, i.e., $X = f(Y_1, Y_2, \dots, Y_k)$. The CPD is defined as:

$$P(X = x | Y_1 = y_1, Y_2 = y_2, \dots, Y_k = y_k) = \begin{cases} 1, & x = f(y_1, y_2, \dots, y_k); \\ 0, & otherwise. \end{cases}$$

4.1.1.2 Example: Multiplexer CPDs

Define $Par(X) = \{A, Z_1, Z_2, \dots, Z_k\},$ where A is a selector taking value from $1, 2, \dots, k$ and X is a copy of Z_A, the CPD is defined as:

$$P(X = x | A = a, Z_1 = z_1, Z_2 = z_2, \dots, Z_k = z_k) = \begin{cases} 1, & x = z_a; \\ 0, & otherwise. \end{cases}$$

4.1.2 Context-Specific CPDs

To define context-specific CPDs, we first define context-specific independence:

P satisfies that X is **context-specific independent** with Y given Z and $C = c$, if any of the following formulas is satisfied:

$$\begin{gathered} P(X, Y | Z, c) = P(X | Z, c) P(Y | Z, c) \\ P(X | Y, Z, c) = P(X | Z, c) \\ P(Y | X, Z, c) = P(Y | Z, c) \end{gathered}$$

, written as $P \models X \perp Y | Z, c$.

If X is context-specific independent with some of its parents Y_i given the value of some of its other parents Y_j, we can represent the CPD $P(X | Par(X))$ as a context-specific CPD. A very commonly used context-specific CPD is **Tree-CPD**. Besides, it turns out that multiplexer CPDs can also be represented as Tree-CPDs.

4.1.3 CPDs satisfying Independence of Causal Influence (ICI)

4.1.3.1 Definition

Independence of Causal Influence: Influence of multiple variables together can be decomposed into multiple influences of single variables. More concretely, for each parent Y_i, there is a variable Z_i caused by it by some probability. The total influence Z is a deterministic function of $(Z_0), Z_1, Z_2, \ldots, Z_k$. Result X is caused by Z by some probability. In this way, we successfully disentangle the influence of all Y_i.

NB. Independence of causal of influence does NOT means independence of variables. Actually we do not care whether the parents variables are independent or not.

4.1.3.2 Example: Noisy-OR/AND/MAX/...

In these models, X is just Z and Z is OR/AND/MAX/... function of $(Z_0), Z_1, Z_2, \ldots, Z_k$.

4.1.3.3 Example: General Linear Model

General Linear Model defines CPDs as

$$E(X = x|Y_1 = y_1, Y_2 = y_2, \ldots, Y_k = y_k) = g^{-1}(w_0 + \sum_i w_i y_i)$$

- If X is a binary variable, we can define **sigmoid CPDs**:

$$P(X|Y_1 = y_1, Y_2 = y_2, \ldots, Y_k = y_k) = sigmoid(w_0 + \sum_i w_i y_i)$$

- For cases X and $Par(X)$ are both continuous variables, we can define **linear Gaussian CPDs**:

$$P(X|Y_1 = y_1, Y_2 = y_2, \ldots, Y_k = y_k) \sim \mathcal{N}(w_0 + \sum_i w_i y_i, \sigma^2)$$

- If some of $Par(X)$ are discrete variables, we can define **conditional linear Gaussian CPDs**:

$$P(X|A = a, Y_1 = y_1, Y_2 = y_2, \ldots, Y_k = y_k) \sim \mathcal{N}(w_{a0} + \sum_i w_{ai} y_i, \sigma_a^2)$$

We can also define CPDs using other link functions and other distributions similarly. Actually it turns out that almost all kinds of discriminative model can be used as CPD.

4.2 Local Structures of Markov Networks

In Markov Networks, we also have local structures models to reduce the amount of parameters and give us a compact representation. The most commonly used model is **Log-Linear Model**, where we re-define the unnormalized distribution as $\tilde{P} = exp(-\sum_i w_i f_i(\boldsymbol{D}_i))$, where $f_i(\cdot)$ is a feature over scope $\boldsymbol{D}_i$.

In log-linear models, having assigned $f(\cdot)$, we only need w to represent the distribution rather than specify all values for all ϕ.

NB. Different features can have the same scope.

There are 2 commonly used features: Indictor functions and metric features.

4.2.1 Indictor functions

We can use Indictor function $\mathcal{I}\{\cdot\}$ as features.

In the trivial case, for a factors $\phi(X_1, X_2, \ldots, X_k)$, we can define

$$f^{x_1,x_2,\ldots,x_k}_{X_1,X_2,\ldots,X_k} = \mathcal{I}\{X_1 = x_1, X_2 = x_2, \ldots, X_k = x_k\}$$

for all assignment $x_1, x_2, \ldots, x_k$ of $X_1, X_2, \ldots, X_k$. In this case, we convert tabular representation of a factor to log-linear representation.

In practice, we may want more non-trivial features to show the superiority of log-linear model. For example, in NLP, we have factors over word variables, which can take all values in the corpus. In this case, a tabular representation is unfeasible. It turns out that we can use some indictors functions like "whether a word is capitalized" to reduce the amount of parameters.

4.2.2 Metric MRFs

If X_i are continuous values, Indictors functions may be problematic sometimes. We can use distance functions as features.

X_i takes values in space $\mathbb{V}$. $\mu : \mathbb{V} \times \mathbb{V} \to \mathbb{R}$ is a **distance function** if,

- Reflexivity: $\forall v,\ \mu(v, v) = 0$;
- Symmetry: $\forall v_1, v_2,\ \mu(v_1, v_2) = \mu(v_2, v_1)$;
- Triangle Inequality: $\forall v_1, v_2, v_3,\ \mu(v_1, v_2) \leq \mu(v_1, v_3) + \mu(v_2, v_3)$.

In Metric MRFs, we define $f_{i,j}(X_i, X_j) = \mu(X_i, X_j)$ and all $w_{ij} > 0$.

5 Sharing Parameters in Template Models

In Graphical Models, we may encounter situations requiring parameters to be shared both between models and within models, which motivates us design Template Models where we can reuse parameters. These template models can be directed or undirected.

In template models, there is a list of classes $Q = [Q_1, Q_2, \ldots, Q_n]$, each with multiple objects (instances). Template models define a set of **Template variables** $A(U_1, U_2, \ldots, U_k)$. Each U_i is correlated to one class $Q[U_i]$, in which way $A(U_1, U_2, \ldots, U_k)$ is instantiated by objects of classes. **NB.** Different U_i and U_j can represent the same class.

Some commonly used template models are shown as follow.

5.1 Temporal Models: DBNs

In DBNs, the set $\{U_1, U_2, \ldots, U_k\}$ has only one element, correlated to the time T.

5.1.1 Assumptions

- Markov Assumption

$$X^{(t+1)} \perp X^{(0:t-1)} | X^{(t)}$$

- Time Invariance

$$\forall t, \;\; P(X^{(t+1)} | X^{(t)}) = P(X' | X)$$

5.1.2 Definitions

- 2-time-slice Bayesian Network (2TBN):
  - A transition model over $X_1, X_2, \ldots, X_k$. The nodes include $X'_1, X'_2, \ldots, X'_k$ and a subset of $X_1, X_2, \ldots, X_k$. Only nodes $X'_1, X'_2, \ldots, X'_k$ have parents and CPDs.
  - 2TBN defines a conditional distribution:

$$P(X' | X) = \prod_{i=1}^{k} P(X'_i | Par(X'_i))$$

- Dynamic Bayesian Network (DBN):

  A Bayesian network over $X_1, X_2, \ldots, X_k$ defined by 2 components:
  - a 2TBN $BN^{(\rightarrow)}$ over $X_1, X_2, \ldots, X_k$ as transition model;
  - a Bayesian Network $BN^{(0)}$ over $X_1^{(0)}, X_2^{(0)}, \ldots, X_k^{(0)}$ as initial state.

5.2 Plate Models

In Plate Models, the set $\{U_1, U_2, \ldots, U_k\}$ can have multiple elements.

5.2.1 Definition

For each template variable $A(U_1, U_2, \dots, U_k)$, a Plate Model defines:

- Template variables $B_1(\boldsymbol{U}_1), B_2(\boldsymbol{U}_2), \dots, B_m(\boldsymbol{U}_m)$ as m parents of A. Each $\boldsymbol{U}_i$ is a subset of $\{U_1, U_2, \dots, U_k\}$;
- A CPD $P(A|B_1, B_2, \dots, B_m)$.

5.2.2 Example

For each student and each class, define template variable $G(S, C)$ representing Grades. Here the set $\{U_1, U_2, \dots, U_k\} = \{S, C\}$. $G(S, C)$ has 2 parents $I(S)$ and $D(C)$, representing Intelligence of a student and Difficulty of a class.

For students $s_1, s_2, \dots, s_m$, $I(S)$ is instantiated m times; for classes $c_1, c_2, \dots, c_n$, $D(C)$ is instantiated n times. For each pair $I(s_i), D(c_j)$, there exists a $G(s_i, c_j)$ (the value of these variables can be unseen because only a student taking this course G is observed). All $G(s_i, c_j)$ share the same CPD $P(G|S, C)$.

In the ground network of this example, there are $m + n + mn$ nodes and $m + n$ edges. By template model, we significantly reduce the amount of nodes and edges.

5.3 Shared Features in Log-Linear Models

In Markov Nets, parameters sharing is achieved by shared features and weights in log-linear models.

Concretely, for each feature f_j, we will reuse it multiple times over scopes $\boldsymbol{D}_1, \boldsymbol{D}_2, \dots, \boldsymbol{D}_k$. For each $\boldsymbol{D}_j \in \{\boldsymbol{D}_1, \boldsymbol{D}_2, \dots, \boldsymbol{D}_k\}$ we have a term $w_j f_j(\boldsymbol{D}_j)$ in the energy function.