Chapter 7. Latent Variable Models

7.1 Factor Graphs

Directed vs. Undirected Graphs

- Both graphical models
  - Specify a factorization (how to express the joint distribution)
  - Define a set of conditional independence properties

\[ P(A, B, C, D, E) = P(A)P(B)P(C | A, B)P(D | A)P(E | C) \]

Fig. 7-1. Parent-child local conditional distribution
Relation of Directed and Indirected Graphs

- Converting a directed graph to an undirected graph
  - **Case 1: straight line**
    
    \[
p(x) = p(x_1)p(x_2|x_1)p(x_3|x_2) \cdots p(x_N|x_{N-1}).
    \]

  
  \[
p(x) = \frac{1}{Z} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \cdots \psi_{N-1,N}(x_{N-1}, x_N).
    \]

  
  \[
  \begin{align*}
  \psi_{1,2}(x_1, x_2) &= p(x_1)p(x_2|x_1) \\
  \psi_{2,3}(x_2, x_3) &= p(x_3|x_2) \quad \vdots \\
  \psi_{N-1,N}(x_{N-1}, x_N) &= p(x_N|x_{N-1})
  \end{align*}
  \]

  - In this case, the partition function
    
    \[
    Z = 1
    \]

  - **Case 2: general case.** Moralization = ‘marrying the parents’
    
    - Add additional undirected links between all pairs of parents
    - Drop the arrows

    \[
p(x) = p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3).
    \]

  - Results in the **moral graph**
    
    - Fully connected \(\rightarrow\) no conditional independence properties, in contrast to the original directed graph

  - We should add the fewest extra links to retain the maximum number of independence properties
Factor Graphs

- A factor graph is a bipartite graph representing a joint distribution in the form of a product of factors.
  - Factors in directed/undirected graphs
  - Introducing additional nodes for the factors themselves
  - Explicit decomposition /factorization

\[ p(x) = \prod_s f_s(x_s) \]

\[ p(x) = p(x_1)p(x_2|x_1)p(x_3|x_2) \ldots p(x_N|x_{N-1}) \]

\[ p(x) = \frac{1}{Z} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \ldots \psi_{N-1,N}(x_{N-1}, x_N) \]

(Factor graphs are bipartite)

\[ p(x) = f_d(x_1, x_2)f_b(x_1, x_2)f_c(x_2, x_3)f_d(x_3) \]

- Definition. Given a factorization of a function \( g(X_1, X_2, \ldots, X_n) \)

\[ g(X_1, X_2, \ldots, X_n) = \prod_{j=1}^{m} f_j(S_j), \]

where \( S_j \subseteq \{X_1, X_2, \ldots, X_n\} \) the corresponding factor graph \( G = (X, F, E) \) consists of variable vertices \( X = \{X_1, X_2, \ldots, X_n\} \), factor vertices \( F = \{f_1, f_2, \ldots, f_m\} \), and edges \( E \). The edges depend on the factorization as follows: there is an undirected edge between factor vertex \( f_i \) and variable vertex \( X_i \) when \( X_i \subseteq S_j \). The function is assumed to be real-valued, i.e. \( g(X_1, X_2, \ldots, X_n) \in \mathbb{R} \).

- Factor graphs can be combined with message passing algorithms to efficiently compute certain characteristics of the function \( g(X_1, X_2, \ldots, X_n) \), such as the marginals.

- Example. Consider a function that factorizes as follows:

\[ g(X_1, X_2, X_3) = f_1(X_1)f_2(X_1, X_2)f_3(X_1, X_2)f_4(X_2, X_3), \]

with a corresponding factor graph
This factor graph has a cycle. If we merge \( f_2(X_1, X_2) \) and \( f_3(X_1, X_2) \) into a single factor, the resulting factor graph will be a tree. This is an important distinction, as message passing algorithms are usually exact for trees, but only approximate for graphs with cycles.

**Inferences on factor graphs.**

- **Sum-product algorithm**: evaluating local marginals over nodes or subsets of nodes

\[
p(x) = \sum_{x} p(x) = \prod_{s \in \text{Net}(x)} F_s(x, X_s)
\]

\[
p(x) = \sum_{x} p(x) = \sum_{x} \left[ \prod_{s \in \text{Net}(x)} F_s(x, X_s) \right] = \prod_{s \in \text{Net}(x)} \left[ \sum_{X_s} F_s(x, X_s) \right] = \prod_{s \in \text{Net}(x)} \mu_{f_s \rightarrow x}(x)
\]

The messages \( \mu_{f_s \rightarrow x}(x) \) from the factor node \( f_s \) to the variable node \( x \) are computed in the vertices and passed along the edges.

- **Max-sum algorithm**: finding the most probable state

The Hammersley–Clifford theorem shows that other probabilistic models such as Markov networks and Bayesian networks can be represented as factor graphs.

Factor graph representation is frequently used when performing inference over such networks using belief propagation.

On the other hand, Bayesian networks are more naturally suited for generative models, as they can directly represent the causalities of the model.

**Properties of Factor Graphs**
Converting directed and undirected graphs into factor graphs

- **undirected graph $\rightarrow$ factor graph**

\[
f(x_1, x_2, x_3) = \psi(x_1, x_2, x_3)
\]

\[
f_a(x_1, x_2, x_3) f_b(x_1, x_2) = \psi(x_1, x_2, x_3)
\]

Note: For a given fully connected undirected graph, two (or more) different factor graphs are possible. Factor graphs are more specific than the undirected graphs.

- **directed graph $\rightarrow$ factor graph**

\[
f(x_1, x_2, x_3) = p(x_1)p(x_2)p(x_3 | x_1, x_2)
\]

\[
f_a(x_1) = \tilde{p}(x_1), \quad f_b(x_2) = \tilde{p}(x_2)
\]

\[
f_c(x_1, x_2, x_3) = p(x_3 | x_1, x_2)
\]

For the same directed graph, two or more factor graphs are possible. There can be multiple factor graphs all of which correspond to the same undirected/directed graph.

- Converting a directed/undirected tree to a factor graph
  - The result is again a tree (no loops, one and only one path connecting any two nodes)

- Converting a directed polytree to a factor graph
  - The results in a tree.
  - Cf. Converting a directed polytree into an undirected graph results in loops due to the moralization step.
• Local cycles in a directed graph can be removed on conversion to a factor graph

• Factor graphs are more specific about the precise form of the factorization

For a fully connected undirected graph, two (or more) factor graphs are possible.

• Directed and undirected graphs can express different conditional independence properties
7.2 Probabilistic Latent Semantic Analysis

Latent Variable Models

- Latent variables
  - Variables that are not directly observed but are rather inferred from other variables that are observed and directly measured
- Latent variable models
  - Explain the statistical properties of the observed variables in terms of the latent variables
- General formulation

PLSA

\[
R(w|d) = \sum P(w|z)R(z|d)
\]

d: data, z: topic, w: visual word
7.3 Gaussian Mixture Models

- **Graphical representation of a mixture model**
  A binary random variable $z$ having a 1-of-$K$ representation

\[
Z 
\]

- **Gaussian mixture distribution** can be written as a linear superposition of Gaussians

\[
p(x) = \sum_{k=1}^{K} \pi_k N(x | \mu_k, \Sigma_k)
\]

An equivalent formulation of the Gaussian mixture involving an explicit latent variable

\[
p(z) = \prod_{k=1}^{K} \pi_k^{z_k}
\]

\[
p(x \mid z = 1) = N(x \mid \mu_z, \Sigma_z)
\]

\[
p(x \mid z) = \prod_{k=1}^{K} N(x \mid \mu_k, \Sigma_k)^{z_k}
\]
\[ p(x) = \sum_z p(x, z) = \sum_z p(z)p(x | z) = \sum_z \left[ \prod_{k=1}^K \pi_k \prod_{i=1}^{K} N(x | \mu_k, \Sigma_k) \right] \]
\[ = \sum_{k=1}^K \pi_k N(x | \mu_k, \Sigma_k) \]
\[ p(z_k = 1) = \pi_k \]
\[ \sum_{k=1}^K \pi_k = 1 \]

- The marginal distribution of \( x \) is a Gaussian mixture of the form (*)
  - for every observed data point \( x \), there is a corresponding latent variable \( z \)

\[ p(x) = \sum_z p(x, z) \]
\[ \gamma(z_k) \equiv p(z_k = 1 | x) \]
\[ = \frac{p(z_k = 1) p(x | z_k = 1)}{\sum_{j=1}^K p(z_j = 1) p(x | z_j = 1)} \]
\[ = \frac{\pi_k N(x | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x | \mu_j, \Sigma_j)} \]

- \( \gamma(z_k) \) can also be viewed as the responsibility that component \( k \) takes for explaining the observation \( x \)

- Generating random samples distributed according to the Gaussian mixture model
  - Generating a value for \( z \), which denoted as \( z \) from the marginal distribution \( p(z) \) and then generate a value for \( x \) from the conditional distribution \( p(x | z) \)

a. The three states of \( z \), corresponding to the three components of the mixture, are depicted in red, green, blue
b. The corresponding samples from the marginal distribution \( p(x) \)

c. The same samples in which the colors represent the value of the responsibilities \( \gamma(z_{nk}) \) associated with data point
   ✗ Illustrating the responsibilities by evaluating the posterior probability for each component in the mixture distribution which this data set was generated
   ♦ Distribution

● **Graphical representation of a Gaussian mixture model** for a set of \( N \) i.i.d. data points \( \{x_n\} \), with corresponding latent points \( \{z_n\} \)
   - Data set: \( X \) (\( N \times D \) matrix) with \( n \)-th row \( x_n^T \)
   - Latent variables: \( Z \) (\( N \times K \) matrix) with rows \( z_n^T \)

![Graphical representation of a Gaussian mixture model](image)

- The log of the likelihood function
  \[
  \ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^{N} \ln \left( \sum_{k=1}^{K} \pi_k N(x_n | \mu_k, \Sigma_k) \right)
  \]

### 7.4 Learning Gaussian Mixtures by EM

- The Gaussian mixture models can be learned by the expectation-maximization (EM) algorithm.
  Repeat
  ♦ Expectation step: calculate posterior or responsibilities using the current parameters
  ♦ Maximization step: re-estimate the parameters based on the responsibilities

- Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters
  1. Initialize the means \( \mu_k \), covariance \( \Sigma_k \) and mixing coefficients \( \pi_k \)
  2. **E-step**: evaluate the posterior probabilities or responsibilities using the current value for the parameters
     \[
     \gamma(z_{nk}) = \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n | \mu_j, \Sigma_j)}
     \]
  3. **M-step**: re-estimate the means, covariances, and mixing coefficients using the result of E-step.
\[ \mu^{\text{new}}_k = \frac{1}{N_k} \sum_{n \in \mathcal{D}} \gamma(z_{nk}) x_n \]

\[ \sum_{k=1}^{K} \frac{1}{N_k} \sum_{n \in \mathcal{D}} \gamma(z_{nk}) (x_n - \mu^{\text{old}}_k) (x_n - \mu^{\text{new}}_k)^	op \]

\[ \pi^{\text{new}}_k = \frac{N_k}{N} \]

\[ N_k = \sum_{n \in \mathcal{D}} \gamma(z_{nk}) \]

4. Evaluate the log likelihood

\[ \ln p(X | \mu, \Sigma, \pi) = \sum_{n = 1}^{N} \ln \left( \sum_{k = 1}^{K} \pi_k N(x_n | \mu_k, \Sigma_k) \right) \]

If converged, terminate; otherwise, go to Step 2.

**The General EM Algorithm**

- In maximizing the log likelihood function \( \ln p(X | \theta) \), the summation prevents the logarithm from acting directly on the joint distribution.
- Instead, the log likelihood function for the complete data set \( \{X, Z\} \) is straightforward.
- In practice since we are not given the complete data set, we consider instead its expected value \( Q \) under the posterior distribution \( p(Z|X, \Theta) \) of the latent variable.
- **General EM Algorithm**
  1. Choose an initial setting for the parameters \( \theta^{\text{old}} \)
  2. **E step** Evaluate \( p(X | Z, \Theta^{\text{old}}) \)
  3. **M step** Evaluate \( \Theta^{\text{new}} \) given by

![Graphs](image-url)
\[ \Theta^{\text{new}} = \arg \max_{\Theta} Q(\Theta, \Theta^{\text{old}}) \]

\[ Q(\Theta, \Theta^{\text{old}}) = \sum_Z p(Z | X, \Theta^{\text{old}}) \ln p(X, Z | \Theta) \]

4. If the covariance criterion is not satisfied, then let \( \Theta^{\text{old}} \leftarrow \Theta^{\text{new}} \) and return to Step 2.

- The EM algorithm can also be used for finding MAP (maximum a posteriori) using the modified M-step

\[ Q(\theta, \Theta^{\text{old}}) + \ln p(\theta) \]