Learning from Sensor Data: Set II

Behnaam Aazhang
J.S. Abercombie Professor
Electrical and Computer Engineering
Rice University
6. Data Representation

- The approach for learning from data
  - Probabilistic modeling and algebraic manipulation
- Diagrammatic representation is often extremely useful
  - Probabilistic graphical modeling
    - Visualize the structure
    - Infer dependence based on inspection of the graph
    - Simplify complex computations
• Examples of graphical modeling in engineering problems
  
  • Circuit diagrams
  
  • Signal flow diagrams
  
  • Trellis diagrams
  
  • Block diagrams
• A graph can be viewed as the simplest way to represent a complex system where
  
  • Vertices are simplest units of the system
  
  • Edges represent their mutual interactions
• Elements

  • Nodes or vertices
    • A random variable (data) or a group of random variables

  • Links or edges
    • Probabilistic relationships between the variables
• Examples of graphs

\[ X_1 \xrightarrow{C_{X_1,X_2}} X_2 \]

\[ X_1 \xrightarrow{I(X_1;X_2)} X_2 \]

\[ X_1 \xrightarrow{I(X_1 \rightarrow X_2)} X_2 \]
• A typical graph representing data
• RAH2 node is influencing several nodes
• Another example
• does neuron 3 excite neuron 8?
• does neuron 3 excite neuron 8?
• Did neuron 3 causally influence firing of neuron 8?

<table>
<thead>
<tr>
<th>Neuron 3</th>
<th>...00011110000001001100001100000001100001...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Neuron 8</td>
<td>...0000011110000001011110000110000001111...</td>
</tr>
</tbody>
</table>
\( I(X_3 \rightarrow X_8) \)
• Learning from graphs
  
  • Identifying important features of data from graphs
  
  • A graph $G = (V,E)$ with $V$ as the set of vertices and $E$ as the set of edges
  
  • A graph is simple if it has no parallel edges and no loops
  
  • Adjacent edges and adjacent vertices are defined as the terms suggest
  
  • The degree of vertex $v$ is $d(v)$ as the number of edges with $v$ as the end
  
  • A pendant vertex is a vertex with degree 1.
• A graph is called regular if all vertices have the same degree

• In an undirected graph each edge is an unordered pair of vertices \((u, v)\)

• In a directed graph each edge is an ordered pair of vertices \((u, v)\)
• In degree of vertex $v$ in a directed graph is the number of edges with $v$ as the end

• Out degree of vertex $v$ in a directed graph is the number of edges with $v$ as the tail

• An isolated vertex is one with degree 0. In degree and out degree 0 in a directed graph.
• For undirected graph we define the following concepts and properties

  • Some definitions can be extended to directed graphs

  • Minimum degree of a graph $\delta(G)$

  • Maximum degree of a graph $\Delta(G)$
• It can be shown that for a graph $G = (V, E)$ with $n$ vertices and $m$ edges then

$$\sum_{i=1}^{n} d(v_i) = 2m$$

• A graph $G = (V, E)$ is a subgraph of graph $H = (W, F)$ if $V$ is a subset of $W$ and every edge in $E$ is also an edge in $F$.

• A complete graph is a simple graph with all the possible edges

• A complete subgraph of graph $G$ is called a clique.
• The density of a graph $G = (V, E)$ is defined as

$$\rho(G) = \frac{m}{\binom{n}{2}} \text{ for } n \geq 2$$

where $\binom{n}{2} = \frac{n!}{2!(n-2)!}$

• The density of a complete graph is 1

• The adjacency matrix of graph $G$ is a $n \times n$ matrix

$$A_G = \begin{pmatrix}
    a_{11} & \cdots & a_{1n} \\
    \vdots & \ddots & \vdots \\
    a_{n1} & \cdots & a_{nn}
\end{pmatrix}$$

where $a_{uv} = \begin{cases} 
    1 & \text{if there is an edge between } u \text{ and } v \\
    0 & \text{otherwise}
\end{cases}$
• The spectrum of graph $G = (V, E)$ is the set of eigenvalues of the adjacency matrix and their eigenvectors.

• The Laplacian matrix of graph $G = (V, E)$ is defined as

\[
L = D - A_G
\]

• where the diagonal degree matrix, $D$ is defined as

\[
D = \begin{pmatrix}
d(v_1) & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & d(v_n)
\end{pmatrix}
\]
The normalized Laplacian is

\[ \mathcal{L} = D^{-1/2}LD^{-1/2} = I - D^{-1/2}A_GD^{-1/2} \]

The Laplacian matrix carries some of the key properties of a graph.

Since the adjacency and Laplacian matrices are symmetric their eigenvalues are real.

The eigenvalues of the normalized Laplacian are in \([0, 2]\).

This fact makes it convenient to compare the spectral properties of two graphs.
Two graphs are isomorphic if any two vertices of one are adjacent if and only if the equivalent vertices in the other graph are also adjacent.

\[ f(a) = 1 \]
\[ f(b) = 6 \]
\[ f(c) = 8 \]
\[ f(d) = 3 \]
\[ f(g) = 5 \]
\[ f(h) = 2 \]
\[ f(i) = 4 \]
\[ f(j) = 7 \]
Graphs that have the same spectrum are referred to as *cospectral* (or *isospectral*).

If two graphs have the same eigenvalues but different eigenvectors they are referred to as *weakly cospectral*.

Although adjacency matrix of a graph depends on the labeling of the vertices, the spectrum of a graph is independent of labeling.

Isomorphic graphs are cospectral but not all cospectral graphs are isomorphic.
• The complement of graph $G = (V, E)$ is $\bar{G} = (V, \bar{E})$

• where the edges in complement graph are the ones not in $E$

• Common binary and linear operations can be defined for graphs

• Complement, union, intersection, ring sum, …

• examples of commutative and associative operations.
• A community is a group of vertices that “belong together” according to some criterion that could be measured

• An example, a group of vertices where the density of edges between the vertices in the group is higher than the average edge density in the graph

• In some literature a community is also referred to as a module or a cluster.
• Examples
In summary, the eigenvalues of the adjacency matrix are denoted as matrices. The set of all n eigenvalues of the adjacency and laplacian matrices.

Because both the adjacency and laplacian matrices are symmetric, we are guaranteed to only get real eigenvalues. Similarly, the set of eigenvalues of the laplacian matrix is referred to as the laplacian spectrum.

Example 6.1.

\[
D = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix} \quad A = \begin{bmatrix}
0 & 1 & 0 & 0 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix} \quad L = \begin{bmatrix}
1 & -1 & 0 & 0 \\
-1 & 2 & -1 & 0 \\
0 & -1 & 1 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

- The eigenvalues of the adjacency matrix are \((\sqrt{2}, 0, 0, -\sqrt{2})\).
- The eigenvalues of the Laplacian matrix are \((3, 1, 1, 0)\).
- One isolated vertex and two pendant vertices.
Example 6.1

Alternative adjacency and Laplacian matrices are

\[ A = \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 \\
0 & 0 & 1 & 0
\end{bmatrix} \quad \text{and} \quad
L = \begin{bmatrix}
1 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 \\
-1 & 0 & 2 & -1 \\
0 & 0 & -1 & 1
\end{bmatrix} \]

Eigenvalues of \( A \) and \( L \) are

\( (\sqrt{2}, 0, 0, -\sqrt{2}) \) \quad \text{and} \quad \( (3, 1, 1, 0) \)
• Example 6.2

• The bipartite graph
• Example 6.2

• The bipartite graph

• The adjacency metric

  \[
  A_G = \begin{pmatrix}
  0 & 0 & 0 & 1 & 1 & 1 \\
  0 & 0 & 0 & 1 & 1 & 0 \\
  0 & 0 & 0 & 0 & 1 & 0 \\
  1 & 1 & 0 & 0 & 0 & 0 \\
  1 & 1 & 1 & 0 & 0 & 0 \\
  1 & 0 & 0 & 0 & 0 & 0
  \end{pmatrix}
  \]

• The Laplacian

  \[
  L = \begin{pmatrix}
  3 & 0 & 0 & -1 & -1 & -1 \\
  0 & 2 & 0 & -1 & -1 & 0 \\
  0 & 0 & 1 & 0 & -1 & 0 \\
  -1 & -1 & 0 & 2 & 0 & 0 \\
  -1 & -1 & -1 & 0 & 3 & 0 \\
  -1 & 0 & 0 & 0 & 0 & 1
  \end{pmatrix}
  \]
• Example 6.3

• A complete graph

• The diagonal degree matrix is \( D = 4xI \) where \( I \) is a 5x5 identity matrix

• The Laplacian is

\[
L = \begin{pmatrix}
4 & -1 & -1 & -1 & -1 \\
-1 & 4 & -1 & -1 & -1 \\
-1 & -1 & 4 & -1 & -1 \\
-1 & -1 & -1 & 4 & -1 \\
-1 & -1 & -1 & -1 & 4 \\
\end{pmatrix}
\]
• Example 6.4

• A regular graph with $D = 2xI$ where $I$ is a $4x4$ identity matrix
• Graphs can be used to
  
  • efficiently compute different functions of data
  
  • represent data
    
    • identify which vertices-data are significant
      
      • reduce dimensionality and only focus on important vertices-data
Defining a suitable centrality metric (or index of significance) is important.

Centrality

- Closeness
- Betweenness
- Degree
- Eigenvector
- Katz
- PageRank
• Degree centrality

• The degree vector $d = Ae$ where $A$ is the adjacency matrix of the graph and $e$ is the all 1 vector.
• Degree centrality

• For directed graphs
  • In degree centrality
  • Out degree centrality
• Degree centrality

• For directed graphs
  • In degree centrality
  • Out degree centrality
• Degree centrality

• For directed graphs
  • In degree centrality
  • Out degree centrality
• Eigenvector centrality

• Identifying important vertices in a large network is a critical problem with numerous applications.

• A vertex is important if its adjacent vertices are important.
• Eigenvector centrality

• Identifying important vertices in a large network is critical problem with numerous applications.

• A vertex is important if its adjacent vertices are important

which one is more important?
• Eigenvector centrality

• Identifying important vertices in a large network is a critical problem with numerous applications.

• A vertex is important if its adjacent vertices are important

• Centrality is proportional to the centrality of adjacent vertices

\[ E_{v_i} \propto \sum_{j \in \mathcal{N}_i} E_{v_j} = \sum_j a_{ij} E_{v_j} \]

• A system of equations with \( n \) unknowns
Eigenvector centrality

\[ \lambda E_v = A_G E_v \]

The eigenvector of the adjacency matrix

\[
E_{v_i} \propto \sum_{j \in N_i} E_{v_j} = \sum_j a_{ij} E_{v_j}
\]
• Eigenvector centrality

\[ \lambda E_v = A_G E_v \]

• The eigenvector of the adjacency matrix

\[ A_G = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \]
• Intuition starts with degree centrality

\[ X = \begin{pmatrix} 2 \\ 3 \\ 2 \\ 3 \\ 2 \\ 2 \\ 2 \end{pmatrix} \]

• Degree vector

\[ A_G X = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 2 \\ 3 \\ 2 \\ 2 \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \\ 6 \\ 6 \\ 5 \\ 5 \end{pmatrix} \]
• The process of adjusting the significance of a node based on the significance of neighbors can continue till the adjustment settles

\[
\begin{pmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\begin{pmatrix}
5 \\
6 \\
6 \\
6 \\
5 \\
5 \\
5
\end{pmatrix}
=
\begin{pmatrix}
11 \\
16 \\
12 \\
16 \\
11 \\
11 \\
11
\end{pmatrix}
\]

• Leading to an eigenvector of the matrix \( A \)

\[
\lambda E_v = A G E_v
\]
• The set of eigenvalues are -1.81 -1.00 -1.00 -1.00 0.47 2.00 2.34

• The eigenvector corresponding to the largest eigenvalue will have non-negative elements since the adjacency matrix has non-negative elements (from the Perron-Frobenius theorem)

• That is also the best lower rank approximation of the matrix $A$

• Eigenvalue 2.34 and the corresponding eigenvector

$$E_v = \begin{pmatrix}
0.33 \\
0.45 \\
0.38 \\
0.45 \\
0.33 \\
0.33 \\
0.33 \\
\end{pmatrix}$$
$E_v = \begin{pmatrix} 0.33 \\ 0.45 \\ 0.38 \\ 0.45 \\ 0.33 \\ 0.33 \\ 0.33 \end{pmatrix}$
• Eigenvalue 2.34 and the corresponding eigenvector

• The average degree of vertices is 2.28

• It can be shown that $2.28 < 2.34 < 3$, that is, the value of the largest eigenvalue of $A$ is between the average degree and the maximum degree of the vertices

• The consequence of eigenvector centrality is to only focus on critical vertices and reduce the dimensionality of the problem.
• Graphs to better understand dynamics of networks
Aphasia

- An impairment of language, affecting the production or comprehension of speech and ...

- Often due to injury to the brain
  - Most commonly from a stroke ...
The language system

- Unique to human

- Impact of aphasia
  - How we process visual information
  - How we recall
  - How we articulate
  - How we speak
Our understanding today

- Inferences based on responses in high gamma power
  - >60 Hz
Our understanding today

• Visual cortex
Our understanding today

- Left temporal cortex (processing of semantics)
Our understanding today

- Broca region (speech production)
Our understanding today

- Motor cortex
Our curiosity

- Inference based on responses in high gamma power
  - High gamma >60 Hz

- What are the underlying mechanisms of our language region?

- Are there causal relations among recorded signals?

- Are there coupling (coherency) among recordings in different frequencies?

- How are the network dynamics as language is produced?
The experiment

- 0 sec image shown
- 2 sec fixation cross shown
- 5 sec

Trial 1

- stimulus onset

Trial 2

- start of articulation
Recordings

- Electro-cortico-graphy (ecog)

- Learn language production

  - 7 epileptic patients
Recordings

• Local field potentials LFP (time series)

• Spatio-temporal analysis
  • 100-300 time series
  • 200-500 trials
Graphs

- Spatial relationships
- Undirected
  - Coherency of time series
  - Coherency in high gamma
- Directed
  - Causal relation
  - Information flow
Back to language production

• Electrodes as vertices

• Edges
  • Undirected: coupling at different frequencies
  • Directed: causal relation

• Graph dynamics as language is produced
Graphical analysis-undirected

- edges
  - undirected: coupling at high gamma
Graphical analysis-undirected

- Edges
  - Undirected: coupling at high gamma

- Graph density

\[ \rho(G) = \frac{1}{2} \sum_{i=1}^{n} d(v_i) \]

- The degree of vertex \( v \) is \( d(v) \) as the number of edges of \( v \)
Graphical analysis-undirected

- Edges
  - Undirected: coupling at high gamma
Multiscale graphical analysis-directed

• **Coarse scale:** graph density

• **Intermediate scale:** louvain community

• **Fine scale:** in degree and out degree
Multiscale graphical analysis-directed

- **Coarse scale:** graph density \( \rho(G) = \frac{1}{2} \sum_{i=1}^{n} d(v_i) \)

- Increase in graph density prior to articulation

![Graph Density vs Time](image)
Multiscale graphical analysis-directed

- **Coarse scale**: graph density \( \rho(G) = \frac{1/2 \sum_{i=1}^{n} d(v_i)}{\binom{n}{2}} \)

- Increase in graph density prior to articulation

![Graph density and articulation time](image)
Multiscale graphical analysis-directed

- **Coarse scale:** graph density $\rho(G)$
- **Intermediate scale:** louvain clusters
multiscale graphical analysis-directed

- **Coarse scale:** graph density \( \rho(G) \)

- **Intermediate scale:** louvain clusters
  - identifying significant clusters
  - a practical algorithm to find “best” clustering
  - density of intra-cluster edges to inter cluster edges
Multiscale graphical analysis-directed

- **Coarse scale:** graph density \( \rho(G) \)

- **Intermediate scale:** louvain clusters
Multiscale graphical analysis-directed

- **Coarse scale**: graph density $\rho(G)$

- **Intermediate scale**: louvain clusters
Multiscale graphical analysis-directed

- **Coarse scale**: graph density \( \rho(G) \)

- **Intermediate scale**: louvain community

- **Fine scale**: in degree and out degree

articulation time
Multiscale graphical analysis-directed

- **Coarse scale**: graph density $\rho(G)$
- **Intermediate scale**: louvain community
- **Fine scale**: in degree and out degree

articulation time
take home message

• building a framework to understand language production

• increased functional and effective connectivity
  
  • onset of stimulus

  • articulation

• heavier clusters at articulation
• A graph can also capture the way joint probability distributions of all variables can be decomposed and then computed

• Different graphical models for inference

  • **Bayesian networks**

  • Markov random fields

  • Factor graph
• Example 6.5

• A common motivating example

• Difficulty of an exam, intelligence of the student, grade in a class, student’s SAT exam results, professor’s letter of recommendation

  • Denoted as D, i, g, S, l, respectively

  • How is the dependency structure of all these variables?
• Example 6.5

• How is the dependency structure of all these variables?

• Intuitively
Example 6.5

- The joint probability

\[ p_{D,g,i,S,l} = p_D p_i p_S|i p_g|D,i p_l|g \]

- Let's find out how
Example 6.6

Some basic concepts for two random variables, that is, two data sets

\[
F_{X_1,X_2}(a, b) = Pr\{X_1 \leq a, X_2 \leq b\}
\]

\[
F_{X_1,X_2}(a, b) = Pr\{X_1 \leq a, X_2 \leq b\}
= Pr\{A = \{w \in \Omega | X_1(w) \leq a\} \cap B = \{w \in \Omega | X_2(w) \leq b\}\}
= Pr\{B|A\} Pr\{A\}
\]
• Example 6.6

• Some basic concepts for two random variables, that is, two data sets

\[
F_{X_1, X_2}(a, b) = \Pr \{ X_1 \leq a, X_2 \leq b \}
\]

\[
F_{X_1, X_2}(a, b) = F_{X_2|X_1}(b|a)F_{X_1}(a)
= \Pr \{ X_1 \leq a, X_2 \leq b \} = \Pr \{ X_2 \leq b|X_1 \leq a \} \Pr \{ X_1 \leq a \}
\]

\[
F_{X_2}(b) = \lim_{a \to +\infty} F_{X_1, X_2}(a, b) = \lim_{a \to +\infty} \Pr \{ X_1 \leq a, X_2 \leq b \}
\]
• If the data sets are discrete valued then probability mass functions (pmf’s) are defined and we will have similar implications

\[ p_{X_1, X_2}(a, b) = Pr\{X_1 = a, X_2 = b\} \]

\[ p_{X_1, X_2}(a, b) = p_{X_2|X_1}(b|a)p_{X_1}(a) = Pr\{X_1 = a, X_2 = b\} = Pr\{X_2 = b|X_1 = a\}Pr\{X_1 = a\} \]

\[ p_{X_2}(b) = \sum_a p_{X_1, X_2}(a, b) = \sum_a Pr\{X_1 = a, X_2 = b\} \]
• If the data sets were continuous valued then probability density functions (pdf’s) will be defined and we will have similar implications

\[
F_{X_1, X_2}(a, b) = Pr\{X_1 \leq a, X_2 \leq b\} = \int_{\infty}^{b} \int_{\infty}^{a} f_{X_1, X_2}(x_1, x_2) \, dx_1 \, dx_2
\]

\[
f_{X_1, X_2}(x_1, x_2) = f_{X_2|X_1}(x_2|x_1) f_{X_1}(x_1)
\]

\[
f_{X_2}(x_2) = \int_{-\infty}^{+\infty} f_{X_1, X_2}(x_1, x_2) \, dx_1 = \int_{-\infty}^{+\infty} f_{X_2|X_1}(x_2|x_1) f_{X_1}(x_1) \, dx_1
\]
• Efficient graphical models

• Compute joint distribution of data—global function of multiple variables

\[ F_X = F_{X_1, X_2, X_3, X_4, X_5} \]

• Marginalize

\[ F_{X_3}(x_3) = \lim_{x_1 \to +\infty} \lim_{x_2 \to +\infty} \lim_{x_4 \to +\infty} \lim_{x_5 \to +\infty} F_{X_1, X_2, X_3, X_4, X_5} \]
• Efficient graphical models

• Compute joint distribution of data—global function of multiple variables

\[ f_\mathbf{x} = f_{X_1, X_2, X_3, X_4, X_5} \]

• Marginalize

\[ f_{X_3}(x_3) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f_{X_1, X_2, X_3, X_4, X_5}(x_1, x_2, x_3, x_4, x_5) \, dx_1 \, dx_2 \, dx_4 \, dx_5 \]
• Efficient graphical models

• Compute joint distribution of data—global function of multiple variables

\[ p_x(x) = p_{X_1, X_2, X_3, X_4, X_5} \]

• Marginalize

\[ p_{X_3}(x_3) = \sum_{x_1} \sum_{x_2} \sum_{x_4} \sum_{x_5} p_{X_1, X_2, X_3, X_4, X_5}(x_1, x_2, x_3, x_4, x_5) \]
• Critical for inference problems

• The global function factorizing into local functions

\[ F_{X_1, X_2}(a, b) = F_{X_2|X_1}(b|a)F_{X_1}(a) \]

\[ f_{X_1, X_2}(x_1, x_2) = f_{X_2|X_1}(x_2|x_1)f_{X_1}(x_1) \]

\[ p_{X_1, X_2}(a, b) = p_{X_2|X_1}(b|a)p_{X_1}(a) \]

• Graphical models are powerful tools in representing these expressions
• Bayesian network—directed graphs

• Consider three variables and their joint distribution

\[ F_{X_1, X_2, X_3}(x_1, x_2, x_3) = F_{X_3|X_1, X_2}(x_3|x_1, x_2)F_{X_1, X_2}(x_1, x_2) \]
\[ = F_{X_3|X_1, X_2}(x_3|x_1, x_2)F_{X_2|X_1}(x_2|x_1)F_{X_1}(x_1) \]
- Bayesian network—directed graphs

- Consider three variables and their joint distribution

\[
F_{X_1, X_2, X_3}(x_1, x_2, x_3) = F_{X_3|X_1, X_2}(x_3|x_1, x_2)F_{X_1, X_2}(x_1, x_2) \\
= F_{X_3|X_1, X_2}(x_3|x_1, x_2)F_{X_2|X_1}(x_2|x_1)F_{X_1}(x_1)
\]

\[
X_1 \quad X_2 \quad X_3
\]
• Bayesian network—directed graphs

• Consider three variables and their joint distribution

\[ F_{X_1, X_2, X_3}(x_1, x_2, x_3) = F_{X_3|X_1, X_2}(x_3|x_1, x_2)F_{X_1, X_2}(x_1, x_2) \]
\[ = F_{X_3|X_1, X_2}(x_3|x_1, x_2)F_{X_2|X_1}(x_2|x_1)F_{X_1}(x_1) \]
• Bayesian network—directed graphs

• Consider three variables and their joint distribution

\[
F_{X_1, X_2, X_3}(x_1, x_2, x_3) = F_{X_3|X_1, X_2}(x_3|x_1, x_2)F_{X_1, X_2}(x_1, x_2) \\
= F_{X_3|X_1, X_2}(x_3|x_1, x_2)F_{X_2|X_1}(x_2|x_1)F_{X_1}(x_1)
\]
• Bayesian network—directed graphs

• Consider three variables and their joint distribution

\[
F_{X_1, X_2, X_3}(x_1, x_2, x_3) = F_{X_3|X_1, X_2}(x_3|x_1, x_2)F_{X_1, X_2}(x_1, x_2) \\
= F_{X_3|X_1, X_2}(x_3|x_1, x_2)F_{X_2|X_1}(x_2|x_1)F_{X_1}(x_1)
\]
• If $X_2$ and $X_1$ are independent

$$F_{X_1, X_2, X_3}(x_1, x_2, x_3) = F_{X_3|X_1, X_2}(x_3|x_1, x_2)F_{X_2}(x_2)F_{X_1}(x_1)$$
• If $X_2$ and $X_1$ are independent

$$F_{X_1,X_2,X_3}(x_1, x_2, x_3) = F_{X_3|X_1,X_2}(x_3|x_1, x_2)F_{X_2}(x_2)F_{X_1}(x_1)$$

• Here $X_1, X_2$ are the parents of $X_3$

• The computation of joint density

  • Decomposed

  • Tractable
An alternative order of conditioning would lead to

\[
F_{X_1, X_2, X_3} = F_{X_1 | X_2, X_3} F_{X_2, X_3} = F_{X_1 | X_2, X_3} F_{X_2 | X_3} F_{X_3}
\]
• If $X_2$ and $X_1$ are independent

• This graph will not be reduced

\[ F_{X_1,X_2,X_3} = F_{X_1|X_2,X_3} F_{X_2,X_3} \]
\[ = F_{X_1|X_2,X_3} F_{X_2|X_3} F_{X_3} \]
• If $X_2$ and $X_1$ are independent
  - This graph will not be reduced
  • Since $X_1$ and $X_2$ may not be independent conditioned on $X_3$

\[
F_{X_1,X_2,X_3} = F_{X_1|X_2,X_3}F_{X_2,X_3} = F_{X_1|X_2,X_3}F_{X_2|X_3}F_{X_3}
\]
• If $X_2$ and $X_1$ are independent

• This graph will not be reduced

Since $X_1$ and $X_2$ may not be independent conditioned on $X_3$

• Can we construct a counter example to show?

\[
F_{X_1, X_2, X_3} = F_{X_1 | X_2, X_3} F_{X_2, X_3} = F_{X_1 | X_2, X_3} F_{X_2 | X_3} F_{X_3}
\]
• However, if $X_2$ and $X_3$ were independent then

• The graph will be reduced

\[
\begin{align*}
F_{X_1, X_2, X_3} &= F_{X_1 | X_2, X_3} F_{X_2, X_3} \\
&= F_{X_1 | X_2, X_3} F_{X_2 | X_3} F_{X_3}
\end{align*}
\]
• If the graph is

![Graph Diagram]
• If the graph is

\[ \begin{align*}
X_1 & \rightarrow X_2 \\
X_1 & \rightarrow X_4 \\
X_4 & \rightarrow X_2 \\
X_4 & \rightarrow X_3 \\
X_3 & \rightarrow X_5 \\
X_5 & \rightarrow X_6 \\
X_5 & \rightarrow X_7 \\
X_6 & \rightarrow X_7 \\
\end{align*} \]

• Then,

\[
F_{X_1, X_2, \ldots, X_7} = F_{X_1}F_{X_2}F_{X_3}F_{X_4|X_1, X_2, X_3}F_{X_5|X_1, X_3}F_{X_6|X_4}F_{X_7|X_4, X_5}
\]
• If the graph is
• If the graph is

\[ Y \]
\[ X_1 \]
\[ X_2 \]
\[ \ldots \]
\[ X_n \]

• Then,

\[ F_{X_1, X_2, \ldots, X_n, Y} = F_Y F_{X_1|Y} F_{X_2|Y} \ldots F_{X_n|Y} \]
• The repetition could be simplified by defining a *plate*
Graphical probabilistic model with deterministic parameters

\[ F_{X,Y|s,\alpha,\sigma^2} = F_{Y|\alpha} \prod_{i=1}^{n} F_{X_i|Y,s_i,\sigma^2} \]
• Graphical probabilistic model with deterministic parameters

\[
F_{X,Y|s,\alpha,\sigma^2} = F_Y|\alpha \prod_{i=1}^{n} F_{X_i|Y,s_i,\sigma^2}
\]

• For example, \( F_{X_i|Y,s_i,\sigma^2} \) Gaussian
Graphical probabilistic model with observed variables

\[ F_{X,Y|s,\alpha,\sigma^2} = F_Y|_\alpha \prod_{i=1}^{n} F_{X_i|Y,s_i,\sigma^2} \]

observed variables
Graphical probabilistic model with observed variables

\[ F_{X,Y|s,\alpha,\sigma^2} = F_{Y|\alpha} \prod_{i=1}^{n} F_{X_i|Y,s_i,\sigma^2} \]

latent variable
• Can we infer independence or conditional independence from Bayesian graphs? Let us investigate via a few simple examples.

• The joint pmf of these variables using the graph is

\[ p_{X_1, X_2, X_3} = p_{X_3} p_{X_1|X_3} p_{X_2|X_3} \]
• Can we infer independence or conditional independence from Bayesian graphs? Let us investigate via a few simple examples.

\[ p_{X_1, X_2, X_3} = p_{X_3} p_{X_1|X_3} p_{X_2|X_3} \]

\[ p_{X_1, X_2|X_3} = \frac{p_{X_1, X_2, X_3}}{p_{X_3}} = p_{X_1|X_3} p_{X_2|X_3} \]

• They are independent conditioned on \( X_3 \)

Node \( X_3 \) is tail-to-tail with respect to path from \( X_1 \) to \( X_2 \)
• Can we infer independence or conditional independence from Bayesian graphs? Let us investigate via a few simple examples.

• The joint pmf of these variables using the graph

\[ p_{X_1, X_2, X_3} = p_{X_3} p_{X_1|X_3} p_{X_2|X_3} \]

• \( X_1 \) and \( X_2 \) are independent conditioned on \( X_3 \)

• Are \( X_1, X_2 \) independent?
Can we infer independence or conditional independence from Bayesian graphs? Let us investigate via a few simple examples.

\[
px_1, x_2, x_3 = px_3 px_1 | x_3 px_2 | x_3
\]

\[
px_1, x_2 = \sum_{x_3} px_1, x_2, x_3 = \sum_{x_3} px_3 px_1 | x_3 px_2 | x_3 \neq px_1, px_2
\]

• They are not independent unless

\[
X_1 \text{ and } X_3 \text{ as well as } X_2 \text{ and } X_3 \text{ are independent}
\]
Another example

\[
p_{X_1, X_2, X_3} = p_{X_1} p_{X_3 | X_1} p_{X_2 | X_3}
\]

\[
p_{X_1, X_2} = \sum_{X_3} p_{X_1, X_2, X_3} = p_{X_1} \sum_{X_3} p_{X_3 | X_1} p_{X_2 | X_3} \neq p_{X_1} p_{X_2}
\]

They are not independent
• Another example

\[ p_{X_1,X_2,X_3} = p_{X_1} p_{X_3|X_1} p_{X_2|X_3} \]

\[ p_{X_1,X_2|X_3} = \frac{p_{X_1,X_2,X_3}}{p_{X_3}} = \frac{p_{X_1} p_{X_3|X_1} p_{X_2|X_3}}{p_{X_3}} = p_{X_1|X_3} p_{X_2|X_3} \]

• They are independent conditioned on \( X_3 \)

Node \( X_3 \) is head-to-tail with respect to path from \( X_1 \) to \( X_2 \)
Another example

\[ p_{X_1,X_2,X_3} = p_{X_1} p_{X_2} p_{X_3 | X_1,X_2} \]

- Are \( X_1, X_2 \) independent?
• Another example

\[ p_{X_1, X_2, X_3} = p_{X_1} p_{X_2} p_{X_3 | X_1, X_2} \]

Are \(X_1, X_2\) independent?

\[ p_{X_1, X_2} = \sum_{x_3} p_{X_1, X_2, X_3} = p_{X_1} p_{X_2} \sum_{x_3} p_{X_3 | X_1, X_2} = p_{X_1} p_{X_2} \]

• Yes they are independent
• Another example

\[ p_{X_1, X_2, X_3} = p_{X_1} p_{X_2} p_{X_3 | X_1, X_2} \]

Are \( X_1, X_2 \) conditioned on \( X_3 \) independent?

\[ p_{X_1, X_2 | X_3} = \frac{p_{X_1, X_2, X_3}}{p_{X_3}} = \frac{p_{X_1} p_{X_2} p_{X_3 | X_1, X_2}}{p_{X_3}} \neq p_{X_1 | X_3} p_{X_2 | X_3} \]

• They are not independent conditioned on \( X_3 \)

Node \( X_3 \) is head-to-head with respect to path from \( X_1 \) to \( X_2 \)
• Example 6.7

• Two random variable are independent

• Conditioned on a third random variable then they are not.

• Assume $X$ and $Y$ are independent random binary data (that is basically a coin flip experiment).

• Equally likely to 0 or 1.
• Example 6.7

• Then by assumption they are independent.

• Define $Z$ to be another random variable as $Z = X + Y$

• $X$ and $Y$ are dependent conditioned on $Z = 1$

\[ P(X = 1, Y = 1 | Z = 1) = 0 \text{ however } P(X = 1 | Z = 1)P(Y = 1 | Z = 1) = 1/2 \times 1/2 = 1/4 \]
• Summary of $X_1$ and $X_2$ independence

  • Conditionally independent but not independent
    • not blocked unless the node on the path is observed
  
  • Conditionally independent but not independent
    • not blocked unless the node on the path is observed
  
  • Independent but not conditionally independent
    • blocked unless the blocking node is observed
• Bayesian networks

• A tail-to-tail node or head-to-tail node “leaves” a path unblocked unless the node is observed (that is, the distribution is conditioned on that variable). In that case it blocks the path

• Conditionally independent but not independent
• Bayesian networks

• A tail-to-tail node or head-to-tail node “leaves” a path unblocked unless the node is observed (that is, it is conditioned on that variable). In that case it blocks the path

• A head-to-head node blocks the path if it is unobserved

  • If the node, and/or at least one of its descendants, is observed then the path becomes unblocked

  • Independent but not conditionally independent
• Bayesian networks

• A tail-to-tail node or head-to-tail node “leaves” a path unblocked unless the node is observed (that is, it is conditioned on that variable). In that case it blocks the path

• A head-to-head node blocks the path if it is unobserved

  • If the node, and/or at least one of its descendants, is observed then the path becomes unblocked

• When the path between two nodes is blocked then the two nodes (the variables) are independent
• These rules apply to larger networks and to sets of nodes

• The path between $X_1$ and $X_2$
  • Unblocked by $X_5$
  • Tail-to-tail
  • Blocked by $X_3$
  • Head-to-head

$X_1$ and $X_2$ are independent
• If the path between two nodes is blocked then the nodes are independent—conditioned on the variable that blocked the path
• These rules apply to larger networks and to sets of nodes

• The path between $X_1$ and $X_2$
  
  • Blocked by $X_5$
  
  • Conditioned
  
  • Tail-to-tail
  
  • Blocked by $X_3$
  
  • Head-to-head

$X_1$ and $X_2$ are independent
• These rules apply to larger networks and to sets of nodes

• The path between $X_1$ and $X_2$
  - Unblocked by $X_3$
    - Head-to-head
    - Conditioned on its descendent
      - Unblocked by $X_5$
        - Tail-to-tail

$X_1$ and $X_2$ are not independent
In general, Bayesian networks can be represented as

\[ p_x = \prod_{k=1}^{K} p_{X_k | p_a(k)} \text{ where } p_a(k) \text{ is the set of parent’s of node } k \]

Note that Bayesian graphs do not have cycles

- Directed acyclic graph

- Invalid

\[ p_{X_1 | X_2} p_{X_2 | X_3} p_{X_3 | X_1} \]
Graphical modeling for inference

- Bayesian networks

- Markov random fields

- Factor graphs
• Conditional independence is often difficult to infer from directed graphs.

• Undirected graphs are also powerful tools
  
  • Markov undirected networks
    
    • Clique
      
      • A group of nodes fully connected

    • Maximal clique
      
      • Cliques that can not be expanded
• Cliques
• Cliques
• Cliques
• Maximal clique
• Maximal clique
• Not a clique
The probability distribution can be written as

\[ F_X = \frac{1}{Z} \prod_c \psi_c(X) \]

where \( \psi_c \) is the “potential function” of clique

An example,

\[ F_X = F_{X_1,X_2,\ldots,X_n} = F_{X_1}F_{X_2|X_1}F_{X_3|X_2} \cdots F_{X_n|X_{n-1}} \]

\[ F_X = \frac{1}{Z} \psi_{1,2}(X_1, X_2)\psi_{2,3}(X_2, X_3) \cdots \psi_{n-1,n}(X_{n-1}, X_n) \]
• The network

\[
\begin{align*}
\psi_{1,2}(X_1, X_2) &= F_{X_1}F_{X_2|X_1} \\
\psi_{2,3}(X_2, X_3) &= F_{X_3|X_2} \\
&\vdots \\
\psi_{n-1,n}(X_{n-1}, X_n) &= F_{X_n|X_{n-1}}
\end{align*}
\]

\[
F_X = \frac{1}{Z} \psi_{1,2}(X_1, X_2)\psi_{2,3}(X_2, X_3) \ldots \psi_{n-1,n}(X_{n-1}, X_n)
\]
• A less obvious example

\[ F_X = F_{X_1,X_2,X_3,X_4,X_5} = F_{X_1} F_{X_2} F_{X_3|X_1,X_2} F_{X_4|X_3} F_{X_5|X_3} \]
\[ F_x = F_{X_1} F_{X_2} F_{X_3 | X_1, X_2} F_{X_4 | X_3} F_{X_5 | X_3} \]

- Let’s recall the rules on independence

\[ X_1 \text{ and } X_2 \text{ are independent} \]

\[ X_4 \text{ and } X_5 \text{ are not independent} \]
\[ F_x = F_{X_1} F_{X_2} F_{X_3|X_1,X_2} F_{X_4|X_3} F_{X_5|X_3} \]

- Let's recall the rules on independence

- \( X_1 \) and \( X_2 \) are not independent conditioned on \( X_3 \)

- \( X_4 \) and \( X_5 \) are independent conditioned on \( X_3 \)
To convert a directed graph to an undirected graph

- Moralization
  - Remove directionality in all links
  - Add links to all pairs of parents of each node

\[
F_{\mathbf{x}} = F_{X_1} F_{X_2} F_{X_3|X_1, X_2} F_{X_4|X_3} F_{X_5|X_3}
\]
To convert a directed graph to an undirected graph

- Moralization
  - Remove directionality in all links
  - Add links to all pairs of parents of each node
  - Identify maximal cliques

\[
F_X = F_{X_1} F_{X_2} F_{X_3 | X_1, X_2} F_{X_4 | X_3} F_{X_5 | X_3}
\]
To convert a directed graph to an undirected graph

- Moralization
  - Remove directionality in all links
  - Add links to all pairs of parents of each node
  - Identify maximal cliques

\[
F_X = F_{X_1} F_{X_2} F_{X_3}_{X_1,X_2} F_{X_4}_{X_3} F_{X_5}_{X_3}
\]
To convert a directed graph to an undirected graph

- Moralization

  - Remove directionality in all links
  
  - Add links to all pairs of parents of each node
  
- Identify maximal cliques

\[
F_X = F_{X_1} F_{X_2} F_{X_3|X_1, X_2} F_{X_4|X_3} F_{X_5|X_3}
\]
To convert a directed graph to an undirected graph

- Moralization

\[ F_X = \frac{1}{Z} \psi_{1,2,3}(X_1, X_2, X_3) \psi_{3,4}(X_3, X_4) \psi_{3,5}(X_3, X_5) \]

- Remove directionality in all links

- Add links to all pairs of parents of each node

- Identify maximal cliques

- Maximal cliques form potentials
To convert a directed graph to an undirected graph

- Moralization

\[ F_X = \frac{1}{Z} \psi_{1,2,3}(X_1, X_2, X_3)\psi_{3,4}(X_3, X_4)\psi_{3,5}(X_3, X_5) \]

- Remove directionality in all links

- Add links to all pairs of parents of each node

- Identify maximal cliques

- Maximal cliques form potentials
\[ F_X = F_{X_1} F_{X_2} F_{X_3|X_1, X_2} F_{X_4|X_3} F_{X_5|X_3} \]

- Moralization
  \[ F_X = \frac{1}{Z} \psi_{1,2,3}(X_1, X_2, X_3) \psi_{3,4}(X_3, X_4) \psi_{3,5}(X_3, X_5) \]

- Remove directionality in all links

- Add links to all pairs of parents of each node

- Identify maximal cliques

- Maximal cliques form potential functions

- Adjust with parameter \( Z \)
- The less obvious example

\[ F_{\mathbf{X}} = F_{X_1, X_2, X_3, X_4, X_5} = F_{X_1} F_{X_2} F_{X_3|X_1, X_2} F_{X_4|X_3} F_{X_5|X_3} \]
• The less obvious example

\[ F_x = F_{X_1, X_2, X_3, X_4, X_5} = F_{X_1} F_{X_2} F_{X_3|X_1, X_2} F_{X_4|X_3} F_{X_5|X_3} \]

\[ F_x = \frac{1}{Z} \psi_{1,2,3}(X_1, X_2, X_3) \psi_{3,4}(X_3, X_4) \psi_{3,5}(X_3, X_5) \]
• The less obvious example

\[ p\mathbf{x} = p_{X_1}p_{X_2}p_{X_3|X_1, X_2}p_{X_4|X_3}p_{X_5|X_3} \]

\[ p\mathbf{x} = \frac{1}{Z} \psi_{1,2,3}(X_1, X_2, X_3)\psi_{3,4}(X_3, X_4)\psi_{3,5}(X_3, X_5) \]

• where

\[ Z = \sum_{\mathbf{x}} \psi_{1,2,3}(X_1, X_2, X_3)\psi_{3,4}(X_3, X_4)\psi_{3,5}(X_3, X_5) \]
• The less obvious example

\[ F_{\mathbf{x}} = \frac{1}{Z} \psi_{1,2,3}(X_1, X_2, X_3)\psi_{3,4}(X_3, X_4)\psi_{3,5}(X_3, X_5) \]
• The less obvious example

$X_1$ and $X_2$ are not independent conditioned on $X_3$

$X_4$ and $X_5$ are independent conditioned on $X_3$
• The less obvious example

\[ X_1 \text{ and } X_2 \text{ are not independent conditioned on } X_3 \]

\[ X_4 \text{ and } X_5 \text{ are independent conditioned on } X_3 \]

• The path between the two vertices is blocked
Another illustrative example

\[ p_X = p_{X_1} p_{X_2} p_{X_3} p_{X_4} | X_1 X_2 X_3 \]
• Another illustrative example

\[ p_x = p_{X_1} p_{X_2} p_{X_3} p_{X_4} \mid X_1 X_2 X_3 \]

\[ p_x = \frac{1}{Z} \psi_{1,2,3,4}(X_1, X_2, X_3, X_4) \]
• Another illustrative example

• Conditional independence is not present since all vertices are connected
• Markov random fields and Bayesian networks are not prefect

• Consider this directed graph
• Markov random fields and Bayesian networks are not prefect

• Consider this directed graph

• Now a moralized Markov random field
• Markov random fields and Bayesian networks are not prefect

• Consider this directed graph

• Now a moralized Markov random field

\[X_1 \text{ and } X_2 \text{ are independent}\]
• Markov random fields and Bayesian networks are not prefect

• The moralized Markov random field is not very useful
• Markov random network offers a powerful tool to identify conditional independence
• Markov random network offers a powerful tool to identify conditional independence

• Conditioned on observed nodes
• Markov random network offers a powerful tool to identify conditional independence

• Conditioned on observed nodes

• Nodes in these sets are independent

• This graphical representation is indeed powerful
• Graphical modeling for inference

  • Bayesian networks

  • Markov random fields

  • Factor graphs
• Factor graphs

• Allow a global function of several variables be expressed as a product of factors of subsets of these variables

\[ p_X = \prod_s f_s(X_s) \]
Factor graphs

- Allow a global function of several variables be expressed as a product of factors of subsets of these variables

\[ p_X = f_a(X_1)f_b(X_1, X_2)f_c(X_1, X_2)f_d(X_2, X_3) \]
• Factor graphs

• Allow a global function of several variables be expressed as a product of factors of subsets of these variables

\[ p_X = \prod_{s} f_s(X_s) \]

• They could simplify computation of complex functions

• They are generalizations of Bayesian and Markov graphs.

• The factor graphs are more explicit than Bayesian and Markov graphs.

• By construction, factor graphs are bipartite graphs
• By construction, factor graphs are bipartite graphs

\[ p_X = f_a(X_1)f_b(X_1, X_2)f_c(X_1, X_2)f_d(X_2, X_3) \]
By construction, factor graphs are bipartite graphs

\[ p_x = f_a(X_1)f_b(X_1, X_2)f_c(X_1, X_2)f_d(X_2, X_3) \]
By construction, factor graphs are bipartite graphs

\[ p_X = f_a(X_1) f_b(X_1, X_2) f_c(X_1, X_2) f_d(X_2, X_3) \]
• Example 6.8

• A general function factorized

\[ F_{X_1, X_2, X_3} = F_1(X_1)F_2(X_1, X_2)F_3(X_1, X_2)F_4(X_2, X_3) \]
Example 6.8

A function factorized

\[ F_{X_1, X_2, X_3} = F_1(X_1) F_2(X_1, X_2) F_3(X_1, X_2) F_4(X_2, X_3) \]
Example 6.8

A function factorized

\[ F_{X_1, X_2, X_3} = F_1(X_1)F_2(X_1, X_2)F_3(X_1, X_2)F_4(X_2, X_3) \]
Example 6.8

A function factorized

\[ F_{X_1, X_2, X_3} = F_1(X_1)F_2(X_1, X_2)F_3(X_1, X_2)F_4(X_2, X_3) \]

edge if it is a variable in that factor
Example 6.8

A function factorized

\[ F_{X_1, X_2, X_3} = F_1(X_1)F_2(X_1, X_2)F_3(X_1, X_2)F_4(X_2, X_3) \]
• Factor graphs are bipartite

• A generalization of Tanner graphs

  • Tanner graphs were developed to describe decoding of low density parity check codes (LDPC)

  • Factor graphs are particularly useful for decoding of modern error correcting codes

• Factor graph can unify seemingly and historically different computations/processing of data
• Factor graphs unify
  • Kalman filtering
  • Statistical physics via Markov random fields
  • Recursive least-squared filters
  • Hidden Markov models
  • Viterbi decoding
  • Bayesian and Markov networks can be represented as factor graphs
• Recall an earlier example

\[ F_X = F_{X_1, X_2, X_3, X_4, X_5} = F_{X_1} F_{X_2} F_{X_3 | X_1, X_2} F_{X_4 | X_3} F_{X_5 | X_3} \]

• Markov and Bayesian networks
Recall

\[ F_\mathbf{x} = F_{X_1, X_2, X_3, X_4, X_5} = F_{X_1} F_{X_2} F_{X_3 | X_1, X_2} F_{X_4 | X_3} F_{X_5 | X_3} \]

\[ = F_A(X_1) F_B(X_2) F_C(X_1, X_2, X_3) F_D(X_3, X_4) F_E(X_3, X_5) \]
• Recall

\[
F_X = F_{X_1, X_2, X_3, X_4, X_5} = F_{X_1} F_{X_2} F_{X_3|X_1, X_2} F_{X_4|X_3} F_{X_5|X_3} = F_A(X_1) F_B(X_2) F_C(X_1, X_2, X_3) F_D(X_3, X_4) F_E(X_3, X_5)
\]
• Recall

\[ F_X = F_{X_1, X_2, X_3, X_4, X_5} = F_{X_1} F_{X_2} F_{X_3|X_1, X_2} F_{X_4|X_3} F_{X_5|X_3} \]

\[ = F_A(X_1) F_B(X_2) F_C(X_1, X_2, X_3) F_D(X_3, X_4) F_E(X_3, X_5) \]

• Alternative factor graph representation
• Cycles in a graph
\[
F_{X_1, X_2, X_3} = F_1(X_1)F_2(X_1, X_2)F_3(X_1, X_2)F_4(X_2, X_3)
\]
\[ F_{X_1, X_2, X_3} = F_1(X_1)F_2(X_1, X_2)F_3(X_1, X_2)F_4(X_2, X_3) \]

If \( F_2 \) and \( F_3 \) were combined

\[ F_5(X_1, X_2) = F_2(X_1, X_2)F_3(X_1, X_2) \]
\[ F_{X_1, X_2, X_3} = F_1(X_1)F_2(X_1, X_2)F_3(X_1, X_2)F_4(X_2, X_3) \]
• A graph with no cycles (or loops) is a tree where there is one and only one path connecting two nodes.
• A Bayesian network can be presented as a factor graph

\[ p_{X_1, X_2, X_3} = p_{X_1} p_{X_2} p_{X_3 | X_1, X_2} \]
A Bayesian network can be presented as a factor graph

\[ p_{X_1, X_2, X_3} = p_{X_1} p_{X_2} p_{X_3 | X_1, X_2} \]
• The Bayesian network can be moralized to yield a Markov graph

\[ p_{X_1, X_2, X_3} = p_{X_1} p_{X_2} p_{X_3 | X_1, X_2} \]

• Then, directed and undirected graphs are
• A factor graph
\[ p_{X_1, X_2, X_3} = p_{X_1} p_{X_2} p_{X_3 \mid X_1, X_2} \]

• Conversion of directed graph to undirected resulted in cycles (loops)

  • Moralization step

• Conversion to factor graph did not result in cycles
• A factor graph \[ p_{X_1, X_2, X_3} = p_{X_1} p_{X_2} p_{X_3 | X_1, X_2} \]

• Conversion of directed graph to undirected resulted in cycles (loops)
  
  • Moralization step

• Conversion to factor graph did not result in cycles
• A factor graph  \( p_{X_1, X_2, X_3} = p_{X_1} p_{X_2} p_{X_3 | X_1, X_2} \)

• Conversion of directed graph to undirected resulted in cycles (loops)
  
  • Moralization step

• Conversion to factor graph did not result in cycles
• Example 6.9

\[ p_X = F_a(X_1, X_2)F_b(X_2, X_3)F_c(X_2, X_4) \]

\[ p_{X_2} = \sum_{x_1, x_3, x_4} p_X = \sum_{x \setminus x_2} F_a(x_1, x_2)F_b(x_2, x_3)F_c(x_2, x_4) \]

• Computing marginals is critical for inference

• Direct computation is prohibitively expensive
• Marginalization

\[ p_X = F_a(X_1, X_2)F_b(X_2, X_3)F_c(X_2, X_4) \]

\[ p_{X_2} = \sum_{x_1, x_3, x_4} p_X = \sum_{x \setminus x_2} F_a(x_1, x_2)F_b(x_2, x_3)F_c(x_2, x_4) \]

\[ = \left\{ \sum_{x_1} F_a(x_1, x_2) \right\} \left\{ \sum_{x_3} F_b(x_2, x_3) \right\} \left\{ \sum_{x_4} F_c(x_2, x_4) \right\} \]

• Distributive law

\[ (x+y)(a+b) = xa + xb + ya + yb \]

• 3 operations versus 7 operations
The marginalization can be implemented efficiently with the “sum-product” algorithm on the factor graph:

\[
p_{X_2} = \left\{ \sum_{x_1} F_a(x_1, x_2) \right\} \left\{ \sum_{x_3} F_b(x_2, x_3) \right\} \left\{ \sum_{x_4} F_c(x_2, x_4) \right\}
\]

- Distributive law

- Efficient reuse of intermediate sum values

- Iterative data flow
• The sum-product algorithm on the factor graph

\[ p(x_2) = \left\{ \sum_{x_1} F_a(x_1, x_2) \right\} \left\{ \sum_{x_3} F_b(x_2, x_3) \right\} \left\{ \sum_{x_4} F_c(x_2, x_4) \right\} \]

• The root is the variable of interest and leaves are marginalized
• The sum-product algorithm on the factor graph

\[ p_X(x_2) = \left\{ \sum_{x_1} F_a(x_1, x_2) \right\} \left\{ \sum_{x_3} F_b(x_2, x_3) \right\} \left\{ \sum_{x_4} F_c(x_2, x_4) \right\} \]

• The root is the variable of interest and leaves are marginalized
• The sum-product algorithm on the factor graph

\[ p_{x_2} = \sum_{x_1} F_a(x_1, x_2) \sum_{x_3} F_b(x_2, x_3) \sum_{x_4} F_c(x_2, x_4) \]

• Message passing

\[ \mu_{x_1 \rightarrow F_a}(x_1) = 1 \]
\[ \mu_{F_a \rightarrow x_2}(x_2) = \sum_{x_1} F_a(x_1, x_2) \]
\[ \mu_{x_4 \rightarrow F_c}(x_4) = 1 \]
\[ \mu_{F_c \rightarrow x_2}(x_2) = \sum_{x_4} F_c(x_2, x_4) \]
\[ \mu_{x_3 \rightarrow F_b}(x_3) = 1 \]
\[ \mu_{F_b \rightarrow x_2}(x_2) = \sum_{x_3} F_b(x_2, x_3) \]
• The sum-product algorithm on the factor graph

\[
p_{X_2} = \left\{ \sum_{x_1} F_a(x_1, x_2) \right\} \left\{ \sum_{x_3} F_b(x_2, x_3) \right\} \left\{ \sum_{x_4} F_c(x_2, x_4) \right\}
\]

• Message passing

\[
\mu_{x_1 \rightarrow F_a}(x_1) = 1
\]

\[
\mu_{F_a \rightarrow x_2}(x_2) = \sum_{x_1} F_a(x_1, x_2)
\]

\[
\mu_{x_4 \rightarrow F_c}(x_4) = 1
\]

\[
\mu_{F_c \rightarrow x_2}(x_2) = \sum_{x_4} F_c(x_2, x_4)
\]

\[
\mu_{x_3 \rightarrow F_b}(x_3) = 1
\]

\[
\mu_{F_b \rightarrow x_2}(x_2) = \sum_{x_3} F_b(x_2, x_3)
\]
• The sum-product algorithm on the factor graph

\[ p_{X_2} = \left\{ \sum_{x_1} F_a(x_1, x_2) \right\} \left\{ \sum_{x_3} F_b(x_2, x_3) \right\} \left\{ \sum_{x_4} F_c(x_2, x_4) \right\} \]

• Message passing

\[ \mu_{x_1 \rightarrow F_a(x_1)} = 1 \]

\[ \mu_{F_a \rightarrow x_2(x_2)} = \sum_{x_1} F_a(x_1, x_2) \]

\[ \mu_{x_4 \rightarrow F_c(x_4)} = 1 \]

\[ \mu_{F_c \rightarrow x_2(x_2)} = \sum_{x_4} F_c(x_2, x_4) \]

\[ \mu_{x_3 \rightarrow F_b(x_3)} = 1 \]

\[ \mu_{F_b \rightarrow x_2(x_2)} = \sum_{x_3} F_b(x_2, x_3) \]
• The sum-product algorithm on the factor graph

\[ p(x_2) = \left\{ \sum_{x_1} F_a(x_1, x_2) \right\} \left\{ \sum_{x_3} F_b(x_2, x_3) \right\} \left\{ \sum_{x_4} F_c(x_2, x_4) \right\} \]

• Message passing

\[ \mu_{x_1 \rightarrow F_a}(x_1) = 1 \]

\[ \mu_{F_a \rightarrow x_2}(x_2) = \sum_{x_1} F_a(x_1, x_2) \]

\[ \mu_{x_4 \rightarrow F_c}(x_4) = 1 \]

\[ \mu_{F_c \rightarrow x_2}(x_2) = \sum_{x_4} F_c(x_2, x_4) \]

\[ \mu_{x_3 \rightarrow F_b}(x_3) = 1 \]

\[ \mu_{F_b \rightarrow x_2}(x_2) = \sum_{x_3} F_b(x_2, x_3) \]
• Message passing is done

\begin{align*}
\mu_{x_1 \rightarrow F_a}(x_1) &= 1 \\
\mu_{F_a \rightarrow x_2}(x_2) &= \sum_{x_1} F_a(x_1, x_2) \\
\mu_{x_4 \rightarrow F_c}(x_4) &= 1 \\
\mu_{F_c \rightarrow x_2}(x_2) &= \sum_{x_4} F_c(x_2, x_4) \\
\mu_{x_3 \rightarrow F_b}(x_3) &= 1 \\
\mu_{F_b \rightarrow x_2}(x_2) &= \sum_{x_3} F_b(x_2, x_3)
\end{align*}

\[ p_{X_2} = \mu_{F_a \rightarrow x_2}(x_2) \mu_{F_b \rightarrow x_2}(x_2) \mu_{F_c \rightarrow x_2}(x_2) \]

\[ = \left\{ \sum_{x_1} F_a(x_1, x_2) \right\} \left\{ \sum_{x_3} F_b(x_2, x_3) \right\} \left\{ \sum_{x_4} F_c(x_2, x_4) \right\} \]
• The sum-product algorithm on the factor graph with different root

\[ p_{X_3} = \left\{ \sum_{x_1} \sum_{x_2} \sum_{x_4} F_c(x_2, x_4)F_a(x_1, x_2) \right\}\left\{ \sum_{x_2} F_b(x_2, x_3) \right\} \]

• The root is the variable of interest and leaves are marginalized
• The sum-product algorithm on the factor graph with different root

\[ p_{X_3} = \left\{ \sum_{x_1} \sum_{x_2} \sum_{x_4} F_c(x_2, x_4) F_a(x_1, x_2) \right\} \left\{ \sum_{x_2} F_b(x_2, x_3) \right\} \]

• The message passing

\[
\begin{align*}
\mu_{x_1 \rightarrow F_a}(x_1) &= 1 \\
\mu_{F_a \rightarrow x_2}(x_2) &= \sum_{x_1} F_a(x_1, x_2) \\
\mu_{x_4 \rightarrow F_c}(x_4) &= 1 \\
\mu_{F_c \rightarrow x_2}(x_2) &= \sum_{x_4} F_c(x_2, x_4) \\
\mu_{x_2 \rightarrow F_b}(x_2) &= \mu_{F_a \rightarrow x_2}(x_2) \mu_{F_c \rightarrow x_2}(x_2) \\
\mu_{F_b \rightarrow x_3}(x_3) &= \sum_{x_2} F_b(x_2, x_3) \mu_{x_2 \rightarrow F_b}(x_2)
\end{align*}
\]
• The sum-product algorithm on the factor graph with different root

\[ p_{X_3} = \left\{ \sum_{x_1} \sum_{x_2} \sum_{x_4} F_c(x_2, x_4) F_a(x_1, x_2) \right\} \left\{ \sum_{x_2} F_b(x_2, x_3) \right\} \]

• The message propagates from root back to leaf nodes

\[ \mu_{x_3 \rightarrow F_b}(x_3) = 1 \]
\[ \mu_{F_b \rightarrow x_2}(x_2) = \sum_{x_3} F_b(x_2, x_3) \]
\[ \mu_{x_2 \rightarrow F_a}(x_2) = \mu_{F_b \rightarrow x_2}(x_2) \mu_{F_c \rightarrow x_2}(x_2) \]
\[ \mu_{F_a \rightarrow x_1}(x_1) = \sum_{x_2} F_a(x_1, x_2) \mu_{x_2 \rightarrow F_a}(x_2) \]
\[ \mu_{x_2 \rightarrow F_c}(x_2) = \mu_{F_a \rightarrow x_2}(x_2) \mu_{F_b \rightarrow x_2}(x_2) \]
\[ \mu_{F_c \rightarrow x_4}(x_4) = \sum_{x_2} F_c(x_2, x_4) \mu_{x_2 \rightarrow F_c}(x_2) \]
• The sum-product algorithm on the factor graph with different root

\[ p_{X_3} = \left\{ \sum_{x_1} \sum_{x_2} \sum_{x_4} F_c(x_2, x_4) F_a(x_1, x_2) \right\} \left\{ \sum_{x_2} F_b(x_2, x_3) \right\} \]

• The message passing

\[ p_{X_3} = \mu_{F_b \rightarrow x_3}(x_3) \]
Another example from earlier pages in this set

\[ p_X = p_{X_1, X_2, \ldots, X_n} = p_{X_1} p_{X_2 \mid X_1} p_{X_3 \mid X_2} \cdots p_{X_n \mid X_{n-1}} \]

\[ p_X = \frac{1}{Z} \psi_{1,2}(X_1, X_2) \psi_{2,3}(X_2, X_3) \cdots \psi_{n-1,n}(X_{n-1}, X_n) \]
Another example from earlier pages in this set

\[ p_x = \frac{1}{Z} \psi_{1,2}(X_1, X_2) \psi_{2,3}(X_2, X_3) \cdots \psi_{n-1,n}(X_{n-1}, X_n) \]

\[ p_{X_k} = \sum_{x \setminus x_k} p_x = \sum_{x \setminus x_k} p_{X_1, X_2, \ldots, X_n} = \sum_{x \setminus x_k} p_{X_1} p_{X_2 | X_1} p_{X_3 | X_2} \cdots p_{X_n | X_{n-1}} \]
• Another example from earlier pages in this set

\[
p_{X_k} = \sum_{x \setminus x_k} p_x = \sum_{x \setminus x_k} p_{X_1, X_2, \ldots, X_n} = \sum_{x \setminus x_k} p_{X_1} p_{X_2|X_1} p_{X_3|X_2} \cdots p_{X_n|X_{n-1}}
\]

\[
p_{X_k} = \frac{1}{Z} \left[ \sum_{x_{k-1}} \psi_{k-1,k}(X_{k-1}, X_k) \cdots \sum_{x_2} \psi_{2,3}(X_2, X_3) \sum_{x_1} \psi_{1,2}(X_1, X_2) \right] \left[ \sum_{x_{k+1}} \psi_{k,k+1}(X_k, X_{k+1}) \cdots \sum_{x_n} \psi_{n-1,n}(X_{n-1}, X_n) \right]
\]
• If each variable takes $K$ possible values the complexity is $O(nK^2)$

• A naive computation will be exponential rather than linear

• Message passing

\[
\sum_{x_1} \psi_{1,2}(x_1, x_2)
\]
• If each variable takes $K$ possible values the complexity is $O(nK^2)$

• A naive computation will be exponential rather than linear

• Message passing

\[
\sum_{x_1} \psi_{1,2}(x_1, x_2) \rightarrow \sum_{x_2} \psi_{2,3}(x_2, x_3) \rightarrow \sum_{x_1} \psi_{1,2}(x_1, x_2)
\]

\[
X_1 \rightarrow X_2 \rightarrow \cdots \rightarrow X_{n-1} \rightarrow X_n
\]
• If each variable takes $K$ possible values the complexity is $O(nK^2)$

• A naive computation will be exponential rather than linear

• Message passing
• Recall that factor graphs are ideal tools to describe $p_x$

• Note that the sum-product algorithm is ideal for computing marginals $p_{x_2}$

• The max-sum algorithm is ideal for computing

$$\mathbf{x}^* = \arg \max_x p_x$$

$$p_{x^*} = \max_x p_x$$
The max-sum algorithm is ideal for computing

\[ p_{X_k}^* = \max_{X \setminus x_k} p_X \]

Then, similar to distributive law

\[ \arg \max_X p_X = (\arg \max_{X_1} p_{X_1}^* \arg \max_{X_2} p_{X_2}^* \ldots \arg \max_{X_n} p_{X_n}^*) \]

Note that the probability mass function could be factored

\[ p_X = \prod_s f_s(X_s) \]

Leading to an efficient implementation
• Graphical modeling for inference

  • Bayesian networks

  • Markov random fields

  • Factor graphs
• Example 6.10