Lecture 11: Graphical Models

Recommended reading: Bishop: Chapter 8

Marc Deisenroth
Department of Computing
Imperial College London

February 10, 2017
Three types of probabilistic graphical models

- Bayesian networks (directed graphical models)
- Markov random fields (undirected graphical models)
- Factor graphs

- **Nodes**: (Sets of) random variables
- **Edges**: Probabilistic/functional relations between variables

Graph captures the way in which the joint distribution over all random variables can be decomposed into a product of factors depending only on a subset of these variables.
Why are they useful?

- Simple way to **visualize the structure** of a probabilistic model
- **Insights into properties** of the model (e.g., conditional independence) by inspection of the graph
- Can be used to **design/motivate new models**
- Complex computations for inference and learning can be expressed in terms of **graphical manipulations**
Importance of Visualization

\[ P(\{y_g, \gamma_g, t_{gk}, \beta_{gk}, l_d, f_g, z_n, i_{ng}\}|\{w_{nd}\}) = \prod_{g} p(y_g|\rho)p(\gamma_g|\sigma)p(f_g|\alpha). \]

\[ \prod_{k} p(t_{gk}|\gamma_g)p(\beta_{gk}|t_{gk}, y_g)p(\kappa|\alpha) \prod_{d} p(l_d|\kappa)p(\pi|\alpha) \prod_{n} p(z_n|\pi) \]

\[ \prod_{n} \prod_{g} p(i_{ng}|\beta, z_n) \prod_{n} \prod_{d} p(w_{nd}|i_{ng}, f, l_d) \]

From Kim et al. (NIPS, 2015)
Bayesian Networks (Directed Graphical Models)
We are given a data set 
\((x_1, y_1), \ldots, (x_N, y_N)\) where

\[ y_i = f(x_i) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \]

with \(f\) unknown.

Find a (regression) model that explains the data.

- **Consider polynomials** \( f(x) = \sum_{j=0}^{M} w_j x^j \) with parameters \( \mathbf{w} = [w_0, \ldots, w_M]^\top \).

- **Bayesian linear regression:** Place a conjugate Gaussian prior on the parameters: \( p(\mathbf{w}) = \mathcal{N}(0, \alpha^2 I) \)

From PRML (Bishop, 2006)
Graphical Model for Linear Regression

\[ p(y|x) = \mathcal{N}(y | f(x), \sigma^2) \]

\[ f(x) = \sum_{j=0}^{M} w_j x^j \]

\[ p(w) = \mathcal{N}(0, \alpha^2 I) \]
(Conditional) independence allows for a factorization of the joint distribution

\[ a \indep b \mid c \iff p(a \mid b, c) = p(a \mid c) \]
\[ \iff p(a, b \mid c) = p(a \mid c)p(b \mid c) \]

- (Conditional) independence allows for a factorization of the joint distribution
  - More efficient inference
- Conditional independence properties of the joint distribution can be read directly from the graph
- No analytical manipulations required.

\textbf{d-separation} (Pearl, 1988)
**D-Separation (Directed Graphs)**

Directed, acyclic graph in which $A$, $B$, $C$ are arbitrary, non-intersecting sets of nodes. Does $A \perp B | C$ hold?

Note: $C$ is observed if we condition on it (and the nodes in the GM are shaded)

Consider all possible paths from any node in $A$ to any node in $B$.

Any such **path is blocked** if it includes a node such that either

- Arrows on the path meet either **head-to-tail** or **tail-to-tail** at the node, and the node is in the set $C$ or
- Arrows meet **head-to-head** at the node and neither the node nor any of its descendants is in the set $C$

If all paths are blocked, then $A$ is **d-separated** from $B$ by $C$, and the joint distribution satisfies $A \perp B | C$. 

Graphical Models
A path is **blocked** if it includes a node such that either

- The arrows on the path meet either head-to-tail or tail-to-tail at the node, and the node is in the set $C$ (observed) or
- The arrows meet head-to-head at the node, and neither the node nor any of its descendants is in the set $C$ (observed)
Markov Random Fields (Undirected Graphical Models)
Markov Random Fields

Graphical Models

DAPI, Lecture 11

February 10, 2017
Joint Distribution

- Express joint distribution $p(x_1, \ldots, x_n) =: p(x)$ as a product of functions defined on subsets of variables that are local to the graph

- If $x_i, x_j$ are not connected directly by a link then $x_i \perp x_j | x \setminus \{x_i, x_j\}$ (conditionally independent given everything else)
Factorization of the Joint Distribution

- If $x_i \perp x_j | x \setminus \{x_i, x_j\}$ then $x_i, x_j$ never appear in a common factor in the factorization of the joint
- Joint distribution as a product of cliques (fully connected subgraphs)
- Define factors in the decomposition of the joint to be functions of the variables in (maximum) cliques:

$$p(x) \propto \prod_C \psi_C(x_C)$$

Example: $p(a, b, c, d) \propto \psi_1(a) \psi_2(b, c, d)$
Factorization of the Joint Distribution

\[ p(x) = \frac{1}{Z} \prod_C \psi_C(x_C) \]

- \( C \): maximal clique
- \( x_C \): all variables in this clique
- \( \psi_C(x_C) \): clique potential
- \( Z = \sum_x \prod_C \psi_C(x_C) \): normalization constant
Clique Potentials

\[ p(x) = \frac{1}{Z} \prod_{C} \psi_C(x_C) \]

Clique potentials \( \psi_C(x_C) \):

- \( \psi_C(x_C) \geq 0 \)
- Unlike directed graphs, no probabilistic interpretation necessary (e.g., marginal or conditional).
- If we convert a directed graph into an MRF, the clique potentials may have a probabilistic interpretation
Normalization Constant

\[ p(x) = \frac{1}{Z} \prod_C \psi_C(x_C) \]

- Gives us \textit{flexibility} in the definition the factorization in an MRF
- Normalization constant (also: partition function) \( Z \) is required for parameter learning (not covered in this course)
- In a \textit{discrete model} with \( M \) discrete nodes each having \( K \) states, the evaluation \( Z \) requires summing over \( K^M \) states
  - \textit{Exponential} in the size of the model
- In a \textit{continuous model}, we need to solve integrals
  - \textit{Intractable} in many cases

\textbf{Major limitation of MRFs}
Two easy checks for conditional independence:

- $A \indep B | C$ if and only if all paths from $A$ to $B$ pass through $C$. (Then, all paths are blocked)
- Alternative: Remove all nodes in $C$ from the graph. If there is a path from $A$ to $B$ then $A \indep B | C$ does not hold
Potentials as Energy Functions

- Look only at potential functions with $\psi_C(x_C) > 0$
  - $\psi_C(x_C) = \exp(-E(x_C))$ for some energy function $E$
- Joint distribution is the product of clique potentials
  - Total energy is the sum of the energies of the clique potentials
Example: Image Restoration

- Binary image, corrupted by 10% binary noise (pixel values flip with probability 0.1).
- Objective: Restore noise-free image

Pairwise MRF that has all its variables joined in cliques of size 2.
MRF-based approach

- Latent variables $x_i \in \{-1, +1\}$ are the binary noise-free pixel values that we wish to recover
- Observed variables $y_i \in \{-1, +1\}$ are the noise-corrupted pixel values
Two types of clique potentials:

- $\log \psi_{xy}(x_i, y_i) = E(x_i, y_i) = -\eta x_i y_i, \quad \eta > 0$
  
  ▶ Strong correlation between observed and latent variables

- $\log \psi_{xx}(x_i, x_j) = E(x_i, x_j) = -\beta x_i x_j, \quad \beta > 0$
  for neighboring pixels $x_i, x_j$

  ▶ Favor similar labels for neighboring pixels (smoothness prior)
Energy Function

Total energy:

\[ E(x, y) = -\eta \sum_i x_i y_i - \beta \sum_{i,j} x_i x_j + h \sum_i x_i \]

- Bias term places a prior on the latent pixel values, e.g., +1.
- Joint distribution \( p(x, y) = \frac{1}{Z} \exp(-E(x, y)) \)
- Fix \( y \)-values to the observed ones ➤ Implicitly define \( p(x|y) \)
- Example of an Ising model ➤ Statistical physics
ICM Algorithm for Image Restoration

Iterated Conditional Modes (ICM, Kittler & Föglein, 1984)

1. Initialize all $x_i = y_i$
2. Pick any $x_j$: Evaluate total energy
   
   $E(x \setminus j \cup \{+1\}, y), \quad E(x \setminus j \cup \{-1\}, y)$

3. Set $x_j$ to whichever state ($\pm 1$) has the lower energy
4. Repeat

Local optimum
Relation to Directed Graphs

- Directed and undirected graphs express different conditional independence properties
- Left: $a \perp b \mid \emptyset$, $a \perp b \mid c$ has no MRF equivalent
- Center: $a \perp b \mid \emptyset$, $c \perp d \mid a \cup b$, $a \perp b \mid c \cup d$ has no Bayesnet equivalent
Factor Graphs

Good references:


(Un)directed graphical models express a global function of several variables as a product of factors over subsets of those variables.

Factor graphs make this decomposition explicit by introducing additional nodes for the factors themselves.
Factorizing the Joint

The joint distribution is a product of factors:

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

- \( x = (x_1, \ldots, x_n) \)
- \( x_s: \) Subset of variables
- \( f_s: \) Factor; non-negative function of the variables \( x_s \)

- Building a factor graph as a bipartite graph:
  - Nodes for all random variables (same as in (un)directed graphical models)
  - Additional nodes for factors (black squares) in the joint distribution
  - Undirected links connecting each factor node to all of the variable nodes the factor depends on
Example

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

Efficient inference algorithms for factor graphs (e.g., sum-product algorithm, see Appendix for more information)
Applications of Inference in Graphical Models

- **Ranking**: TrueSkill (Herbrich et al., 2007)
- **Computer vision**: de-noising, segmentation, semantic labeling, ... (e.g., Sucar & Gillies, 1994; Shotton et al., 2006; Szeliski et al., 2008)
- **Coding theory**: Low-density parity-check codes, turbo codes, ... (e.g., McEliece et al., 1998)
- **Linear algebra**: Solve linear equation systems (Shental et al., 2008)
- **Signal processing**: Iterative state estimation (e.g., Bickson et al., 2007; Deisenroth & Mohamed, 2012)
Appendix
Revision: From Joints to Graphs

Consider the joint distribution

\[ p(a, b, c) = p(c|a, b)p(b|a)p(a) \]

Building the corresponding graphical model:

1. Create a node for all random variables
2. For each conditional distribution, we add a directed link (arrow) to the graph from the nodes corresponding to the variables on which the distribution is conditioned on

Graph layout depends on the choice of factorization
Joint distribution is the product of a set of conditionals, one for each node in the graph.

Each conditional is conditioned only on the parents of the corresponding node in the graph.

\[
p(x_1, x_2, x_3, x_4, x_5) = p(x_1) p(x_5) p(x_2|x_5) p(x_3|x_1, x_2) p(x_4|x_2)
\]

In general:
\[
p(x) = \prod_{k=1}^{K} p(x_k|\text{pa}_k)
\]
MRF → Factor Graph

1. Take variable nodes from MRF
2. Create additional factor nodes corresponding to the maximal cliques $x_s$
3. The factors $f_s(x_s)$ equal the clique potentials
4. Add appropriate links

Not unique
Directed Graph $\rightarrow$ MRF

- **Moralization:**
  - Add additional undirected links between all pairs of parents for each node in the graph
  - Drop arrows on original links
- Identify (maximum) cliques
- Initialize all clique potentials to 1
- Take each conditional distribution factor in the directed graph, multiply it into one of the clique potentials
Example: MRF → Factor Graph

- MRF with clique potential $\psi(x_1, x_2, x_3)$
- Factor graph with factor $f(x_1, x_2, x_3) = \psi(x_1, x_2, x_3)$
- Factor graph with factors, such that
  
  $f_a(x_1, x_2, x_3)f_b(x_2, x_3) = \psi(x_1, x_2, x_3)$
Directed Graphical Model $\rightarrow$ Factor Graph

1. Take variable nodes from Bayesian network
2. Create additional factor nodes corresponding to the conditional distributions
3. Add appropriate links

Not unique
Example: Directed Graph → Factor Graph

- Directed graph with factorization \( p(x_1)p(x_2)p(x_3|x_1, x_2) \)
- Factor graph with factor \( f(x_1, x_2, x_3) = p(x_1)p(x_2)p(x_3|x_1, x_2) \)
- Factor graph with factors \( f_a = p(x_1), f_b = p(x_2), f_c = p(x_3|x_1, x_2) \)
Removing Cycles

- Local cycles in an (un)directed graph (due to links connecting parents of a node) can be removed on conversion to a factor graph.
Sum-Product Algorithm for Factor Graphs

- Factor graphs give a uniform treatment to message passing
- Two different types of messages:
  - Messages $\mu_{x \rightarrow f}(x)$ from variable nodes to factors
  - Messages $\mu_{f \rightarrow x}(x)$ from factors to variable nodes
- Factors transform messages into evidence for the receiving node.
Variable-to-Factor Message

\[
\mu_{x_m \rightarrow f_s}(x_m) = \prod_{l \in \text{ne}(x_m) \setminus f_s} \mu_{f_l \rightarrow x_m}(x_m)
\]

- Take the product of all incoming messages along all other links.
- A variable node can send a message to a factor node once it has received messages from all other neighboring factors.
- The message that a node sends to a factor is made up of the messages that it receives from all other factors.
Factor-to-Variable Message

\[ \mu_{f_s \to x}(x) = \sum_{x_1} \cdots \sum_{x_M} f_s(x, x_1, \ldots, x_M) \prod_{m \in \text{ne}(f_s) \setminus x} \mu_{x_m \to f_s}(x_m) \]

- Take the product of the incoming messages along all other links coming into the factor node
- Multiply by the factor associated with that node
- Marginalize over all of the variables associated with the incoming messages
Initialization

- If the leaf node is a variable nodes, initialize the corresponding messages to 1:

\[ \mu_{x \rightarrow f}(x) = 1 \]

- If the leaf node is a factor node, the message should be

\[ \mu_{f \rightarrow x}(x) = f(x) \]
Example (1)

\[ \begin{align*}
\mu_{x_1 \rightarrow f_a}(x_1) &= 1 \\
\mu_{f_a \rightarrow x_2}(x_2) &= \sum_{x_1} f_a(x_1, x_2) \cdot 1 \\
\mu_{x_4 \rightarrow f_c}(x_4) &= 1 \\
\mu_{f_c \rightarrow x_2}(x_2) &= \sum_{x_4} f_c(x_2, x_4) \cdot 1 \\
\mu_{x_2 \rightarrow f_b}(x_2) &= \mu_{f_a \rightarrow x_2}(x_2) \mu_{f_c \rightarrow x_2}(x_2) \\
\mu_{f_b \rightarrow x_3}(x_3) &= \sum_{x_2} f_b(x_2, x_3) \mu_{x_2 \rightarrow f_b}(x_2)
\end{align*} \]

From PRML (Bishop, 2006)
Example (2)

From PRML (Bishop, 2006)

\[
\begin{align*}
\mu_{x_3 \rightarrow f_b}(x_3) &= 1 \\
\mu_{f_b \rightarrow x_2}(x_2) &= \sum_{x_3} f_b(x_2, x_3) \cdot 1 \\
\mu_{x_2 \rightarrow f_a}(x_2) &= \mu_{f_b \rightarrow x_2}(x_2) \mu_{f_c \rightarrow x_2}(x_2) \\
\mu_{f_a \rightarrow x_1}(x_1) &= \sum_{x_2} f_a(x_1, x_2) \mu_{x_2 \rightarrow f_a}(x_2) \\
\mu_{x_2 \rightarrow f_c}(x_2) &= \mu_{f_a \rightarrow x_2}(x_2) \mu_{f_b \rightarrow x_2}(x_2) \\
\mu_{f_c \rightarrow x_4}(x_4) &= \sum_{x_2} f_c(x_2, x_4) \mu_{x_2 \rightarrow f_c}(x_2)
\end{align*}
\]
For a single variable node the marginal is given as the product of all incoming messages:

\[ p(x) = \prod_{f_i \in \text{ne}(x)} \mu_{f_i \rightarrow x}(x) \]
References I


