MSA101/MVE187 2020 Lecture 10.1
Introduction to graphical models

Petter Mostad

Chalmers University

September 25, 2020

Graphical representations of conditional independencies

- In complex models with many variables, it is crucial to model and keep track of how variables depend on each other.
- Idea: Represent dependencies in a graph.
  - Helpful for visualization.
  - May use graph theory in connection with computations.
- We will look at two examples of graphical models:
  - Bayesian networks: Represent the probability density as a product of conditional densities:

$$\pi(x, y, z, v, w) = \pi(x \mid y, z) \cdot \pi(y \mid z) \cdot \pi(z \mid v, w) \cdot \pi(v) \cdot \pi(w)$$

  - Markov random fields: Represent the probability density as a product of factors:

$$\pi(x, y, z, v, w) = C \cdot f_1(x, y, z) \cdot f_2(y, z) \cdot f_3(z, v, w) \cdot f_4(v) \cdot f_5(w)$$

Bayesian networks

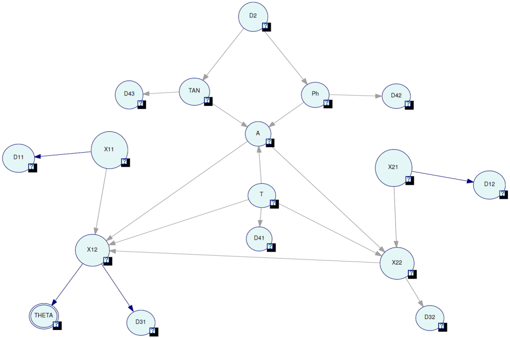
- Any joint density can be written as a product over conditional densities:

$$\pi(x_1, \ldots, x_n) = \pi(x_1)\pi(x_2 \mid x_1)\pi(x_3 \mid x_1, x_2) \ldots \pi(x_n \mid x_1, \ldots, x_{n-1})$$

- Given a specific model, we might be able to drop the conditioning on some of the variables in some factors. The representation then conveys the structure of the model.
- Re-ordering the variables will often give a different representation!
- The graph with an arrow $x \to y$ for each of the conditionings $\pi(y \mid \ldots x \ldots)$ in the representation above is the Bayesian Network representation. x is "parent", y is "child".
- Note that, following the arrows, you can never get a cycle. Thus the graph is a *directed acyclic graph* (DAG).
- Conversely, given any DAG and conditional densities for each child given its parents, the product of these gives a joint probability density.

Bayesian networks for visualization

- To the right: An example of a specific graphical network.
- Hierarchical models are, by definition, specified as a series of conditional distributions. The graph represents essential model information.
- Visualizations may use "plates" to represent repeated components.
- Note: Get a sample from the unconditional joint density by "propagating" simulation through network.



Conditional independence

- If x and y become independent when we fix the value of z we say that x and y are conditionally independent given z. We write $x \coprod y \mid z$.
- Equivalent formulations:
  - $\pi(x, y \mid z) = \pi(x \mid z)\pi(y \mid z)$
  - $\pi(x \mid y, z) = \pi(x \mid z)$
  - $\pi(y \mid x, z) = \pi(y \mid z)$
- We use the same definitions and notation when X, Y and Z are *disjoint groups of variables*.
- Example: When the data x_1, x_2, x_3 is *iid* given the parameter θ, we get for example $\{x_1, x_2\} \coprod x_3 \mid \theta$.

Reading off conditional independencies from a Bayesian network

- Some conditional independence statements can be “read off” the DAG of a Bayesian network.
- Note: Conditioning on children generally creates dependencies between their parents.
- Note: A “v-structure” is a part of a network consisting of a child with two parents.
- Is there a general way to prove that two sets of variables are conditionally independent given a third set based only on the Bayesian network graph?

d-separation

- A "trail" in a DAG is an *undirected path* in the graph.
- Assume X, Y, Z are sets of variables. An "active trail" from X to Y given Z is one where for every v-structure $x_{i-1} \rightarrow x_i \leftarrow x_{i+1}$ in the trail x_i or a decendant is in Z, and no other node in the trail is in Z.
- We say X and Y are *d-separated* given Z if there is no active trail between any $x \in X$ and $y \in Y$ given Z.
- Theorem: If X and Y are d-separated given Z in a Bayesian network representation of a stochastic model, then $X \coprod Y \mid Z$.
- Theorem: If X and Y are *not* d-separated given Z in a DAG, then there exists a stochastic model where X and Y are not conditionally independent given Z that has the DAG as a Bayesian network.
- See Koller & Friedman: "Probabilistic Graphical Models" for more details.

A way to check d-separation

- Note: The dependency between X and Y given Z is not changed if you remove from a network a child that is not i X, Y, or Z and has no children on its own.
- Doing this repeatedly will lead to a network where all nodes that do not have children are either in X, Y, or Z.
- Note: The dependency between X and Y given Z is not changed if you remove from the network the links from nodes in Z to their children.
- After you have done the two changes above, you can check whether X and Y are d-separated given Z simply by checking if there is an (undirected) path in the network from a node of X to a node of Y.

MSA101/MVE187 2020 Lecture 10.2
Markov networks

Petter Mostad

Chalmers University

September 26, 2020

Markov networks

- For many models, the probability (density) function may be written as a product of positive factors where each involves only a subset of the variables. Example:

$$\pi(x, y, z, v, w) = C \cdot f_1(x, y, z) \cdot f_2(y, z) \cdot f_3(z, v, w) \cdot f_4(v) \cdot f_5(w)$$

- Note: The f_i functions are *not* necessarily densities (i.e., do not necessarily integrate to 1).
- Assume the representation is maximally reduced, i.e., for any pair of variables x, y occuring in a factor, the factor cannot be written as a product of two factors where the first does not contain x and the second does not contain y.
- The corresponding Markov network contains an *undirected* edge between x and y for all nodes x and y occurring together in a factor.
- A Bayesian network may generally be converted into a Markov network using a process called *moralization*.

Conditional independence in Markov networks

- For a variable x, its *Markov blanket* Z is the set of variables directly connected to x in the Markov network representation.
- We then have $x \coprod Y \mid Z$ for any set Y of variables not containing x or Z.
- We define in the same way the Markov blanket of a set of variables X; the same conclusion about conditional independence holds.
- A way to specify a stochastic model on a set of variables is to construct a graph connecting the variables in some way and specify the conditional distribution of each variable given values of the variables it is connected to.
  - This is different from a Bayesian Network in that we might specify dependencies that go in opposite directions!
  - This does not necessarily result in a *proper* distribution!

Simulation in Markov networks using Gibbs sampling

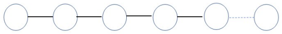
- With a Markov network representation of a posterior, we can set up a Gibbs sampling from the posterior by iteratively simulating from the conditional distribution of each node given its Markov blanket.
- Explicitly: Write down the joint density of all variables, and for each variable θ_i in sequence:
  - Regard all other variables as constants, throw away all factors not depending on θ_i.
  - Interpret the remaining function of θ_i as a standard density, or use it in some more advanced simulation method.
- Note: You need to check that the joint density is *proper*.
- We may simulate from a posterior represented as a Bayesian network by converting it to a Markov network (using moralization) and then simulate as above.
- Widely used programs like BUGS (WinBugs, OpenBugs), Jags (Just Another Gibbs Sampler), and **Stan** offer "black box" implementations of Gibbs sampling on wide classes of Bayesian Networks.

Gaussian Markov random fields (GMRF)

- A density $\pi(x_1, \dots, x_n)$ can be considered a GMRF if it can be written as
$$\pi(x_1, \dots, x_n) = \exp\left(-f(x_1, \dots, x_n)\right)$$
where $f(x_1, \dots, x_n)$ is a quadratic polynomial.
- We can then always re-write the density on $x = (x_1, \dots, x_n)$ so that
$$\pi(x) = \exp\left(-\frac{1}{2}(x-\mu)^t P(x-\mu) + C\right).$$
where μ is a vector, P is a symmetric matrix, and C is a constant.
- The density is *proper* if and only if P is *positive definite*. In this case we can re-write the density as
$$\pi(x) = \frac{1}{|2\pi P^{-1}|} \exp\left(-\frac{1}{2}(x-\mu)^t P(x-\mu)\right),$$
so that $x \sim \text{Normal}(\mu, P^{-1})$.
- In many cases it may be useful to consider the Markov network for the GMRF.

GMRF and precision matrices

- For a GMRF and two variables x_i and x_j, the following are equivalent:
  1. There is no line between x_i and x_j in the Markov network.
  2. In the term $a_{ij}x_ix_j$ in the quadratic polynomial f defining the density, we have $a_{ij} = 0$.
  3. In the precision matrix P, the ij-th entry p_{ij} is zero.
- Thus, we can read off the Markov network directly from the precision matrix: Its non-zero terms correspond to edges in the Markov network.
- Example: If P is zero everywhere except along the main diagonal and the diagonals closest to it (i.e., $p_{ij} = 0$ unless $|i - j| \leq 1$) then the Markov network looks like the graph below (with number of nodes corresponding to number of variables).



MSA101/MVE187 2020 Lecture 10.3
Inference for graphical models

Petter Mostad

Chalmers University

September 28, 2020

Inference for graphical models (BNs or Markov networks)

- Two types of inference:
  - Given a network, and given observed values for some variables, how can we make predictions for (or simulate from) some remaining variables using the conditional distribution?
  - Given observations for some variables, how do we find a graphical model for these variables from the data?
- For the first question, we have seen that Gibbs sampling is a good general (approximative) solution.
- However, for some models, exact solutions (not using Markov chain approximations) are possible. In particular when variables have a finite number of possible values.
- Below, we look briefly at exact inference for graphical models. BUT we mainly save this for the special case of *Hidden Markov Models* in the next lecture.
- Learning networks from data is often extremely difficult. Active area of research.

Exact posterior inference for graphical models

- We want to fix some variables (called *data*) and compute the posterior distribution of *some* other variables of interest.
- For a Markov network, fixing some variables produces directly another similar Markov network.
- A Bayesian Network may first be converted to a Markov network, using moralization.
- Then: A direct way to obtain the marginal distribution for the variables of interest in a Markov network is *variable elimination*:
  - Integrating (or summing) out variables in factors.
  - Multiplying together factors.
- Can lead to expression with an "explosion" in the number of terms in many cases, but the problem may be contained when variables have only a finite number of values.
- Any inference algorithm depends on the basic operations above, but they can be "scheduled" and organized in smart ways, using e.g. a "message passing" algorithms. See the "sum-product" algorithm in Bishop (not core course material).