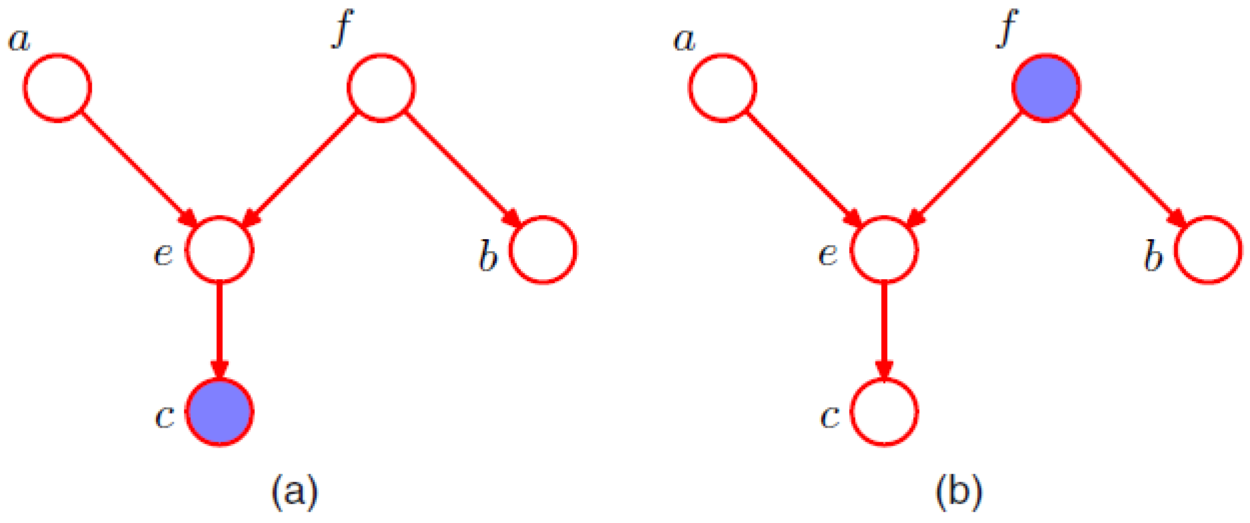
Week 2 Tutorial: Examples of Directed Graphical Models

Goal of this tutorial:

1. (Re)familiarize you with representing probability distribution as directed-graphical models **(DGM)**.
2. Go through examples of DGMs: definition, learning, and inference.

Examples = on Bayes Ball (PRML 8.2.2)

Recall the Bayes ball algorithm for verifying A ⊥ B | C:
(a) 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, or
(b) the arrows meet head-to-head at the node, and neither the node, nor any of its descendants, is in the set C.



1. What is the joint distribution factorization implied by the graph above (ignore the observed node for now)?
2. Is **a** independent from **b** when:
   a) conditioned on **c** ?
   b) conditioned on **f**?

If you find Bayes Ball confusing, try to use moralization instead.

Naive Bayes Model

Consider the inference problem of text classification into spam/not spam:
Let R.V. C denote whether a text is ($C = 1$) or isn't spam ($C = 0$).

Problem Setting

We'll use a "bag of words" representation for text:
Suppose we have dictionary of D words $\mathcal{D} = \{W_1, \ldots, W_D\}$ as an indexable set, a text x is a set of words in the dictionary, i.e. $x = \{W \in \mathcal{D}\}, x \subseteq \mathcal{D}$, which can be equivalently be represented as a set of indices $x' = \{i : W_i \in x\}$.
Fancy way of saying "apperance of word matters, repetition and order doesn't matter".

Example: $\mathcal{D} = \{\text{hello}, \text{world}, \text{test}, \text{is}, \text{this}, \text{a}\}$, $D = 6$

- "hello world" $\overset{x}{=} \{\text{hello}, \text{world}\} \overset{x'}{=} \{1, 2\}$
- "this is a test" $\overset{x}{=} \{\text{test}, \text{is}, \text{this}, \text{a}\} \overset{x'}{=} \{3, 4, 5, 6\}$
- "hello hello hello world" $\overset{x'}{=} \{1, 2\} =$ "hello world" = "world hello".

Let $\mathrm{X} = (X_1, \ldots, X_D)$, $X_i \in \{0, 1\}$ be a binary random vector denoting the appearance of i'th word in the text. (e.g. $\mathrm{X}(\text{hello world}) = (1, 1, 0, 0, 0, 0)$).

Our goal is to compute the posterior $p(C|\mathrm{X})$. Similar to lectures, we'll use p to mean probability mass function when its argument is discrete, and density function when its argument is continuous.

A general model

Using bayes theorem, we can write the posterior as:

$$p(C|\mathrm{X}) = \frac{p(C, \mathrm{X})}{p(\mathrm{X})}.$$

Since the denominator $p(X)$ does not depend on specific outcome of C, we have $p(C|\mathrm{X}) \propto p(C, \mathrm{X})$.
In general, we can further factorize $p(C, \mathrm{X})$ into its components with baye's rule:

$$p(C, \mathrm{X}) = p(C)p(\mathrm{X}|C) = p(C)p(X_1|C)p(X_2|X_1, C)\ldots p(X_d|X_1, \ldots X_{d-1}, C)$$
$$= p(C)p(X_1|C)\Pi_{i=2}^{d} p(X_i|X_1, \ldots, X_{i-1}, C).$$

How would this factorization appear as a DGM? Since we have ordered the terms above such that each term is only conditioned on variables that have appeareed to its left, we can draw the graphical model accordingly:

“../img/general_fac.png” could not be found.

Few observations:

- This graph has $d+1$ nodes (X_1 to X_d, and C).
- The degree of each node is the same and equal to d - thus, this graph is fully connected! Every node is a neighbour of every other node.
- For node X_i, # of input edges = i.
- Size of conditional probability table of each node $= 2^{\#\text{ input edges}+1}$, which requires $2^{\#\text{ input edges}}$ parameters.
- Total # of parameters: $1+\sum_{i=1}^{d} 2^i = 1 + (2^{d+1}-2) = 2^{d+1}-1$ parameters, which is equal to the number of parameters needed to specify the joint tensor over $d+1$ binary random variables - this factorization is indeed general

Reducing complexity with Naive Bayes model

Learning $2^{d+1}-1$ parameters is very expensive (computationally and learning-theoretically).
Goal: Reduce parameter through simplifying the graphical model.
Method: Remove all edges between (X_i, X_j), only keep edges originating from C.

"../img/naive_bayes.png" could not be found.

What factorization does this model imply?

$$p(\mathrm{X}, C) = p(C)\Pi_{i=1}^{d} p(X_i|C)$$

i.e. $p(X_i|X_1, \ldots X_{i-1}, C) = p(X_i|C)$: X_i is independent from X_j for all $j \neq i$ given C. **We can manipulate the joint distribution through manipulating the DGM!**
Number of parameters: $1+2d$; complexity scale linearly instead of exponentially.

Learning the Naive Bayes model with MLE

Parameterize the model as follow: $p(C=1)=\pi$, $p(X_j=1|C)=\theta_{j,c}$.
Suppose we have N texts $x^i \in \{0,1\}^D$ with labels c^i, and wish to learn the parameters. We will use maximum likelihood estimation as done in CSC311.

1. Factorize the log likelihood function:

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{N} \log p(c^{(i)}, \mathbf{x}^{(i)}) = \sum_{i=1}^{N} \log \left\{ p(\mathbf{x}^{(i)} | c^{(i)}) p(c^{(i)}) \right\}$$

$$= \sum_{i=1}^{N} \log \left\{ p(c^{(i)}) \prod_{j=1}^{D} p(x_j^{(i)} \,|\, c^{(i)}) \right\}$$

$$= \sum_{i=1}^{N} \left[\log p(c^{(i)}) + \sum_{j=1}^{D} \log p(x_j^{(i)} \,|\, c^{(i)}) \right]$$

$$= \underbrace{\sum_{i=1}^{N} \log p(c^{(i)})}_{\substack{\text{Bernoulli log-likelihood} \\ \text{of labels}}} + \sum_{j=1}^{D} \underbrace{\sum_{i=1}^{N} \log p(x_j^{(i)} \,|\, c^{(i)})}_{\substack{\text{Bernoulli log-likelihood} \\ \text{for feature } x_j}}$$

2. Derive the first term:

$$p(c^{(i)}) = \pi^{c^{(i)}} (1 - \pi)^{1 - c^{(i)}}$$

$$\sum_{i=1}^{N} \log p(c^{(i)}) = \sum_{i=1}^{N} c^{(i)} \log \pi + \sum_{i=1}^{N} (1 - c^{(i)}) \log(1 - \pi)$$

3. Factorize and derive the second term:

$$p(x_j^{(i)} \,|\, c) = \theta_{jc}^{x_j^{(i)}} (1 - \theta_{jc})^{1 - x_j^{(i)}}$$

$$\sum_{i=1}^{N} \log p(x_j^{(i)} \,|\, c^{(i)}) = \sum_{i=1}^{N} c^{(i)} \left\{ x_j^{(i)} \log \theta_{j1} + (1 - x_j^{(i)}) \log(1 - \theta_{j1}) \right\}$$

$$+ \sum_{i=1}^{N} (1 - c^{(i)}) \left\{ x_j^{(i)} \log \theta_{j0} + (1 - x_j^{(i)}) \log(1 - \theta_{j0} \right.$$

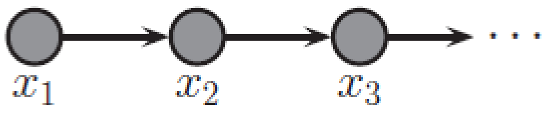
4. Set derivative to zero and solve:

$$\hat{\pi} = \frac{\sum_i \mathbb{1}[c^{(i)} = 1]}{N} = \frac{\#\text{ spams in dataset}}{\text{total }\#\text{ samples}}$$

$$\hat{\theta}_{jc} = \frac{\sum_i \mathbb{I}[x_j^{(i)} = 1 \ \& \ c^{(i)} = c]}{\sum_i \mathbb{I}[c^{(i)} = c]} \overset{\text{for } c = 1}{=} \frac{\#\text{word } j \text{ appears in spams}}{\# \text{ spams in dataset}}$$

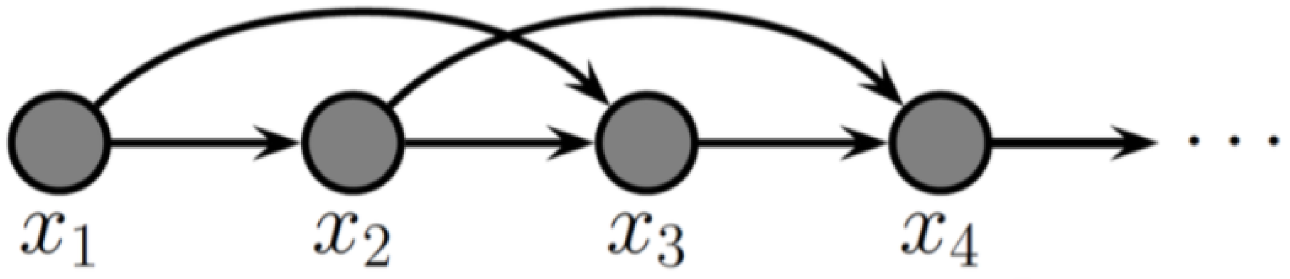
Markov chains

In lecture, you have seen a first order Markov chain. The "order" of Markov chain refers to the number of previous states that the current state could depend on.



$$p(X_{1:T}) = p(X_1)p(X_2|X_1)p(X_3|X_2)\ldots = p(X_1)\Pi_{t=2}^{T} p(X_t|X_{t-1})$$

Second order Markov chain:

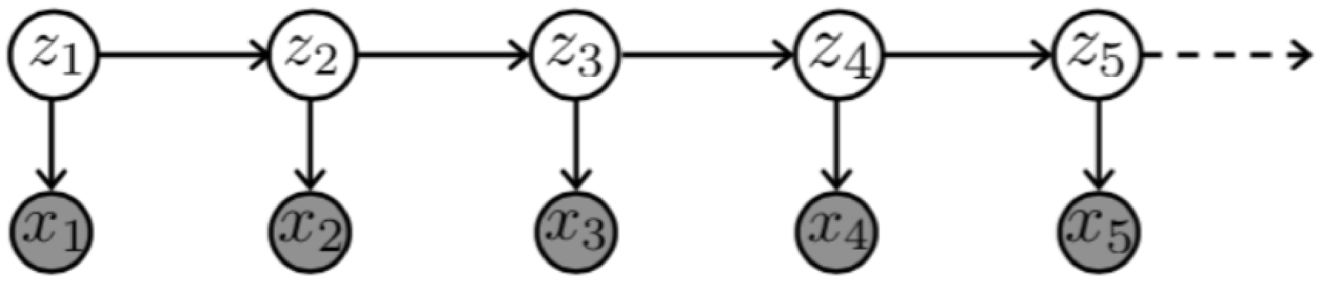


$$p(\mathbf{x}_{1:T}) = p(x_1, x_2)p(x_3|x_1, x_2)p(x_4|x_2, x_3)\ldots = p(x_1, x_2)\prod_{t=3}^{T} p(x_t|x_{t-1}, x_{t-2})$$

The earlier images depicts a *first*-order Markov chain, this is a *second*-order Markov chain.

Hidden Markov Models (HMMs)

Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (i.e. hidden) states. It is a very popular type of latent variable model



where

- Z_t are *hidden states* taking on one of K discrete values
- X_t are *observed variables* taking on values in any space

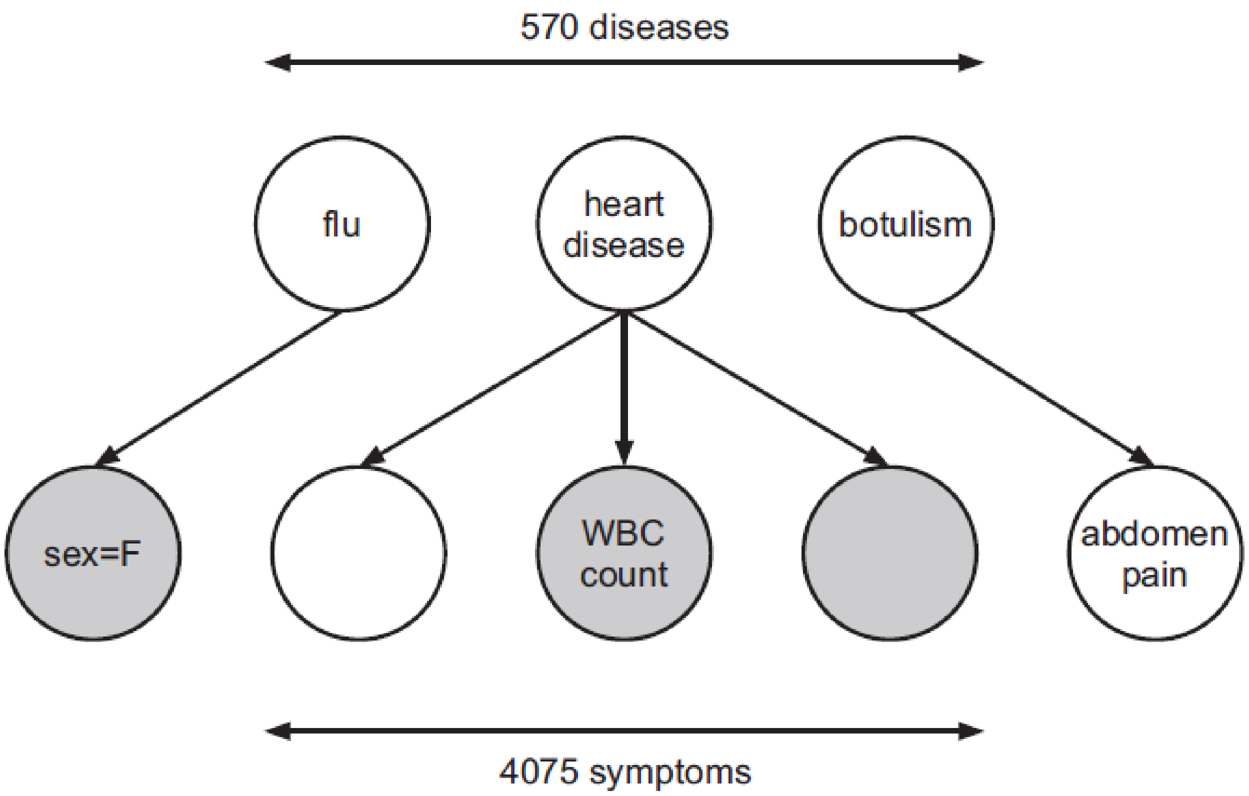
the joint probability represented by the graph factorizes according to

$$p(X_{1:T}, Z_{1:T}) = p(Z_{1:T})p(X_{1:T}|Z_{1:T}) = p(Z_1)\prod_{t=2}^{T} p(Z_t|Z_{t-1})\prod_{t=1}^{T} p(X_t|Z_t)$$

Medical diagnosis

In the models above, we designed generic DGMs based on (usually wrong, but useful) assumptions. Next, we will see examples where the models are designed based on domain knowledge (PPML 10.2.3).

Quick Medical Reference has a bipartite structure, with diseases as hidden nodes, and symptom and other observables as visible nodes. All nodes are binary



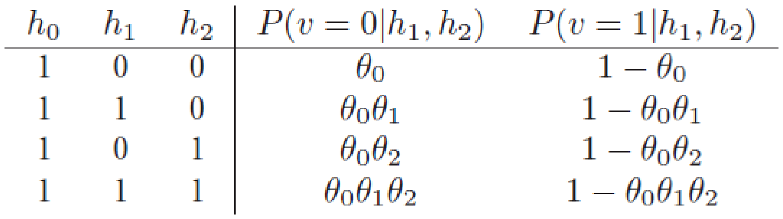
Let $\mathbf{h} = \{h_s\}_{s=0}^{570}$ denote the hidden nodes, and $\mathbf{v} = \{v_t\}_{t=0}^{4075}$ denote the visible variables. Their joint distribution is can be factorized as:

$$p(\mathbf{v}, \mathbf{h}) = \prod_s p(h_s) \prod_t p(v_t|\mathbf{h}_{\mathrm{pa}(t)})$$

The conditional probability of the symptoms $p(v_t|h_{pa(t)})$ follow a **noisy OR** model - if any parent of v_t is positive, then v_t is also likely to be positive.
More precisely:

$$p(v_t = 0|\mathbf{h}) = \prod_{s \in \text{pa}(t)} \theta_{st}^{\mathbb{I}(h_s=1)}$$

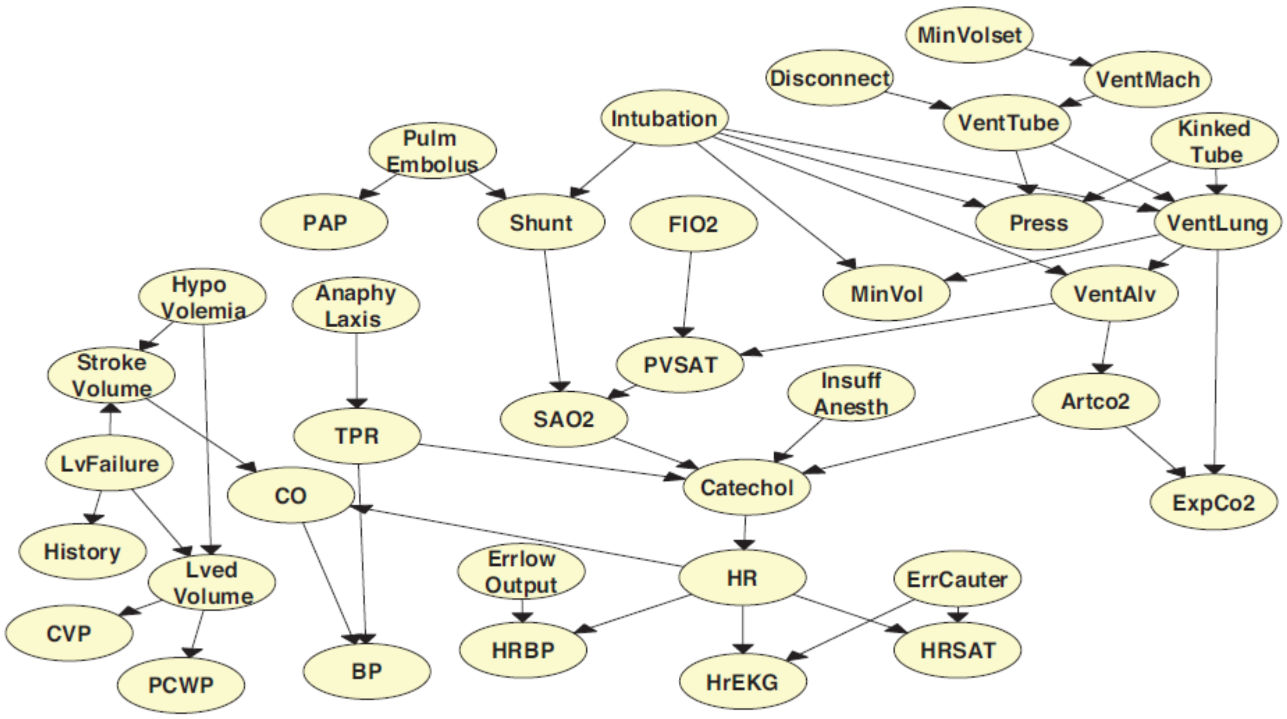
Where $\theta_{st} = p(v_t = 0|h_s = 1, h_{/s} = 0)$. One way to visualize this is to take a coin flip with head probability of θ_{st} for each disease that is positive, and if all of the coins are heads, then the symptom will be negative. If any coin flipped tails, then the symptom will be positive.
A "dummy" node h_0 is added to represent all "unknown diseases" and is always set to 1. This allows the model to give non-zero probability to patients who have symptoms but no included diseases.



h_0	h_1	h_2	$P(v=0\|h_1,h_2)$	$P(v=1\|h_1,h_2)$
1	0	0	θ_0	$1-\theta_0$
1	1	0	$\theta_0\theta_1$	$1-\theta_0\theta_1$
1	0	1	$\theta_0\theta_2$	$1-\theta_0\theta_2$
1	1	1	$\theta_0\theta_1\theta_2$	$1-\theta_0\theta_1\theta_2$

Table 10.1 Noisy-OR CPD for 2 parents augmented with leak node. We have omitted the t subscript for brevity.

The **Alarm Network**, with 37 random variables relating to vital signs, conditions, and symptoms, was designed to monitor ICU patients. Each random variable is discrete, with up to 4 states. Since the graph is sparsely connected, the total number of parameters in the graph is only 504 (much less than 2^37-1). It is small enough to allow inference of marginal distributions of unobserved nodes when conditioned on sufficient observed nodes.
*You wil see algorithms that perform this inference later in the course.



The connections in this graph are made based on domain knowledge - causal relations that are known in medicine. For instance, Hypovolemia is a low level of extracellular fluid. The extracellular fluid is fluid thats drained from the blood into body tissue in the capillaries. They traverse the lymphatic system, which carries these fluid back into the blood stream through the superior vena cava. Reduction in this fluid volume can reduce volume of blood reaching the heart, which decreases stroke volume. The stroke volume, multiplied with the heart rate, determines cardia output, which in turn determines blood pressure.

Optional reading

I got a bunch of questions about the proof of correctness of the Bayes Ball and the moralization algorithm. For the proof that the moralization algorithm works, check Proposition 5.13 in this book

Cowell, R. G., Dawid, P., Lauritzen, S. L., & Spiegelhalter, D. J. (2007). *Probabilistic networks and expert systems: Exact computational methods for Bayesian networks*. Springer Science & Business Media.

For the Bayes Ball algorithm you can refer to the original paper