Tutorial 2

CSC412: Probabilistic Machine Learning

Stephan Rabanser stephan@cs.toronto.edu

UNIVERSITY OF TORONTO

VECTOR INSTITUTE

January 16, 2025

Goal of This Tutorial

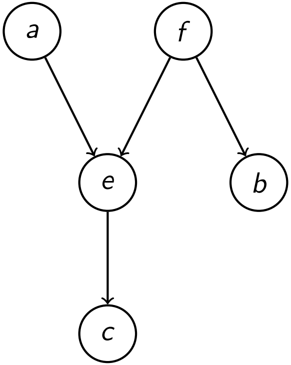
Goals:

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

Topics Overview:

- Bayes Ball
- Naive Bayes
- Markov Chains
- Hidden Markov Models (HMMs)
- Medical Diagnosis Example

Directed Acyclic Graphs and Joint Probability



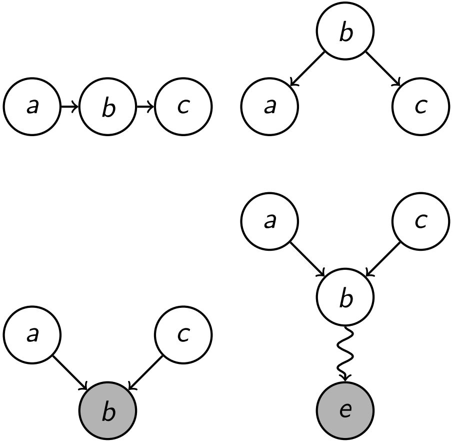
Let's first remember how to write down the joint probability distribution of this DAG. How do we do that?

$$p(a, b, c, e, f) = \prod_{n \in \{a,b,c,e,f\}} p(n|\text{parents}(n))$$

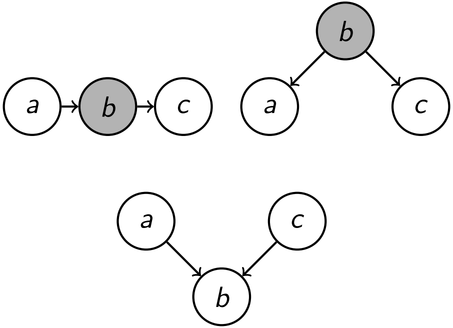
$$p(a, b, c, e, f) = p(a)p(f)p(e|a, f)p(b|f)p(c|e)$$

Bayes Ball

Active Triples



Inactive Triples



A path is active if each triple in a path is active. If there are no active paths → independence.

d-separation and Bayes Ball Reminder

Two sets of variables X and Y are said to be conditionally independent given a third set Z *if and only if* there is *no active path* from any node in X to any node in Y once we account for the conditioning on Z. Formally:

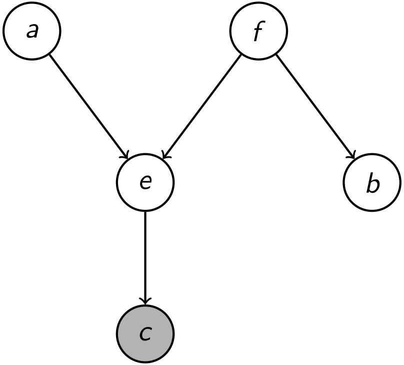
$$X \perp Y \mid Z \quad \Longleftrightarrow \quad \text{There is no active path between } X \text{ and } Y \text{ given } Z.$$

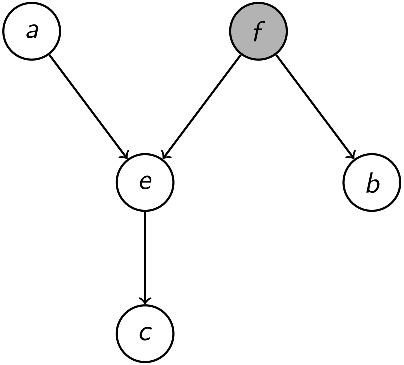
Terminology Recap:

1. **Active path**: A path that remains open under the conditioning set Z. In the Bayes Ball perspective, one can think of the ball being able to roll from a node in X to a node in Y without being blocked by the conditioning or collider structures.
2. **d-separation**: A graphical criterion that tells us when X and Y are independent given Z. If there is no path open (active) between X and Y once we consider the effect of Z, then we say $X \perp Y \mid Z$.
3. **Bayes Ball**: An algorithmic way to check d-separation by rolling a ball in the directed graph and seeing if it can make it from X to Y once the conditioning set Z is taken into account.

Bayes Ball Example

Is a independent from b?





Naive Bayes

- 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 $(C = 0)$.
- We'll use a "bag of words" representation for text:
  - Suppose we have a dictionary of d words $\mathcal{D} = \{W_1, \dots, W_d\}$ as an indexable set, a text x is a set of words in the dictionary, i.e., $x \subseteq \mathcal{D}$, which can be equivalently be represented as a set of indices $x' = \{i : W_i \in x\}$.
  - *Fancy way of saying: "Appearance of word matters, repetition and order doesn't."*
- **Example:** $\mathcal{D} = \{\text{hello, world, test, is, this, a}\}$,
  - "hello world" $x = \{\text{hello, world}\}$ $x' = \{1, 2\}$
  - "this is a test" $x = \{\text{test, is, this, a}\}$ $x' = \{3, 4, 5, 6\}$
  - "hello hello hello world" $x' = \{1, 2\}$ = "hello world" = "world hello"

Naive Bayes: A General Model

- Let $\mathbf{X} = (X_1, \ldots, X_d)$, $X_i \in \{0, 1\}$ be a binary random vector denoting the appearance of the ith word in the text (e.g., $\mathbf{X}(\text{hello world}) = (1, 1, 0, 0, 0, 0)$).
- Our goal is to compute the posterior $p(C|\mathbf{X})$.
- Using Bayes' theorem, we can write the posterior as:

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

- Since the denominator $p(\mathbf{X})$ does not depend on specific outcome of C, we have

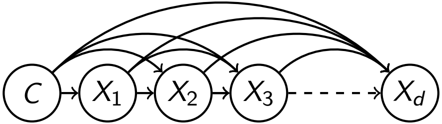
$$p(C|\mathbf{X}) \propto p(C, \mathbf{X}).$$

- In general, we can further factorize $p(C, \mathbf{X})$ into its components with Bayes' rule:

$$\begin{aligned} p(C, \mathbf{X}) = p(C)p(\mathbf{X}|C) &= p(C)p(X_1|C)p(X_2|X_1, C) \ldots p(X_d|X_1, \ldots, X_{d-1}, C) \\ &= p(C)p(X_1|C)\prod_{i=2}^{d} p(X_i|X_1, \ldots, X_{i-1}, C). \end{aligned}$$

Naive Bayes: A General Model

$$p(C)p(X_1|C)\prod_{i=2}^{d} p(X_i|X_1,\ldots,X_{i-1},C)$$



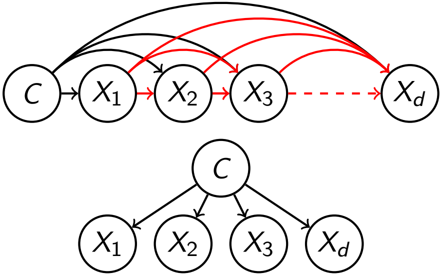
Few Observations:

- 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 the conditional probability table of each node $= 2^{\#\text{input edges}+1}$, which requires $2^{\#\text{input edges}}$ parameters.
- Total $\#$ of parameters: $1+\sum_{i=1}^{d} 2^i = 1+(2^{d+1}-2) = 2^{d+1}-1$, 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.

Naive Bayes: Reducing Complexity

Learning $2^{d+1} - 1$ parameters is very expensive (comp & lear-theo).

Goal: Reduce parameters through simplifying the graphical model.
Method: Remove all edges between features, only keep edges from C.



What factorization does this imply?

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

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 scales linearly, not exponentially.

Naive Bayes: Learning with Maximum Likelihood Estimation

Parameterize the model as follows: $p(C = 1) = \pi$, $p(X_j = 1|C = c) = \theta_{j,c}$.

Suppose we have N texts x^i with labels c^i, and wish to learn the parameters.

1. Factorize the log-likelihood function:

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

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

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

Naive Bayes: Learning with Maximum Likelihood Estimation

2. Derive the first term:

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

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

Naive Bayes: Learning with Maximum Likelihood Estimation

3. Factorize and derive the second term:

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

$$\begin{aligned}\sum_{i=1}^{N} \log p(x_j^{(i)} \mid c^{(i)}) &= \sum_{i=1}^{N} c^{(i)} \left\{ x_j^{(i)} \log \theta_{j,1} + (1 - x_j^{(i)}) \log(1 - \theta_{j,1}) \right\} \\ &\quad + \sum_{i=1}^{N} (1 - c^{(i)}) \left\{ x_j^{(i)} \log \theta_{j,0} + (1 - x_j^{(i)}) \log(1 - \theta_{j,0}) \right\}\end{aligned}$$

Naive Bayes: Learning with Maximum Likelihood Estimation

4. Set derivative to zero and solve:

$$\frac{\partial \ell(\theta)}{\partial \pi} = \sum \frac{c^{(i)}}{\pi} + \sum \frac{1 - c^{(i)}}{\pi - 1} \overset{!}{=} 0,$$

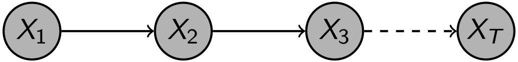
$$\frac{\partial \ell(\theta)}{\partial \theta_{jc}} = \sum c^{(1)} \left\{ \frac{x_j^{(i)}}{\theta_{jc}} + \frac{1 - x_j^{(i)}}{\theta_{jc} - 1} \right\} \overset{!}{=} 0$$

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

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

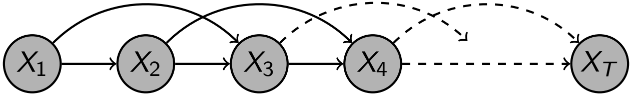
Markov Chains

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



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

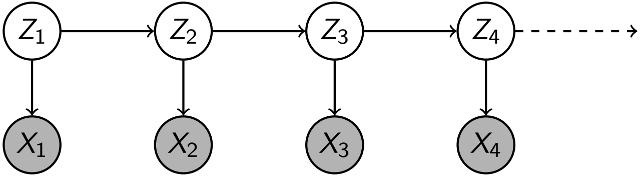
Second-order Markov chain:



$$p(X_{1:T}) = p(X_1, X_2)p(X_3 \mid X_1, X_2)p(X_4 \mid X_2, X_3)\ldots = p(X_1, X_2)\prod_{t=3}^{T} p(X_t \mid X_{t-1}, X_{t-2})$$

Hidden Markov Model

A **Hidden Markov Model (HMM)** is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved states.



Where:

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

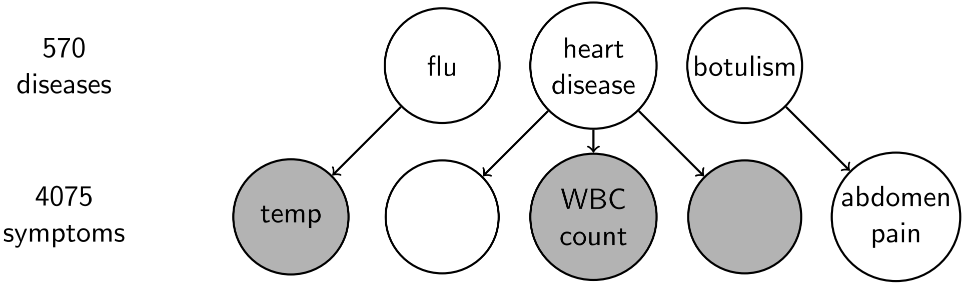
The joint probability represented by the graph factorizes according to:

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

Medical Diagnosis Example

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 symptoms and other observables as visible nodes. All nodes are binary.



Medical Diagnosis Example

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 can be factorized as:

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

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

$$p(v_t = 0 \mid h_{\mathrm{pa}(t)}) = \prod_{s \in \mathrm{pa}(t)} \theta_{st}^{\mathbb{I}[h_s = 1]}$$

where $\theta_{st} = p(v_t = 0 \mid h_s = 1, h_{/s} = 0)$. 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.

Medical Diagnosis Example

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

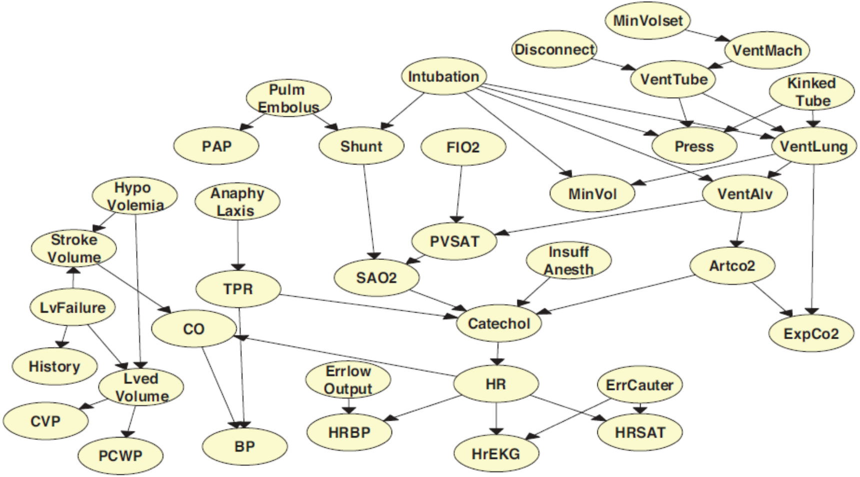
Table 10.1: Noisy-OR CPD for 2 parents augmented with a leak node. We have omitted the t subscript for brevity.

Medical Diagnosis Example

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 will see algorithms that perform this inference later in the course.

Medical Diagnosis Example



Medical Diagnosis Example

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.