CSE 150A-250A AI: Probabilistic Models

Lecture 8

Fall 2025

Trevor Bonjour
Department of Computer Science and Engineering
University of California, San Diego

Slides adapted from previous versions of the course (Prof. Lawrence, Prof. Alvarado, Prof Berg-Kirkpatrick)

Agenda

Review

Learning in BNs

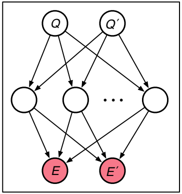
Markov models

Naive Bayes models

Review

MCMC - Gibbs Sampling

- **Initialization**
  Fix evidence nodes to observed values e, e'.
  Initialize non-evidence nodes to random values.



- **Repeat** N **times**
  Pick a non-evidence node X at random.
  Use **Bayes rule** to compute $P(X|B_X)$.
  Resample $x \sim P(X|B_X)$.
  Take a snapshot of all the nodes in the BN.

- **Estimate**
  Count the snapshots $N(q, q') \leq N$ with $Q = q$ and $Q' = q'$.

$$P(Q = q, Q' = q' | E = e, E' = e') \approx \frac{N(q, q')}{N}$$

Properties of MCMC

Under reasonable conditions...

1. This sampling procedure defines an ergodic (**irreducibile** and **aperiodic**) Markov chain over the non-evidence nodes of the BN.
2. The stationary distribution of this Markov chain is equal to the BN's posterior distribution over its non-evidence nodes.
3. Theoretical guarantees for mixing time, in practice we use burn in time.
4. The estimates from MCMC converge in the limit:

$$\lim_{N \to \infty} \frac{N(q, q')}{N} \rightarrow P(Q = q, Q' = q' | E = e, E' = e')$$

MCMC versus likelihood weighting (LW)

- How they sample

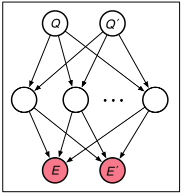
$$\left.\begin{array}{r}\text{LW}\\ \text{MCMC}\end{array}\right\}\ \text{samples non-evidence nodes from}\ \left\{\begin{array}{l}P(X|\mathrm{pa}(X))\\ P(X|B_X)\end{array}\right.$$

- Cost per sample

LW can read off $P(X|\mathrm{pa}(X))$ from each CPT.
MCMC must compute $P(X|B_X)$ before each sample.

- Convergence

LW is slow for rare evidence in leaf nodes.
MCMC can be much faster in this situation.



Learning in BNs

Learning in BNs

- **Where do BNs come from?**

  Sometimes an expert can provide the DAG and CPTs.
  But not always — especially not in very complex domains.

- **What is the alternative?**

  With sufficient data, we can estimate useful models.
  This is the central idea of *machine learning*.

- **What are some applications?**

  Language modeling
  Visual object recognition
  Recommender systems

Maximum likelihood (ML) estimation

- **Here's a simple idea:**

  Model data by the BN that assigns it the highest probability.
  In other words, choose the DAG and CPTs to **maximize**

  $$P(\text{observed data} \mid \text{DAG \& CPTs}).$$

  This probability is known as the **likelihood**.

- **But is this too simple?**

  The data may be unrepresentative or too limited.
  This is one failure mode of ML estimation.

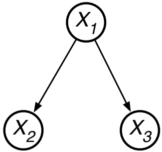
Learning with complete data and tabular CPTs

ASSUMPTIONS

1. The DAG is fixed (and known) over a finite set of discrete random variables $\{X_1, X_2, \ldots, X_n\}$.
2. The data consists of T complete (or fully observed) instantiations of all the nodes in the BN.
3. CPTs enumerate $P(X_i = x | \mathrm{pa}(X_i) = \pi)$ as lookup tables; each must be estimated for all values of x and π.

Example

- Fixed DAG over discrete random variables



$$
\begin{aligned}
X_1 &\in \{1,2,3\} \\
X_2 &\in \{1,2,3,4\} \\
X_3 &\in \{1,2,3,4,5\}
\end{aligned}
$$

- Data set

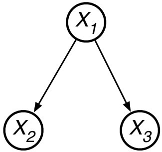
example	x_1	x_2	x_3
1	1	4	5
2	3	2	4
3	2	1	3
⋮	⋮	⋮	⋮
T	1	3	2

Note that if T is sufficiently large, some rows are destined to repeat.

We can also denote the data set as $\left\{\left(x_1^{(t)}, x_2^{(t)}, x_3^{(t)}\right)\right\}_{t=1}^{T}$.

Example

- Fixed DAG over discrete random variables



$$X_1 \in \{1, 2, 3\}$$
$$X_2 \in \{1, 2, 3, 4\}$$
$$X_3 \in \{1, 2, 3, 4, 5\}$$

- Data set

example	x_1	x_2	x_3
1	1	4	5
2	3	2	4
3	2	1	3
⋮	⋮	⋮	⋮
T	1	3	2

How to choose the CPTs so that the BN maximizes the probability of this data set?

ML estimation

- **IID assumption**

  The examples are assumed to be *independent and identically distributed* (IID) from the joint distribution of the BN.

- **Probability of IID data**

$$P(data) \;=\; \prod_{t=1}^{T} P\left(X_1 = x_1^{(t)}, X_2 = x_2^{(t)}, \ldots, X_n = x_n^{(t)}\right)$$

- **Probability of t^{th} example**

$$\begin{aligned} & P\left(X_1 = x_1^{(t)}, X_2 = x_2^{(t)}, \ldots, X_n = x_n^{(t)}\right) \\ &= \prod_{i=1}^{n} P\left(X_i = x_i^{(t)} \,\middle|\, X_1 = x_1^{(t)}, \ldots, X_{i-1} = x_{i-1}^{(t)}\right) \quad \boxed{\text{product rule}} \\ &= \prod_{i=1}^{n} P\left(X_i = x_i^{(t)} \,\middle|\, \mathrm{pa}(X_i) = \mathrm{pa}_i^{(t)}\right) \quad \boxed{\text{conditional independence}} \end{aligned}$$

Computing the log-likelihood

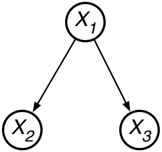
$$\begin{aligned}
\mathcal{L} &= \log P(\mathrm{data}) \\
&= \log \prod_{t=1}^{T} P\left(x_1^{(t)}, x_2^{(t)}, \ldots, x_n^{(t)}\right) && \boxed{\text{IID}} \\
&= \log \prod_{t=1}^{T} \prod_{i=1}^{n} P\left(x_i^{(t)} \middle| \mathrm{pa}_i^{(t)}\right) && \boxed{\text{product rule \& CI}} \\
&= \sum_{t=1}^{T} \sum_{i=1}^{n} \log P\left(x_i^{(t)} \middle| \mathrm{pa}_i^{(t)}\right) && \boxed{\log pq = \log p + \log q} \\
&= \sum_{i=1}^{n} \underbrace{\sum_{t=1}^{T} \log P\left(x_i^{(t)} \middle| \mathrm{pa}_i^{(t)}\right)}_{\text{sum over examples}} && \boxed{\text{sums can be reordered}}
\end{aligned}$$

Counting co-occurrences

- Counts

  Let $\text{count}(X_i = x, \text{pa}_i = \pi)$ denote the number of examples where $X_i = x$ and $\text{pa}_i = \pi$.

- Example



X_1	X_2	X_3
1	4	5
3	2	4
2	1	3
2	1	4
1	3	5
1	3	2

$$\begin{aligned} \text{count}(X_1 = 1) &= 3 \\ \text{count}(X_1 = 2) &= 2 \\ \text{count}(X_1 = 3) &= 1 \\ \text{count}(X_2 = 1, X_1 = 2) &= 2 \\ \text{count}(X_2 = 3, X_1 = 1) &= 2 \\ &\vdots \\ \text{count}(X_3 = 5, X_1 = 1) &= 2 \end{aligned}$$

Note: these counts can be compiled in one pass through the data set.

Computing the log-likelihood

Next: replace the unweighted sum over examples at each node by a weighted sum over its values and those of its parents.

$$\mathcal{L} = \sum_{i=1}^{n} \sum_{t=1}^{T} \log P\left(x_i^{(t)} \middle| \mathrm{pa}_i^{(t)}\right) \quad \boxed{\text{unweighted}}$$

$$= \sum_{i=1}^{n} \sum_{x} \sum_{\pi} \mathrm{count}(X_i = x, \mathrm{pa}_i = \pi) \log P(X_i = x | \mathrm{pa}_i = \pi)$$

$$\boxed{\text{weighted}}$$

These two expressions compute the exact same sum!

But the latter has a much more appealing form ...

Interpreting the log-likelihood

$$\mathcal{L} = \sum_i \sum_x \sum_\pi \overbrace{\mathrm{count}(X_i = x, \mathrm{pa}_i = \pi)}^{\text{constants of the data}} \underbrace{\log P(X_i = x | \mathrm{pa}_i = \pi)}_{\text{CPTs to optimize}}$$

- The log-likelihood for complete data is a triple sum over

  i — the nodes in the BN
  x — the values of each node X_i
  π — the values π of the parents of X_i

- How to optimize?

  Intuitively, the larger the $\mathrm{count}(X_i = x, \mathrm{pa}_i = \pi)$,
  the larger we should choose $P(X_i = x | \mathrm{pa}_i = \pi)$.

Decomposing the log-likelihood

- Log-likelihood for BN

$$\mathcal{L} = \sum_i \sum_\pi \sum_x \mathrm{count}(X_i = x, \mathrm{pa}_i = \pi) \log P(X_i = x | \mathrm{pa}_i = \pi)$$

- Contribution from row π of i^{th} node's CPT

$$\mathcal{L}_{i\pi} = \sum_x \mathrm{count}(X_i = x, \mathrm{pa}_i = \pi) \log P(X_i = x | \mathrm{pa}_i = \pi)$$

- Divide and conquer

  The overall optimization over $\mathcal{L}$ reduces to many simpler and smaller optimizations over each $\mathcal{L}_{i\pi}$.

This is a special property of ML estimation for complete data.

ML Estimation

- Problem

For each node X_i in the BN, and for each row π of its CPT, our goal is to maximize

$$\mathcal{L}_{i\pi} \;=\; \sum_x \mathrm{count}(X_i\!=\!x, \mathrm{pa}_i\!=\!\pi)\, \log P(X_i\!=\!x|\mathrm{pa}_i\!=\!\pi)$$

subject to two constraints:

1. $\sum_x P(X_i\!=\!x|\mathrm{pa}_i\!=\!\pi) = 1$ *(normalized)*
2. $P(X_i\!=\!x|\mathrm{pa}_i\!=\!\pi) \geq 0$ *(nonnegative)*

- Shorthand

$$\begin{aligned} C_\alpha &= \mathrm{count}(X_i\!=\!\alpha, \mathrm{pa}_i\!=\!\pi) \\ p_\alpha &= P(X_i\!=\!\alpha|\mathrm{pa}_i\!=\!\pi) \end{aligned}$$

ML Estimation

- **Problem**

  For each node X_i in the BN, and for each row π of its CPT, our goal is to maximize

$$\mathcal{L}_{i\pi} = \sum_x \mathrm{count}(X_i = x, \mathrm{pa}_i = \pi) \log P(X_i = x | \mathrm{pa}_i = \pi)$$

  subject to two constraints:

  1. $\sum_x P(X_i = x | \mathrm{pa}_i = \pi) = 1$ *(normalized)*
  2. $P(X_i = x | \mathrm{pa}_i = \pi) \geq 0$ *(nonnegative)*

- **Shorthand**

$$C_\alpha = \mathrm{count}(X_i = \alpha, \mathrm{pa}_i = \pi)$$
$$p_\alpha = P(X_i = \alpha | \mathrm{pa}_i = \pi)$$

$\Longrightarrow$

How to maximize $\sum_\alpha C_\alpha \log p_\alpha$ such that $\sum_\alpha p_\alpha = 1$ and $p_\alpha \geq 0$?

Maximizing the likelihood

- **Compute the normalized counts:**

  Define $q_\alpha = \frac{C_\alpha}{\sum_\beta C_\beta}$ so that $\sum_\alpha q_\alpha = 1$.

  Note that q_α is itself a distribution.

- **All these problems have the same solution:**

Maximize	$\sum_\alpha C_\alpha \log p_\alpha$	such that	$\sum_\alpha p_\alpha = 1,\ p_\alpha \geq 0.$
Minimize	$\sum_\alpha C_\alpha \log \frac{1}{p_\alpha}$	such that	$\sum_\alpha p_\alpha = 1,\ p_\alpha \geq 0.$
Minimize	$\sum_\alpha C_\alpha \log \frac{C_\alpha}{p_\alpha}$	such that	$\sum_\alpha p_\alpha = 1,\ p_\alpha \geq 0.$
Minimize	$\sum_\alpha q_\alpha \log \frac{q_\alpha}{p_\alpha}$	such that	$\sum_\alpha p_\alpha = 1,\ p_\alpha \geq 0.$

Maximizing the likelihood

- **Compute the normalized counts:**

  Define $q_\alpha = \frac{C_\alpha}{\sum_\beta C_\beta}$ so that $\sum_\alpha q_\alpha = 1$.

  Note that q_α is itself a distribution.

- **All these problems have the same solution:**

Maximize	$\sum_\alpha C_\alpha \log p_\alpha$	such that	$\sum_\alpha p_\alpha = 1,\ p_\alpha \geq 0.$
Minimize	$\sum_\alpha C_\alpha \log \frac{1}{p_\alpha}$	such that	$\sum_\alpha p_\alpha = 1,\ p_\alpha \geq 0.$
Minimize	$\sum_\alpha C_\alpha \log \frac{C_\alpha}{p_\alpha}$	such that	$\sum_\alpha p_\alpha = 1,\ p_\alpha \geq 0.$
Minimize	$\underbrace{\sum_\alpha q_\alpha \log \frac{q_\alpha}{p_\alpha}}_{\mathrm{KL}(q,p)}$	such that	$\sum_\alpha p_\alpha = 1,\ p_\alpha \geq 0.$

$\leftarrow$ KL distance

Solution: $p_\alpha = q_\alpha$

ML solution from normalized counts

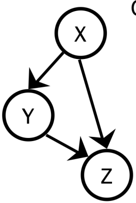
$$P_{\mathrm{ML}}(X_i = x | \mathrm{pa}_i = \pi) = \frac{\mathrm{count}(X_i = x, \mathrm{pa}_i = \pi)}{\sum_{x'} \mathrm{count}(X_i = x', \mathrm{pa}_i = \pi)}$$

- For nodes with parents:

$$P_{\mathrm{ML}}(X_i = x | \mathrm{pa}_i = \pi) = \frac{\mathrm{count}(X_i = x, \mathrm{pa}_i = \pi)}{\mathrm{count}(\mathrm{pa}_i = \pi)}$$

- For root nodes:

$$P_{\mathrm{ML}}(X_i = x) = \frac{\mathrm{count}(X_i = x)}{T}$$



X, Y and Z are Boolean variables

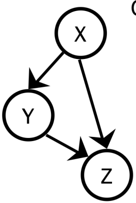
Observed data:

X	Y	Z
0	0	1
0	1	0
0	1	1
0	1	0
1	0	0
1	0	0
0	1	1
1	0	0
0	1	1
0	0	1
0	1	1
1	0	0

Q. Which of the following is a parameter we would like to estimate?

A. P(X=1)
B. P(Y=1)
C. P(X=1|Y=1)
D. More than one of these
E. None of these

ML Example



X, Y and Z are Boolean variables

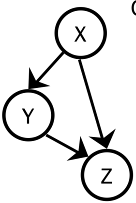
Observed data:

X	Y	Z
0	0	1
0	1	0
0	1	1
0	1	0
1	0	0
1	0	0
0	1	1
1	0	0
0	1	1
0	0	1
0	1	1
1	0	0

Q. Not including complements (e.g. P(X=1) and P(X=0)), how many different parameters are there to estimate?

A. 3
B. 4
C. 5
D. 7
E. More than 7

ML Example



X, Y and Z are Boolean variables

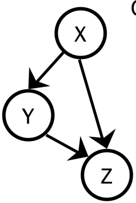
Observed data:

X	Y	Z
0	0	1
0	1	0
0	1	1
0	1	0
1	0	0
1	0	0
0	1	1
1	0	0
0	1	1
0	0	1
0	1	1
1	0	0

Q. What is the ML estimate for $P(Z=1|X=0, Y=0)$?

A. 0
B. 1/6
C. 1/2
D. 1
E. None of the above

ML Example



X, Y and Z are Boolean variables

Observed data:

X	Y	Z
0	0	1
0	1	0
0	1	1
0	1	0
1	0	0
1	0	0
0	1	1
1	0	0
0	1	1
0	0	1
0	1	1
1	0	0

Q. Which parameter has an undefined ML estimate?

A. P(X=1)
B. P(Y=1|X=0)
C. P(Z=1|X=0, Y=0)
D. P(Z=1|X=1, Y=1)
E. More than one of the above

Properties of ML solution

- **Asymptotically correct:**

  The more data you have, the better your estimates.
  If $P(x_1, x_2, \ldots, x_n) > 0$, then

$$\lim_{T \to \infty} P_{\mathrm{ML}}(x_1, x_2, \ldots, x_n) \;=\; P(x_1, x_2, \ldots, x_n)$$

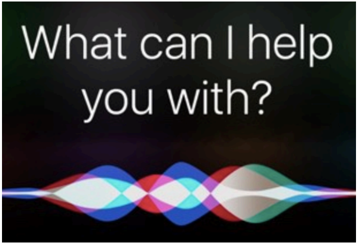
- **But problematic for sparse data:**

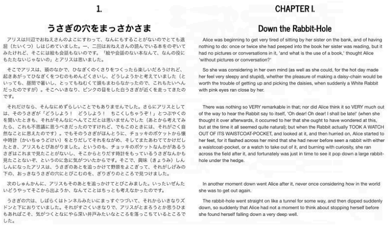
$$P_{\mathrm{ML}}(X_i = x | \mathrm{pa}_i = \pi) \;=\; \frac{\mathrm{count}(X_i = x, \mathrm{pa}_i = \pi)}{\mathrm{count}(\mathrm{pa}_i = \pi)}$$

  This is undefined when $\mathrm{count}(\mathrm{pa}_i = \pi) = 0$.
  Otherwise it is zero when $\mathrm{count}(X_i = x, \mathrm{pa}_i = \pi) = 0$.

Markov models

Let w_ℓ denote the ℓ^{th} word in a sentence (or text).
How to model $P(w_1, w_2, \ldots, w_L)$?





1.

うさぎの穴をまっさかさま

アリスは川辺でおねえさんのよこにすわって、なんにもすることがないのでとても退屈（たいくつ）しはじめていました。一、二回はおねえさんの読んでいる本をのぞいてみたけれど、そこには絵も会話もないのです。「絵や会話のない本なんて、なんの役にもたたないじゃないの」とアリスは思いました。

そこでアリスは、頭のなかで、ひなぎくのくさりをつくったら楽しいだろうけれど、起きあがってひなぎくをつむのもめんどくさいし、どうしようかと考えていました（といっても、昼間で暑いし、とってもねむくて頭もまわらなかったので、これもたいへんだったのですが）。そこへいきなり、ピンクの目をした白うさぎが近くを走ってきたのです。

それだけなら、そんなにめずらしいことでもありませんでした。さらにアリスとしては、そのうさぎが「どうしよう！　どうしよう！　ちこくしちゃうぞ！」とつぶやくのを聞いたときも、それがそんなにへんてこだとは思いませんでした（あとから考えてみたら、これも不思議に思うべきだったのですけれど、でもこのときには、それがごく自然なことに思えたのです）。でもそのうさぎがほんとうに、チョッキのポケットから懐中時計（かいちゅうどけい）をとりだしてそれをながめ、そしてまたあわててかけだしたとき、アリスもとびあがりました。というのも、チョッキのポケットなんかがあるうさぎはこれまで見たことがないし、そこからとりだす時計をもっているうさぎなんかも見たことないぞ、というのに急に気がついたからです。そこで、興味（きょうみ）しんしんになったアリスは、うさぎのあとを追っかけて野原をよこぎって、それがしげみの下の、おっきなうさぎの穴にとびこむのを、ぎりぎりのところで見つけました。

次のしゅんかんに、アリスもそのあとを追っかけてとびこみました。いったいぜんたいどうやってそこから出ようか、なんてことはちっとも考えなかったのです。

うさぎの穴は、しばらくはトンネルみたいにまっすぐつづいて、それからいきなりズドンと下におりていました。それがすごくいきなりで、アリスがとまろうとか思うひまもあればこそ、気がつくとなにやら深い井戸みたいなところを落っこちているところでした。

CHAPTER I.

Down the Rabbit-Hole

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, 'and what is the use of a book,' thought Alice 'without pictures or conversation?'

So she was considering in her own mind (as well as she could, for the hot day made her feel very sleepy and stupid), whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.

There was nothing so VERY remarkable in that; nor did Alice think it so VERY much out of the way to hear the Rabbit say to itself, 'Oh dear! Oh dear! I shall be late!' (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural); but when the Rabbit actually TOOK A WATCH OUT OF ITS WAISTCOAT-POCKET, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket, or a watch to take out of it, and burning with curiosity, she ran across the field after it, and fortunately was just in time to see it pop down a large rabbit-hole under the hedge.

In another moment down went Alice after it, never once considering how in the world she was to get out again.

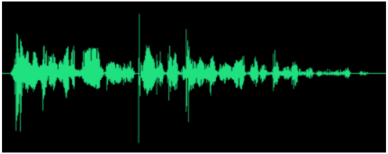
The rabbit-hole went straight on like a tunnel for some way, and then dipped suddenly down, so suddenly that Alice had not a moment to think about stopping herself before she found herself falling down a very deep well.

automatic speech recognition

machine translation

Context and expectations in language

“It’s hard to wreck a nice beach.”



“It’s hard to recognize speech.”

Simplifying assumptions

1. **Finite context**

   To predict the ℓ^{th} word, it is sufficient to consider a *finite* number of words that precede it:

   $$P(w_\ell | w_1, w_2, \ldots, w_{\ell-1}) \;=\; P(w_\ell | \underbrace{w_{\ell-(n-1)}, \ldots, w_{\ell-1}}_{n-1 \text{ previous words}})$$

2. **Position invariance**

   Predictions should not depend on where the context occurs in the sentence or text:

   $$P(W_\ell = w' | w_{\ell-(n-1)}, \ldots, w_{\ell-1})$$

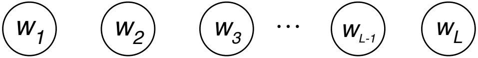
   $$= \quad P(W_{s+\ell} = w' | W_{s+\ell-(n-1)} = w_{\ell-(n-1)}, \ldots, W_{s+\ell-1} = w_{\ell-1})$$

Markov models

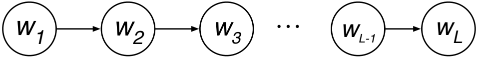
$$
\begin{aligned}
P(w_1, w_2, \ldots, w_L) \\
&= \prod_{\ell} P(w_\ell | w_1, w_2, \ldots, w_{\ell-1}) \quad \boxed{\text{product rule}} \\
&= \prod_{\ell} P(w_\ell | w_{\ell-(n-1)}, \ldots, w_{\ell-1}) \quad \boxed{\text{conditional independence}}
\end{aligned}
$$

Models of different orders

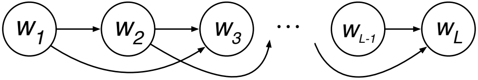
$n = 1$ unigram



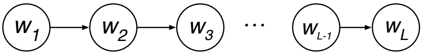
$n = 2$ bigram



$n = 3$ trigram



Bigram models



Note that the same CPT for $P(w_\ell = w' | w_{\ell-1} = w)$ is used at each node (for $\ell > 1$).

How to learn?

Collect a large corpus of text with a well-defined vocabulary.

Count how often word w is followed by the word w'.
Count how often word w is followed by any word.

Estimate from empirical frequencies:

$$P_{\mathrm{ML}}(w_\ell = w' | w_{\ell-1} = w) \;=\; \frac{\mathrm{count}(w \to w')}{\mathrm{count}(w \to *)} \;=\; \frac{\mathrm{count}(w \to w')}{\sum_{w''} \mathrm{count}(w \to w'')}$$

Problems with ML estimates

1. **No generalization to unseen *n*-grams:**

   ML estimates assign **zero** probability to *n*-grams that do not appear in the training corpus.

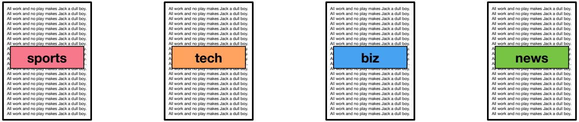
2. **The larger *n*, the worse the problem:**

   n-gram counts become increasingly sparse as *n* increases. Many possible (but improbable) *n*-grams are not observed.

You will explore this problem further in HW 4.

Naive Bayes models

Document classification



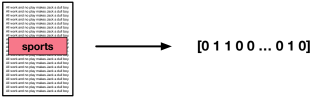
- **Setup**

  Each document can be labeled by one of m topics.
  Each document consists of words from a finite vocabulary.

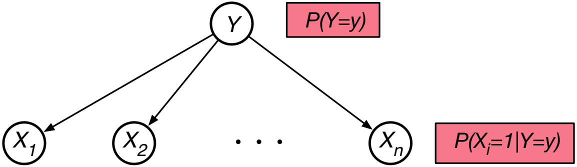
- **Random variables**

  Let $Y \in \{1, 2, \ldots, m\}$ denote the label.
  Let $X_i \in \{0, 1\}$ denote whether the i^{th} word appears.

This representation maps each document to a sparse binary vector of fixed length.



Belief network

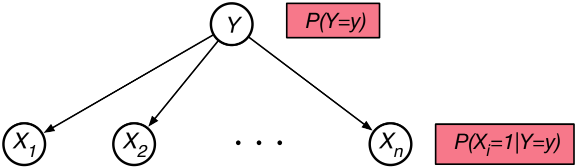


This DAG makes a fairly drastic assumption of conditional independence:

$$P(X_1, \ldots, X_n | Y) = \prod_{i=1}^{n} P(X_i | Y)$$

For this reason it is called a Naive Bayes model.

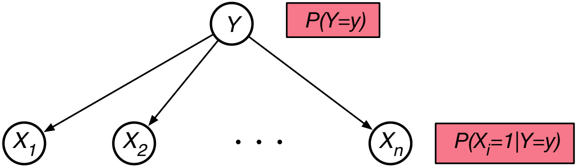
Naive Bayes model



Suppose this DAG is given, but the CPTs are not specified.
How to learn the CPTs from data?

- **Collect** a large corpus of documents.
- **Label** each document by a topic.
- **Estimate** the CPTs by maximizing the likelihood.

ML estimation



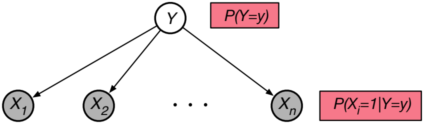
$$P_{\mathrm{ML}}(Y=y) = \text{fraction of documents with label } y \text{ in the corpus}$$

$$P_{\mathrm{ML}}(X_i=1|Y=y) = \text{fraction of documents with label } y \text{ that contain the } i^{\mathrm{th}} \text{ word in the vocabulary}$$

Once the model is learned, what is it good for?

Inference

How to classify an unlabeled document?



$$P(Y=y|X_1, X_2, \ldots, X_n)$$

$$= \frac{P(X_1, X_2, \ldots, X_n|Y=y)\, P(Y=y)}{P(X_1, X_2, \ldots, X_n)}$$

Bayes rule

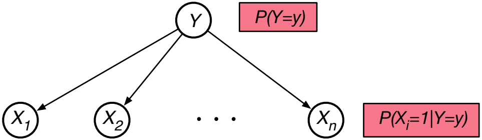
$$= \frac{P(Y=y) \prod_{i=1}^{n} P(X_i|Y=y)}{P(X_1, X_2, \ldots, X_n)}$$

conditional independence

$$= \frac{P(Y=y) \prod_{i=1}^{n} P(X_i|Y=y)}{\sum_{y'} P(Y=y') \prod_{i=1}^{n} P(X_i|Y=y')}$$

normalization

Strengths and weaknesses



Strengths

- Easy to learn from data.
- Easy to classify unlabeled documents.

Weaknesses

- Naive Bayes assumption of conditional independence
- No information about word ordering
- Binarization of word counts
- Etc ...

That’s all folks!