Bayesian Networks

David Rosenberg

New York University

June 9, 2015

Probabilistic Reasoning

- **Represent** system of interest by a set of random variables

$$(X_1, \dots, X_d).$$

- Suppose by research or machine learning, we get a joint probability distribution

$$p(x_1, \dots, x_d).$$

- We'd like to be able to do **inference** on this model – essentially, answer queries:
  1. What is the most likely of value X_1?
  2. What is the most likely of value X_1, given we've observed $X_2 = 1$?
  3. Distribution of (X_1, X_2) given observation of $(X_3 = x_3, \dots, X_d = x_d)$?

Example: Medical Diagnosis

- Variables for each **symptom**
  - fever, cough, fast breathing, shaking, nausea, vomiting
- Variables for each **disease**
  - pneumonia, flu, common cold, bronchitis, tuberculosis
- Diagnosis is performed by **inference** in the model:

$$p(\text{pneumonia} = 1 \mid \text{cough} = 1, \text{fever} = 1, \text{vomiting} = 0)$$

- The QMR-DT (Quick Medical Reference - Decision Theoretic) has
  - 600 diseases
  - 4000 symptoms

Example from David Sontag's *Inference and Representation*, Lecture 1.

Some Notation

- This lecture we'll only be considering **discrete** random variables.
- Capital letters $X_1, \ldots, X_d, Y$, etc. denote **random variables**.
- Lower case letters $x_1, \ldots, x_n, y$ denote the values taken.
- Probability that $X_1 = x_1$ and $X_2 = x_2$ will be denoted

$$\mathbb{P}(X_1 = x_1, X_2 = x_2).$$

- We'll generally write things in terms of the probability mass function:

$$p(x_1, x_2, \ldots, x_d) := \mathbb{P}(X_1 = x_1, X_2 = x_2, \ldots, X_d = x_d)$$

Representing Probability Distributions

- Let's consider the case of discrete random variables.
- Conceptually, everything can be represented with probability tables.
- Variables
  - Temperature $T \in \{\text{hot}, \text{cold}\}$
  - Weather $W \in \{\text{sun}, \text{rain}\}$

t	$p(t)$
hot	0.5
cold	0.5

w	$p(w)$
sun	0.6
rain	0.4

- These are the **marginal** probability distributions.
- To do reasoning, we need the **joint probability distribution**.

Based on David Sontag's *DS-GA 1003 Lectures, Spring 2014*, Lecture 10.

Joint Probability Distributions

- A **joint probability distribution** for T and W is given by

t	w	$p(t,w)$
hot	sun	0.4
hot	rain	0.1
cold	sun	0.2
cold	rain	0.3

- A valid probability distribution if
  - $\forall t,w\colon\ p(t,w) \geqslant 0$
  - $\sum_{t,w} p(t,w) = 1$.

Based on David Sontag's *DS-GA 1003 Lectures, Spring 2014*, Lecture 10.

Conditional Distributions From the Joint Distribution

- We **observe** $T = \text{hot}$. What's the conditional distribution of W?

$$p\left(w \mid T = \text{hot}\right) = ?$$

- Method:
  1. Find entries in joint distribution table where $T = \text{hot}$.

t	w	$p(t,w)$
hot	sun	0.4
hot	rain	0.1

  2. Renormalize to get conditional probability.

t	w	$p(t,w)$	$p(w \mid T = \text{hot})$
hot	sun	0.4	$0.4/0.5 = 0.8$
hot	rain	0.1	$0.1/0.5 = 0.2$

Conditional Distributions From the Joint Distribution

Definition

The **conditional probability** for w given t is

$$p(w \mid t) = \frac{p(w,t)}{p(t)}.$$

t	w	$p(t,w)$	$p(w \mid T = \text{hot})$
hot	sun	0.4	$0.4/0.5 = 0.8$
hot	rain	0.1	$0.1/0.5 = 0.2$

Representing Joint Distributions

- Consider random variables $X_1, \ldots, X_d \in \{0,1\}$.
- How many parameters do we need to represent the joint distribution?
- Joint probability table has 2^d rows.
- For QMR-DT, that's $2^{4600} > 10^{1000}$ rows.
- That's not going to happen.
- Having exponentially many parameters is a problem for
    - storage
    - computation (inference is summing over exponentially many rows)
    - statistical estimation / learning
        - (Estimating 10^{1000} parameters? Nope.)

How to Restrict the Complexity?

- Restrict the space of probability distributions
- We will make various **independence** assumptions.
- Extreme assumption: $X_1, \ldots, X_d$ are **mutually independent**.

Definition

Discrete random variables $X_1, \ldots, X_d$ are **mutually independent** if their joint probability mass function (PMF) factorizes as

$$p(x_1, x_2, \ldots, x_d) = p(x_1)p(x_2)\cdots p(x_d).$$

- Note: We usually just write **independent** for "mutually independent".
- How many parameters to represent the joint distribution, assuming independence?

Assume Full Independence

- How many parameters to represent the joint distribution?
- Say $p(X_i = 1) = \theta_i$, for $i = 1, \dots, d$.
- **Clever representation**: Since $x_i \in \{0, 1\}$, we can write

$$\mathbb{P}(X_i = x_i) = \theta_i^{x_i} (1 - \theta_i)^{1 - x_i}.$$

- Then by independence,

$$p(x_1, \dots, x_d) = \prod_{i=1}^{d} \theta_i^{x_i} (1 - \theta_i)^{1 - x_i}$$

- How many parameters?
- d parameters needed to represent the joint.

Conditional Interpretation of Independence

- Suppose X and Y are independent, then

$$p(x \mid y) = p(x).$$

- Proof:

$$\begin{aligned} p(x \mid y) &= \frac{p(x,y)}{p(y)} \\ &= \frac{p(x)p(y)}{p(y)} = p(x). \end{aligned}$$

- With full independence, we have no relationships among variables.
- Information about one variable says nothing about any other variable.
    - Would mean diseases don't have symptoms.

Conditional Independence

- Consider 3 events:
  1. $W = \{\text{The grass is wet}\}$
  2. $S = \{\text{The road is slippery}\}$
  3. $R = \{\text{It's raining}\}$
- These events are certainly **not** independent.
  - Raining $(R) \implies$ Grass is wet AND The road is slippery $(W \cap S)$
  - Grass is wet $(W) \implies$ More likely that the road is slippery (S)
- Suppose we know that **it's raining**.
  - Then, we learn that **the grass is wet**.
  - Does this tell us anything new about whether **the road is slippery?**
- Once we know R, then W and S become independent.
- This is called **conditional independence**, and we'll denote it as

$$W \perp S \mid R.$$

Conditional Independence

Definition

We say W and S are **conditionally independent** given R, denoted

$$W \perp S \mid R,$$

if the conditional joint factorizes as

$$p(w, s \mid r) = p(w \mid r)p(s \mid r).$$

Also holds when W, S, and R represent **sets of random variables**.

Example: Rainy, Slippery, Wet

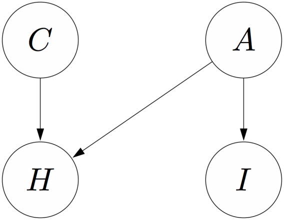
- Consider 3 events:
  1. $W = \{\text{The grass is wet}\}$
  2. $S = \{\text{The road is slippery}\}$
  3. $R = \{\text{It's raining}\}$
- Represent joint distribution as

$$
\begin{aligned}
p(w,s,r) &= p(w,s \mid r)p(r) && \text{(no assumptions so far)} \\
&= p(w \mid r)p(s \mid r)p(r) && \text{(assuming } W \perp S \mid R\text{)}
\end{aligned}
$$

- How many parameters to specify the joint?
  - $p(w \mid r)$ requires two parameters: one for $r = 1$ and one for $r = 0$.
  - $p(s \mid r)$ requires two.
  - $p(r)$ requires one parameter,
- Full joint: 7 parameters. Conditional independence: 5 parameters. Full independence: 3 parameters.

Bayesian Networks: Introduction

- Bayesian Networks are
  - used to specify joint probability distributions that
  - have a particular factorization.



$$p(c,h,a,i) = p(c)p(a) \times p(h \mid c,a)p(i \mid a)$$

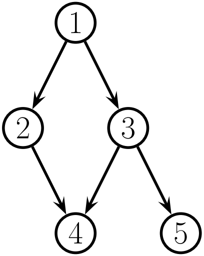
- With practice, one can read conditional independence relationships directly from the graph.

From Percy Liang's "Lecture 14: Bayesian networks II" slides from Stanford's CS221, Autumn 2014.

Directed Graphs

A **directed graph** is a pair $G = (\mathcal{V}, \mathcal{E})$, where

- $\mathcal{V} = \{1, \ldots, d\}$ is a set of **nodes** and
- $\mathcal{E} = \{(s,t) \mid s,t \in \mathcal{V}\}$ is a set of **directed edges**.



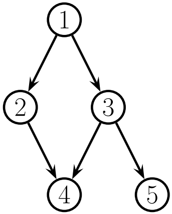
$$
\begin{aligned}
\text{Parents}(5) &= \{3\} \\
\text{Parents}(4) &= \{2,3\} \\
\text{Children}(3) &= \{4,5\} \\
\text{Descendants}(1) &= \{2,3,4,5\} \\
\text{NonDescendants}(3) &= \{1,2\}
\end{aligned}
$$

KPM Figure 10.2(a).

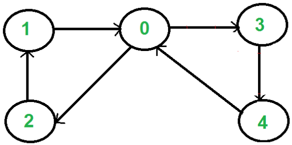
Directed Acyclic Graphs (DAGs)

A **DAG** is a directed graph with **no directed cycles**.

DAG



Not a DAG



Every DAG has a **topological ordering,** in which parents have lower numbers than their children.

http://www.geeksforgeeks.org/wp-content/uploads/SCC1.png and KPM Figure 10.2(a).

Bayesian Networks

Definition

A **Bayesian network** is a

- DAG $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, d\}$, and
- a corresponding set of random variables $X = \{X_1, \ldots, X_d\}$

where

- the joint probability distribution over X factorizes as

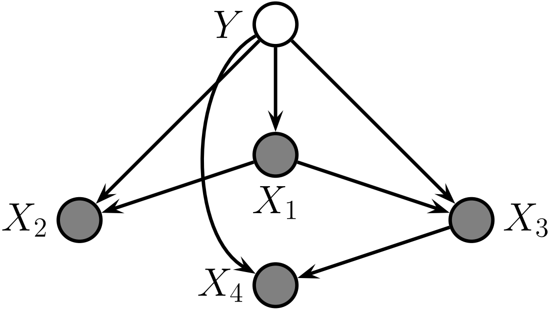
$$p(x_1, \ldots, x_d) = \prod_{i=1}^{d} p(x_i \mid x_{\text{Parents}(i)}).$$

Bayesian networks are also known as

- **directed graphical models**, and
- **belief networks.**

Bayesian Networks: Example

Consider the Bayesian network depicted below:

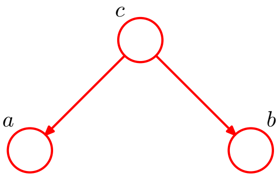


It implies the following factorization for the joint probability distribution:

$$p(x_1, x_2, x_3, x_4, y) = p(y)p(x_1 \mid y)p(x_2 \mid x_1, y)p(x_3 \mid x_1, y)p(x_4 \mid x_3, y)$$

KPM Figure 10.2(b).

Bayesian Networks: "A Common Cause"



$$p(a, b, c) = p(c)p(a \mid c)p(b \mid c)$$

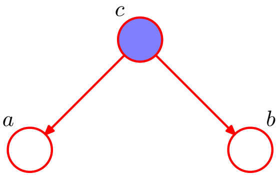
Are a and b independent? (c=Rain, a=Slippery, b=Wet?)

$$p(a, b) = \sum_{c} p(c)p(a \mid c)p(b \mid c),$$

which in general will not be equal to $p(a)p(b)$.

From Bishop's *Pattern recognition and machine learning*, Figure 8.15.

Bayesian Networks: "A Common Cause"



$$p(a, b, c) = p(c)p(a \mid c)p(b \mid c)$$

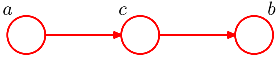
Are a and b independent, conditioned on observing c? (c=Rain, a=Slippery, b=Wet?)

$$\begin{aligned} p(a, b \mid c) &= p(a, b, c)/p(c) \\ &= p(a \mid c)p(b \mid c) \end{aligned}$$

So $a \perp b \mid c$.

From Bishop's *Pattern recognition and machine learning*, Figure 8.16.

Bayesian Networks: "An Indirect Effect"



$$p(a,b,c) = p(a)p(c \mid a)p(b \mid c)$$

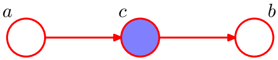
Are a and b independent? (Note: This is a **Markov chain**)
(e.g. a=raining, c=wet ground, b=mud on shoes)

$$\begin{aligned} p(a,b) &= \sum_c p(a,b,c) \\ &= p(a)\sum_c p(c \mid a)p(b \mid c) \end{aligned}$$

So doesn't factorize, thus not independent, in general.

From Bishop's *Pattern recognition and machine learning*, Figure 8.17.

Bayesian Networks: "An Indirect Effect"



$$p(a, b, c) = p(a)p(c \mid a)p(b \mid c)$$

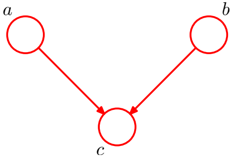
Are a and b independent after observing c?
(e.g. a=raining, c=wet ground, b=mud on shoes)

$$\begin{aligned} p(a, b \mid c) &= p(a, b, c)/p(c) \\ &= p(a)p(c \mid a)p(b \mid c)/p(c) \\ &= p(a \mid c)p(b \mid c) \end{aligned}$$

So $a \perp b \mid c$.

From Bishop's *Pattern recognition and machine learning*, Figure 8.18.

Bayesian Networks: "A Common Effect"



$$p(a,b,c) = p(a)p(b)p(c \mid a,b)$$

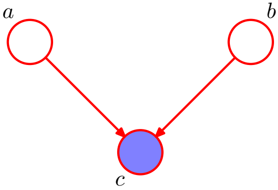
Are a and b independent? (a=course difficulty, b=knowledge, c= grade)

$$\begin{aligned} p(a,b) &= \sum_c p(a)p(b)p(c \mid a,b) \\ &= p(a)p(b)\sum_c p(c \mid a,b) \\ &= p(a)p(b) \end{aligned}$$

So $a \perp b$.

From Bishop's *Pattern recognition and machine learning*, Figure 8.19.

Bayesian Networks: "A Common Effect" or "V-Structure"



$$p(a, b, c) = p(a)p(b)p(c \mid a, b)$$

Are a and b independent, given observation of c? (a=course difficulty, b=knowledge, c= grade)

$$p(a, b \mid c) = p(a)p(b)p(c \mid a, b)/p(c)$$

which does not factorize into $p(a \mid c)p(b \mid c)$, in general.

From Bishop's *Pattern recognition and machine learning*, Figure 8.20.

Conditional Independence from Graph Structure

- In general, given 3 sets of nodes A, B, and C
- How can we determine whether

$$A \perp B \mid C?$$

- There is a purely graph-theoretic notion of "**d-separation**" that is equivalent to conditional independence.
- Suppose we have observed C and we want to do inference on A.
- We could ignore any evidence collected about B, where $A \perp B \mid C$.
- See KPM Section 10.5.1 for details.

Markov Blanket

- Suppose we have a very large Bayesian network.
- We're interested in a single variable A, which we cannot observe.
- To get maximal information about A, do we have to observe all other variables?
- No! We only need to observe the **Markov blanket** of A:

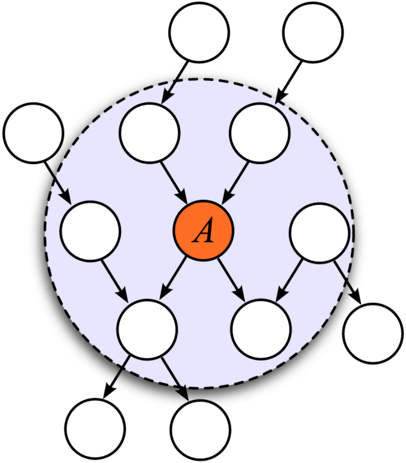
$$p(A \mid \text{all other nodes}) = p(A \mid \text{MarkovBlanket}(A)).$$

- In a Bayesian network, the Markov blanket of A consists of
  - the parents of A
  - the children of A
  - the "co-parents" of A, i.e. the parents of the children of A

(See KPM Sec. 10.5.3 for details.)

Markov Blanket

Markov Blanket of A in a Bayesian Network:



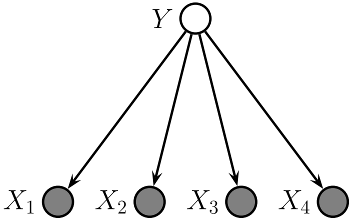
From http://en.wikipedia.org/wiki/Markov_blanket: "Diagram of a Markov blanket" by Laughsinthestocks - Licensed under CC0 via Wikimedia Commons

Bayesian Networks

- Bayesian Networks are great when
  - you know something about the relationships between your variables, or
  - you will routinely need to make inferences with incomplete data.
- Challenges:
  - The naive approach to inference doesn't work beyond small scale.
  - Need more sophisticated algorithm:
    - exact inference
    - approximate inference

Naive Bayes: A Generative Model for Classification

- $\mathcal{X} = \left\{ \left(X_1, X_2, X_3, X_4 \right) \in \{0,1\}^4 \right\}$ $\mathcal{Y} = \{0,1\}$ be a class label.
- Consider the Bayesian network depicted below:



- BN structure implies joint distribution factors as:

$$p(x_1, x_2, x_3, x_4, y) = p(y)p(x_1 \mid y)p(x_2 \mid y)p(x_3 \mid y)p(x_4 \mid y)$$

- Features $X_1, \ldots, X_4$ are independent given the class label Y.

KPM Figure 10.2(a).

Parameters for Naive Bayes

- Generalize to d features.
- Knowing the joint distribution means we need to know

$$p(y),\ p(x_1 \mid y), \ldots p(x_d \mid y).$$

- We could parameterize as:

$$\begin{aligned}\mathbb{P}(Y=1) &= \theta_y \\ \mathbb{P}(X_i=1 \mid Y=1) &= \theta_{i1} \\ \mathbb{P}(X_i=1 \mid Y=0) &= \theta_{i0}\end{aligned}$$

$\Longrightarrow$ $1+2d$ parameters to characterize the joint distribution

Parameterized Expression for Joint

- Parameters:

$$\mathbb{P}(Y=1)=\theta_y \qquad \mathbb{P}(X_i=1\mid Y=1)=\theta_{i1} \qquad \mathbb{P}(X_i=1\mid Y=0)=\theta_{i0}$$

- Joint distribution is

$$\begin{aligned} & p(x_1,\ldots x_d,y) \\ = \;& p(y)\prod_{i=1}^{n}p(x_i\mid y) \\ = \;& (\theta_y)^y(1-\theta_y)^{1-y} \\ & \times\prod_{i=1}^{n}(\theta_{i1})^{yx_i}(1-\theta_{i1})^{y(1-x_i)}(\theta_{i0})^{(1-y)x_i}(1-\theta_{i0})^{(1-y)(1-x_i)} \end{aligned}$$

Naive Bayes

- Suppose we know all conditional distributions:

$$p(y),\ p(x_1 \mid y), \ldots p(x_d \mid y)$$

- We observe $X = (X_1, \ldots, X_d)$. What's the prediction for Y?
- We have a full probability model

$$\begin{aligned} p(y, x_1, \ldots, x_d) &= p(y)p(x_1, \ldots, x_d \mid y) && \text{(no assumptions)} \\ &= p(y)\prod_{i=1}^{d} p(x_i \mid y) && \text{(conditional independence)} \end{aligned}$$

- We can use Bayes rule to compute anything we want...

Posterior Class Probability

- Let $x = (x_1, \ldots, x_d)$, and apply Bayes rule:

$$p(y \mid x) = \frac{p(y,x)}{p(x)} = \frac{p(y)\prod_{i=1}^{d} p(x_i \mid y)}{p(x)}$$

- We know everything except $p(x)$.
- We can compute it explicitly:

$$p(x) = \sum_{y\in\{0,1\}} p(x,y) = \sum_{y\in\{0,1\}} p(x|y)p(y)$$

- So final predicted probability distribution is

$$p(y \mid x) = \frac{p(y)\prod_{i=1}^{d} p(x_i \mid y)}{\sum_{y\in\{0,1\}} p(x|y)p(y)}$$

Dropping Normalization Constant

- Consider $p(y \mid x)$ as a distribution over y, for **fixed** x.

$$p(y \mid x) = p(y,x)/p(x).$$

- With x fixed, $p(x)$ is a constant – let's write it as k to make it clear:

$$\begin{aligned} p(y \mid x) &= k^{-1} p(y,x) \\ \implies p(y \mid x) &\propto p(y,x) \end{aligned}$$

- How to recover value of k? $p(y \mid x)$ must be a distribution on y:

$$\begin{aligned} \sum_{y \in \{0,1\}} p(y \mid x) &= k^{-1} \sum_{y \in \{0,1\}} p(y,x) = 1 \\ \implies k &= \sum_{y \in \{0,1\}} p(y,x) \end{aligned}$$

- So we can always recover the normalizing constant whenever we want.
    - Often no need to keep track of it.

Naive Bayes and Logistic Regression

- Recall the logistic regression prediction function is of the form

$$x \mapsto p(Y = 1 \mid x) = \frac{1}{1 + \exp\left(-w^T x\right)},$$

for some parameter vector $w \in \mathbf{R}^d$.

Theorem

If $p(y, x)$ is any Naive Bayes model with binary x and y, the prediction function

$$x \mapsto p(Y = 1 \mid x)$$

corresponds to logistic regression, for some $w \in \mathbf{R}^d$.

Proof: Homework.

Naive Bayes vs Logistic Regression

- Naive Bayes is a model for the joint distribution $p(y, x)$.
  - We can sample (x, y) pairs from this distribution.
  - Models of the joint distribution are called **generative models.**
- Logistic regression is directly modeling the conditional distribution

$$p(y \mid x).$$

  - No model for the features $x = (x_1, \ldots, x_d)$.
  - Conditional probability models are called **discriminative models.**
- Logistic regression is a specialist in the conditional distribution.
- Naive Bayes is doing more!

Naive Bayes vs Logistic Regression

- **Missing data** is no problem for Naive Bayes.
- Suppose we're missing X_1 and X_2 from the input vector.
- Just predict with

$$\begin{aligned} \mathbb{P}(y \mid x_3, \ldots x_d) &\propto p(y, x_3, \ldots, x_d) \\ &= \sum_{x_1, x_2 \in \{0,1\}} p(y, x) \end{aligned}$$

- For logistic regression? No natural way to predict with missing features.

Naive Bayes vs Logistic Regression

- Logistic regression handles binary or continuous features seamlessly.
- For naive Bayes, you need a different family of conditional distributions, e.g.
$$p(x_i \mid y) = \mathcal{N}\left(x_i \mid \mu_{iy}, \sigma_{iy}^2\right)$$
- Wasted effort to model all features if you only care about $p(y \mid x)$?
- Suppose we're missing X_1 and X_2 from the input vector.
- Just predict with
$$\begin{aligned}\mathbb{P}(y \mid x_3, \ldots x_d) &\propto p(y, x_3, \ldots, x_d) \\ &= \sum_{x_1, x_2 \in \{0,1\}} p(y, x)\end{aligned}$$
- No natural method for missing features with logistic regression.

Easy Estimators for Naive Bayes

- Training set $\mathcal{D} = \left\{ \left(x^1, y^1\right), \ldots (x^n, y^n) \right\}$.
- There are obvious "plug-in" estimators for the Naive Bayes model:

$$
\begin{aligned}
\mathbb{P}(Y = 1) \quad &\approx \quad \hat{\theta}_y = \frac{1}{n} \sum_{i=1}^{n} 1(y^i = 1) \\
\mathbb{P}(X_i = 1 \mid Y = 1) \quad &\approx \quad \hat{\theta}_{i1} = \frac{\sum_{j=1}^{n} 1(y^j = 1 \text{ and } x_i^j = 1)}{\sum_{j=1}^{n} 1(y^j = 1)} \\
\mathbb{P}(X_i = 1 \mid Y = 0) \quad &= \quad \hat{\theta}_{i0} = \frac{\sum_{j=1}^{n} 1(y^j = 0 \text{ and } x_i^j = 1)}{\sum_{j=1}^{n} 1(y^j = 0)}
\end{aligned}
$$

Maximum Likelihood Estimation for Naive Bayes

- Training set $\mathcal{D} = \left\{\left(x^1, y^1\right), \ldots (x^n, y^n)\right\}$.
- More principled: find the MLE for the Naive Bayes model.
- The log-likelihood objective function is

$$J(\theta) = \sum_{i=1}^{n} \log p(y^i, x^i),$$

where we found the likelihood for a single point (x, y) is

$$\begin{aligned} p(x,y) &= (\theta_y)^y (1-\theta_y)^{1-y} \\ &\times \prod_{i=1}^{n} (\theta_{i1})^{yx_i} (1-\theta_{i1})^{y(1-x_i)} \\ &\times \prod_{i=1}^{n} (\theta_{i0})^{(1-y)x_i} (1-\theta_{i0})^{(1-y)(1-x_i)} \end{aligned}$$

- **Theorem**: **MLE is exactly the plug-in estimator.**
- **Proof**: Optional Homework.

Class Prediction

- If we want to predict a single class, we would use

$$y^* = \arg\max_y p(y \mid x).$$

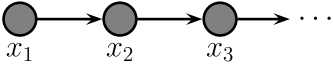
- One approach to this is to write

$$\begin{aligned}
\frac{p(Y=1 \mid x)}{p(Y=0 \mid x)} &= \frac{p(Y=1,x)/p(x)}{p(Y=0,x)/p(x)} = \frac{p(Y=1,x)}{p(Y=0,x)} \\
&= \frac{p(Y=1)\prod_{i=1}^{d} p(x_i \mid Y=1)}{p(Y=0)\prod_{i=1}^{d} p(x_i \mid Y=0)} \\
&= \frac{p(Y=1)}{p(Y=0)} \prod_{i=1}^{d} \frac{p(x_i \mid Y=1)}{p(x_i \mid Y=0)}
\end{aligned}$$

- Compare ratio to 1 to get prediction.

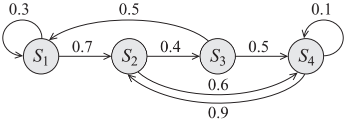
Markov Chain Model

- A Markov chain model has structure:



$$p(x_1, x_2, x_3, \ldots) = p(x_1)p(x_2 \mid x_1)p(x_3 \mid x_2)\cdots$$

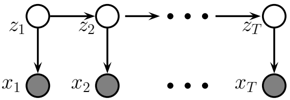
- Conditional distributions $p(x_i \mid x_{i-1})$ is called the **transition model.**
- When conditional distribution independent of i, called **time-homogeneous**.
- 4-state transition model for $X_i \in \{S_1, S_2, S_3, S_4\}$:



KPM Figure 10.3(a) and Koller and Friedman's *Probabilistic Graphical Models* Figure 6.04.

Hidden Markov Model

- A hidden Markov model (HMM) has structure:



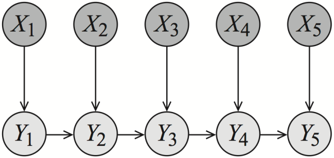
$$p(z_1, z_2, z_3, \ldots) = \underbrace{p(z_1) \prod_{t=2}^{T} p(z_t \mid z_{t-1})}_{\text{Transition Model}} \quad \underbrace{\prod_{t=1}^{T} p(x_t \mid z_t)}_{\text{Observation Model}}$$

- At deployment time, we typically only observe $X_1, \ldots, X_T$.
- Want to infer $Z_1, \ldots, Z_T$.
- e.g. Want to most likely sequence $(Z_1, \ldots, Z_T)$. (Use **Viterbi algorithm**.)

KPM Figure 10.4

Maximum Entropy Markov Model

- A maximum entropy Markov model (MEMM) has structure:



$$p(y_1 \ldots, y_5 \mid x) = \underbrace{p(y_0) \prod_{t=1}^{5} p(y_t \mid y_{t-1}, x)}_{\text{Conditional Transition Model}}$$

- At deployment time, we only observe $X_1, \ldots, X_T$.
- This is a **conditional model.** (And not a generative model).

Koller and Friedman's *Probabilistic Graphical Models* Figure 20.A.1.

Maximum Entropy Markov Model

- The MEMM transition model takes the following form:

$$p(y_i|y_{i-1},x) \quad \propto \quad \exp\left(\sum_k \lambda_k f_k(y_{i-1},y_i) + \sum_r \mu_r g_r(y_i,x)\right)$$

- The functions f_k and g_r are **feature functions.**
- Suppose Y's represent parts-of-speech; X's represent words.
- Could have

$$g_r(y_i,x) = \begin{cases} 1 & \text{if } y_i = \text{"NOUN" and } x_i = \text{"apple"} \\ 0 & \text{otherwise} \end{cases}$$

- For the "transition features", typical would be

$$f_k(y_{i-1},y_i) = \begin{cases} 1 & \text{if } (y_{i-1},y_i) = (\text{ADJ},\text{NOUN}) \\ 0 & \text{otherwise.} \end{cases}$$