CS60050
Machine Learning
Introduction to Bayesian Classifiers

Somak Aditya Sudeshna Sarkar
Department of CSE, IIT Kharagpur
Sep 8, 2023

Outline of lectures

Outline of Maximum likelihood estimation

2 examples of Bayesian classifiers:

- Naïve Bayes
- Logistic regression

Recap: Bayes' Rule

- Two ways to factor a joint distribution over two variables:

$$P(x, y) = P(x|y)P(y) = P(y|x)P(x)$$

- Dividing, we get:

$$P(x|y) = \frac{P(y|x)}{P(y)}P(x)$$

- Why is this at all helpful?
  - Let's us build one conditional from its reverse
  - Often one conditional is tricky but the other one is simple
  - Foundation of many practical systems (e.g. ASR, MT)

- In the running for most important ML equation!

Returning to thumbtack example…

- P(Heads) = θ, P(Tails) = 1-θ

- Flips are *i.i.d.*: $D=\{x_i \mid i=1\ldots n\}, \quad P(D \mid \theta) = \Pi_i P(x_i \mid \theta)$
  - Independent events
  - Identically distributed according to Bernoulli distribution
- Sequence *D* of α_H Heads and α_T Tails

$$P(\mathcal{D} \mid \theta) = \theta^{\alpha_H}(1-\theta)^{\alpha_T}$$

Called the “likelihood” of the data under the model

Maximum Likelihood Estimation

- **Data:** Observed set D of α_H Heads and α_T Tails
- **Hypothesis:** Bernoulli distribution
- **Learning:** finding θ is an optimization problem
  - What's the objective function?

$$P(\mathcal{D} \mid \theta) = \theta^{\alpha_H}(1-\theta)^{\alpha_T}$$

- **MLE:** Choose θ to maximize probability of *D*

$$\begin{aligned} \widehat{\theta} &= \arg\max_{\theta} \; P(\mathcal{D} \mid \theta) \\ &= \arg\max_{\theta} \; \ln P(\mathcal{D} \mid \theta) \end{aligned}$$

Your first parameter learning algorithm

$$\begin{aligned}\hat{\theta} &= \arg\max_{\theta} \ \ln P(\mathcal{D} \mid \theta) \\ &= \arg\max_{\theta} \ \ln \theta^{\alpha_H}(1-\theta)^{\alpha_T}\end{aligned}$$

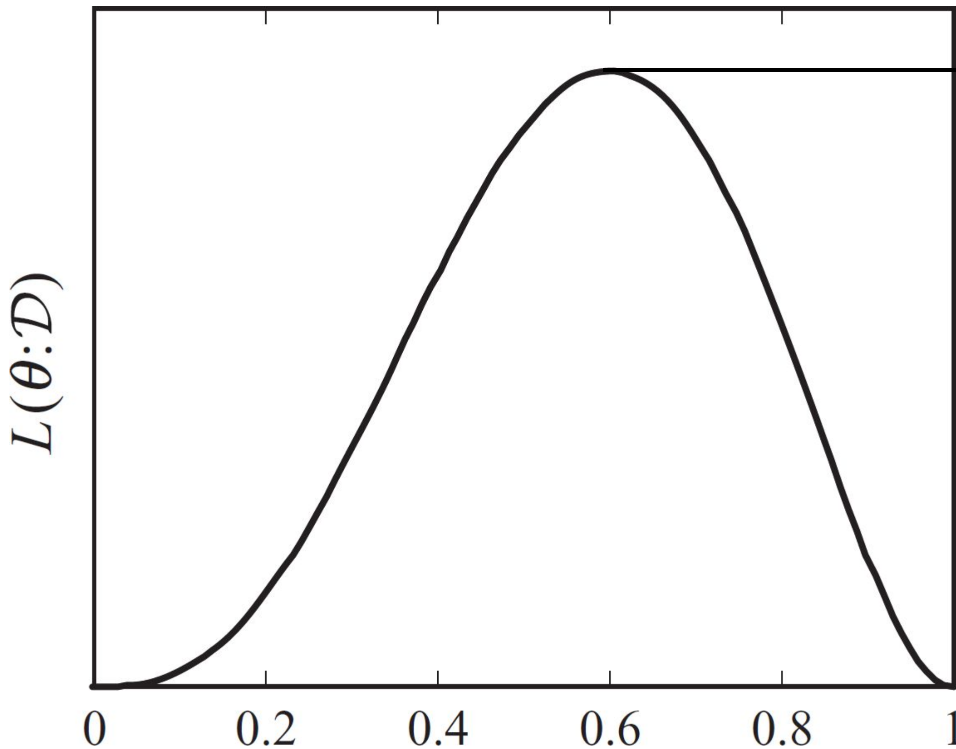
- Set derivative to zero, and solve!

$$\begin{aligned}\frac{d}{d\theta} \ln P(\mathcal{D} \mid \theta) &= \frac{d}{d\theta} [\ln \theta^{\alpha_H}(1-\theta)^{\alpha_T}] \\ &= \frac{d}{d\theta} [\alpha_H \ln \theta + \alpha_T \ln(1-\theta)] \\ &= \alpha_H \frac{d}{d\theta} \ln \theta + \alpha_T \frac{d}{d\theta} \ln(1-\theta) \\ &= \frac{\alpha_H}{\theta} - \frac{\alpha_T}{1-\theta} \quad = 0\end{aligned}$$

$$\boxed{\hat{\theta}_{MLE} = \frac{\alpha_H}{\alpha_H + \alpha_T}}$$

Data

$L(\theta;\, D) \;=\; \ln P(D|\theta)$



$$\boxed{\hat{\theta}_{MLE} \;=\; \frac{\alpha_H}{\alpha_H + \alpha_T}}$$

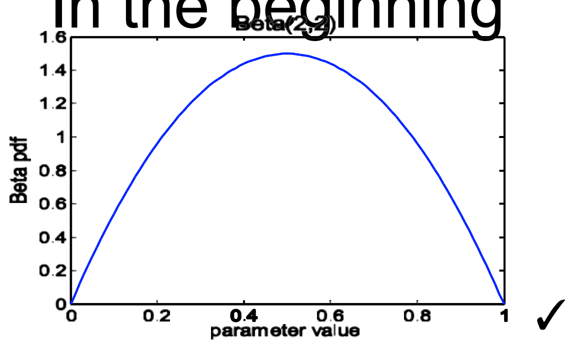
✓

What if I have prior beliefs?

- Billionaire says: Wait, I know that the thumbtack is “close” to 50-50. What can you do for me now?
- **You say: I can learn it the Bayesian way…**
- Rather than estimating a single θ, we obtain a distribution over possible values of θ

In the beginning

$\Pr(\theta)$

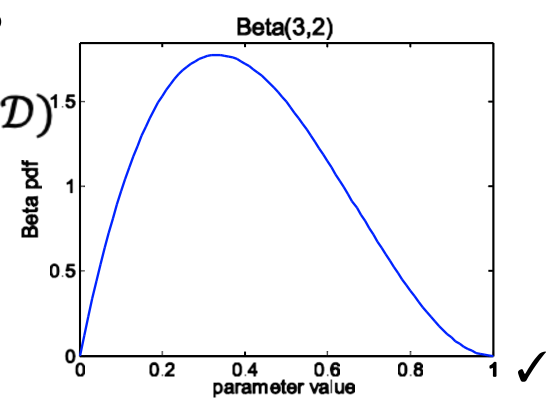


After observations

Observe flips
e.g.: {tails, tails}

$\longrightarrow$

$\Pr(\theta|\mathcal{D})$

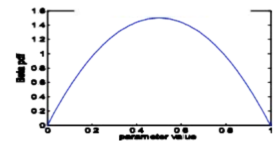


Bayesian Learning

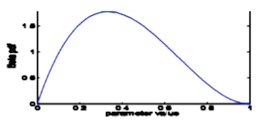
- Use Bayes' rule!

Data Likelihood

Prior



Posterior



$$P(\theta \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \theta) P(\theta)}{P(\mathcal{D})}$$

Normalization

- Or equivalently: $P(\theta \mid \mathcal{D}) \propto P(\mathcal{D} \mid \theta) P(\theta)$
- For *uniform* priors, this reduces to

  maximum likelihood estimation!

$$P(\theta) \propto 1 \qquad P(\theta \mid \mathcal{D}) \propto P(\mathcal{D} \mid \theta)$$

Bayesian Learning for Thumbtacks

$$P(\theta \mid \mathcal{D}) \;\propto\; P(\mathcal{D} \mid \theta)P(\theta)$$

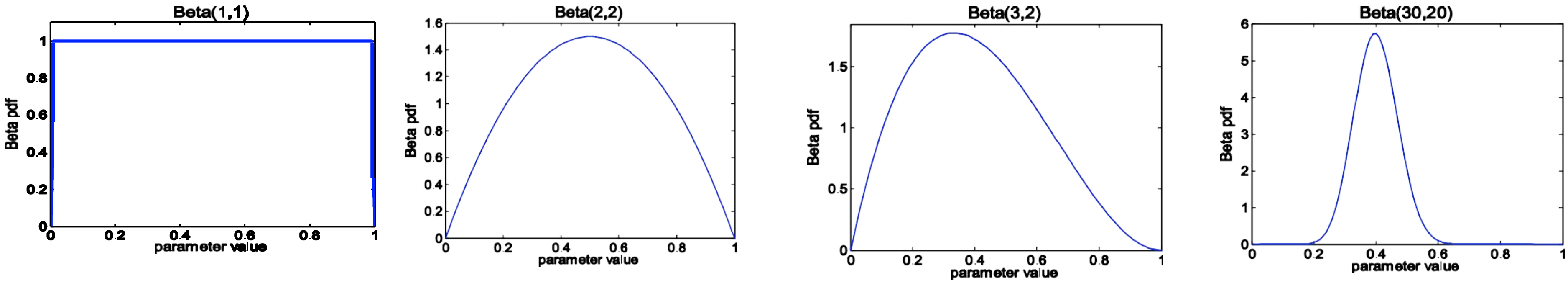
Likelihood: $P(\mathcal{D} \mid \theta) = \theta^{\alpha_H}(1-\theta)^{\alpha_T}$

- What should the prior be?
  - Represent expert knowledge
  - Simple posterior form
- For binary variables, commonly used prior is the Beta distribution:

$$P(\theta) = \frac{\theta^{\beta_H - 1}(1-\theta)^{\beta_T - 1}}{B(\beta_H, \beta_T)} \sim Beta(\beta_H, \beta_T)$$

Beta prior distribution – P(θ)

$$P(\theta) = \frac{\theta^{\beta_H - 1}(1-\theta)^{\beta_T - 1}}{B(\beta_H, \beta_T)} \sim Beta(\beta_H, \beta_T)$$



- Since the Beta distribution is *conjugate* to the Bernoulli distribution, the posterior distribution has a particularly simple form:

$$P(\theta \mid \mathcal{D}) \propto P(\mathcal{D} \mid \theta) P(\theta)$$

$$\propto \theta^{\alpha_H}(1-\theta)^{\alpha_T} \; \theta^{\beta_H - 1}(1-\theta)^{\beta_T - 1}$$

$$= \theta^{\alpha_H + \beta_H - 1}(1-\theta)^{\alpha_T + \beta_T - 1}$$

$$= Beta(\alpha_H + \beta_H, \alpha_T + \beta_T)$$

Using Bayesian inference for prediction

- We now have a **distribution** over parameters
- For any specific *f,* a function of interest, compute the expected value of *f*:

$$E[f(\theta)] = \int_0^1 f(\theta) P(\theta \mid \mathcal{D}) d\theta$$

  - Integral is often hard to compute
  - *As more data is observed, posterior is more concentrated*
- MAP (Maximum a posteriori approximation): use most likely parameter to approximate the expectation

$$\widehat{\theta} = \arg\max_{\theta} P(\theta \mid \mathcal{D})$$

$$E[f(\theta)] \approx f(\widehat{\theta})$$

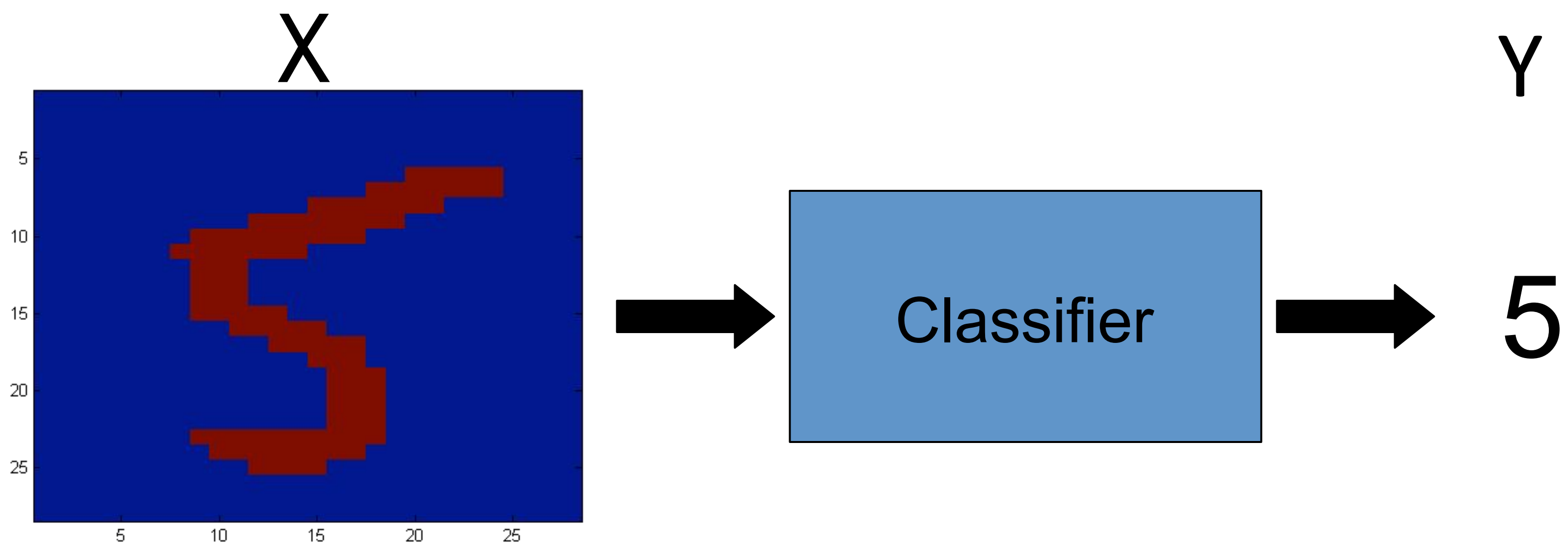
Bayesian Classification

- Problem statement:
  - Given features $X_1, X_2, \ldots, X_n$
  - Predict a label Y

[Next several slides adapted from:
Vibhav Gogate, Jonathan Huang, Luke Zettlemoyer, Carlos Guestrin, and Dan Weld]

Example Application

- **Digit Recognition**



- **$X_1, \ldots, X_n \in \{0,1\}$ (Black vs. White pixels)**
- **$Y \in \{0,1,2,3,4,5,6,7,8,9\}$**

The Bayes Classifier

- If we had the joint distribution on $\mathbf{X_1,\ldots,X_n}$ and $\mathbf{Y}$, could predict using:

$$\arg\max_Y P(Y|X_1,\ldots,X_n)$$

  - (for example: what is the probability that the image represents a 5 given its pixels?)

- So … How do we compute that?

The Bayes Classifier

- Use Bayes Rule!

$$P(Y|X_1,\ldots,X_n) = \frac{\overbrace{P(X_1,\ldots,X_n|Y)}^{\text{Likelihood}}\overbrace{P(Y)}^{\text{Prior}}}{\underbrace{P(X_1,\ldots,X_n)}_{\text{Normalization Constant}}}$$

- Why did this help? Well, we think that we might be able to specify how features are "generated" by the class label

The Bayes Classifier

- Let's expand this for our digit recognition task:

$$
\begin{aligned}
P(Y=5|X_1,\ldots,X_n) &= \frac{P(X_1,\ldots,X_n|Y=5)P(Y=5)}{P(X_1,\ldots,X_n|Y=5)P(Y=5)+P(X_1,\ldots,X_n|Y=6)P(Y=6)} \\
P(Y=6|X_1,\ldots,X_n) &= \frac{P(X_1,\ldots,X_n|Y=6)P(Y=6)}{P(X_1,\ldots,X_n|Y=5)P(Y=5)+P(X_1,\ldots,X_n|Y=6)P(Y=6)}
\end{aligned}
$$

- To classify, we'll simply compute these probabilities, one per class, and predict based on which one is largest

Model Parameters

- How many parameters are required to specify the likelihood, $P(X_1, \dots, X_n|Y)$?
  - (Supposing that each image is 30x30 pixels)

- The problem with explicitly modeling $P(X_1, \dots, X_n|Y)$ is that there are usually way too many parameters:
  - We'll run out of space
  - We'll run out of time
  - And we'll need tons of training data (which is usually not available)

Naïve Bayes

- Naïve Bayes assumption:
  - Features are independent given class:

$$P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y)$$
$$= P(X_1|Y)P(X_2|Y)$$

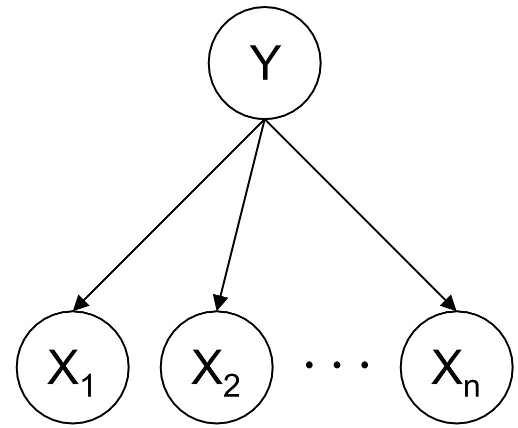
  - More generally:

$$\boxed{P(X_1 ... X_n|Y) = \prod_i P(X_i|Y)}$$

- How many parameters now?
  - Suppose **X** is composed of *n* binary features

The Naïve Bayes Classifier

- Given:
  - Prior P(Y)
  - *n* conditionally independent features $X_1, \ldots, X_1$, given the class Y
  - For each feature *i*, we specify $P(X_i|Y)$



- Classification decision rule:

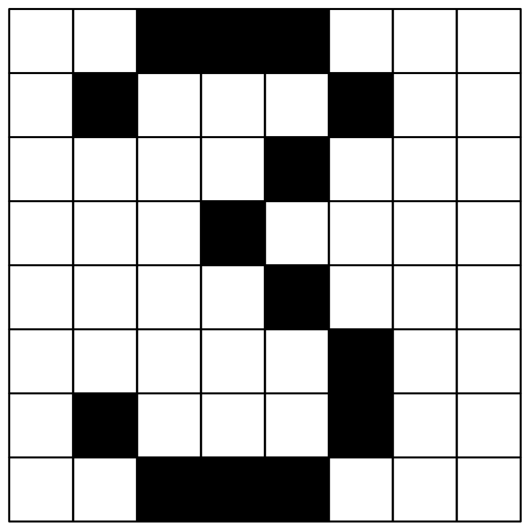
$$y^* = h_{NB}(\mathbf{x}) \;=\; \arg\max_y P(y)P(x_1,\ldots,x_n \mid y)$$

$$\;=\; \arg\max_y P(y)\prod_i P(x_i|y)$$

If certain assumption holds, NB is optimal classifier!
(they typically don't)

A Digit Recognizer

- Input: pixel grids



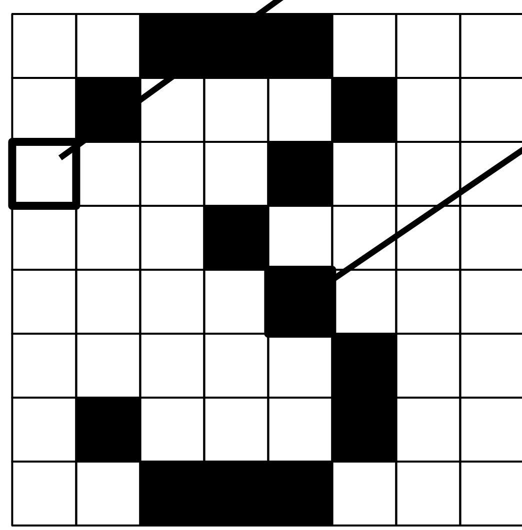
- Output: a digit 0-9

Are the naïve Bayes assumptions realistic here?

What has to be learned?

$P(Y)$

1	0.1
2	0.1
3	0.1
4	0.1
5	0.1
6	0.1
7	0.1
8	0.1
9	0.1
0	0.1



$P(F_{3,1} = on|Y)$

1	0.01
2	0.05
3	0.05
4	0.30
5	0.80
6	0.90
7	0.05
8	0.60
9	0.50
0	0.80

$P(F_{5,5} = on|Y)$

1	0.05
2	0.01
3	0.90
4	0.80
5	0.90
6	0.90
7	0.25
8	0.85
9	0.60
0	0.80

MLE for the parameters of NB

- Given dataset
  - Count(A=a,B=b) □ number of examples where A=a and B=b
- MLE for discrete NB, simply:
  - Prior:

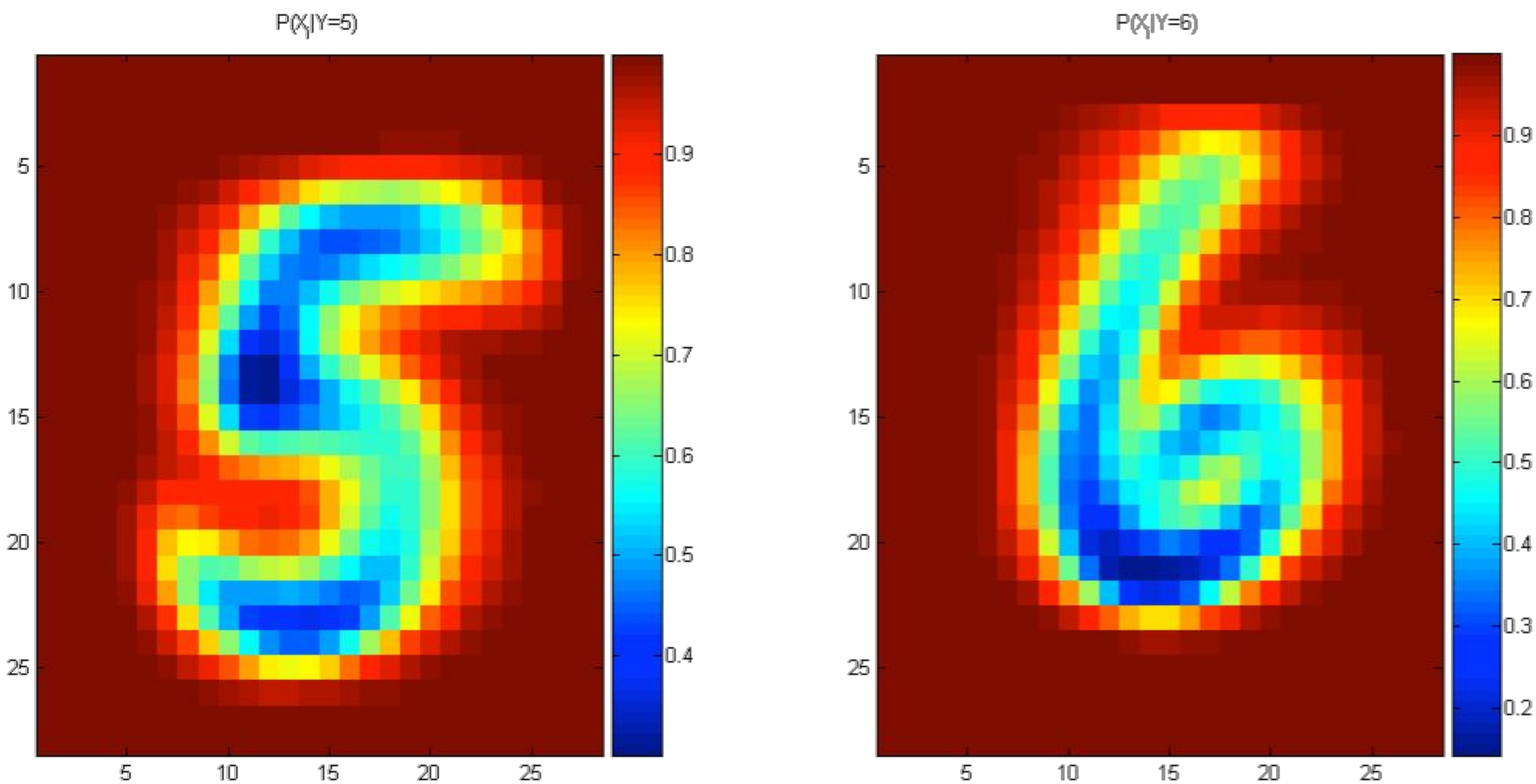
$$P(Y = y) = \frac{Count(Y = y)}{\sum_{y'} Count(Y = y')}$$

  - Observation distribution:

$$P(X_i = x | Y = y) = \frac{Count(X_i = x, Y = y)}{\sum_{x'} Count(X_i = x', Y = y)}$$

MLE for the parameters of NB

- Training amounts to, for each of the classes, averaging all of the examples together:



Using the Naïve Bayes Classifier

- Now, we have

$$P(Y = y_k \mid X = \mathbf{x}_i) = \frac{P(Y = y_k) \prod_{j=1}^{d} P(X_j = x_{i,j} \mid Y = y_k)}{\boxed{P(X = \mathbf{x}_i)}}$$

This is constant for a given instance, and so irrelevant to our prediction

– In practice, we use log-probabilities to prevent underflow

- To classify a new point **x**,

$$h(\mathbf{x}) = \arg\max_{y_k} \; P(Y = y_k) \prod_{j=1}^{d} P(X_j = \underbrace{x_j}_{j^{\text{th}} \text{ attribute value of } \mathbf{x}} \mid Y = y_k)$$

$$= \arg\max_{y_k} \; \log P(Y = y_k) + \sum_{j=1}^{d} \log P(X_j = x_j \mid Y = y_k)$$

The Naïve Bayes Classifier Algorithm

- For each class label y_k
  - Estimate $P(Y = y_k)$ from the data
  - For each value $x_{i,j}$ of each attribute X_i
    - Estimate $P(X_i = x_{i,j} \mid Y = y_k)$
- Classify a new point via:

$$h(\mathbf{x}) = \arg\max_{y_k} \log P(Y = y_k) + \sum_{j=1}^{d} \log P(X_j = x_j \mid Y = y_k)$$

- In practice, the independence assumption doesn't often hold true, but Naïve Bayes performs very well despite this

Naïve Bayes: Subtlety #1

Often the X_i are not really conditionally independent

- We use Naïve Bayes in many cases anyway, and it often works pretty well
  - often the right classification, even when not the right probability (see [Domingos&Pazzani, 1996])
- What is effect on estimated P(Y|X)?
  - Extreme case: what if we add two copies: $X_i = X_k$

Naïve Bayes: Subtlety 2 Zero Counting

- Notice that some probabilities estimated by counting might be zero
  - Possible overfitting!

- Fix by using Laplace Smoothing
  - Adds 1 to each count

$$P(X_j = v \mid Y = y_k) = \frac{c_v + 1}{\sum_{v' \in \text{values}(X_j)} c_{v'} + |\text{values}(X_j)|}$$

where

- c_v is the count of training instances with a value of v for attribute j and class label y_k
- $|\text{values}(X_j)|$ is the number of values X_j can take on

MAP estimation for NB

- Given dataset
  - Count(A=a,B=b) number of examples where A=a and B=b
- MAP estimation for discrete NB, simply:
  - Prior:

$$P(Y=y) = \frac{Count(Y=y)}{\sum_{y'} Count(Y=y')}$$

  - Observation distribution:

$$P(X_i = x|Y=y) = \frac{Count(X_i = x, Y=y) \;\mathbf{+\ a}}{\sum_{x'} Count(X_i = x', Y=y) \;\mathbf{+\ |X\_i|{*}a}}$$

- Called “smoothing”. Corresponds to Dirichlet prior!

Training Naïve Bayes (Example 2)

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

P(play) = ?

P(Sky = sunny | play) = ?

P(Humid = high | play) = ?

...

P($\neg$play) = ?

P(Sky = sunny | $\neg$play) = ?

P(Humid = high | $\neg$play) = ?

...

Training Naïve Bayes with Laplace Smoothing

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting with Laplace smoothing:

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny						*yes*
sunny						*yes*
rainy	cold	high	strong	warm	change	*no*
sunny						*yes*

P(play) = 3/4 P($\neg$play) = 1/4

P(Sky = sunny | play) = 4/5

P(Humid = high | play) = ?

...

P(Sky = sunny | $\neg$play) = ?

P(Humid = high | $\neg$play) = ?

...

Training Naïve Bayes with Laplace Smoothing

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting with Laplace smoothing:

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy						*no*
sunny	warm	high	strong	cool	change	*yes*

P(play) = ¾

P($\neg$play) = 1/4

P(Sky = sunny | play) = 4/5

P(Humid = high | play) = ?

...

P(Sky = sunny | $\neg$play) = 1/3

P(Humid = high | $\neg$play) = ?

...

Training Naïve Bayes with Laplace Smoothing

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting with Laplace smoothing:

Sky	Temp	Humid	Wind	Water	Forecast	Play?
		normal				*yes*
		high				*yes*
rainy	cold	high	strong	warm	change	*no*
		high				*yes*

P(play) = ¾

P(¬play) = 1/4

P(Sky = sunny | play) = 4/5

P(Humid = high | play) = 3/5

P(Sky = sunny | ¬play) = 1/3

P(Humid = high | ¬play) = ?

...

...

Training Naïve Bayes with Laplace Smoothing

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting with Laplace smoothing:

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
		high				
sunny	warm	high	strong	cool	change	*yes*

P(play) = ¾

P(¬play) = 1/4

P(Sky = sunny | play) = 4/5

P(Humid = high | play) = 3/5

...

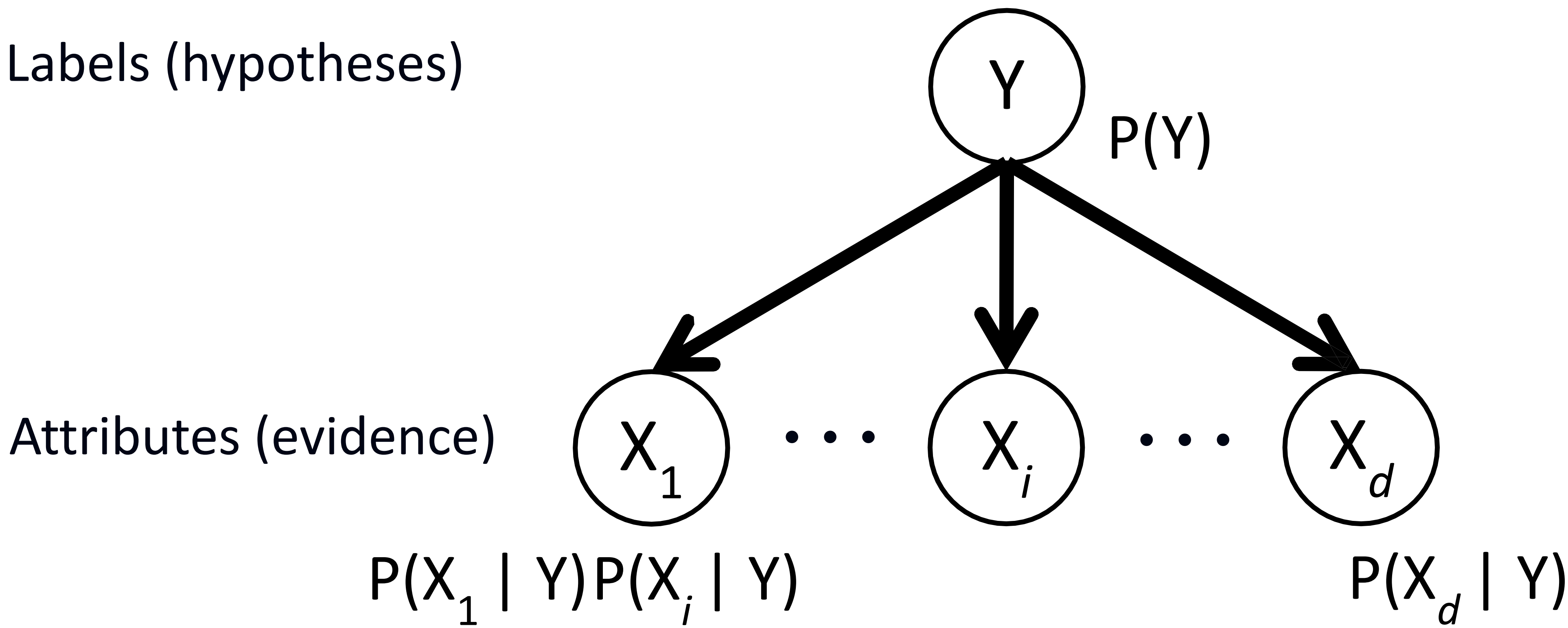
P(Sky = sunny | ¬play) = 1/3

P(Humid = high | ¬play) = 2/3

...

Extra Slides with Full Example

The Naïve Bayes Graphical Model

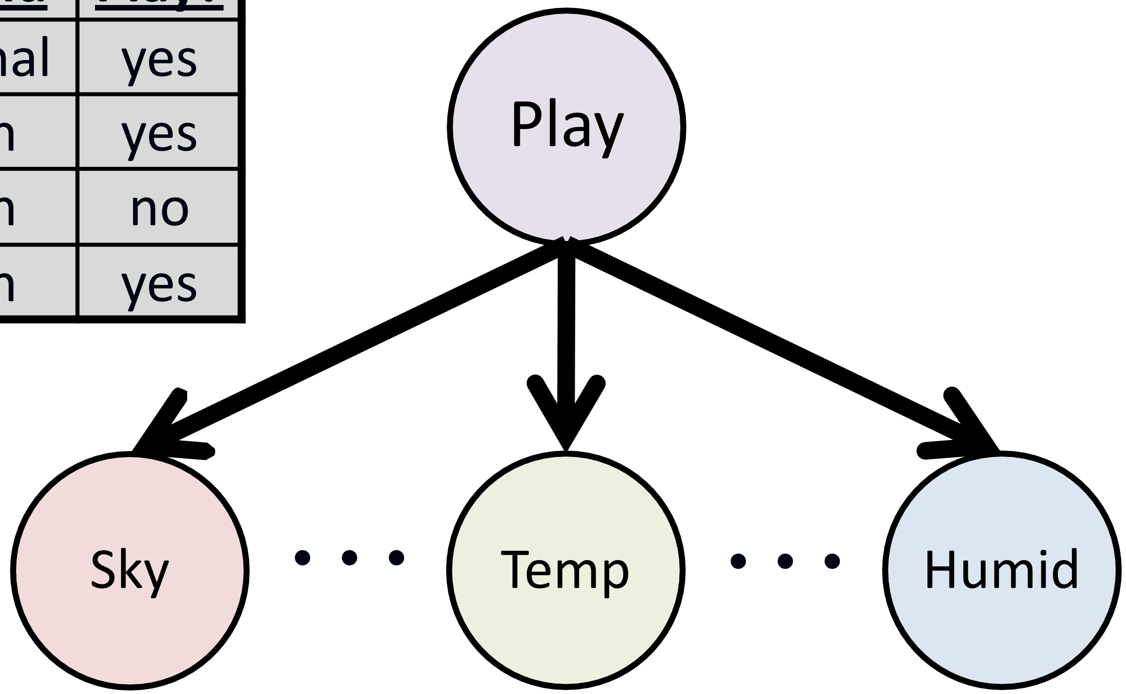


- Nodes denote random variables
- Edges denote dependency
- Each node has an associated conditional probability table (CPT), conditioned upon its parents

Example NB Graphical Model

Data:

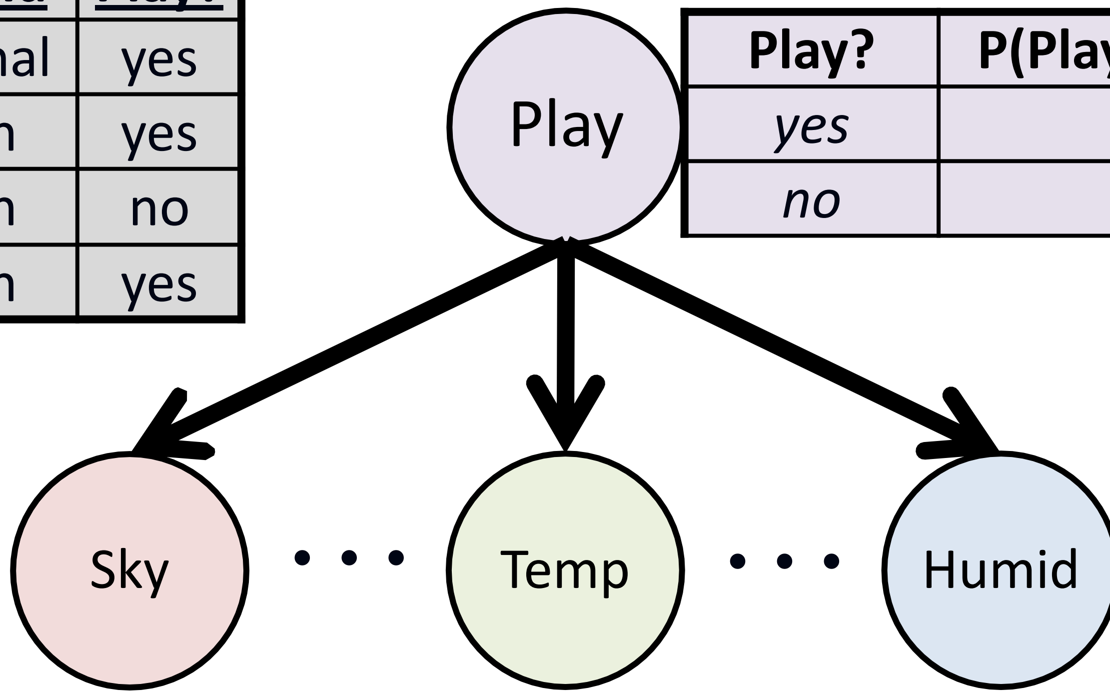
Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes



Example NB Graphical Model

Data:

Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes

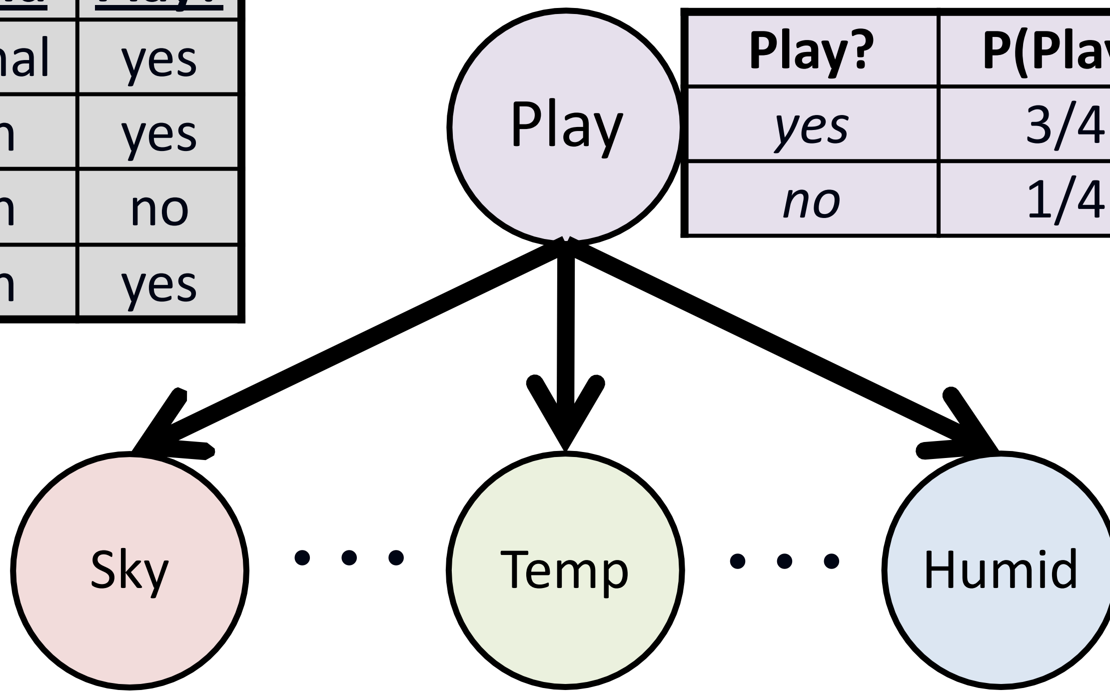


Play?	P(Play)
yes	
no	

Example NB Graphical Model

Data:

<u>**Sky**</u>	<u>**Temp**</u>	<u>**Humid**</u>	<u>**Play?**</u>
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes

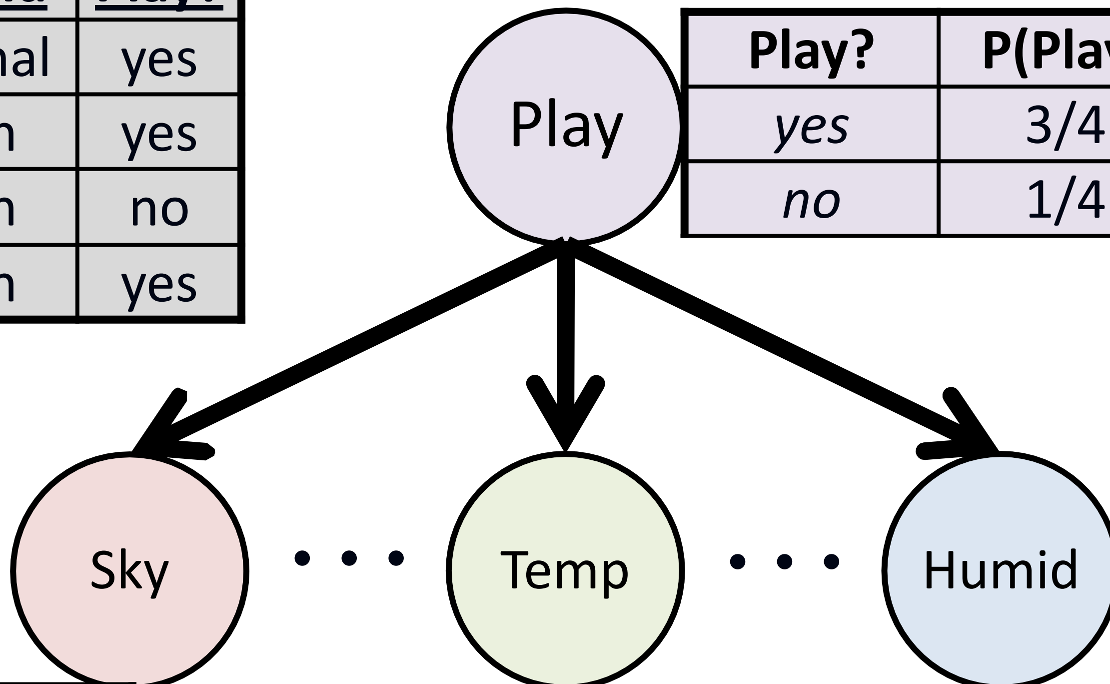


Play?	**P(Play)**
yes	3/4
no	1/4

Example NB Graphical Model

Data:

Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes



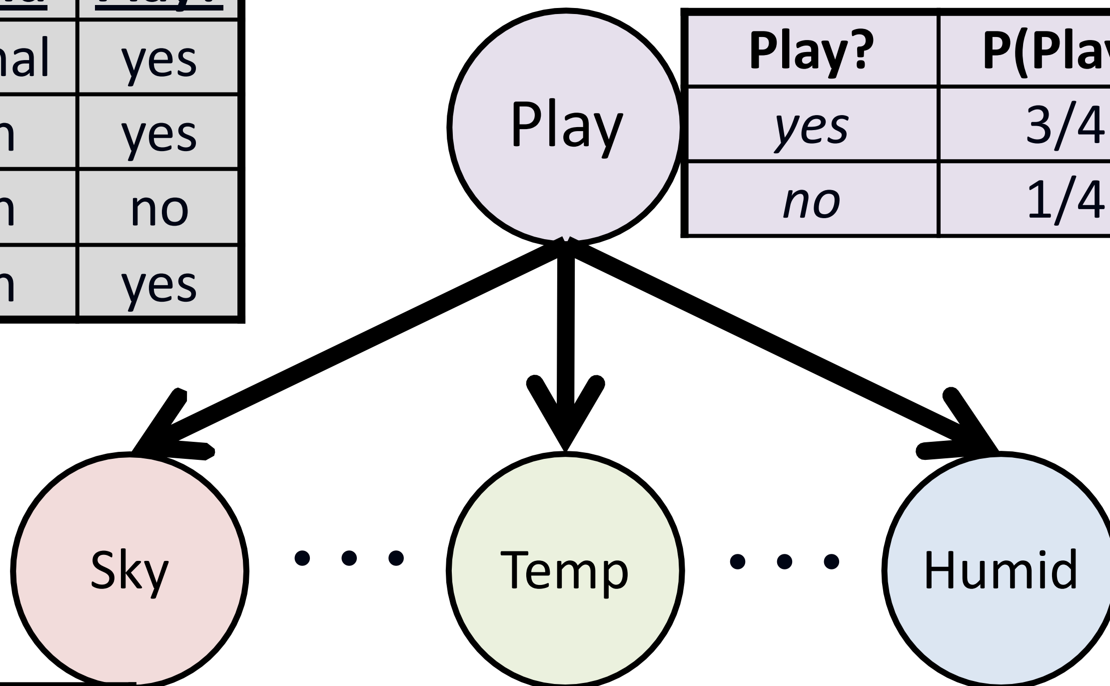
Play?	P(Play)
yes	3/4
no	1/4

Sky	Play?	P(Sky | Play)
sunny	*yes*	
rainy	*yes*	
sunny	*no*	
rainy	*no*	

Example NB Graphical Model

Data:

Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes



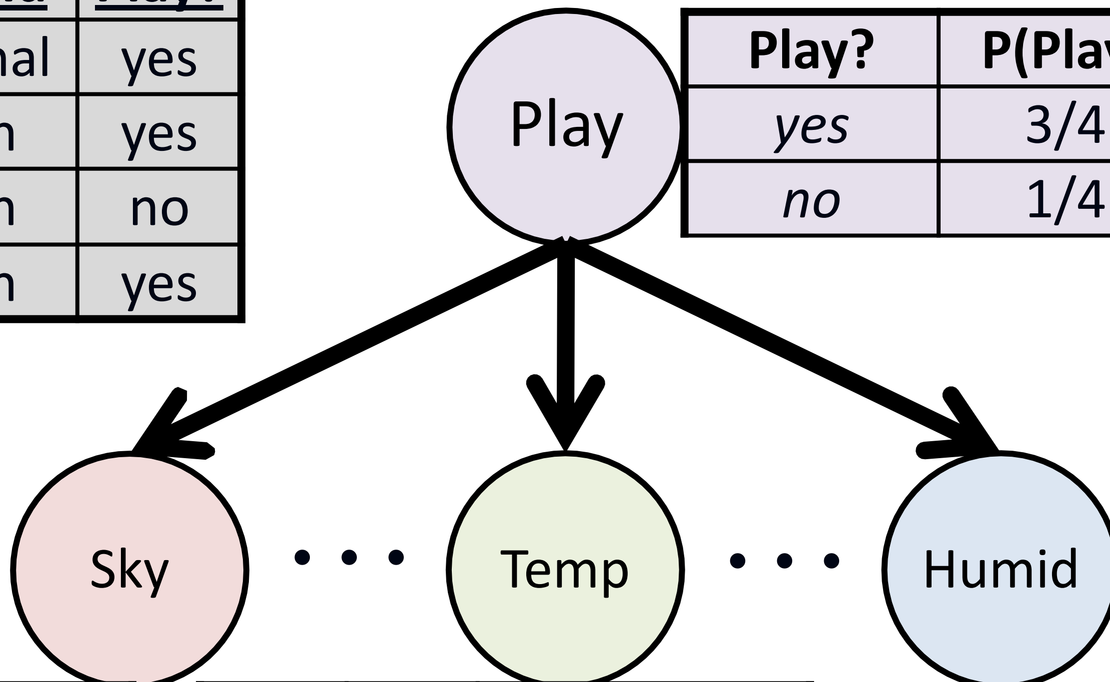
Play?	P(Play)
yes	3/4
no	1/4

Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

Example NB Graphical Model

Data:

Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes



Play?	P(Play)
yes	3/4
no	1/4

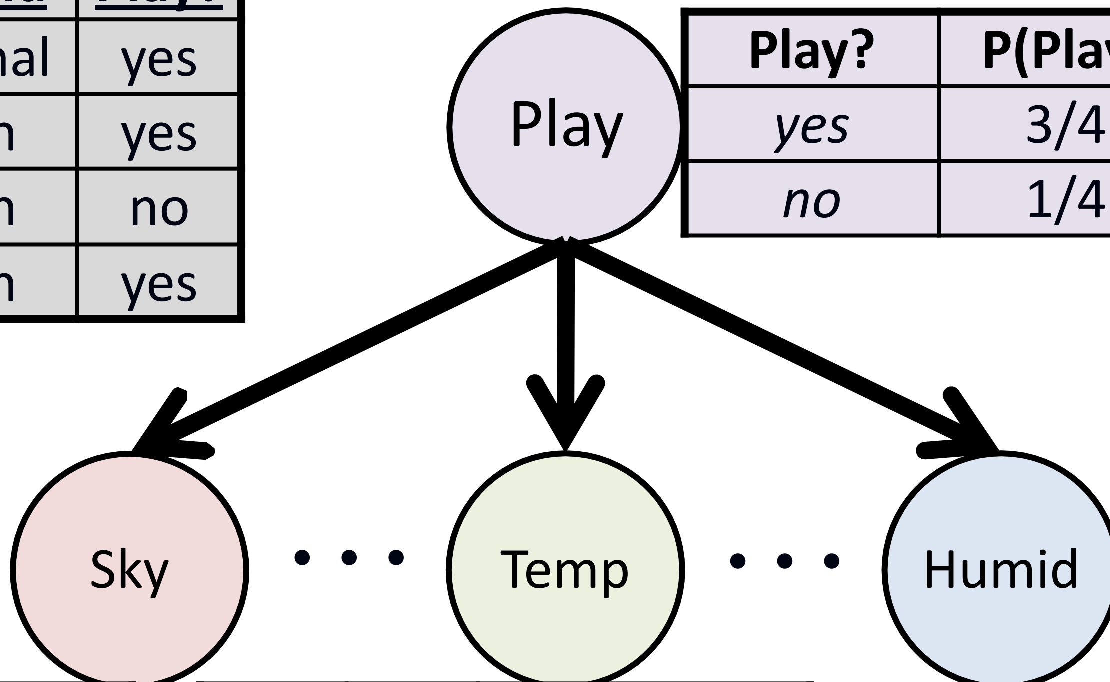
Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

Temp	Play?	P(Temp | Play)
warm	*yes*	
cold	*yes*	
warm	*no*	
cold	*no*	

Example NB Graphical Model

Data:

Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes



Play?	P(Play)
yes	3/4
no	1/4

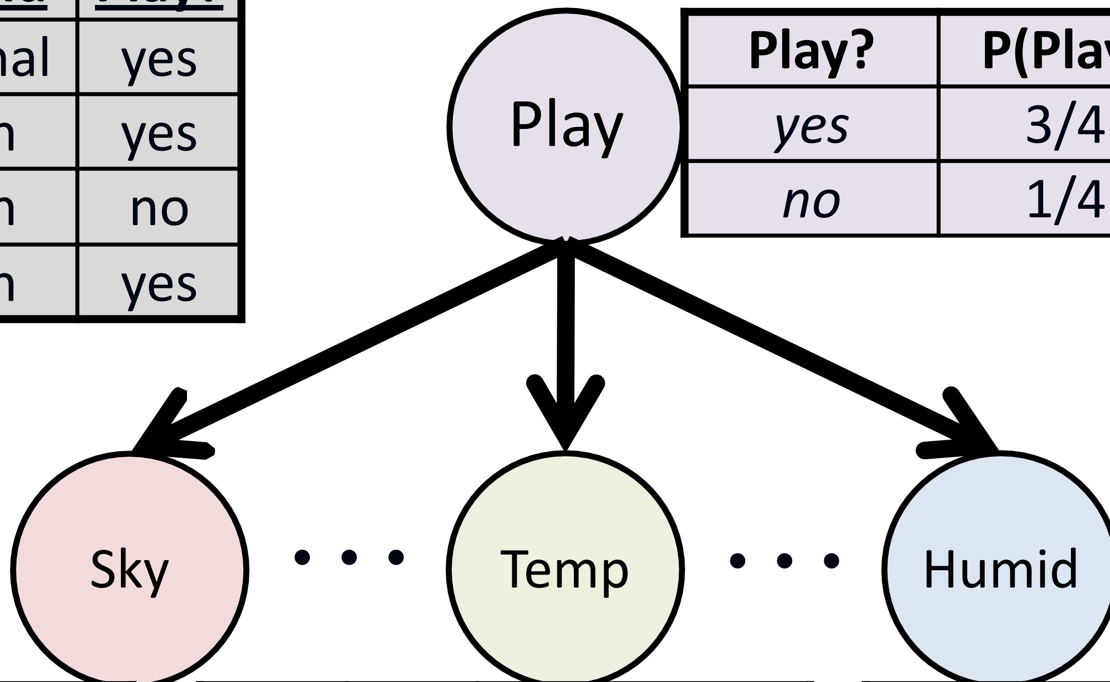
Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

Temp	Play?	P(Temp | Play)
warm	*yes*	4/5
cold	*yes*	1/5
warm	*no*	1/3
cold	*no*	2/3

Example NB Graphical Model

Data:

Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes



Play?	P(Play)
yes	3/4
no	1/4

Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

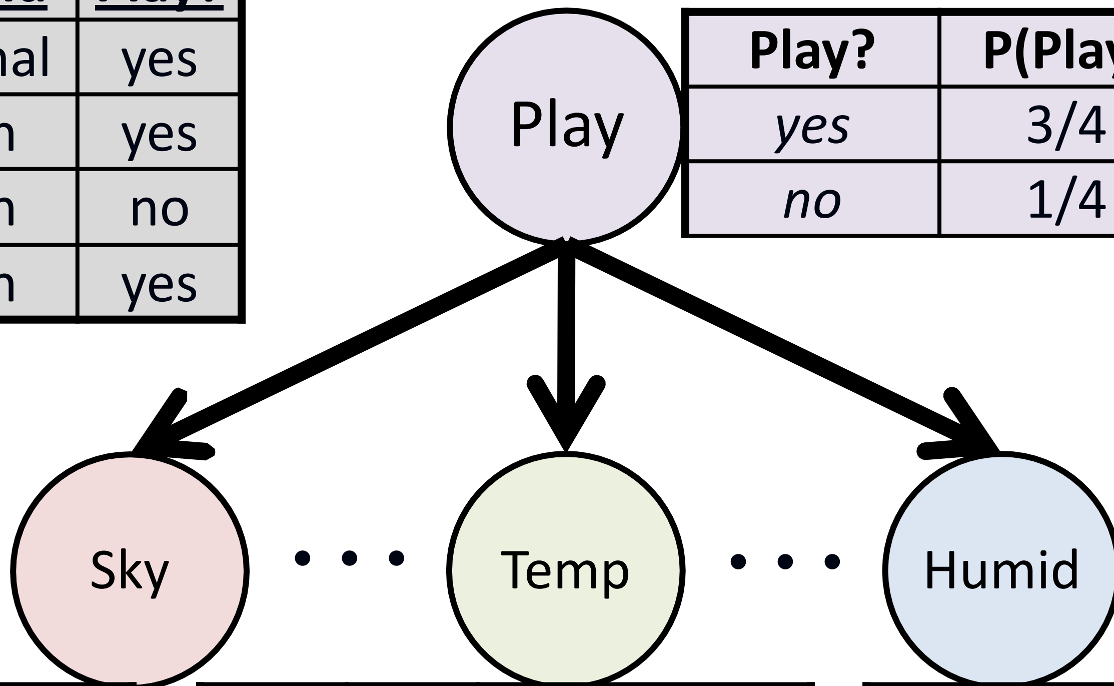
Temp	Play?	P(Temp | Play)
warm	*yes*	4/5
cold	*yes*	1/5
warm	*no*	1/3
cold	*no*	2/3

Humid	Play?	P(Humid | Play)
high	*yes*	
norm	*yes*	
high	*no*	
norm	*no*	

Example NB Graphical Model

Data:

Sky	Temp	Humid	Play?
sunny	warm	normal	yes
sunny	warm	high	yes
rainy	cold	high	no
sunny	warm	high	yes



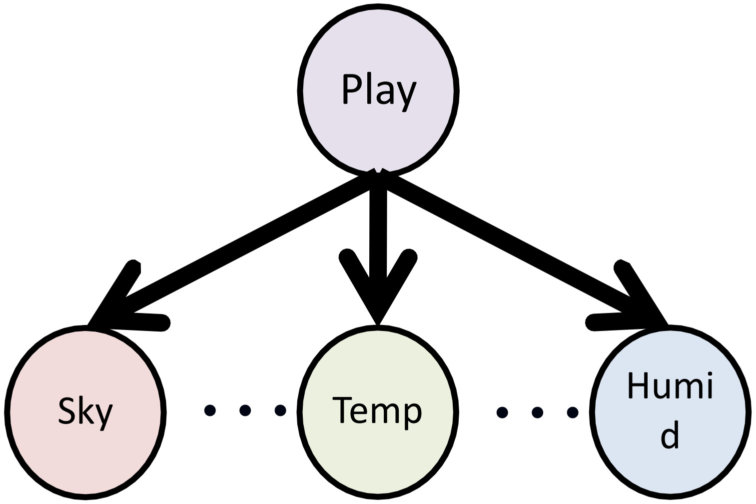
Play?	P(Play)
yes	3/4
no	1/4

Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

Temp	Play?	P(Temp | Play)
warm	*yes*	4/5
cold	*yes*	1/5
warm	*no*	1/3
cold	*no*	2/3

Humid	Play?	P(Humid | Play)
high	*yes*	3/5
norm	*yes*	2/5
high	*no*	2/3
norm	*no*	1/3

Example Using NB for Classification



Play?	P(Play)
yes	3/4
no	1/4

Temp	Play?	P(Temp | Play)
warm	*yes*	4/5
cold	*yes*	1/5
warm	*no*	1/3
cold	*no*	2/3

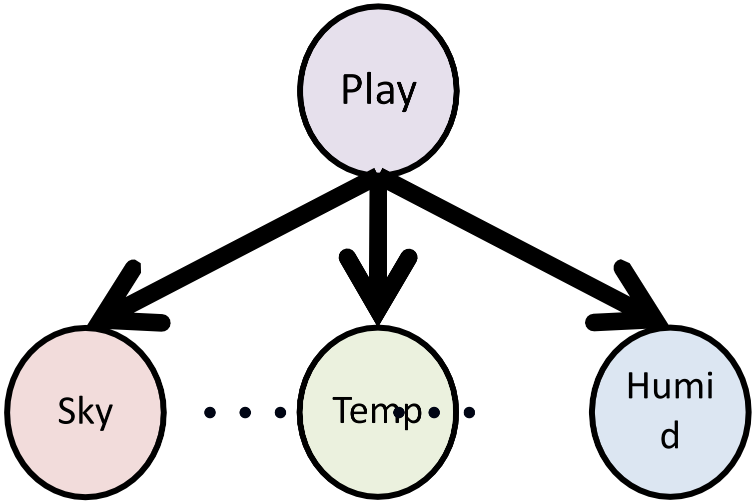
Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

Humid	Play?	P(Humid | Play)
high	*yes*	3/5
norm	*yes*	2/5
high	*no*	2/3
norm	*no*	1/3

$$h(\mathbf{x}) = \arg\max_{y_k} \; \log P(Y = y_k) + \sum_{j=1}^{d} \log P(X_j = x_j \mid Y = y_k)$$

Goal: Predict label for **x** = (rainy, warm, normal)

Example Using NB for Classification



Play?	P(Play)
yes	3/4
no	1/4

Temp	Play?	P(Temp | Play)
warm	*yes*	4/5
cold	*yes*	1/5
warm	*no*	1/3
cold	*no*	2/3

Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

Humid	Play?	P(Humid | Play)
high	*yes*	3/5
norm	*yes*	2/5
high	*no*	2/3
norm	*no*	1/3

Predict label for:

x = (rainy, warm, normal)

$$P(\text{play} \mid \mathbf{x}) \propto \log P(\text{play}) + \log P(\text{rainy} \mid \text{play}) + \log P(\text{warm} \mid \text{play}) + \log P(\text{normal} \mid \text{play})$$

$$\propto \log 3/4 + \log 1/5 + \log 4/5 + \log 2/5 = \boxed{-1.319}$$

predict PLA Y

$$P(\neg\text{play} \mid \mathbf{x}) \propto \log P(\neg\text{play}) + \log P(\text{rainy} \mid \neg\text{play}) + \log P(\text{warm} \mid \neg\text{play}) + \log P(\text{normal} \mid \neg\text{play})$$

$$\propto \log 1/4 + \log 2/3 + \log 1/3 + \log 1/3 = -1.732$$