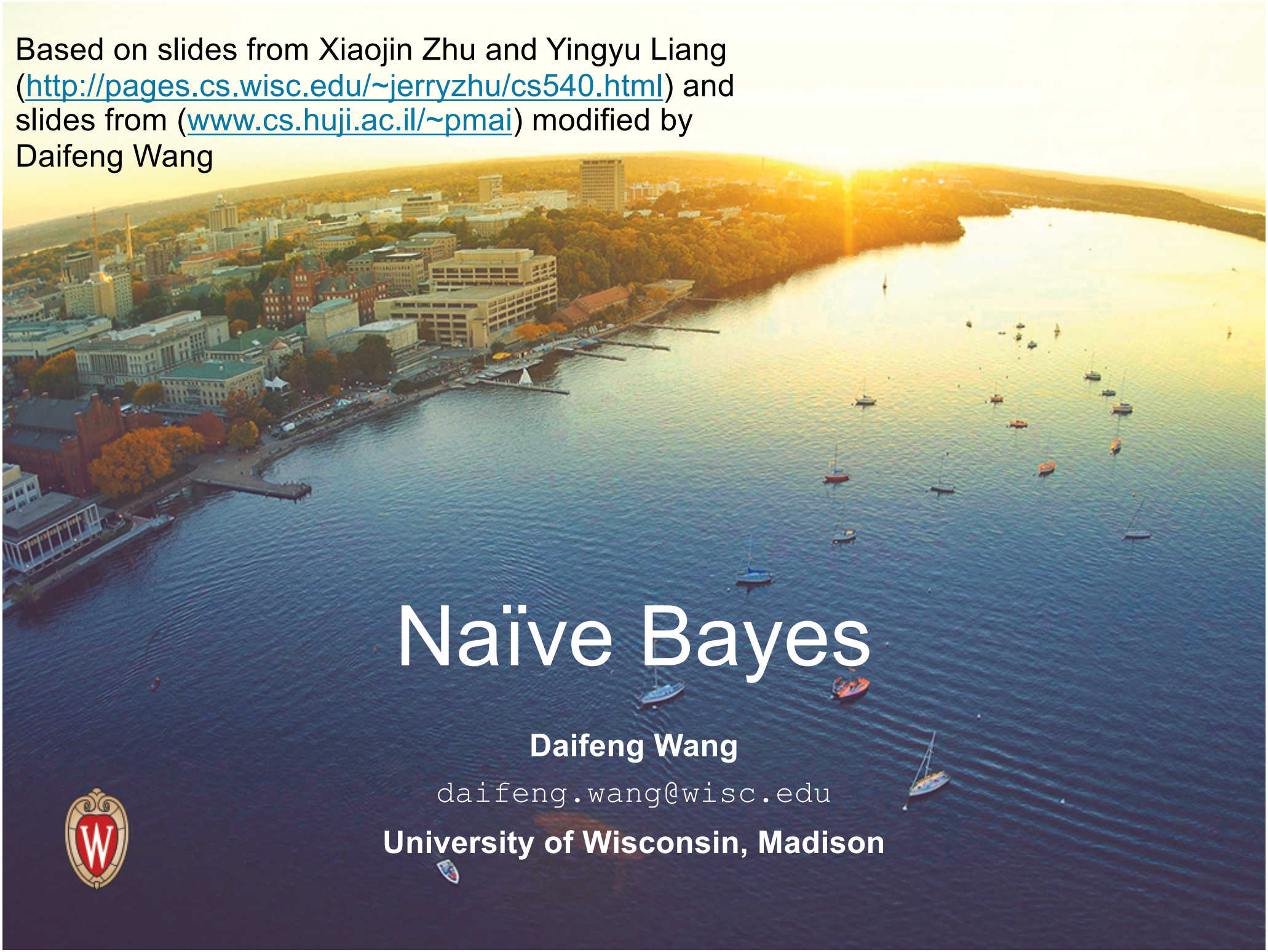
Based on slides from Xiaojin Zhu and Yingyu Liang (http://pages.cs.wisc.edu/~jerryzhu/cs540.html) and slides from (www.cs.huji.ac.il/~pmai) modified by Daifeng Wang



Naïve Bayes

Daifeng Wang

daifeng.wang@wisc.edu

University of Wisconsin, Madison

Outline

- Maximum Likelihood Estimation (MLE)
- Maximum a posteriori (MAP) estimate
- Naïve Bayes
- Various Naïve Bayes models
  - model 1: Bernoulli Naïve Bayes
  - model 2: Multinomial Naïve Bayes
  - model 3: Gaussian Naïve Bayes
  - model 4: Multiclass Naïve Bayes


MLE and MAP

Flip a coin again…

- Flip a coin N=10 times
- N_H= 4 Heads, N_T= 6 Tails
- How can you estimate $\theta = \mathrm{P}(\mathrm{Head})$?
  - Intuitively, $\theta = \frac{4}{10} = 0.4$
  - How do you confirm?
- Any θ can get 4 heads and 6 tails for 10 flips
  - e.g., if $\theta = 0.5$, then HTHHTTTTTHTHH

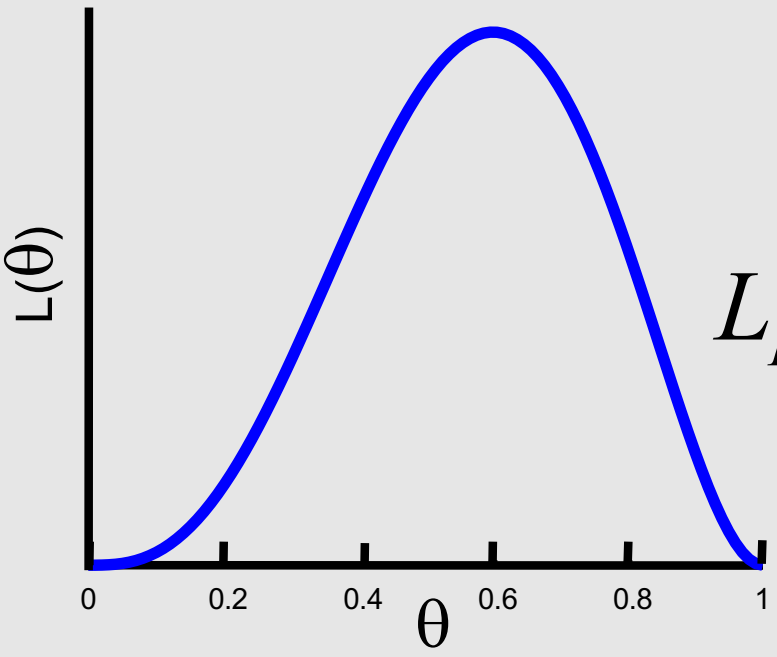
- Given a sequence of toss samples x[1], x[2], …, x[N], we want to estimate the probabilities P(H)= θ and P(T) = 1 - θ
  - Bernoulli distribution

Likelihood Function

- How good is a particular θ? It depends on how likely it is to generate the observed data D={x[i], i=1,2,…,N}

$$L_D(\theta) = P(D \mid \theta) = \prod_m P(x[m] \mid \theta)$$

- The likelihood for the sequence H,T, T, H, H is



$$L_D(\theta) = \theta \cdot (1-\theta) \cdot (1-\theta) \cdot \theta \cdot \theta$$

Log-likelihood function

- To calculate the likelihood in the coin example we only require N_H and N_T (the number of heads and the number of tails)

$$L_D(\theta) = \theta^{N_H} \cdot (1-\theta)^{N_T}$$

- log-likelihood function

$$l_D(\theta) = \log L_D(\theta)$$
$$= N_H \log \theta + N_T \log(1-\theta)$$

Maximum Likelihood Estimation (MLE)

- Find optimal θ^* to maximize the likelihood function (and log-likelihood function)

$$\theta^* = argmax_\theta P(D|\theta)$$

- for flipping a coin

$$\theta^* = argmax_\theta l_D(\theta)$$
$$= argmax_\theta \mathrm{N_H} \log \theta + \mathrm{NT} \log(1-\theta)$$

- $\frac{\partial l_D(\theta)}{\partial \theta} = \frac{N_H}{\theta} - \frac{N_T}{1-\theta} = \frac{N_H - \theta N}{\theta(1-\theta)} = 0 \Rightarrow \theta^* = \frac{N_H}{N}$

which confirms your intuition!

Optional: MLE of Exponential Distribution

- pdf of Exponential(λ): $f(x) = \lambda e^{-\lambda x}$
- Suppose $X_i \sim$ Exponential(λ) for $1 \leq i \leq N$.
- Find MLE for data $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$
- First write down log-likelihood of sample.
- Compute first derivative, set to zero, solve for λ.
- Compute second derivative and check that it is concave down at λ^{MLE}.

Optional: MLE of Exponential Distribution

- First write down log-likelihood of sample.

$$\ell(\lambda) = \sum_{i=1}^{N} \log f(x^{(i)}) \tag{1}$$

$$= \sum_{i=1}^{N} \log(\lambda \exp(-\lambda x^{(i)})) \tag{2}$$

$$= \sum_{i=1}^{N} \log(\lambda) + -\lambda x^{(i)} \tag{3}$$

$$= N \log(\lambda) - \lambda \sum_{i=1}^{N} x^{(i)} \tag{4}$$

Optional: MLE of Exponential Distribution

- Compute first derivative, set to zero, solve for λ.

$$\frac{d\ell(\lambda)}{d\lambda} = \frac{d}{d\lambda} N \log(\lambda) - \lambda \sum_{i=1}^{N} x^{(i)} \quad (1)$$

$$= \frac{N}{\lambda} - \sum_{i=1}^{N} x^{(i)} = 0 \quad (2)$$

$$\Rightarrow \lambda^{\text{MLE}} = \frac{N}{\sum_{i=1}^{N} x^{(i)}} \quad (3)$$

Maximum a posteriori (MAP) estimation

Suppose we have data $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$

$$\theta^*_{MAP} = argmax_\theta \; P(\theta|\mathcal{D})$$

← Posterior

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

Bayes rule

$$= argmax_\theta P(\mathcal{D}|\theta)P(\theta)$$

MLE vs. MAP

Suppose we have data $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$

Maximum Likelihood Estimate (MLE)

$$\boldsymbol{\theta}^{\text{MLE}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta})$$

Maximum *a posteriori* (MAP) estimation

$$\boldsymbol{\theta}^{\text{MAP}} = \underset{\boldsymbol{\theta}}{\text{argmax}} \prod_{i=1}^{N} p(\mathbf{x}^{(i)}|\boldsymbol{\theta}) p(\boldsymbol{\theta})$$

Prior


Naïve Bayes

Play outside or not?

- If weather is sunny, would you like to play outside?

Posterior probability P(Yes|Sunny) vs P(No|Sunny)

- Weather = {Sunny, Rainy, Overcast}
- Play = {Yes, No}
- Observed data {Weather, Play on Day m}, m=1,2,…,N

How can we calculate posterior probabilities?

$$P(Play|Weather) = \frac{P(Weather|Play)P(Play)}{P(Weather)}$$

Bayes rule

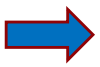
Play outside or not?

- **Step 1**: Convert the data to a frequency table of Weather and Play

- **Step 2**: Based on the frequency table, calculate likelihoods $P(Weather|Play)$ and priors $P(Play)$

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9



Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

$$e.g., P(Play = Yes) = 0.64$$
$$P(Sunny|Yes) = \frac{3}{9} = 0.33$$

https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

Play outside or not?

- **Step 3**: Based on the likelihoods and priors, calculate posteriors $P(\text{Play} \mid \text{Weather})$

- P(Yes|Sunny)
  =P(Sunny|Yes)*P(Yes)/P(Sunny)
  =0.33*0.64/0.36
  =0.6

- P(No|Sunny)
  =P(Sunny|No)*P(No)/P(Sunny)
  =0.4*0.36/0.36
  =0.4

- P(Yes|Sunny)>P(No|Sunny), you should go outside and play!

Bayesian classification

- Given the data $X=\{X_1,X_2,\ldots,X_k\}$ with k attributes
  - e.g., X={Weather, Wind, Traffic, ….}
- L classes you want to classify: Y_1, Y_2, …, Y_L
- Bayesian classification predicts X to Class Y_I if

  $P(Y_I|X)$ is max of $\{P(Y_i|X), i=1, \ldots, L\}$
  - e.g., P(Play=Yes|Sunny) = 0.6
  - However, it is very computationally expensive for

$$P(X_1,\ldots,X_K,Y) = P(X_1,\ldots,X_K|Y)P(Y)$$

Likelihood is hard to calculate for many attributes

Naïve Bayes Assumption

Conditional independence of features:

$$P(X_1, \ldots, X_K, Y) = P(X_1, \ldots, X_K | Y) P(Y)$$

$$= \left(\prod_{k=1}^{K} P(X_k | Y) \right) P(Y)$$

Naïve Bayes Assumption

Assuming conditional independence, the conditional probabilities encode the **same information** as the joint table.

They are very convenient for estimating

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

They are almost as good for computing

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

$$\forall \mathbf{x}, y: P(Y=y|X_1,\ldots,X_n=\mathbf{x})=\frac{P(X_1,\ldots,X_n=\mathbf{x}|Y)P(Y=y)}{P(X_1,\ldots,X_n=\mathbf{x})}$$

Generic Naïve Bayes Model

Support: Depends on the choice of **event model,** $P(X_k|Y)$

Model: Product of **prior** and the event model

$$P(\mathbf{X}, Y) = P(Y) \prod_{k=1}^{K} P(X_k|Y)$$

Training: Find the **class-conditional** MLE parameters
For $P(Y)$, we find the MLE using all the data. For each $P(X_k|Y)$ we condition on the data with the corresponding class.

Classification: Find the class that maximizes the posterior

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, p(y|\mathbf{x})$$

Generic Naïve Bayes Model

Classification:

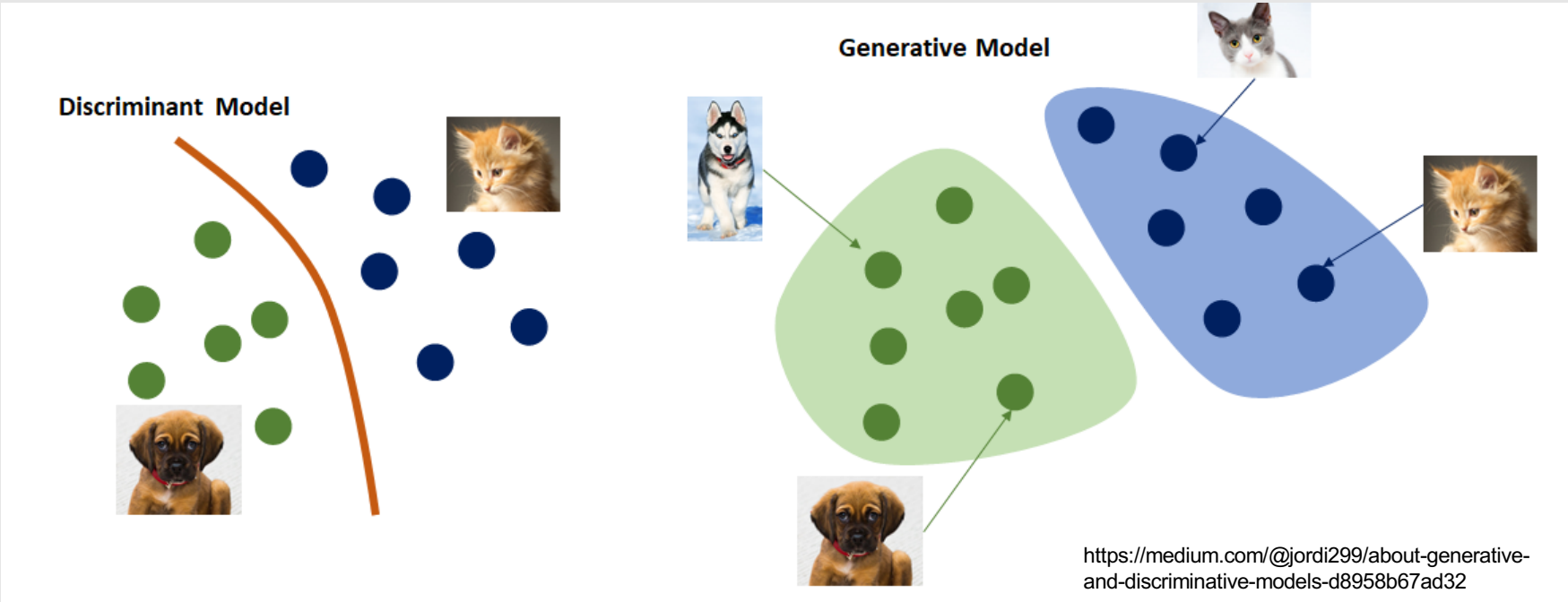
$$\hat{y} = \underset{y}{\operatorname{argmax}}\, p(y|\mathbf{x}) \quad \text{(posterior)}$$

$$= \underset{y}{\operatorname{argmax}}\, \frac{p(\mathbf{x}|y)p(y)}{p(x)} \quad \text{(by Bayes' rule)}$$

$$= \underset{y}{\operatorname{argmax}}\, p(\mathbf{x}|y)p(y)$$

Generative vs. Discriminative models

- Generative approaches model the joint probability p(x,y) for generating data
  - e.g., Naïve Bayes calculates p(y) and p(x|y) and can generate y data from p(y) and x samples from p(x|y)
- Discriminative approaches directly model p(y|x) for classification



Various Naïve Bayes Models

Generic Naïve Bayes Model

Recall...

Classification:

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, p(y|\mathbf{x}) \quad \text{(posterior)}$$

$$= \underset{y}{\operatorname{argmax}}\, \frac{p(\mathbf{x}|y)p(y)}{p(x)} \quad \text{(by Bayes' rule)}$$

$$= \underset{y}{\operatorname{argmax}}\, p(\mathbf{x}|y)p(y)$$

How to define and estimate likelihoods and priors?

Model 1: Bernoulli Naïve Bayes

Support: Binary vectors of length K

$$\mathbf{x} \in \{0,1\}^K$$

Generative Story:

$$Y \sim \text{Bernoulli}(\phi)$$

$$X_k \sim \text{Bernoulli}(\theta_{k,Y}) \; \forall k \in \{1,\ldots,K\}$$

Model:

$$\begin{aligned} p_{\phi,\boldsymbol{\theta}}(\boldsymbol{x}, y) &= p_{\phi,\boldsymbol{\theta}}(x_1, \ldots, x_K, y) \\ &= p_\phi(y) \prod_{k=1}^{K} p_{\boldsymbol{\theta}_k}(x_k|y) \\ &= (\phi)^y (1-\phi)^{(1-y)} \prod_{k=1}^{K} (\theta_{k,y})^{x_k} (1-\theta_{k,y})^{(1-x_k)} \end{aligned}$$

Model 1: Bernoulli Naïve Bayes

- Classify documents into y=1 for 'sports' and y=0 for 'non sports'
  - Bernoulli distribution ϕ=P(y=1)
- A document can be represented by a binary vector x=[x_1, x_2, …, x_K] with K words in the vocabulary
  - x_k = 1 if k^{th} word in the document; x_k=0, otherwise with Bernoulli distributions $\theta_{k,1}$=$P_1(x_k=1)$ for 'sports' and $\theta_{k,0}$ =$P_0(x_k=1)$ for 'non sports'
- Likelihood

$$P(x|y) = \prod_{k=1}^{K} (\theta_{k,y})^{x_k} (1-\theta_{k,y})^{(1-x_k)}$$

- Prior

$$P(y) = (\phi)^{y} (1-\phi)^{(1-y)}$$

How to estimate θ parameters of likelihoods and priors?

Model 1: Bernoulli Naïve Bayes

Support: Binary vectors of length K

$$\mathbf{x} \in \{0,1\}^K$$

Generative Story:

$$Y \sim \text{Bernoulli}(\phi)$$

$$X_k \sim \text{Bernoulli}(\theta_{k,Y}) \; \forall k \in \{1, \ldots, K\}$$

Model: $p_{\phi,\boldsymbol{\theta}}(\boldsymbol{x}, y) = (\phi)^y (1-\phi)^{(1-y)} \prod_{k=1}$

Same as Generic Naïve Bayes

Classification: Find the class that maximizes the posterior

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, p(y|\mathbf{x})$$

Model 1: Bernoulli Naïve Bayes

Training: Find the **class-conditional** MLE parameters

For $P(Y)$, we find the MLE using all the data. For each $P(X_k|Y)$ we condition on the data with the corresponding class.

$$\phi = \frac{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1)}{N}$$

$$\theta_{k,0} = \frac{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0 \wedge x_k^{(i)} = 1)}{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0)}$$

$$\theta_{k,1} = \frac{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1 \wedge x_k^{(i)} = 1)}{\sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 1)}$$

$$\forall k \in \{1, \ldots, K\}$$

Model 1: Bernoulli Naïve Bayes

- Classify documents into y=1 for 'sports' and y=0 for 'non sports'
  - Bernoulli distribution ϕ=P(y=1)
- A document can be represented by a binary vector x=[x_1, x_2, …, x_K] with K words in the vocabulary
  - x_k = 1 if k^{th} word in the document; x_k=0, otherwise with Bernoulli distributions $\theta_{k,1}$=$P_1(x_k=1)$ for 'sports' and $\theta_{k,0}$=$P_0(x_k=1)$ for 'non sports'
- $\hat{\theta}_{k,1}$=Number of sports documents with k^{th} word / Number of sports documents
- $\hat{\theta}_{k,0}$=Number of non-sports documents with k^{th} word / Number of non-sports documents
- $\hat{\phi}$ = Number of sports documents / Number of documents
- Predict classes for a new document x*

$$P(y=1|x^*) \text{ vs. } P(y=0|x^*)$$

Model 2: Multinomial Naïve Bayes

Support: Integer vector (word IDs)

$$\mathbf{x} = [x_1, x_2, \ldots, x_M] \text{ where } x_m \in \{1, \ldots, K\} \text{ a word id.}$$

Generative Story:

for $i \in \{1, \ldots, N\}$**:**

$y^{(i)} \sim \text{Bernoulli}(\phi)$

for $j \in \{1, \ldots, M_i\}$**:** (Assume $M_i = M$ for all i)

$x_j^{(i)} \sim \text{Multinomial}(\boldsymbol{\theta}_{y^{(i)}}, 1)$

Model:

$$p_{\phi,\boldsymbol{\theta}}(\boldsymbol{x}, y) = p_\phi(y) \prod_{k=1}^{K} p_{\boldsymbol{\theta}_k}(x_k|y)$$

$$= (\phi)^y (1-\phi)^{(1-y)} \prod_{j=1}^{M_i} \theta_{y,x_j}$$

Model 3: Gaussian Naïve Bayes

Support: $\mathbf{x} \in \mathbb{R}^K$

Model: Product of **prior** and the event model

$$p(\boldsymbol{x}, y) = p(x_1, \ldots, x_K, y)$$

$$= p(y) \prod_{k=1}^{K} p(x_k|y)$$

Gaussian Naive Bayes assumes that $p(x_k|y)$ is given by a Normal distribution.

Model 4: Multiclass Naïve Bayes

Model:

The only change is that we permit y to range over C classes.

$$p(\boldsymbol{x}, y) = p(x_1, \ldots, x_K, y)$$

$$= p(y) \prod_{k=1}^{K} p(x_k|y)$$

Now, $y \sim$ Multinomial$(\boldsymbol{\phi}, 1)$ and we have a separate conditional distribution $p(x_k|y)$ for each of the C classes.


THANK YOU
Some of the slides in these lectures have been adapted/borrowed from materials developed by Yingyu Liang, Mark Craven, David Page, Jude Shavlik, Tom Mitchell, Nina Balcan, Matt Gormley, Elad Hazan, Tom Dietterich, and Pedro Domingos.