10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

MLE/MAP

+

Naïve Bayes

MLE / MAP Readings:
"Estimating Probabilities"
(Mitchell, 2016)

Naïve Bayes Readings:
"Generative and Discriminative Classifiers: Naive Bayes and Logistic Regression"
(Mitchell, 2016)

Murphy 3
Bishop --
HTF --
Mitchell 6.1-6.10

Matt Gormley
Lecture 5
February 1, 2016

Reminders

- Background Exercises (Homework 1)
  - Release: Wed, Jan. 25
  - Due: Wed, Feb. 1 at 5:30pm
  - ONLY HW1: Collaboration questions not required
- Homework 2: Naive Bayes
  - Release: Wed, Feb. 1
  - Due: Mon, Feb. 13 at 5:30pm

MLE / MAP Outline

- **Generating Data**
  - Natural (stochastic) data
  - Synthetic data
  - Why synthetic data?
  - Examples: Multinomial, Bernoulli, Gaussian
- **Data Likelihood**
  - Independent and Identically Distributed (i.i.d.)
  - Example: Dice Rolls
- **Learning from Data (Frequentist)**
  - Principle of Maximum Likelihood Estimation (MLE)
  - Optimization for MLE
  - Examples: 1D and 2D optimization
  - Example: MLE of Multinomial
  - Aside: Method of Lagrange Multipliers
- **Learning from Data (Bayesian)**
  - *maximum a posteriori* (MAP) estimation
  - Optimization for MAP
  - Example: MAP of Bernoulli—Beta

Last Lecture

This Lecture

Learning from Data (Frequentist)

Whiteboard

- Aside: Method of Langrange Multipliers
- Example: MLE of Multinomial

Learning from Data (Bayesian)

Whiteboard

- *maximum a posteriori* (MAP) estimation
- Optimization for MAP
- Example: MAP of Bernoulli—Beta

Takeaways

- One view of what ML is trying to accomplish is **function approximation**
- The principle of **maximum likelihood estimation** provides an alternate view of learning

- **Synthetic data** can help **debug** ML algorithms
- Probability distributions can be used to **model** real data that occurs in the world
(don’t worry we’ll make our distributions more interesting soon!)

Naïve Bayes Outline

- **Probabilistic (Generative) View of Classification**
  - Decision rule for probability model
- **Real-world Dataset**
  - Economist vs. Onion articles
  - Document → bag-of-words → binary feature vector
- **Naive Bayes: Model**
  - Generating synthetic "labeled documents"
  - Definition of model
  - Naive Bayes assumption
  - Counting # of parameters with / without NB assumption
- **Naïve Bayes: Learning from Data**
  - Data likelihood
  - MLE for Naive Bayes
  - MAP for Naive Bayes
- **Visualizing Gaussian Naive Bayes**

Today's Goal

To define a generative model of emails of two different classes

(e.g. spam vs. not spam)

Spam News

The Economist

La paralización

Spain may be heading for its third election in a year

All latest updates

Stubborn Socialists are blocking Mariano Rajoy from forming a centre-right government

Sep 5th 2016 | MADRID | Europe

Timekeeper | Like 80 | Tweet

BACK in June, after Spain's second indecisive election in six months, the general expectation was that Mariano Rajoy, the prime minister, would swiftly form a new government. Although his conservative People's Party (PP) did not win back the absolute majority it had lost in December, it remained easily the largest party, with 137 of the 350

The Onion

★ ELECTION 2016 ★ MORE ELECTION COVERAGE ▸

Tim Kaine Found Riding Conveyor Belt During Factory Campaign Stop

NEWS IN BRIEF
August 23, 2016

VOL 52 ISSUE 33

Politics · Politicians · Election 2016 · Tim Kaine

AIKEN, SC—Noting that he disappeared for over an hour during a campaign stop meet-and-greet with workers at a Bridgestone tire manufacturing plant, sources confirmed Tuesday that Democratic vice presidential candidate Tim Kaine was finally discovered riding on one of the factory's conveyor belts. "Shortly after we arrived, Tim managed to get out of our sight, but after an extensive search of the facilities, one of our interns found him moving down the assembly line between several radial tires," said senior campaign advisor Mike Henry, adding that Kaine could be seen smiling and laughing as

Real-world Dataset

Whiteboard

– Economist vs. Onion articles

– Document → bag-of-words → binary feature vector

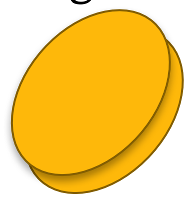
Naive Bayes: Model

Whiteboard

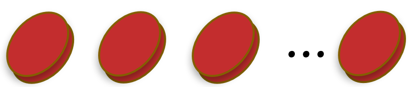
- Generating synthetic "labeled documents"
- Definition of model
- Naive Bayes assumption
- Counting # of parameters with / without NB assumption

Model 1: Bernoulli Naïve Bayes

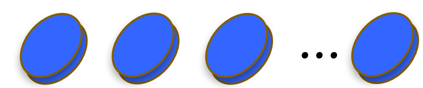
Flip weighted coin



If HEADS, flip
each red coin



If TAILS, flip
each blue coin



y	x_1	x_2	x_3	…	x_M
0	1	0	1	…	1
1	0	1	0	…	1
1	1	1	1	…	1
0	0	0	1	…	1
0	1	0	1	…	0
1	1	0	1	…	0

We can **generate** data in this fashion. Though in practice we never would since our data is **given.**

Instead, this provides an explanation of **how** the data was generated (albeit a terrible one).

Each red coin corresponds to an x_m

Naive Bayes: Model

Whiteboard

- Generating synthetic "labeled documents"
- Definition of model
- Naive Bayes assumption
- Counting # of parameters with / without NB assumption

What's wrong with the Naïve Bayes Assumption?

The features might not be independent!!

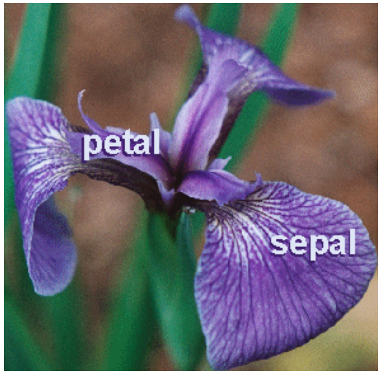
- Example 1:
  - If a document contains the word "Donald", it's extremely likely to contain the word "Trump"
  - These are not independent!

★ ELECTION 2016 ★ MORE ELECTION COVERAGE ▸

Trump Spends Entire Classified National Security Briefing Asking About Egyptian Mummies

NEWS IN BRIEF August 18, 2016
VOL 52 ISSUE 32 · Politics · Politicians · Election 2016 · Donald Trump

- Example 2:
  - If the petal width is very high, the petal length is also likely to be very high



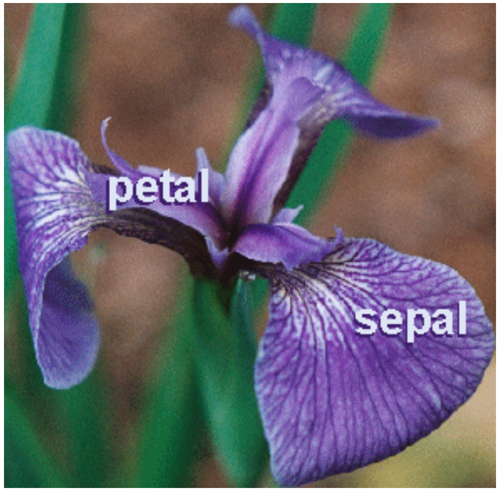
Naïve Bayes: Learning from Data

Whiteboard

- Data likelihood
- MLE for Naive Bayes
- MAP for Naive Bayes

VISUALIZING NAÏVE BAYES

Slides in this section from William Cohen (10-601B, Spring 2016)

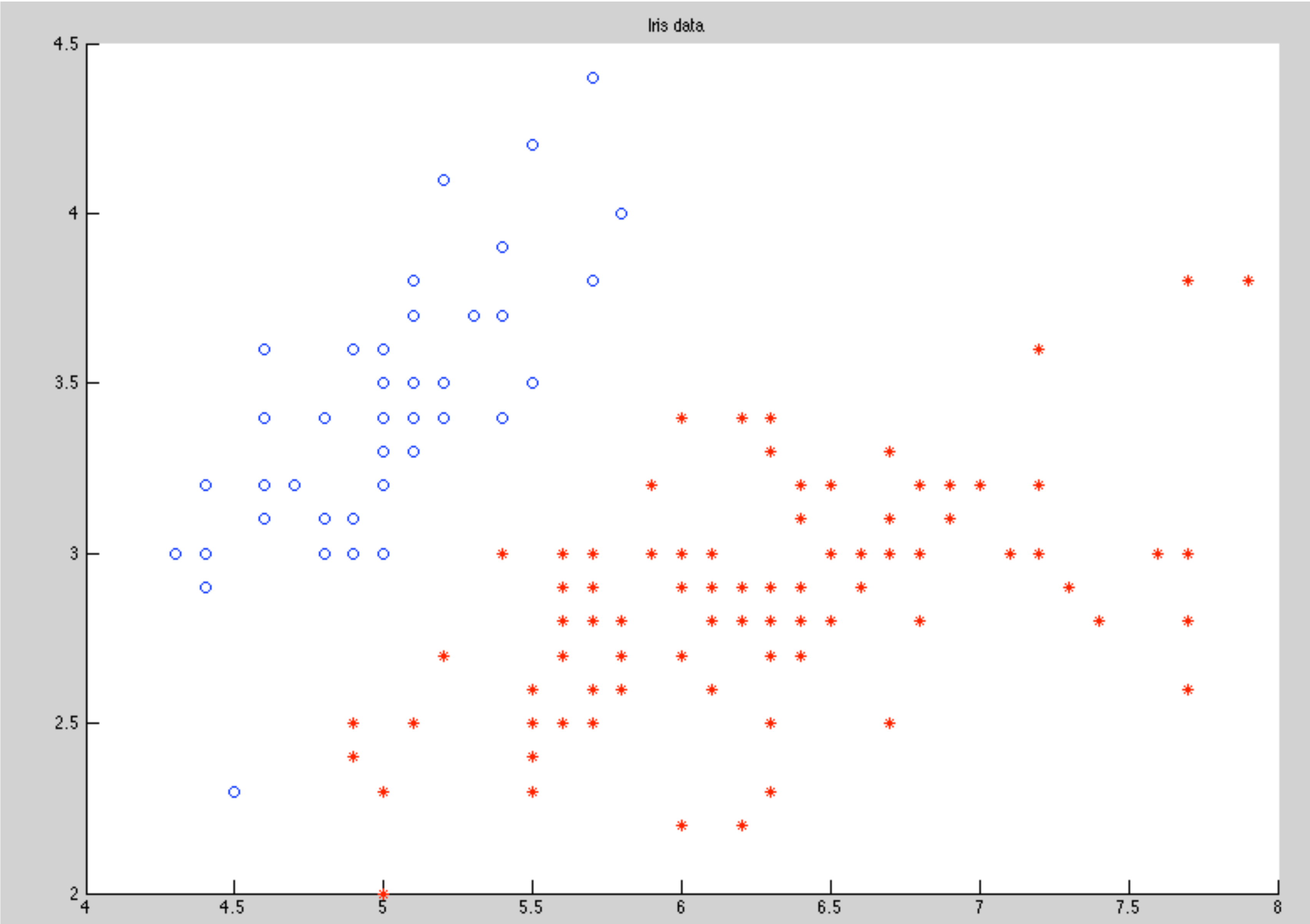

petal
sepal

Fisher Iris Dataset

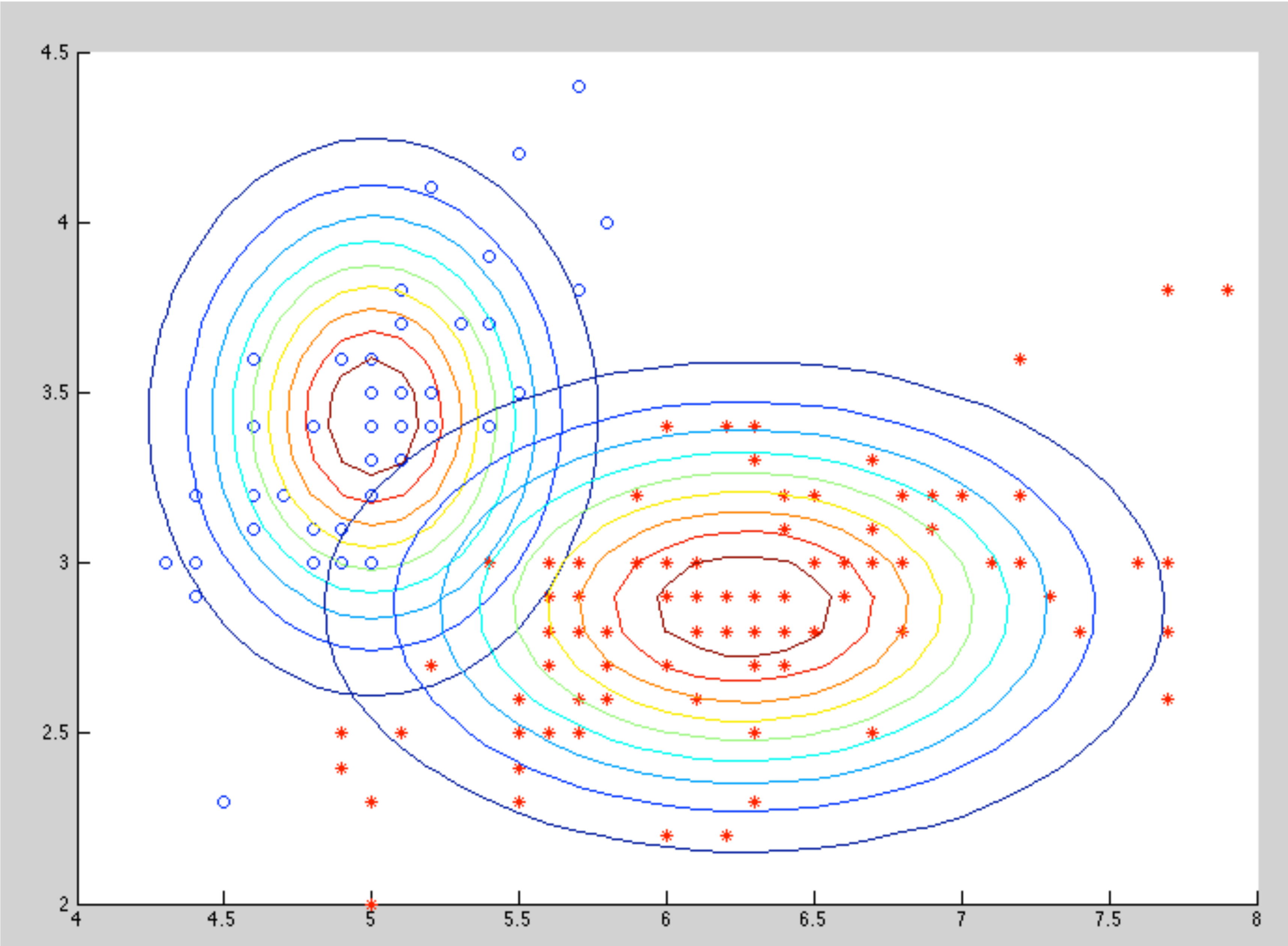
Fisher (1936) used 150 measurements of flowers from 3 different species: Iris setosa (0), Iris virginica (1), Iris versicolor (2) collected by Anderson (1936)

Species	Sepal Length	Sepal Width	Petal Length	Petal Width
0	4.3	3.0	1.1	0.1
0	4.9	3.6	1.4	0.1
0	5.3	3.7	1.5	0.2
1	4.9	2.4	3.3	1.0
1	5.7	2.8	4.1	1.3
1	6.3	3.3	4.7	1.6
1	6.7	3.0	5.0	1.7

Full dataset: https://en.wikipedia.org/wiki/Iris_flower_data_set

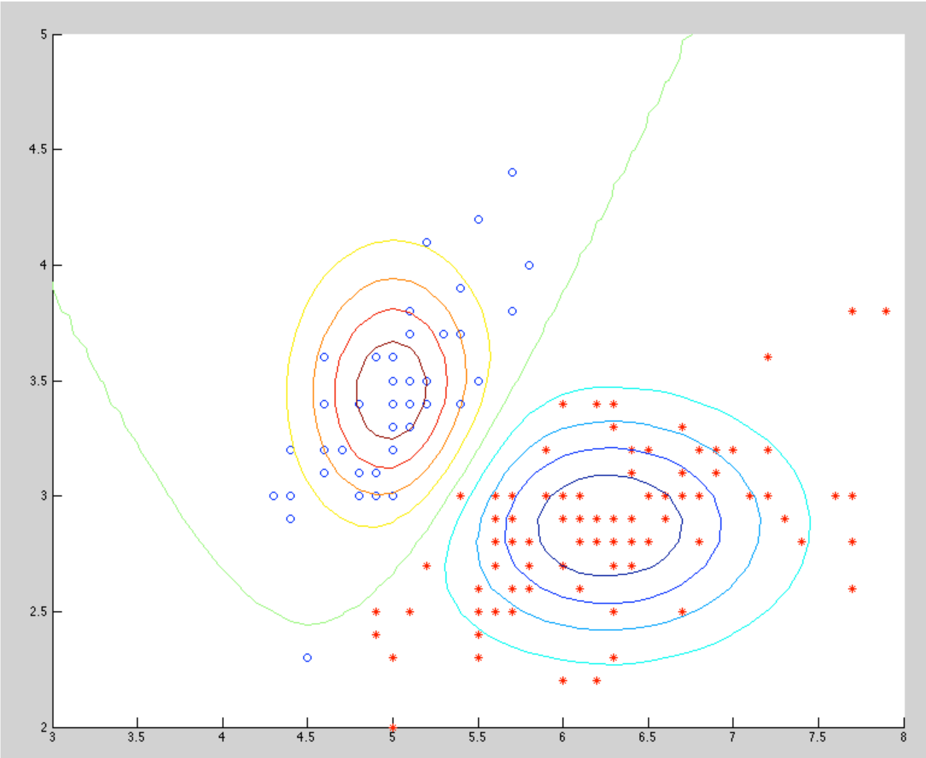


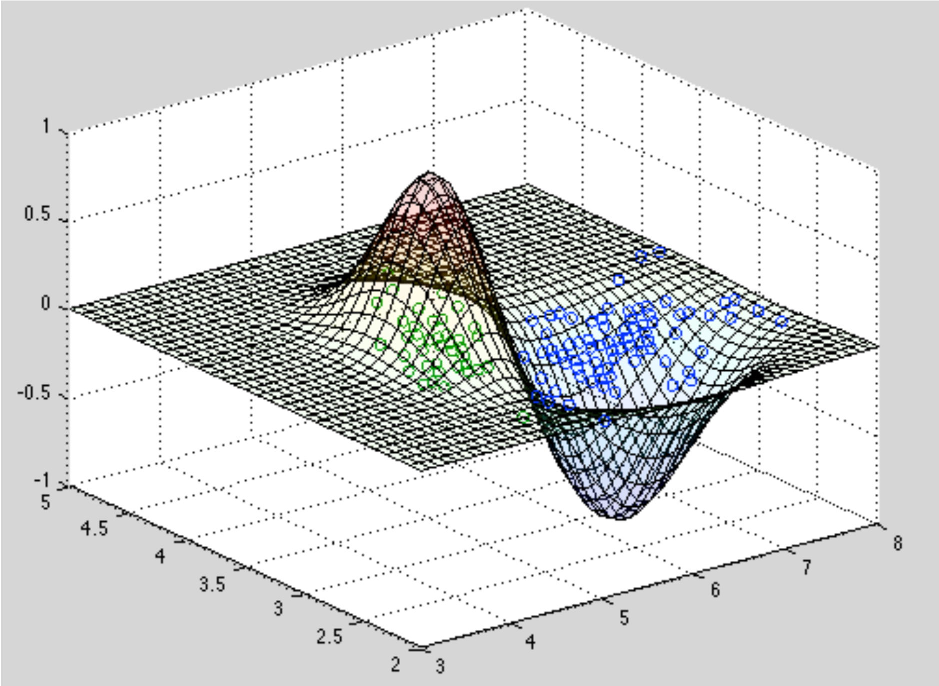
Slide from William Cohen



Slide from William Cohen

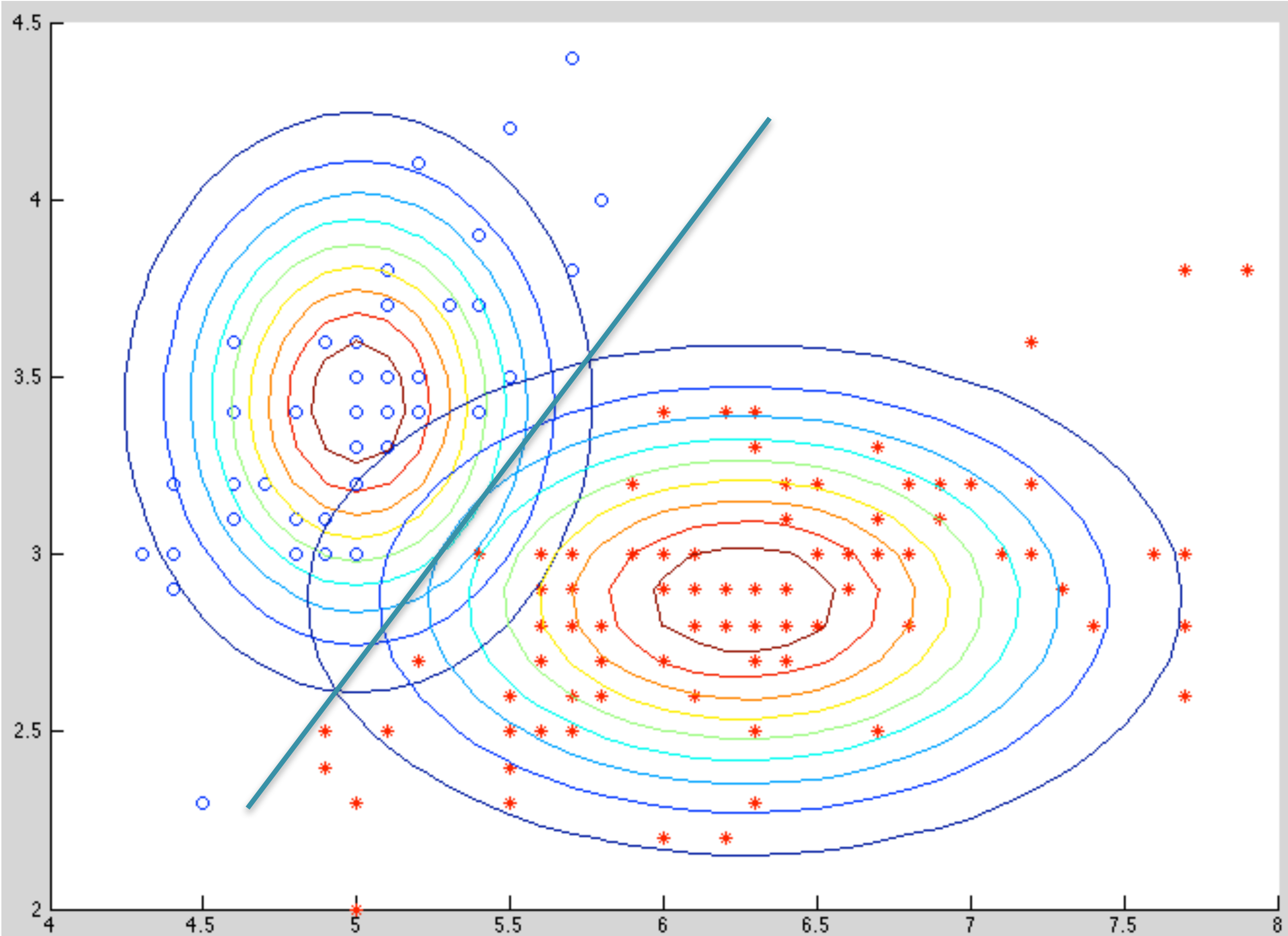
Plot the difference of the probabilities





Slide from William Cohen

Naïve Bayes has a **linear** decision boundary



Slide from William Cohen (10-601B, Spring 2016)

Figure from William Cohen (10-601B, Spring 2016)

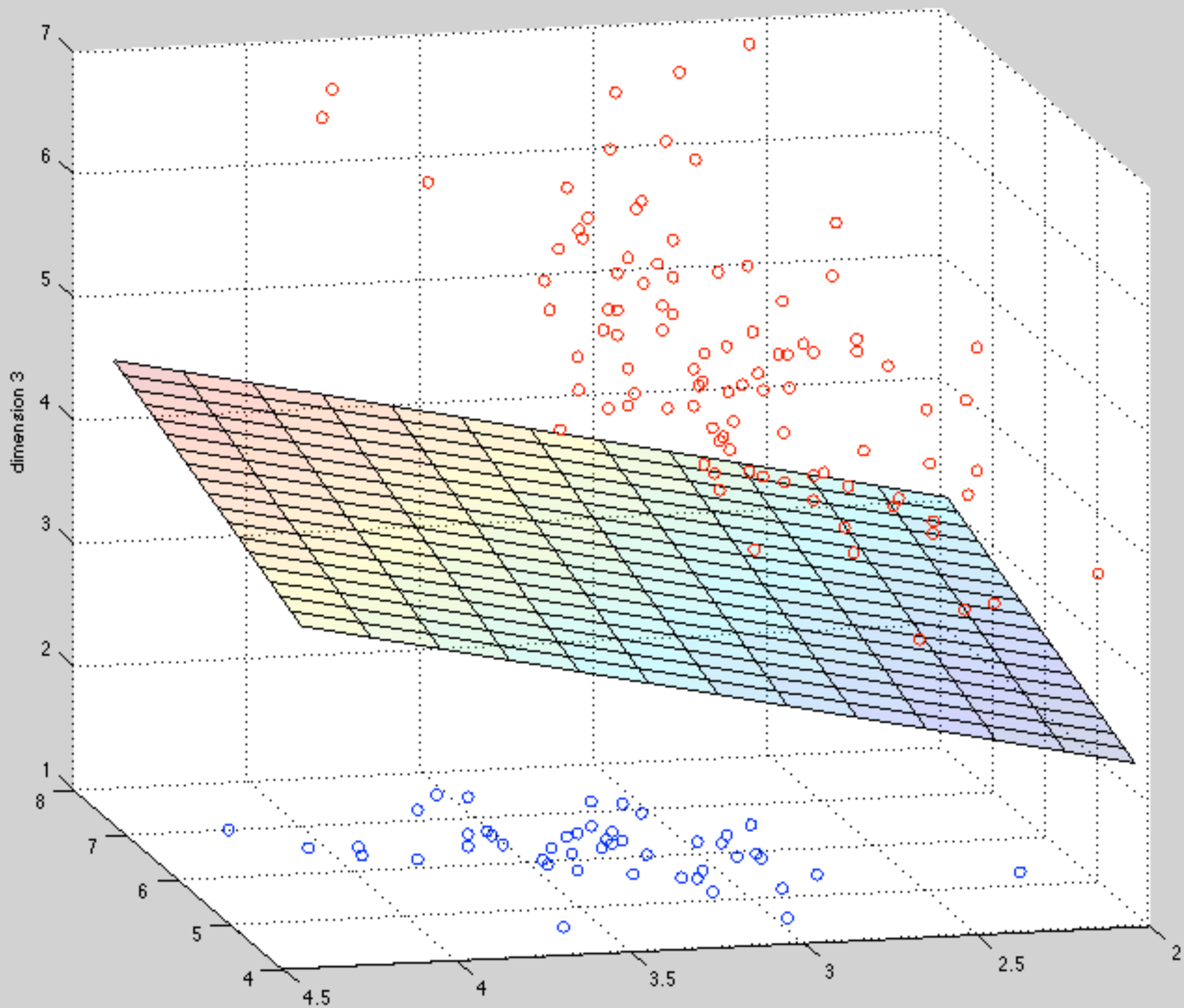
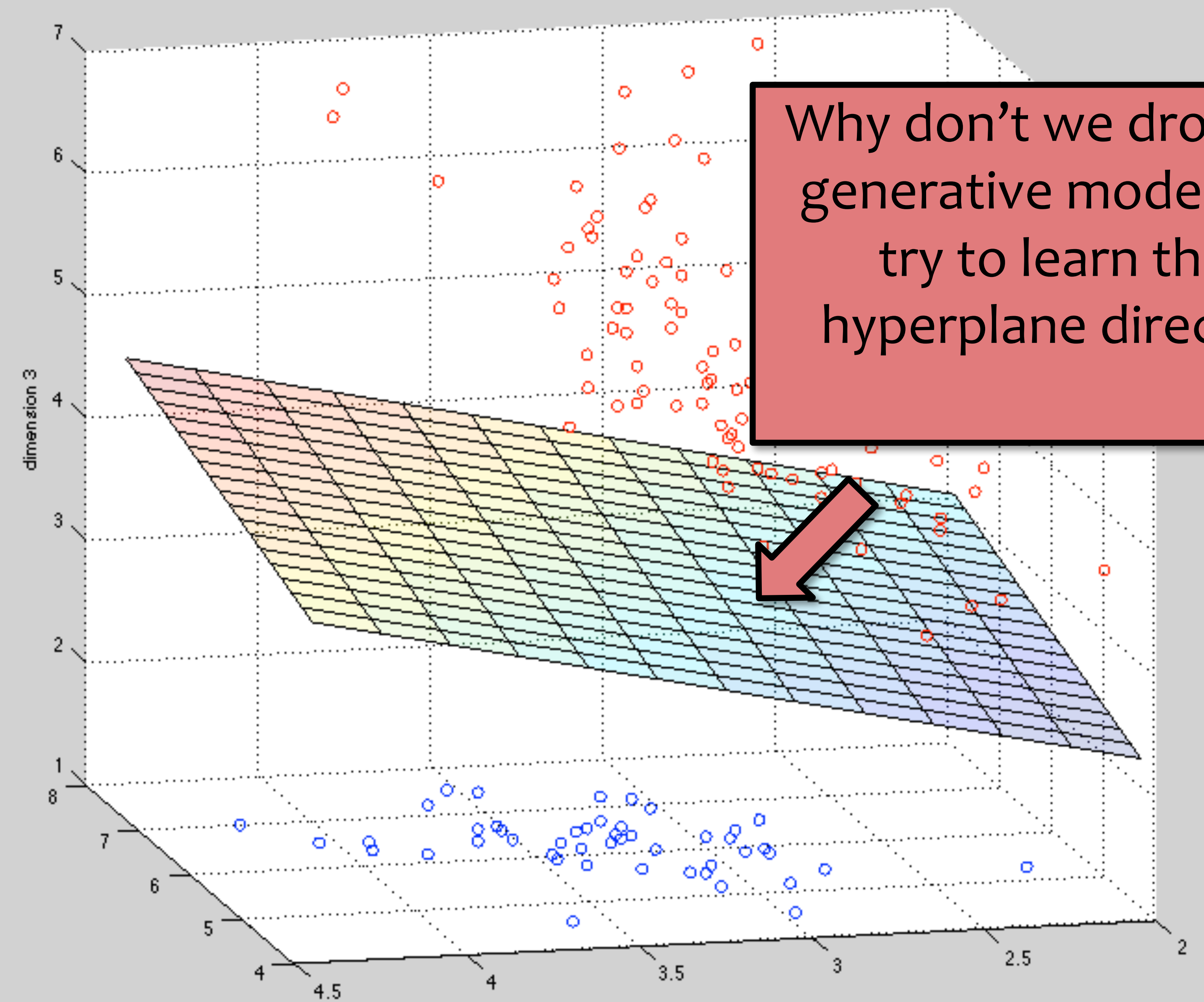


Figure from William Cohen (10-601B, Spring 2016)



Why don't we drop the generative model and try to learn this hyperplane directly?

Beyond the Scope of this Lecture

- **Multinomial** Naïve Bayes can be used for **integer features**
- **Multi-class** Naïve Bayes can be used if your classification problem has > 2 classes

Summary

1. Naïve Bayes provides a framework for **generative modeling**
2. Choose $p(x_m \mid y)$ appropriate to the data (e.g. Bernoulli for binary features, Gaussian for continuous features)
3. Train by **MLE** or **MAP**
4. Classify by maximizing the posterior

EXTRA SLIDES

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

Classification: Find the class that maximizes the posterior

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

Generic Naïve Bayes Model

Classification:

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

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

$$= \operatorname*{argmax}_y p(\mathbf{x}|y)p(y)$$

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

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:

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

Same as Generic Naïve Bayes

Classification: Find the class that maximizes the posterior

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

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

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.

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

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

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

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

Data:

y	x_1	x_2	x_3	…	x_K
0	1	0	1	…	1
1	0	1	0	…	1
1	1	1	1	…	1
0	0	0	1	…	1
0	1	0	1	…	0
1	1	0	1	…	0

Model 2: Multinomial Naïve Bayes

Support: Option 1: Integer vector (word IDs)

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

Generative Story:

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

$y^{(i)} \sim$ Bernoulli(ϕ)

for $j \in \{1, \dots, M_i\}$**:**

$x_j^{(i)} \sim$ 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.

$$
\begin{aligned}
p(\boldsymbol{x}, y) &= p(x_1, \ldots, x_K, y) \\
&= p(y) \prod_{k=1}^{K} p(x_k | y)
\end{aligned}
$$

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

Smoothing

1. Add-1 Smoothing
2. Add-λ Smoothing
3. MAP Estimation (Beta Prior)

MLE

What does maximizing likelihood accomplish?

- There is only a finite amount of probability mass (i.e. sum-to-one constraint)
- MLE tries to allocate **as much** probability mass **as possible** to the things we have observed…

  …**at the expense** of the things we have **not** observed

MLE

For Naïve Bayes, suppose we never observe the word "`serious`" in an Onion article.

In this case, what is the MLE of p(x_k | y)?

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

Now suppose we observe the word "serious" at test time. What is the posterior probability that the article was an Onion article?

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})}$$

1. Add-1 Smoothing

The simplest setting for smoothing simply adds a single pseudo-observation to the data. This converts the true observations $\mathcal{D}$ into a new dataset $\mathcal{D}'$ from we derive the MLEs.

$$\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N} \quad (1)$$

$$\mathcal{D}' = \mathcal{D} \cup \{(\mathbf{0}, 0), (\mathbf{0}, 1), (\mathbf{1}, 0), (\mathbf{1}, 1)\} \quad (2)$$

where $\mathbf{0}$ is the vector of all zeros and $\mathbf{1}$ is the vector of all ones.
This has the effect of pretending that we observed each feature x_k with each class y.

1. Add-1 Smoothing

What if we write the MLEs in terms of the original dataset $\mathcal{D}$?

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

$$\theta_{k,0} = \frac{\boxed{1 +} \sum_{i=1}^N \mathbb{I}(y^{(i)} = 0 \wedge x_k^{(i)} = 1)}{\boxed{2 +} \sum_{i=1}^N \mathbb{I}(y^{(i)} = 0)}$$

$$\theta_{k,1} = \frac{\boxed{1 +} \sum_{i=1}^N \mathbb{I}(y^{(i)} = 1 \wedge x_k^{(i)} = 1)}{\boxed{2 +} \sum_{i=1}^N \mathbb{I}(y^{(i)} = 1)}$$

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

2. Add-λ Smoothing

For the Categorical Distribution

Suppose we have a dataset obtained by repeatedly rolling a K-sided (weighted) die. Given data $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$ where $x^{(i)} \in \{1, \ldots, K\}$, we have the following MLE:

$$\phi_k = \frac{\sum_{i=1}^{N} \mathbb{I}(x^{(i)} = k)}{N}$$

With add-λ smoothing, we add pseudo-observations as before to obtain a smoothed estimate:

$$\phi_k = \frac{\lambda + \sum_{i=1}^{N} \mathbb{I}(x^{(i)} = k)}{k\lambda + N}$$

Recall...

MLE vs. MAP

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

Maximum Likelihood Estimate (MLE)

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

Maximum *a posteriori* (MAP) estimate

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

3. MAP Estimation (Beta Prior)

Generative Story:
The parameters are drawn once for the entire dataset.

for $k \in \{1, \ldots, K\}$**:**
  for $y \in \{0, 1\}$**:**
    $\theta_{k,y} \sim \text{Beta}(\alpha, \beta)$
for $i \in \{1, \ldots, N\}$**:**
  $y^{(i)} \sim \text{Bernoulli}(\phi)$
  for $k \in \{1, \ldots, K\}$**:**
    $x_k^{(i)} \sim \text{Bernoulli}(\theta_{k,y^{(i)}})$

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

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

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

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

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