MACHINE LEARNING DEPARTMENT

10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Naïve Bayes

Matt Gormley
Lecture 21
March 28, 2018

Reminders

- **Homework 6: PAC Learning / Generative Models**
  - **Out: Wed, Mar 28**
  - **Due: Wed, Apr 04 at 11:59pm**

NAÏVE BAYES

Naïve Bayes Outline

- **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**

Fake News Detector

Today's Goal: To define a generative model of emails of two different classes (e.g. real vs. fake news)

The Economist

La paralización

All latest updates

Spain may be heading for its third election in a year

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

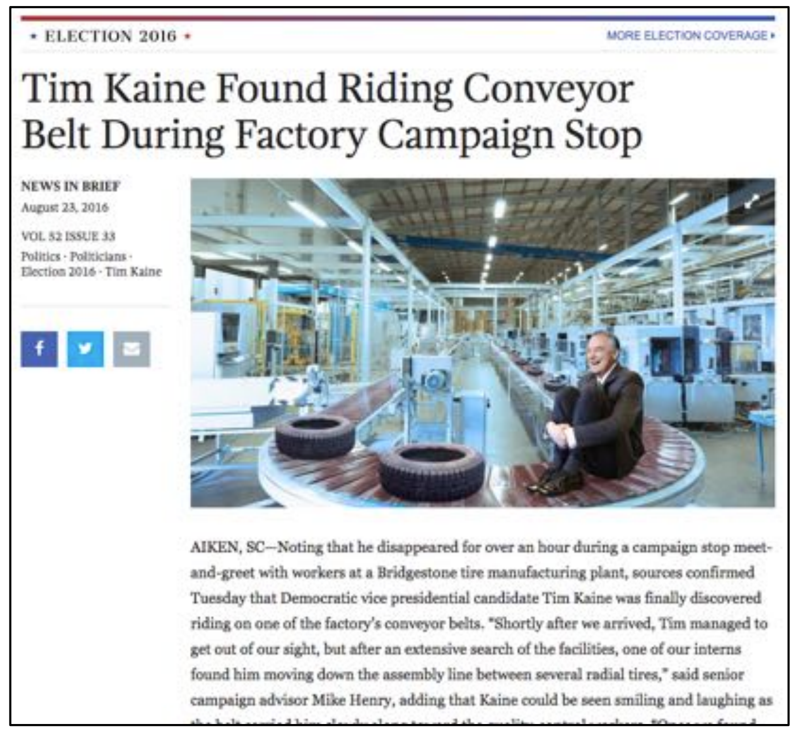
Sep 5th 2016 | MADRID | Europe

Like 60 Tweet

EPA

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

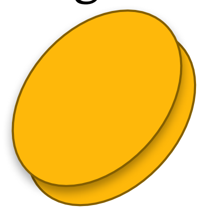
Naive Bayes: Model

Whiteboard

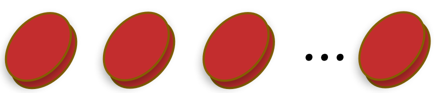
- Document → bag-of-words → binary feature vector
- Generating synthetic "labeled documents"
- Definition of model
- Naive Bayes assumption
- Counting # of parameters with / without NB assumption

Model 1: Bernoulli Naïve Bayes

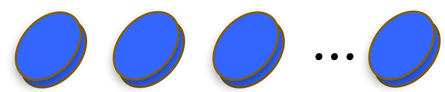
Flip weighted coin



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

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 •

Trump Spends Entire Classified National Security Briefing Asking About Egyptian Mummies

- 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
- Example: MLE for Naïve Bayes with Two Features
- MAP for Naive Bayes

NAÏVE BAYES: MODEL DETAILS

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

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

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.

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

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

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

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

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

Other NB Models

1. **Bernoulli** Naïve Bayes:
   – for **binary features**
2. **Gaussian** Naïve Bayes:
   – for **continuous features**
3. **Multinomial** Naïve Bayes:
   – for **integer features**
4. **Multi-class** Naïve Bayes:
   – for classification problems with > 2 classes
   – **event model** could be any of Bernoulli, Gaussian, Multinomial, depending on features

Model 2: 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 3: Multinomial Naïve Bayes

Support: Option 1: Integer vector (word IDs)

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

Generative Story:

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

$y^{(i)} \sim$ Bernoulli(ϕ)

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

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

Model:

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

Model 5: 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, $y \sim \text{Multinomial}(\phi, 1)$ and we have a separate conditional distribution $p(x_k|y)$ for each of the C classes.

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

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

Generic Naïve Bayes Model

Classification:

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

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

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

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 $p(x_k \mid y)$?

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

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.

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

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

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:

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

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

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

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

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

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

VISUALIZING NAÏVE BAYES

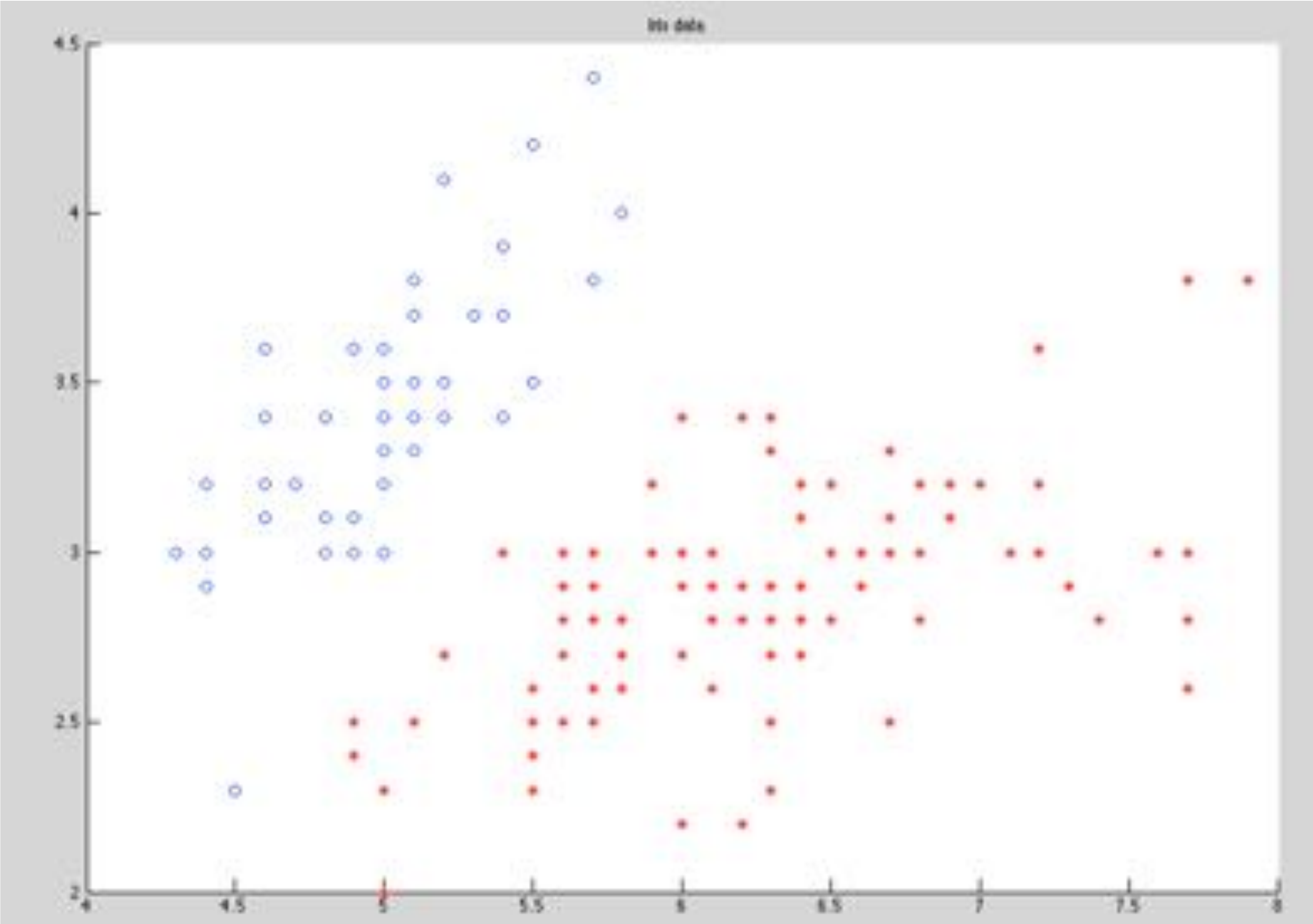

petal
sepal

Fisher Iris Dataset

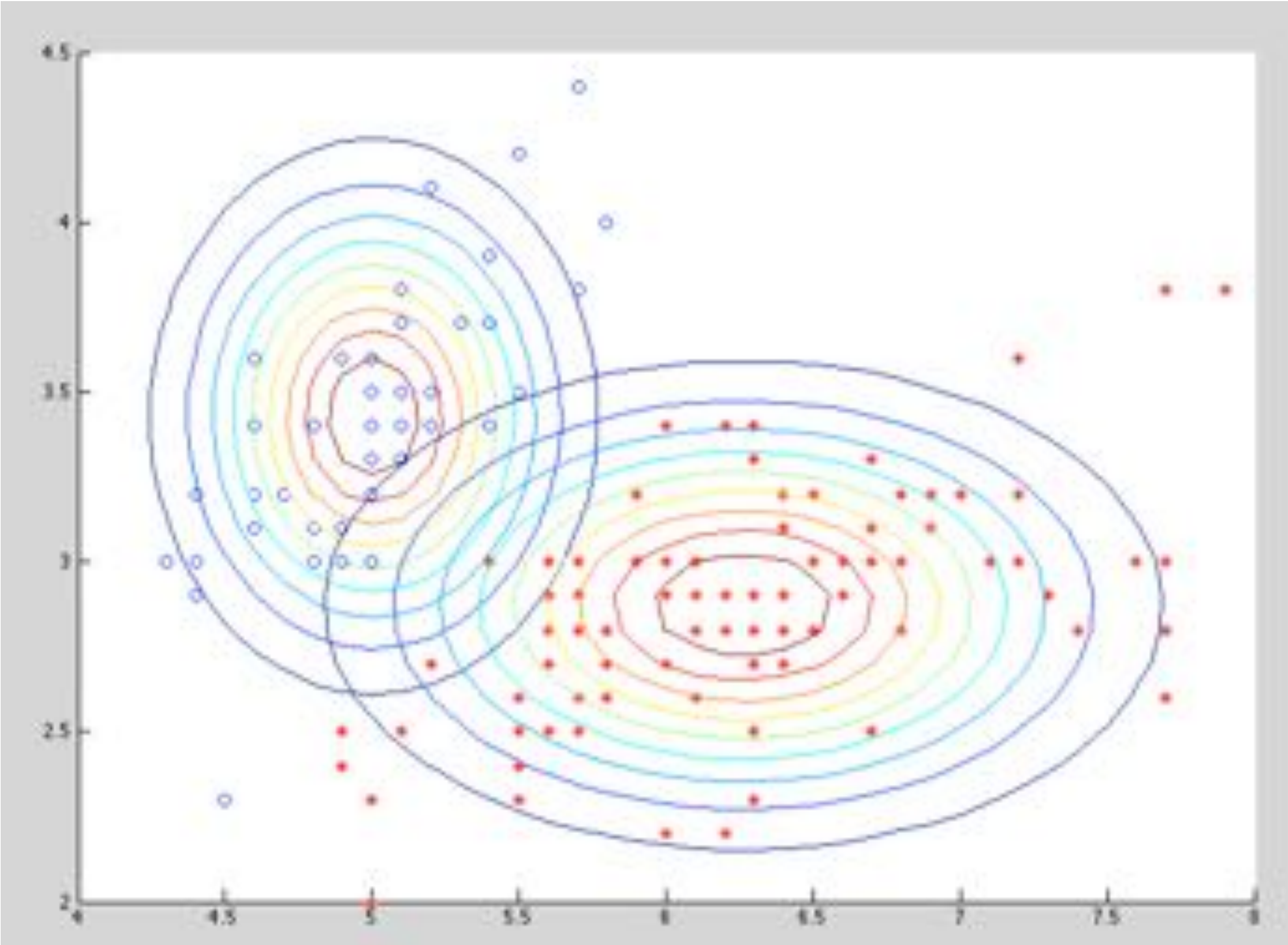
Fisher (1936) used 150 measurements of flowers from 3 different species: Iris setosa (0), Iris virginica (1), Iris versicolor (2) collected by Anderson (1936)

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

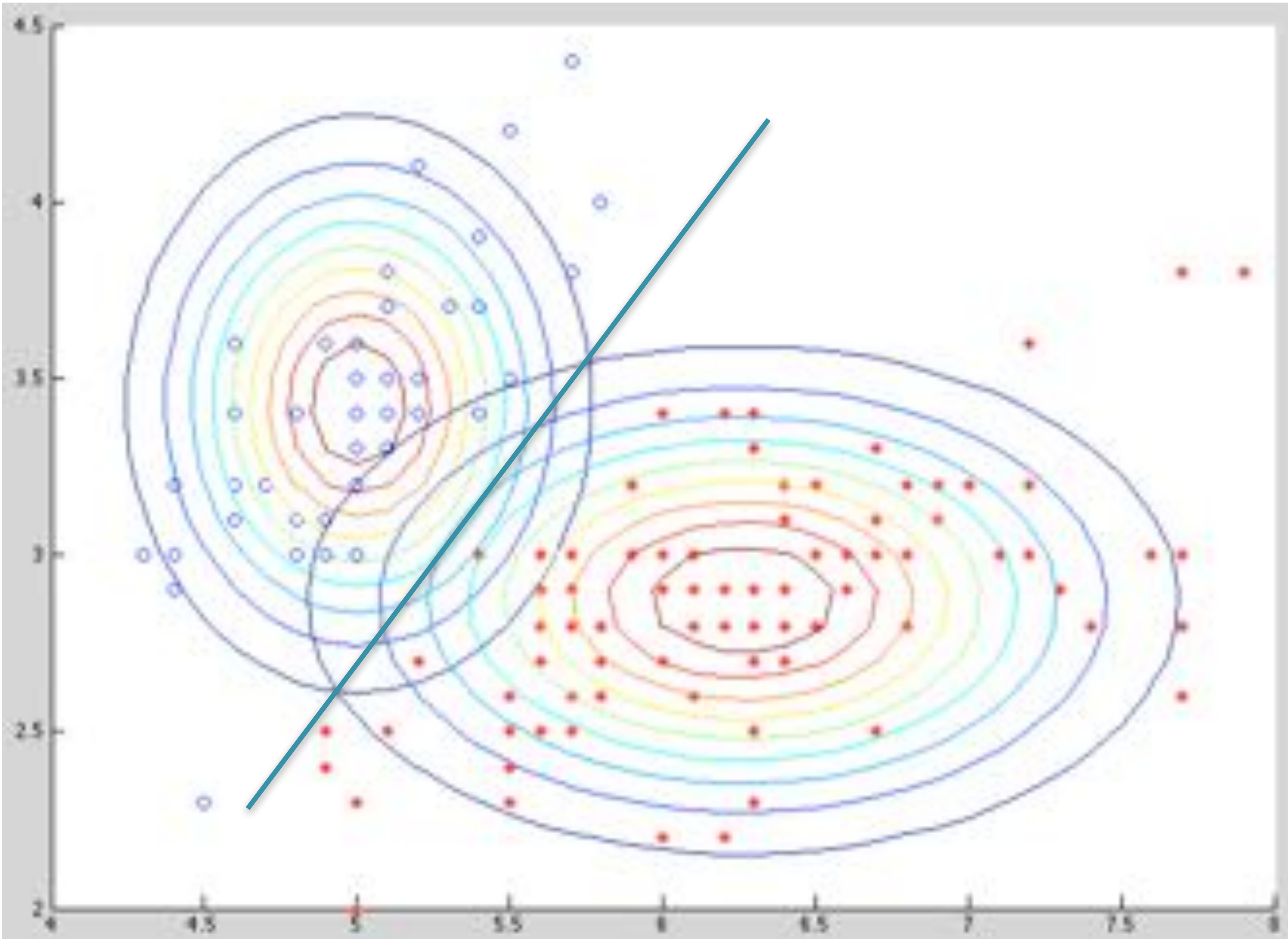


Slide from William Cohen



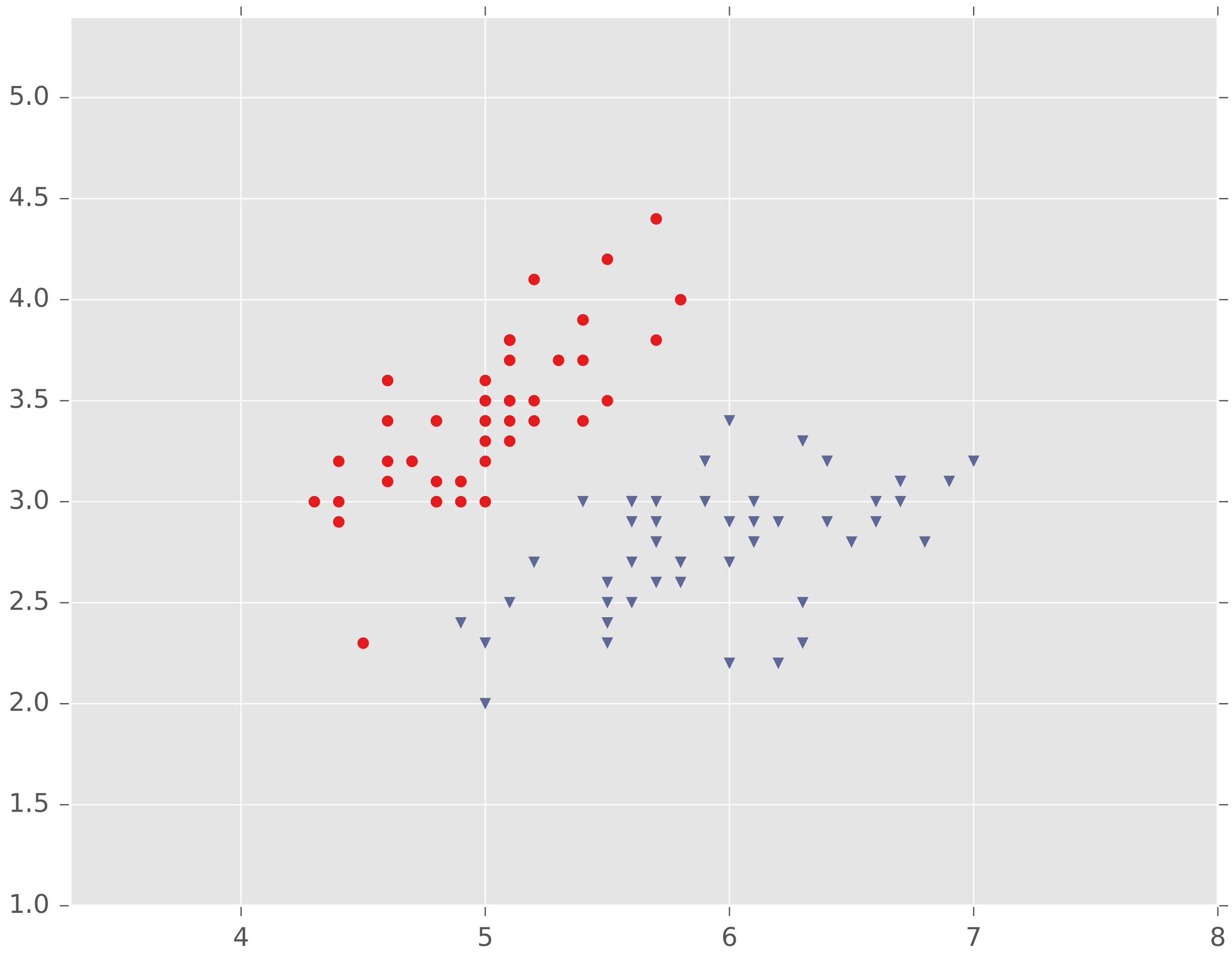
Slide from William Cohen

Naïve Bayes has a **linear** decision boundary if variance (sigma) is constant across classes

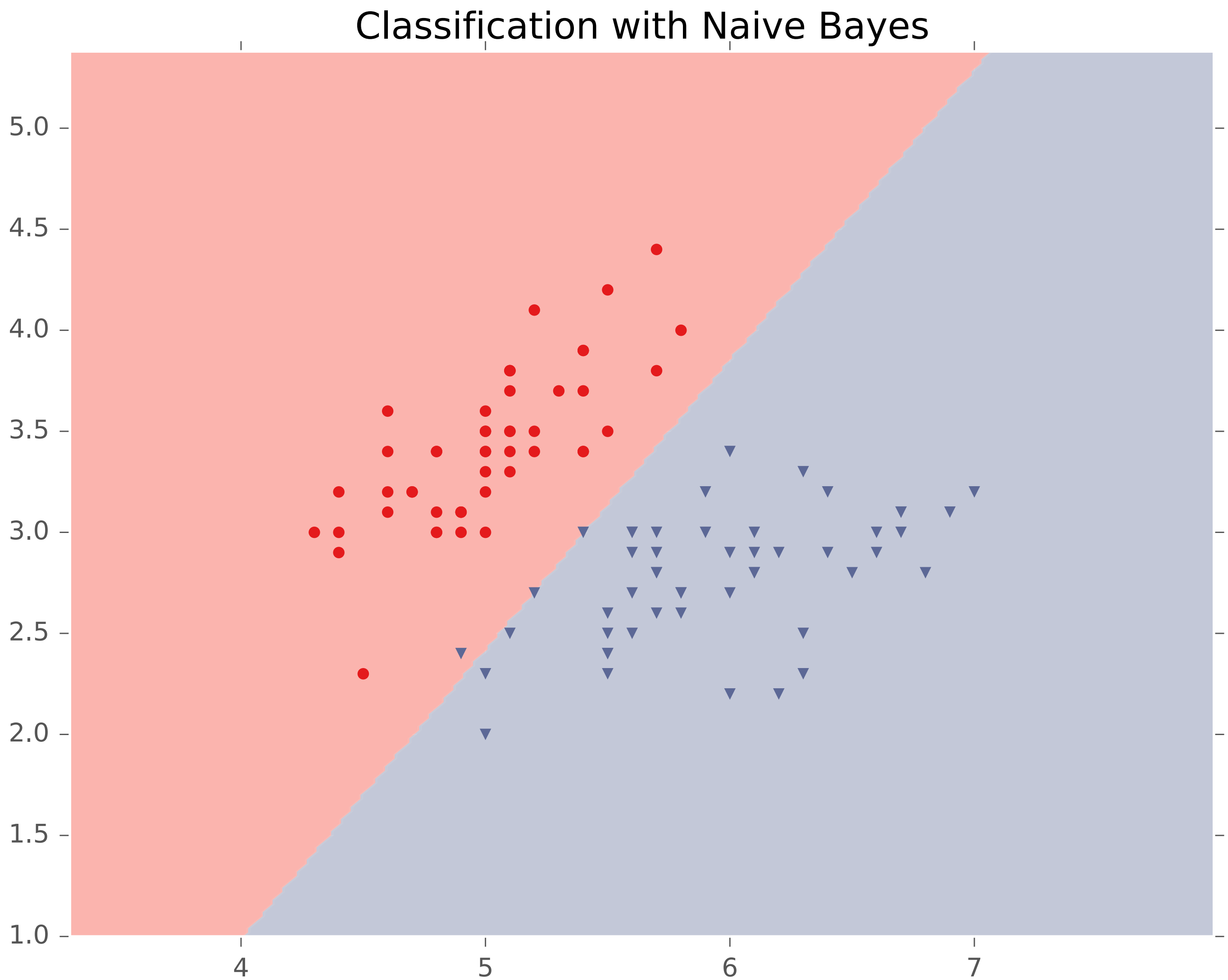


Slide from William Cohen (10-601B, Spring 2016)

Iris Data (2 classes)

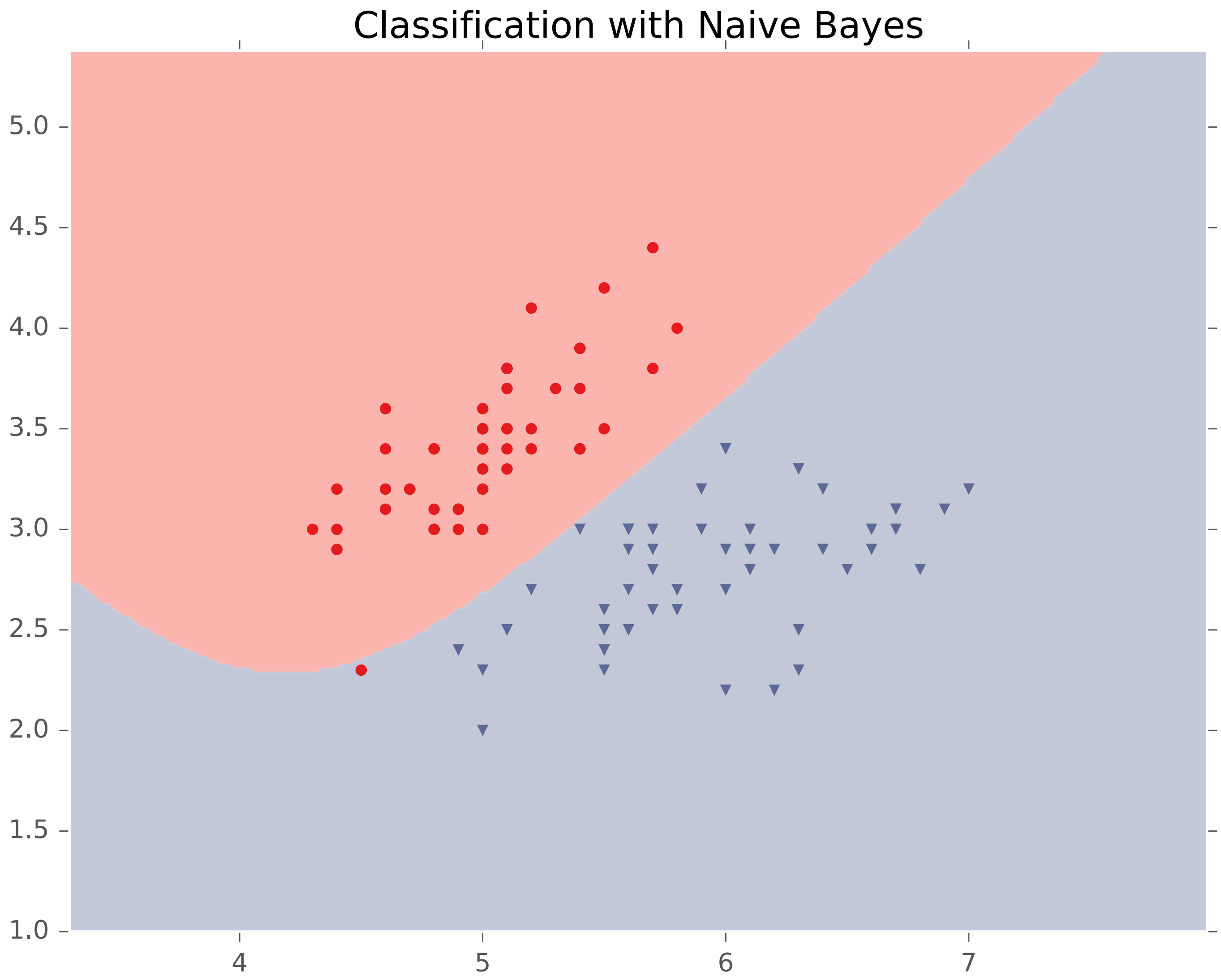


Iris Data (2 classes)



variance = 1

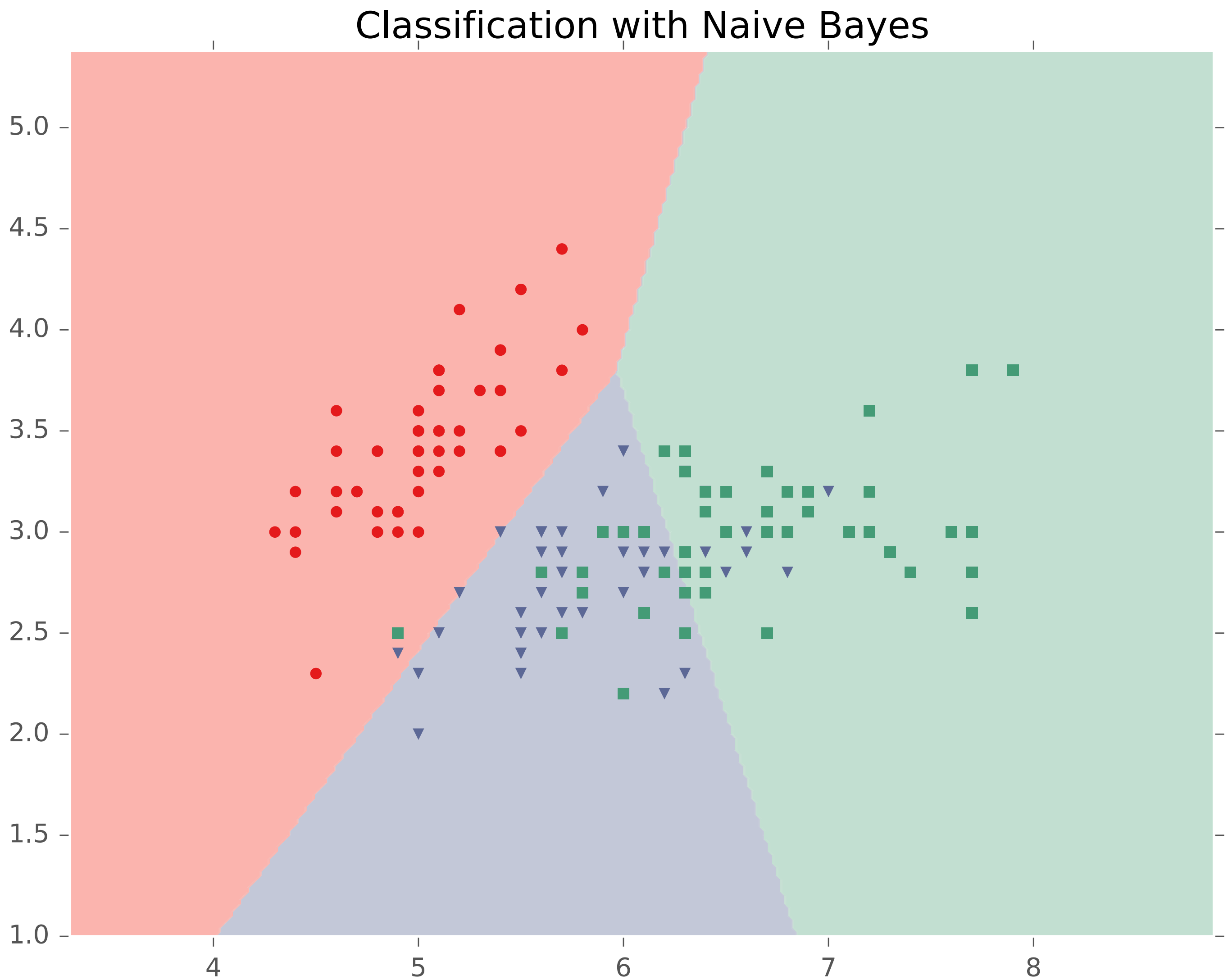
Iris Data (2 classes)



variance learned for each class

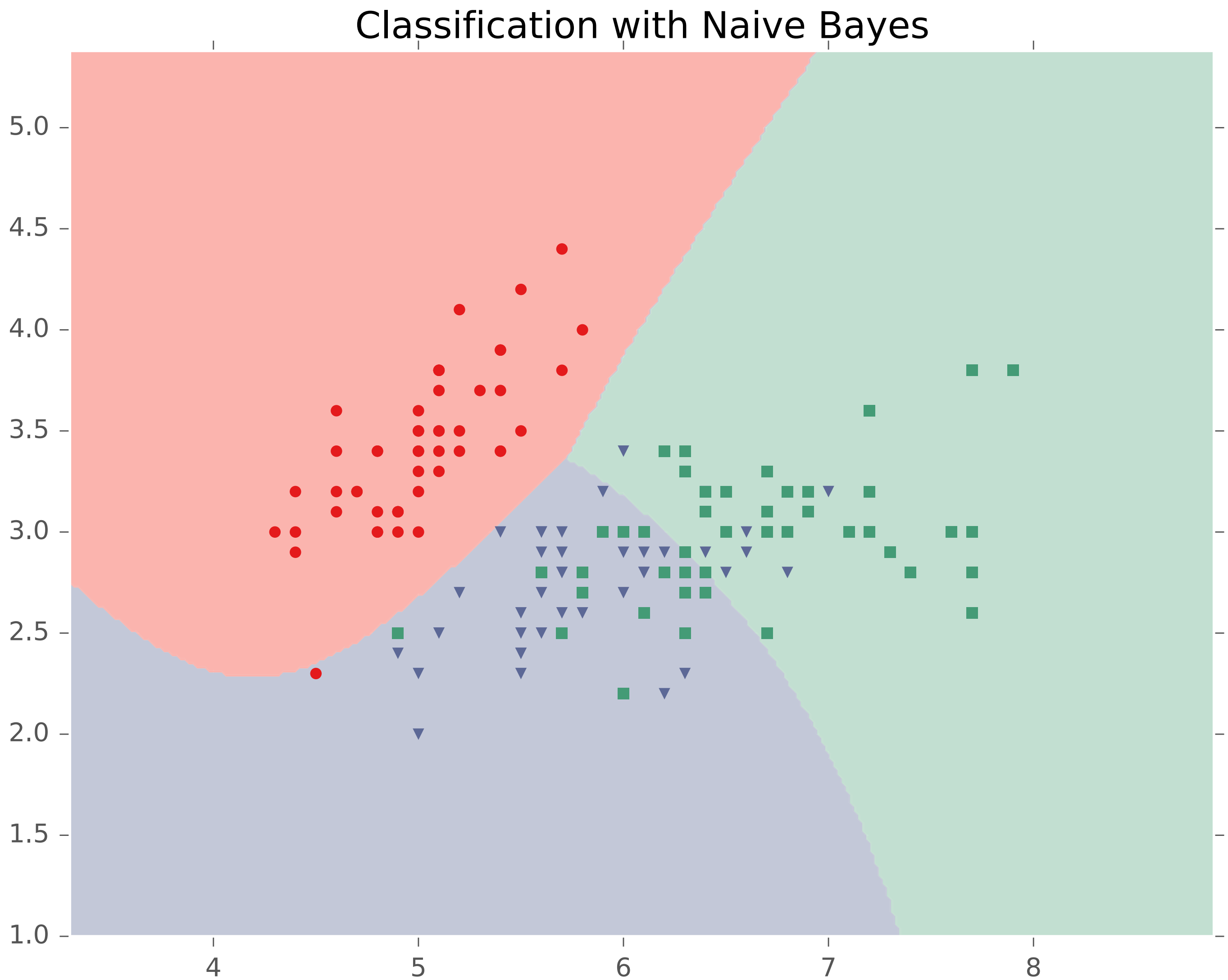
Iris Data (3 classes)

Iris Data (3 classes)



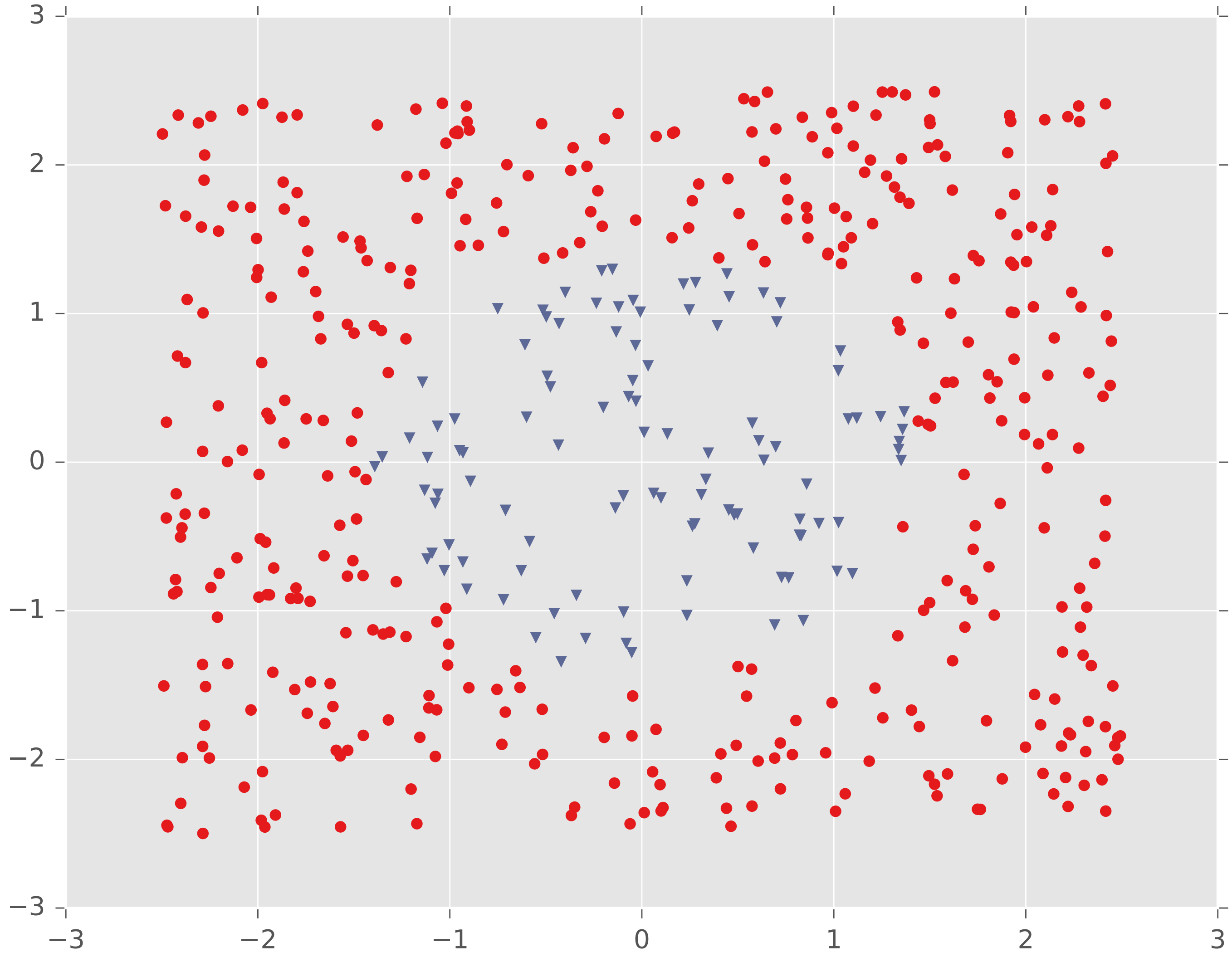
variance = 1

Iris Data (3 classes)

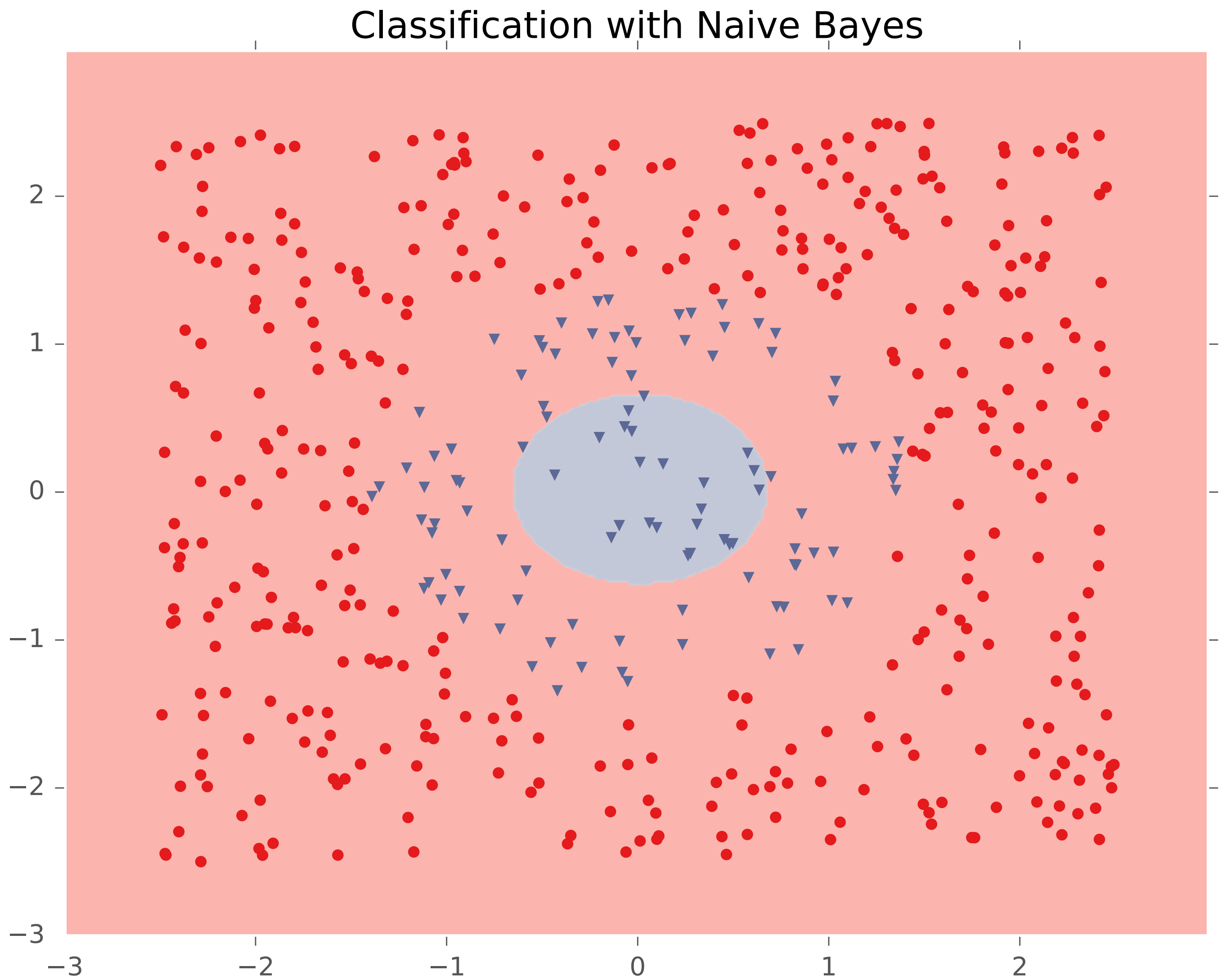


variance learned for each class

One Pocket

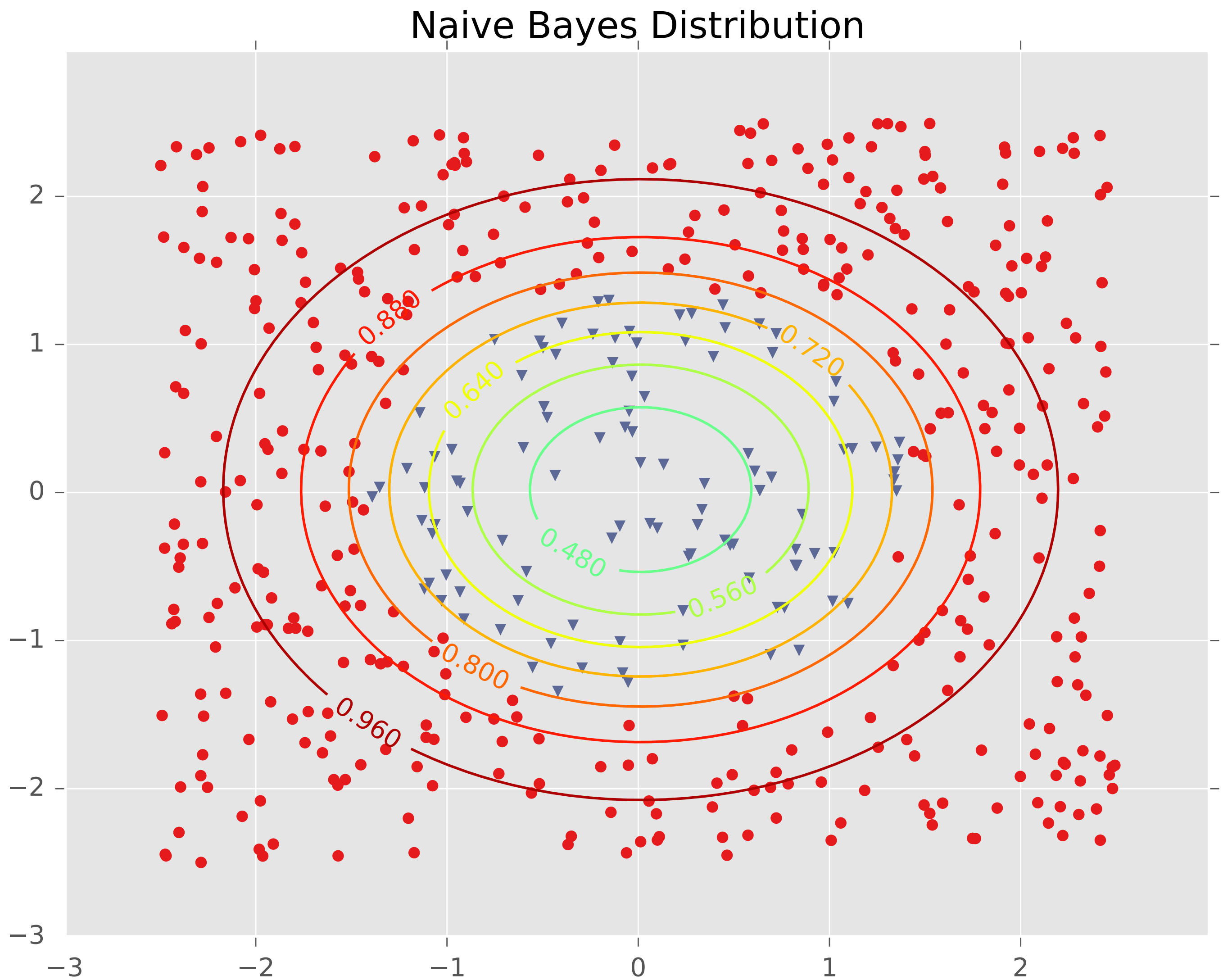


One Pocket



variance learned for each class

One Pocket



variance learned for each class

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

DISCRIMINATIVE AND GENERATIVE CLASSIFIERS

Generative vs. Discriminative

- **Generative Classifiers:**
  - Example: Naïve Bayes
  - Define a joint model of the observations **x** and the labels y: $p(\boldsymbol{x}, y)$
  - Learning maximizes (joint) likelihood
  - Use Bayes' Rule to classify based on the posterior:

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

- **Discriminative Classifiers:**
  - Example: Logistic Regression
  - Directly model the conditional: $p(y|\mathbf{x})$
  - Learning maximizes conditional likelihood

Generative vs. Discriminative

Whiteboard

– Contrast: To model p(x) or not to model p(x)?

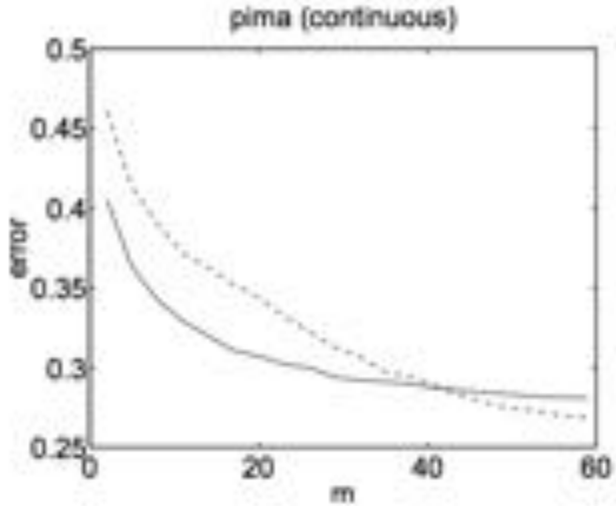
Generative vs. Discriminative

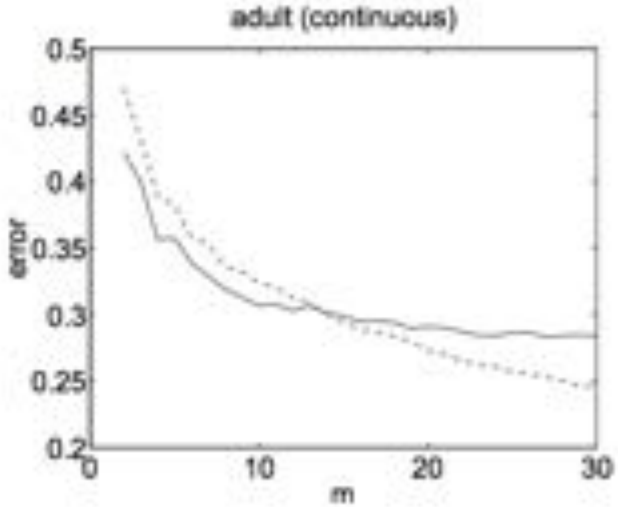
Finite Sample Analysis (Ng & Jordan, 2002)

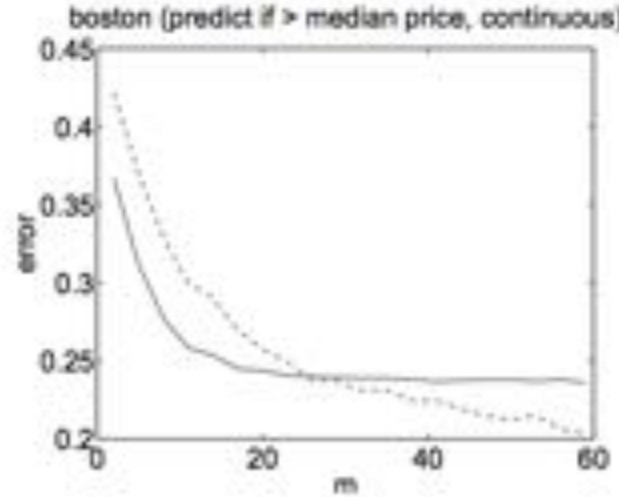
[Assume that we are learning from a finite training dataset]

If model assumptions are correct: Naive Bayes is a more efficient learner (requires fewer samples) than Logistic Regression

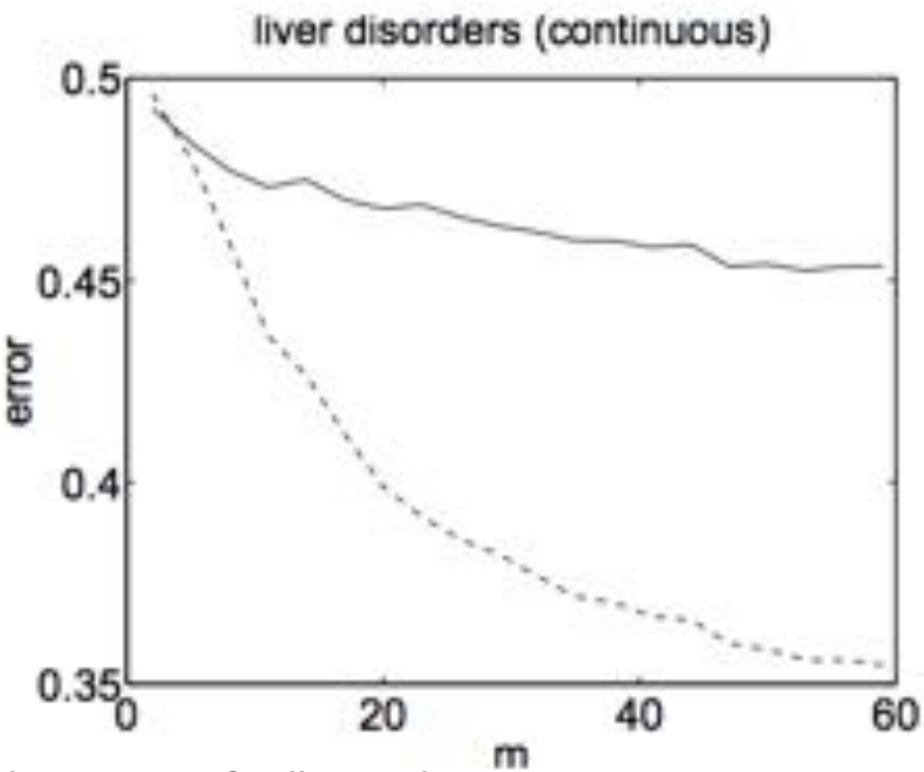
If model assumptions are incorrect: Logistic Regression has lower asymtotic error, and does better than Naïve Bayes

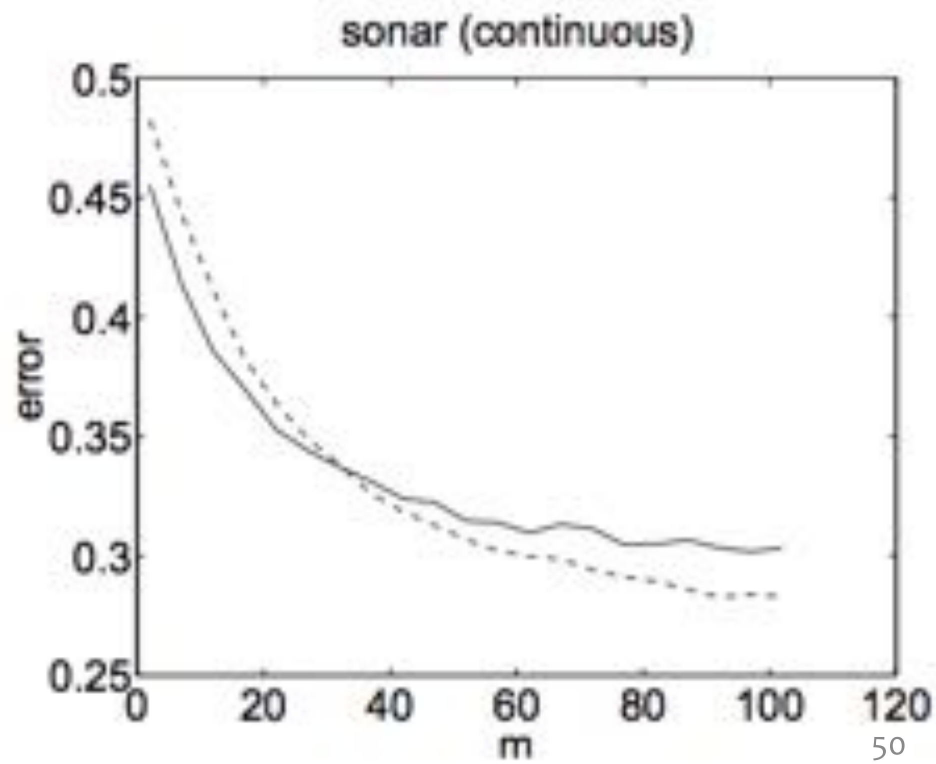






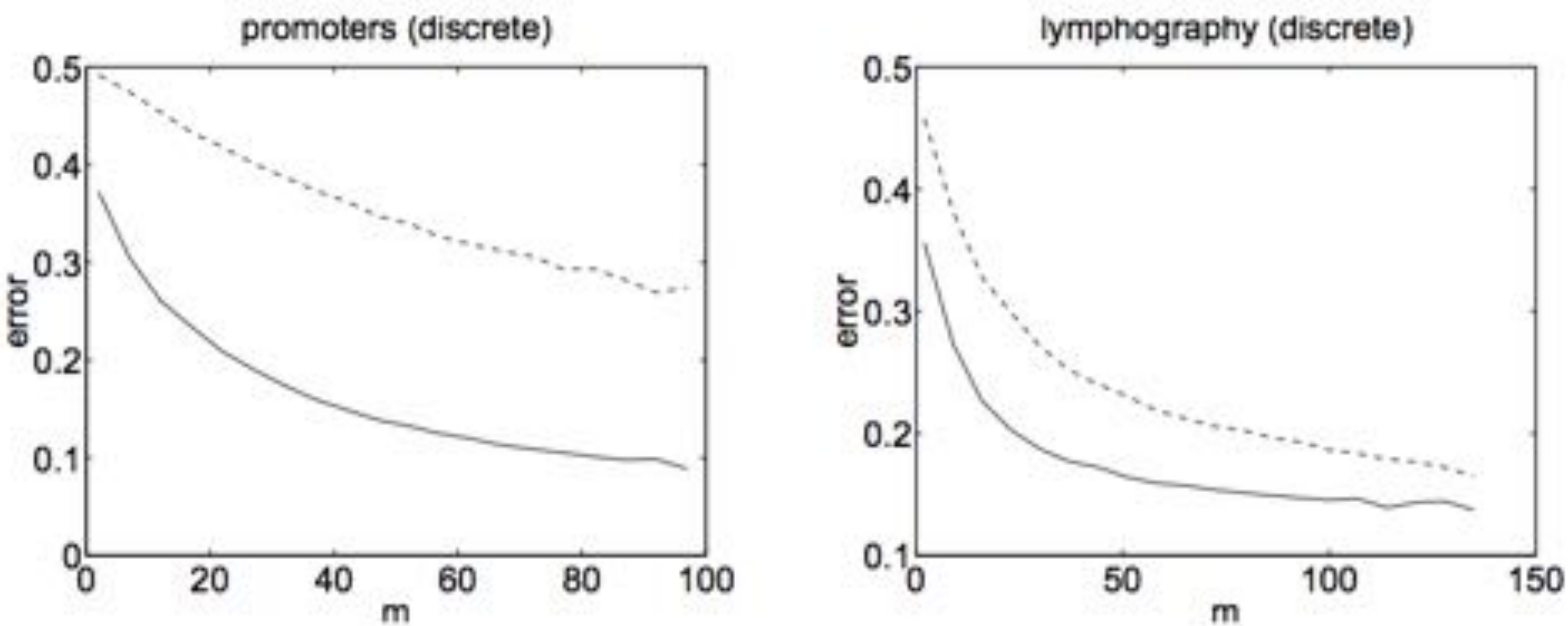
solid: NB dashed: LR





Slide courtesy of William Cohen

solid: NB dashed: LR



Naïve Bayes makes stronger assumptions about the data but needs fewer examples to estimate the parameters

"On Discriminative vs Generative Classifiers: …." Andrew Ng and Michael Jordan, NIPS 2001.

Generative vs. Discriminative

Learning (Parameter Estimation)

Naïve Bayes:
Parameters are decoupled → Closed form solution for MLE

Logistic Regression:
Parameters are coupled → No closed form solution – must use iterative optimization techniques instead

Naïve Bayes vs. Logistic Reg.

Learning (MAP Estimation of Parameters)

Bernoulli Naïve Bayes:
Parameters are probabilities → Beta prior (usually) pushes probabilities away from zero / one extremes

Logistic Regression:
Parameters are not probabilities → Gaussian prior encourages parameters to be close to zero

(effectively pushes the probabilities away from zero / one extremes)

Naïve Bayes vs. Logistic Reg.

Features

Naïve Bayes:
Features $\boldsymbol{x}$ are assumed to be conditionally independent given y. (i.e. Naïve Bayes Assumption)

Logistic Regression:
No assumptions are made about the form of the features $\boldsymbol{x}$. They can be dependent and correlated in any fashion.

Learning Objectives

Naïve Bayes

You should be able to…

1. Write the generative story for Naive Bayes
2. Create a new Naive Bayes classifier using your favorite probability distribution as the event model
3. Apply the principle of maximum likelihood estimation (MLE) to learn the parameters of Bernoulli Naive Bayes
4. Motivate the need for MAP estimation through the deficiencies of MLE
5. Apply the principle of maximum a posteriori (MAP) estimation to learn the parameters of Bernoulli Naive Bayes
6. Select a suitable prior for a model parameter
7. Describe the tradeoffs of generative vs. discriminative models
8. Implement Bernoulli Naives Bayes
9. Employ the method of Lagrange multipliers to find the MLE parameters of Multinomial Naive Bayes
10. Describe how the variance affects whether a Gaussian Naive Bayes model will have a linear or nonlinear decision boundary