10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Naïve Bayes

+

Generative vs. Discriminative

Matt Gormley
Lecture 18
Mar. 25, 2019

Reminders

- Homework 6: Learning Theory / Generative Models
  - Out: Fri, Mar 22
  - Due: Fri, Mar 29 at 11:59pm (1 week)
- Midterm Exam 2
  - Thu, Apr 4 – evening exam, details announced on Piazza
- Homework 7: HMMs
  - Out: Fri, Mar 29
  - Due: Wed, Apr 10 at 11:59pm
- Today's In-Class Poll
  - http://p18.mlcourse.org

Q&A

Q: Why would we use Naïve Bayes? Isn't it too Naïve?

A: Naïve Bayes has one **key advantage** over methods like Perceptron, Logistic Regression, Neural Nets:

Training is lightning fast!

While other methods require slow iterative training procedures that might require hundreds of epochs, Naïve Bayes computes its parameters in closed form by counting.

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**

Naïve Bayes

- Why are we talking about Naïve Bayes?
  - It's **just another decision function** that fits into our "big picture" recipe from last time
  - But it's our first **example of a Bayesian Network** and provides a *clearer* picture of **probabilistic learning**
  - Just like the other Bayes Nets we'll see, it **admits a closed form solution** for MLE and MAP
  - So learning is **extremely efficient** (just counting)

Fake News Detector

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

CNN


U.S. » Crime + Justice | Energy + Environment | Extreme Weather | Space + Science | Live TV | U.S. Edition +

People are petitioning the White House to move Halloween to Saturdays

By Doug Criss, CNN
Updated 10:18 AM ET, Wed October 31, 2018

Trick-or-treating would be a weekend-only affair if an unofficial presidential petition were granted.

(CNN) — Americans long have used petitions to try to solve the nation's thorniest problems, and now a really weighty issue has landed atop the stack: moving Halloween to the weekend.

The change.org petition, addressed to President Donald Trump, wants the US to celebrate Halloween on the last Saturday of October instead of October 31. Nearly 40,000 people had signed it by midmorning Tuesday.

"It's time for a safer, longer, stress-free celebration!"

More from CNN

Longtime conservative columnist tells why he's a 'Never Trumper'

A 5-year-old boy tested positive for meth after...

Paid Content by Outbrain

These Eight Luxury Vacation Tours And Packages Are Incredible! And Under $2000!
Travel Enthusiast | Sponsored Links

Wow - These New Luxury Cars Are Loaded
Faqeo

Watch Parkeon Smart Kiosks Transform City Parking

The Onion


NEWS

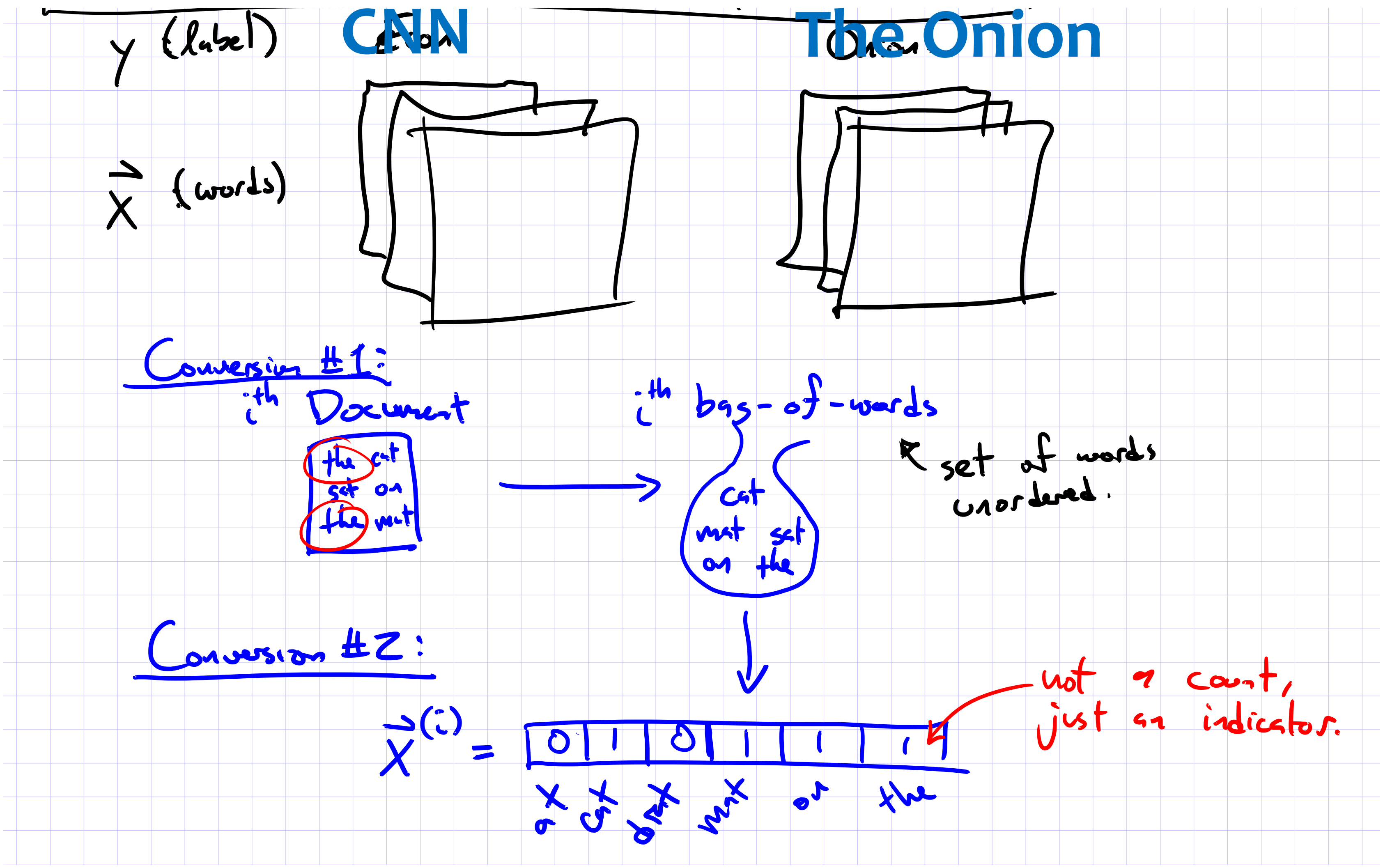
'Take This Grape For It Is The Witch's Eye, Take This Spaghetti For It Is The Witch's Brain,' Says Pope Francis During Halloween-Themed Communion

Today 12:00pm • SEE MORE: HALLOWEEN

Pope Francis administers the unholy sacrament to parishioners in the Haunted House of the Lord.

VATICAN CITY—Standing before his costumed congregants in St. Peter's Basilica, Pope Francis declared, "Take this grape for it is the witch's eye, take this spaghetti for it is the witch's brain," during a Halloween-themed Communion Wednesday, Vatican sources confirmed.

Fake News Detector



We can pretend the natural process generating these vectors is stochastic…

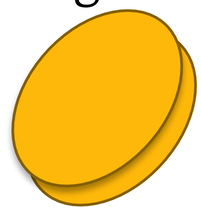
Naive Bayes: Model

Whiteboard

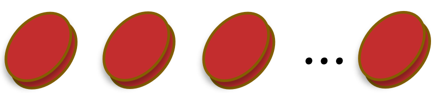
- Document → bag-of-words → binary feature vector
- Generating synthetic "labeled documents"
- Definition of model
- Naive Bayes assumption
- Counting # of parameters with / without NB assumption

Model 1: Bernoulli Naïve Bayes

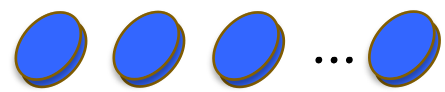
Flip weighted coin



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 •

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

Recipe for Closed-form MLE

1. Assume data was generated i.i.d. from some model (i.e. write the generative story)

   $x^{(i)} \sim p(x|\boldsymbol{\theta})$

2. Write log-likelihood

   $\ell(\boldsymbol{\theta}) = \log p(x^{(1)}|\boldsymbol{\theta}) + \ldots + \log p(x^{(N)}|\boldsymbol{\theta})$

3. Compute partial derivatives (i.e. gradient)

   $\partial\ell(\boldsymbol{\theta})/\partial\theta_1 = \ldots$

   $\partial\ell(\boldsymbol{\theta})/\partial\theta_2 = \ldots$

   …

   $\partial\ell(\boldsymbol{\theta})/\partial\theta_M = \ldots$

4. Set derivatives to zero and solve for $\boldsymbol{\theta}$

   $\partial\ell(\boldsymbol{\theta})/\partial\theta_m = 0$ for all $m \in \{1, \ldots, M\}$

   $\boldsymbol{\theta}^{MLE}$ = solution to system of M equations and M variables

5. Compute the second derivative and check that $\ell(\boldsymbol{\theta})$ is concave down at $\boldsymbol{\theta}^{MLE}$

NAÏVE BAYES: MODEL DETAILS

Model 1: Bernoulli Naïve Bayes

Data: Binary feature vectors, Binary labels

$$\mathbf{x} \in \{0,1\}^M \qquad y \in \{0,1\}$$

Generative Story:

$$y \sim \text{Bernoulli}(\phi)$$
$$x_1 \sim \text{Bernoulli}(\theta_{y,1})$$
$$x_2 \sim \text{Bernoulli}(\theta_{y,2})$$
$$\vdots$$
$$x_M \sim \text{Bernoulli}(\theta_{y,M})$$

Model:

$$\begin{aligned}
p_{\phi,\boldsymbol{\theta}}(\boldsymbol{x}, y) &= p_{\phi,\boldsymbol{\theta}}(x_1, \ldots, x_M, y) \\
&= p_\phi(y) \prod_{m=1}^{M} p_{\boldsymbol{\theta}}(x_m|y) \\
&= \Big[(\phi)^y (1-\phi)^{(1-y)} \\
&\quad \prod_{m=1}^{M} (\theta_{y,m})^{x_m} (1-\theta_{y,m})^{(1-x_m)}\Big]
\end{aligned}$$

Model 1: Bernoulli Naïve Bayes

Maximum Likelihood Estimation

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

Count Variables:

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

$$N_{y=0} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0)$$

$$N_{y=0,x_m=1} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0 \wedge x_m^{(i)} = 1)$$

$$\dots$$

Maximum Likelihood Estimators:

$$\phi = \frac{N_{y=1}}{N}$$

$$\theta_{0,m} = \frac{N_{y=0,x_m=1}}{N_{y=0}}$$

$$\theta_{1,m} = \frac{N_{y=1,x_m=1}}{N_{y=1}}$$

$$\forall m \in \{1, \dots, M\}$$

Model 1: Bernoulli Naïve Bayes

Maximum Likelihood Estimation

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

Count Variables:

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

$$N_{y=0} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0)$$

$$N_{y=0,x_m=1} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0 \wedge x_m^{(i)} = 1)$$

$$\ldots$$

Maximum Likelihood Estimators:

$$\phi = \frac{N_{y=1}}{N}$$

$$\theta_{0,m} = \frac{N_{y=0,x_m=1}}{N_{y=0}}$$

$$\theta_{1,m} = \frac{N_{y=1,x_m=1}}{N_{y=1}}$$

$$\forall m \in \{1, \ldots, M\}$$

Data:

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

Question 1:

What is the MLE of ϕ?

(A) 0/6 (B) 1/6 (C) 2/6 (D) 3/6
(E) 4/6 (F) 5/6 (G) 6/6 (H) None of the above

Model 1: Bernoulli Naïve Bayes

Maximum Likelihood Estimation

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

Count Variables:

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

$$N_{y=0} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0)$$

$$N_{y=0,x_m=1} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0 \wedge x_m^{(i)} = 1)$$

$$\ldots$$

Maximum Likelihood Estimators:

$$\phi = \frac{N_{y=1}}{N}$$

$$\theta_{0,m} = \frac{N_{y=0,x_m=1}}{N_{y=0}}$$

$$\theta_{1,m} = \frac{N_{y=1,x_m=1}}{N_{y=1}}$$

$$\forall m \in \{1, \ldots, M\}$$

Data:

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

Question 2:

What is the MLE of $\theta_{0,1}$?

(A) 0/6 (B) 1/6 (C) 2/6 (D) 3/6

(E) 4/6 (F) 5/6 (G) 6/6 (H) None of the above

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

A Shortcoming of 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?

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

Model 1: Bernoulli Naïve Bayes

MAP Estimation (Beta Prior)

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

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

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

$$N_{y=0} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0)$$

$$N_{y=0,x_m=1} = \sum_{i=1}^{N} \mathbb{I}(y^{(i)} = 0 \wedge x_m^{(i)} = 1)$$

$\ldots$

2. Likelihood:

$$\ell_{MAP}(\phi, \boldsymbol{\theta}) = \log \left[p(\phi, \boldsymbol{\theta} | \alpha, \beta) p(\mathcal{D} | \phi, \boldsymbol{\theta}) \right] = \log \left[\left(p(\phi | \alpha, \beta) \prod_{m=1}^{M} p(\theta_{0,m} | \alpha, \beta) \right) \left(\prod_{i=1}^{N} p(\mathbf{x}^{(i)}, y^{(i)} | \phi, \boldsymbol{\theta}) \right) \right]$$

3. MAP Estimates: $(\phi^{MAP}, \boldsymbol{\theta}^{MAP}) = \underset{\phi, \boldsymbol{\theta}}{\text{argmax}}\, \ell_{MAP}(\phi, \boldsymbol{\theta})$

Take derivatives, set to zero and solve…

$$\phi = \frac{N_{y=1}}{N}$$

$$\theta_{0,m} = \frac{(\alpha - 1) + N_{y=0,x_m=1}}{(\alpha - 1) + (\beta - 1) + N_{y=0}}$$

$$\theta_{1,m} = \frac{(\alpha - 1) + N_{y=1,x_m=1}}{(\alpha - 1) + (\beta - 1) + N_{y=1}}$$

$$\forall m \in \{1, \ldots, M\}$$

Other NB Models

1. **Bernoulli** Naïve Bayes:
   – for **binary features**
2. **Multinomial** Naïve Bayes:
   – for **integer features**
3. **Gaussian** Naïve Bayes:
   – for **continuous 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: 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.

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

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

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

Generic Naïve Bayes Model

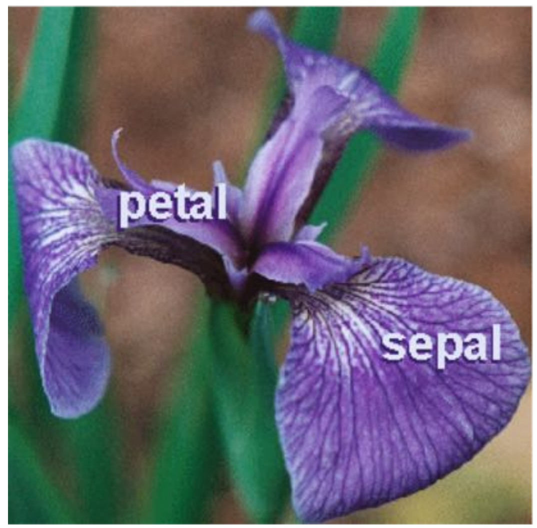
Classification:

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

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

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

VISUALIZING GAUSSIAN NAÏVE BAYES

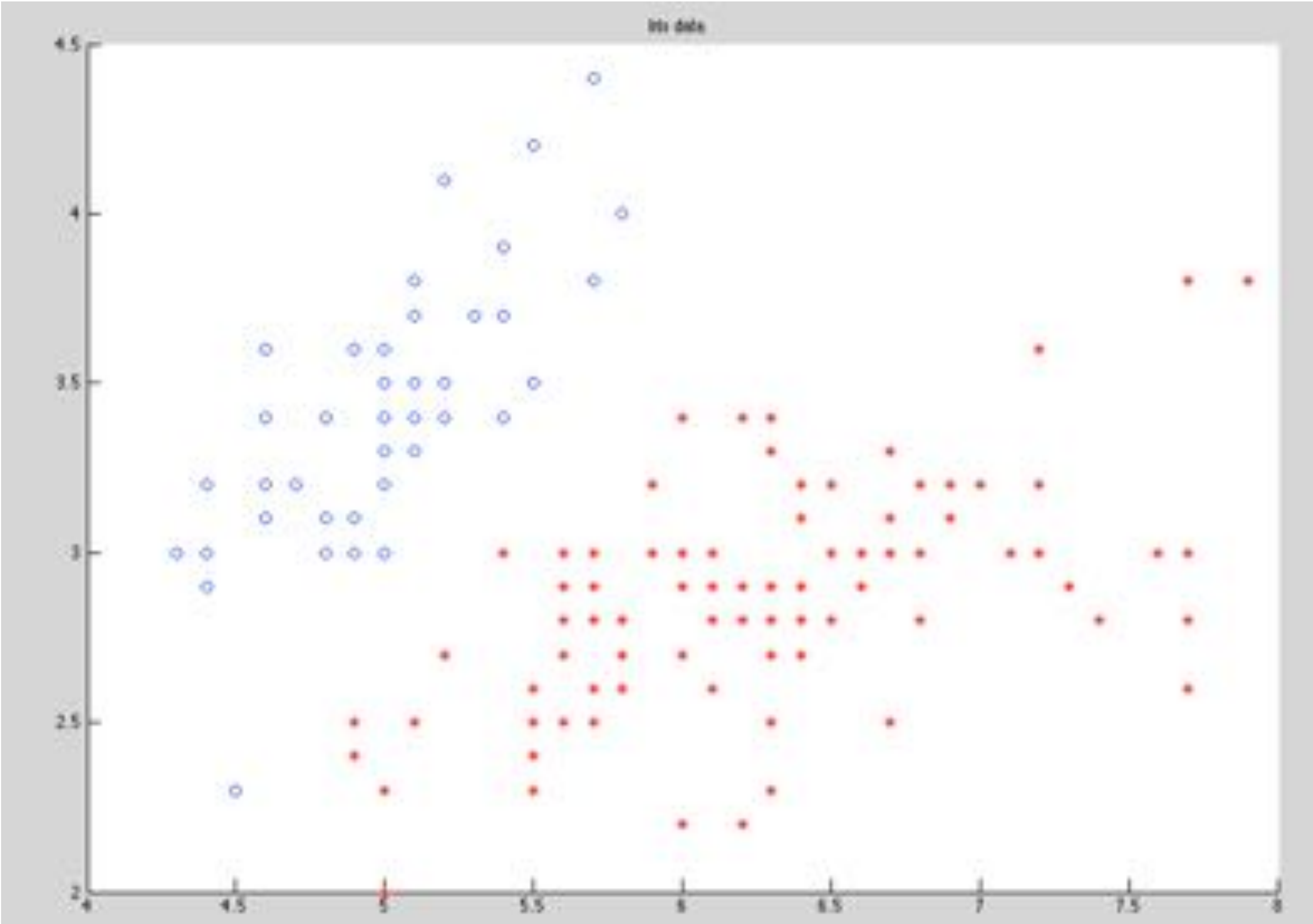

petal
sepal

Fisher Iris Dataset

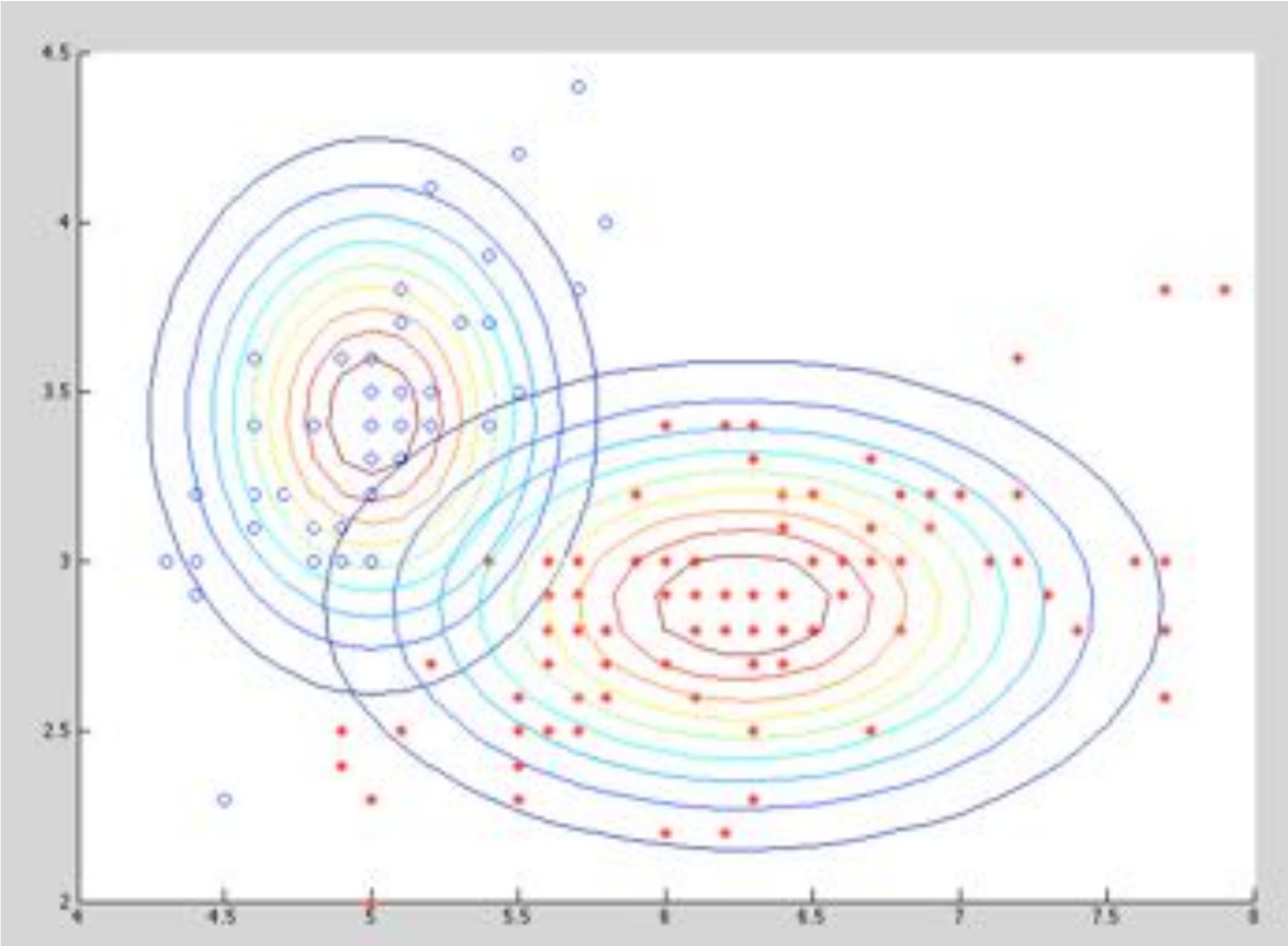
Fisher (1936) used 150 measurements of flowers from 3 different species: Iris setosa (0), Iris virginica (1), Iris versicolor (2) collected by Anderson (1936)

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

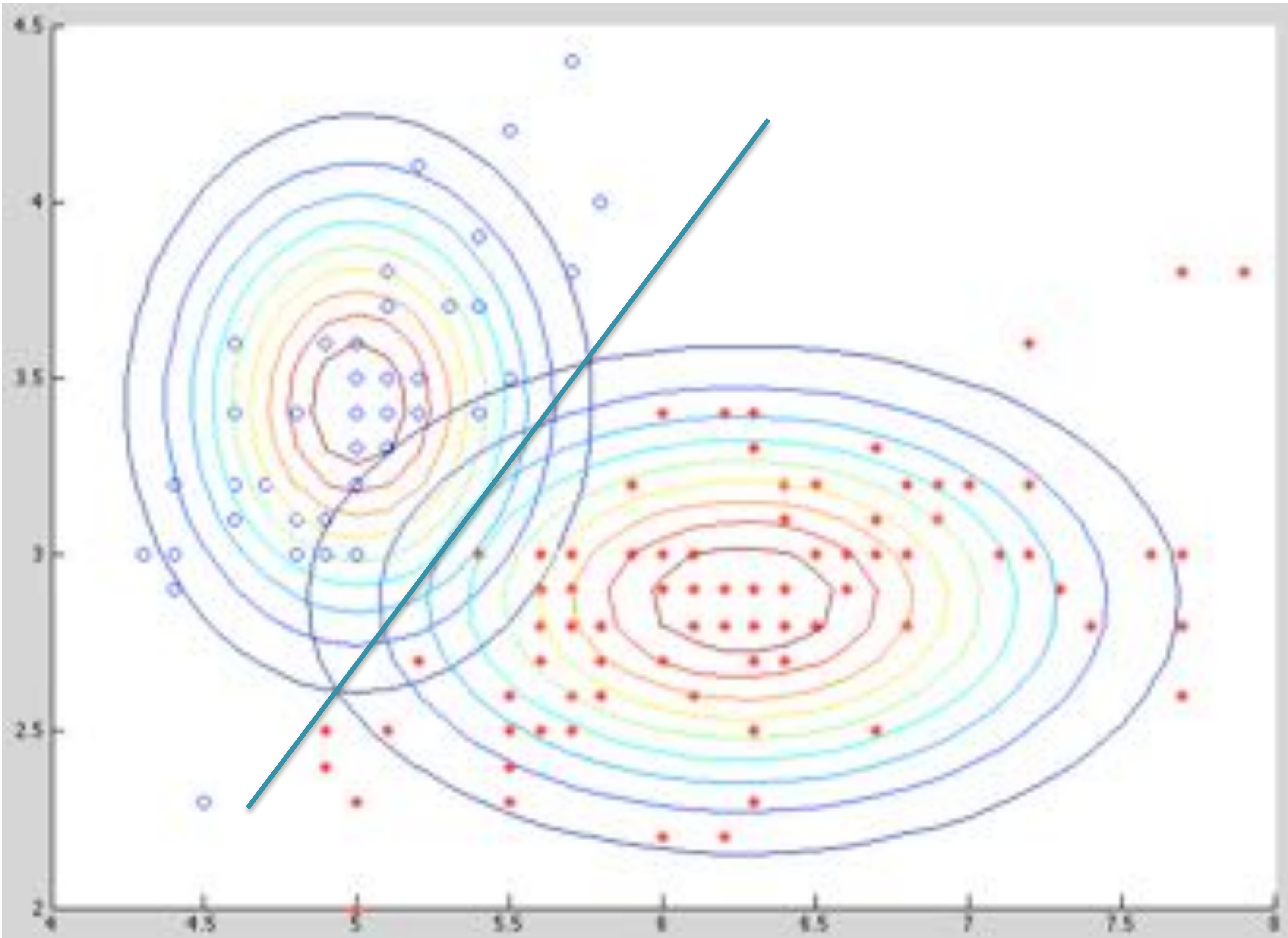


Slide from William Cohen



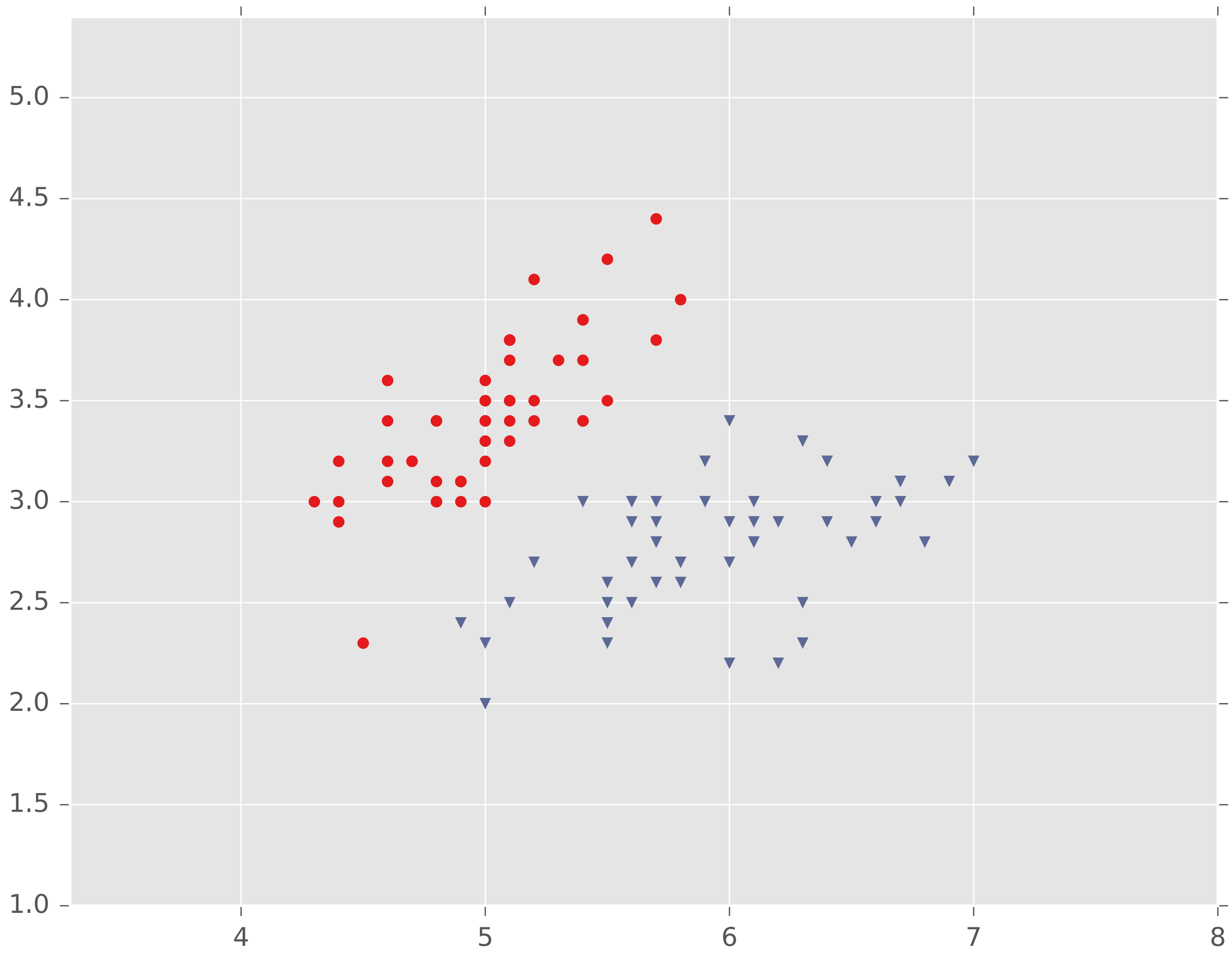
Slide from William Cohen

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

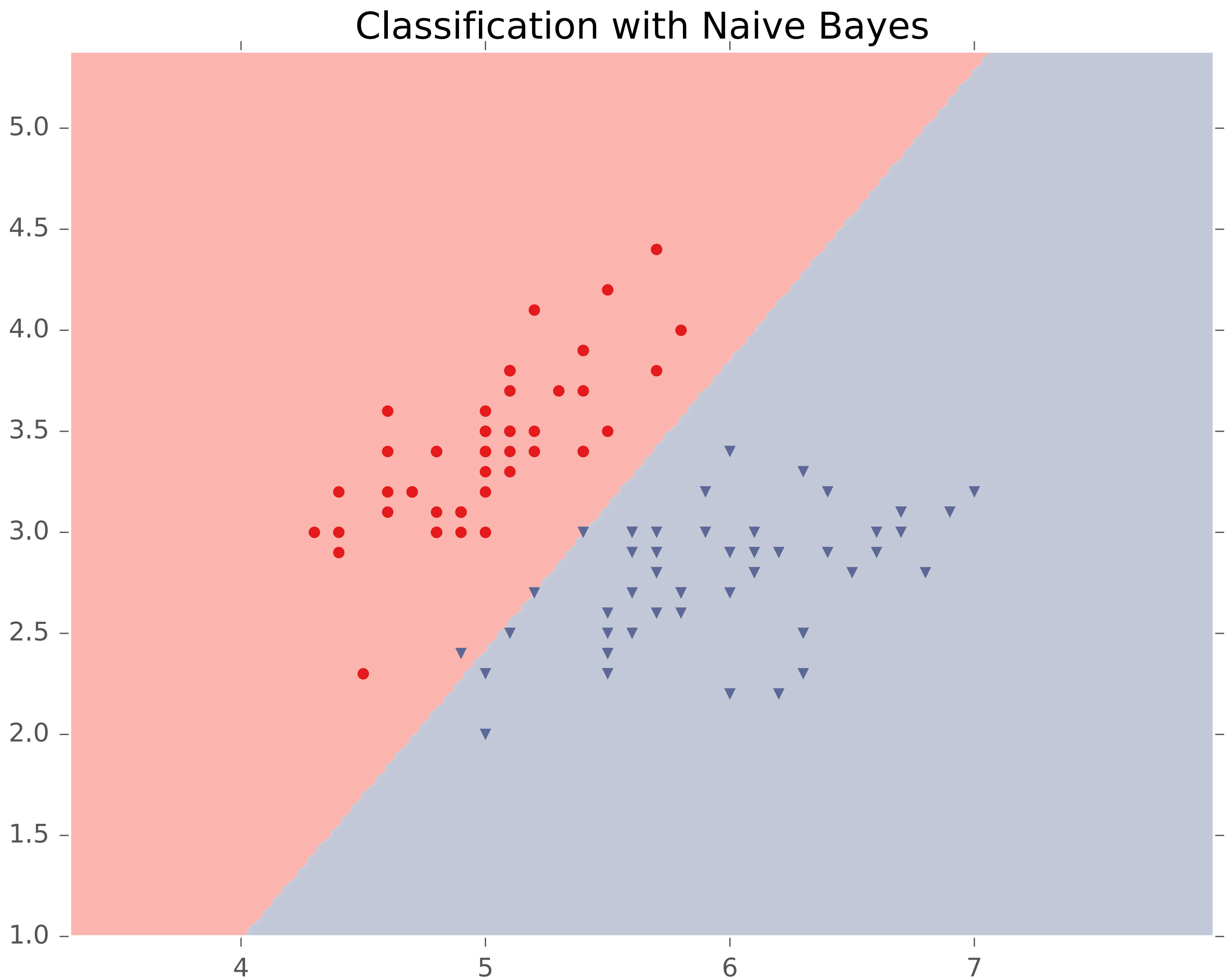


Slide from William Cohen (10-601B, Spring 2016)

Iris Data (2 classes)

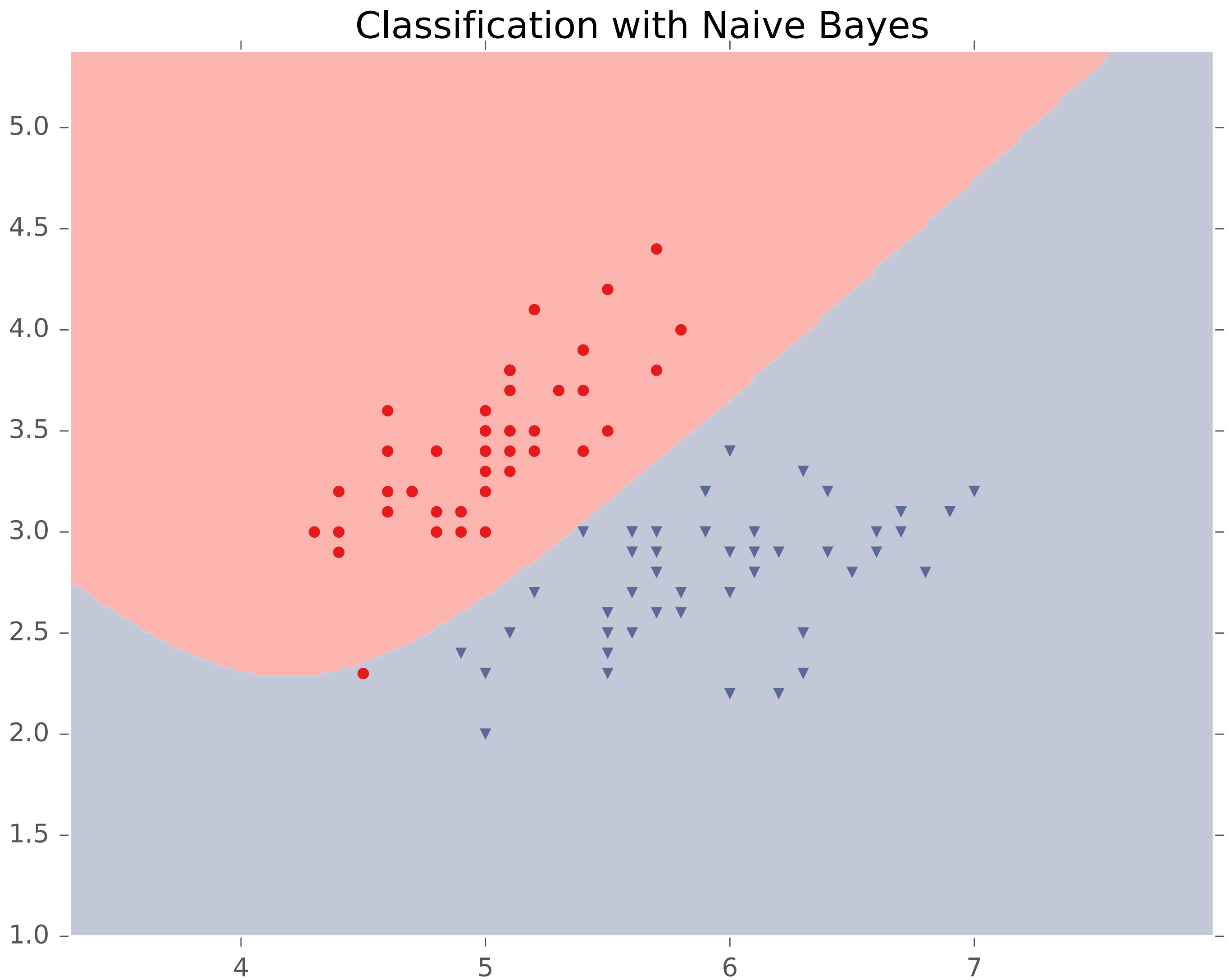


Iris Data (2 classes)



variance = 1

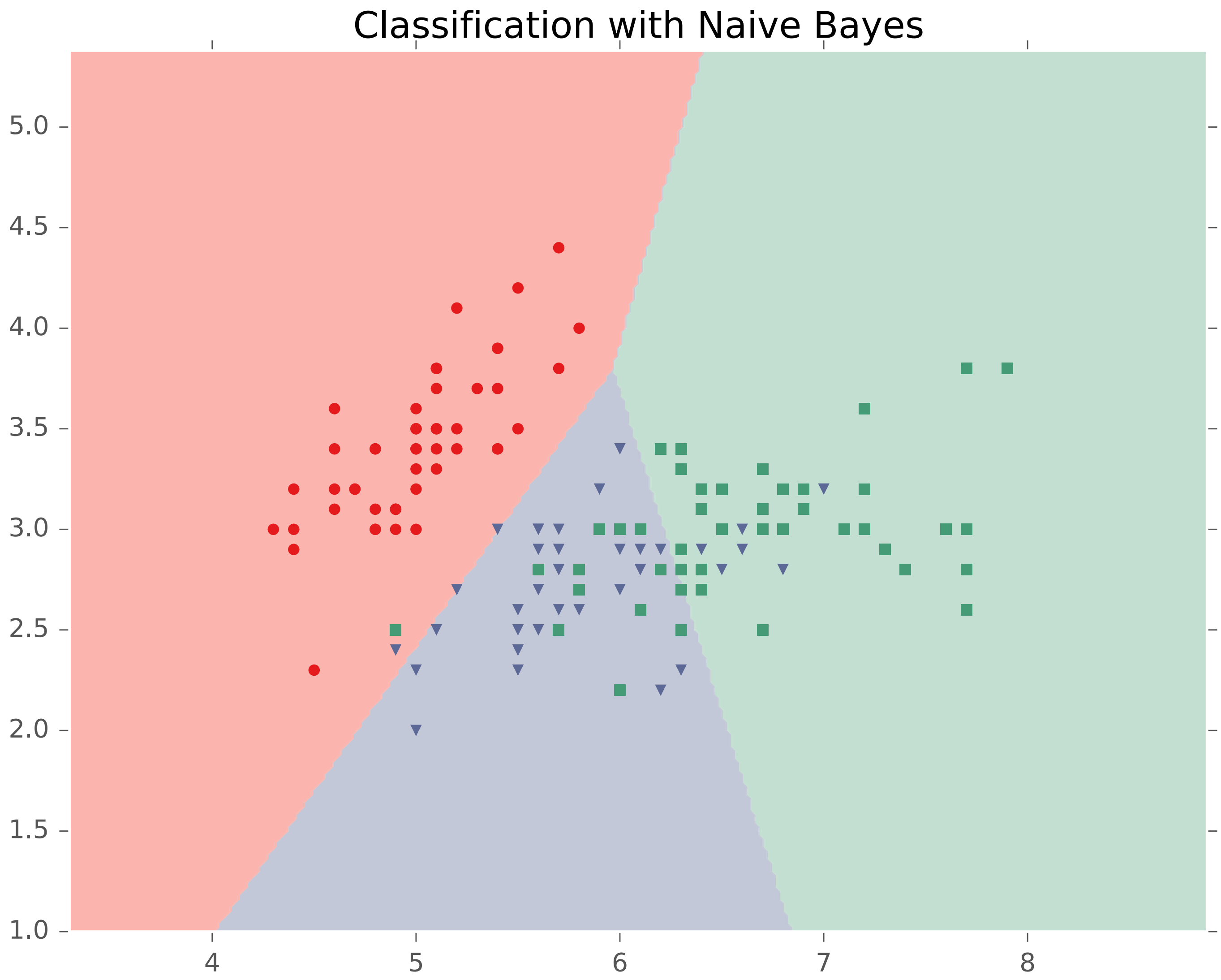
Iris Data (2 classes)



variance learned for each class

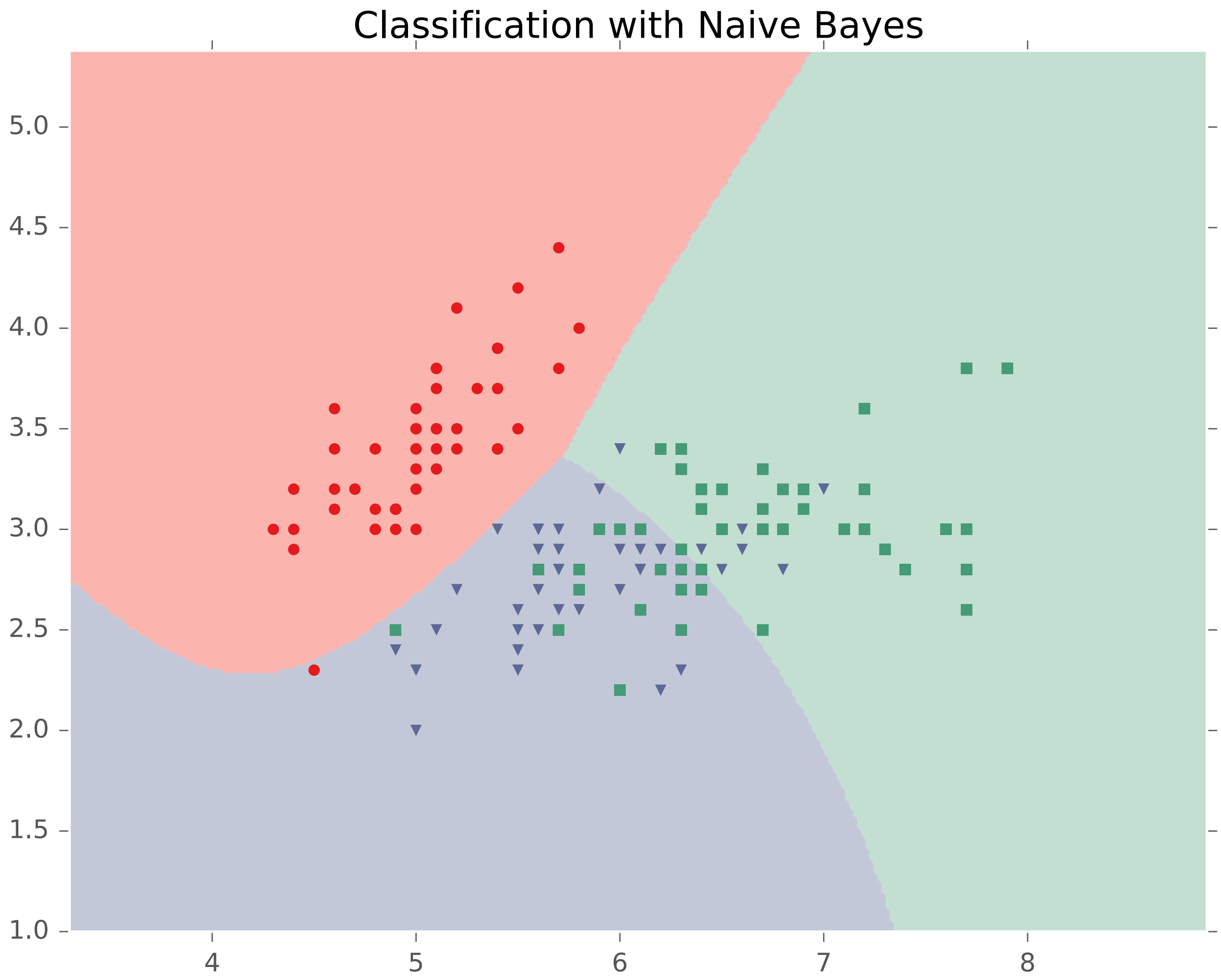
Iris Data (3 classes)

Iris Data (3 classes)



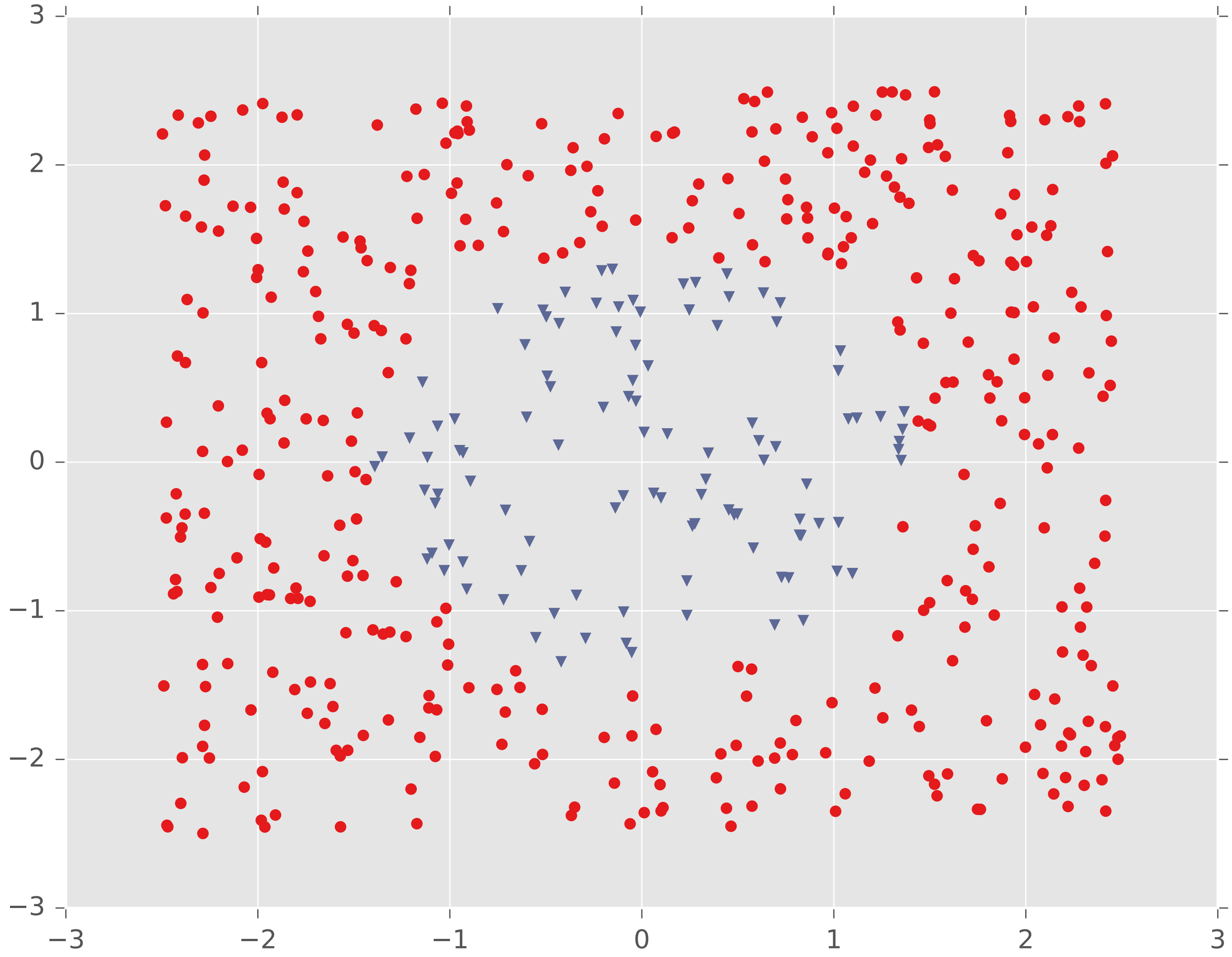
variance = 1

Iris Data (3 classes)

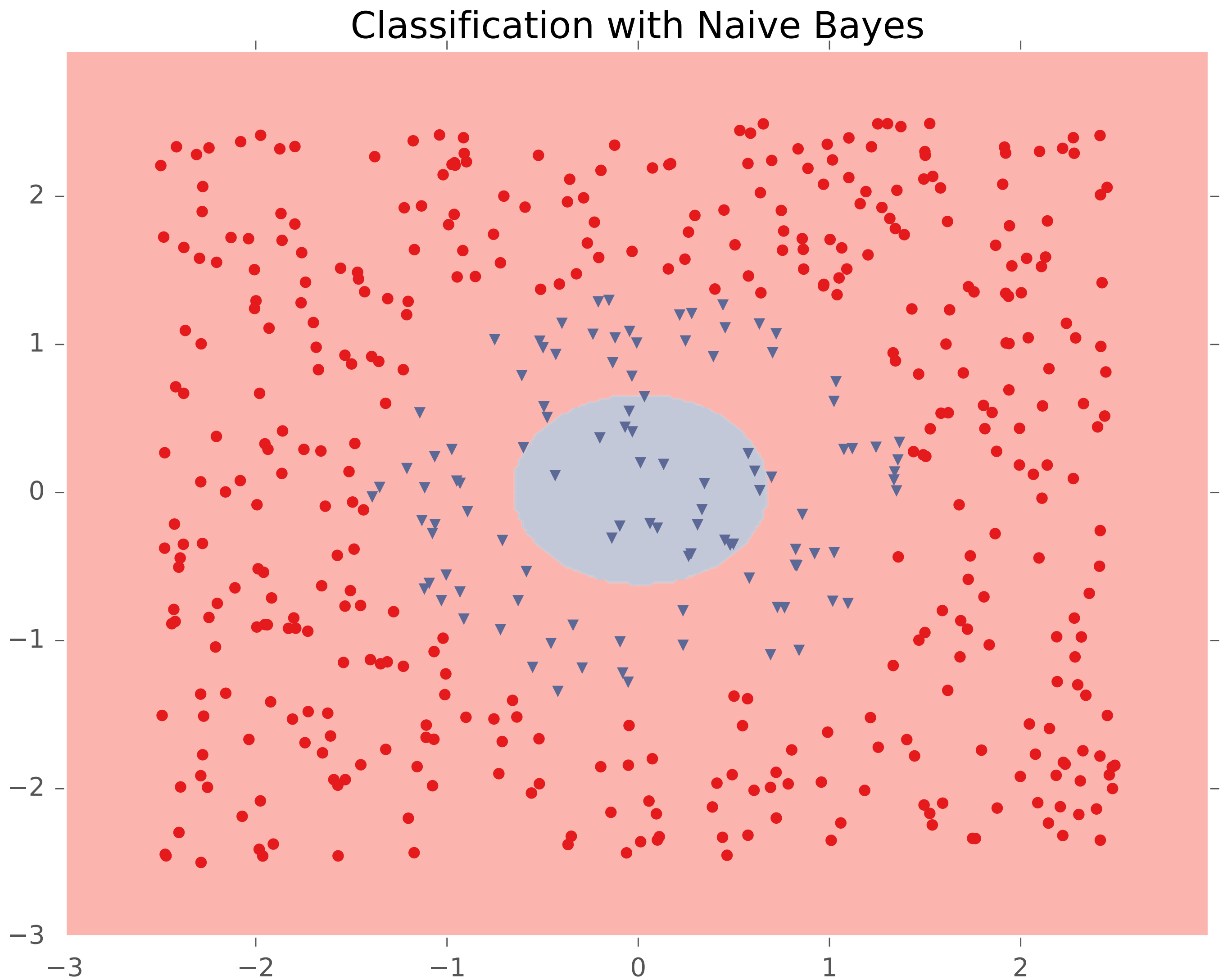


variance learned for each class

One Pocket

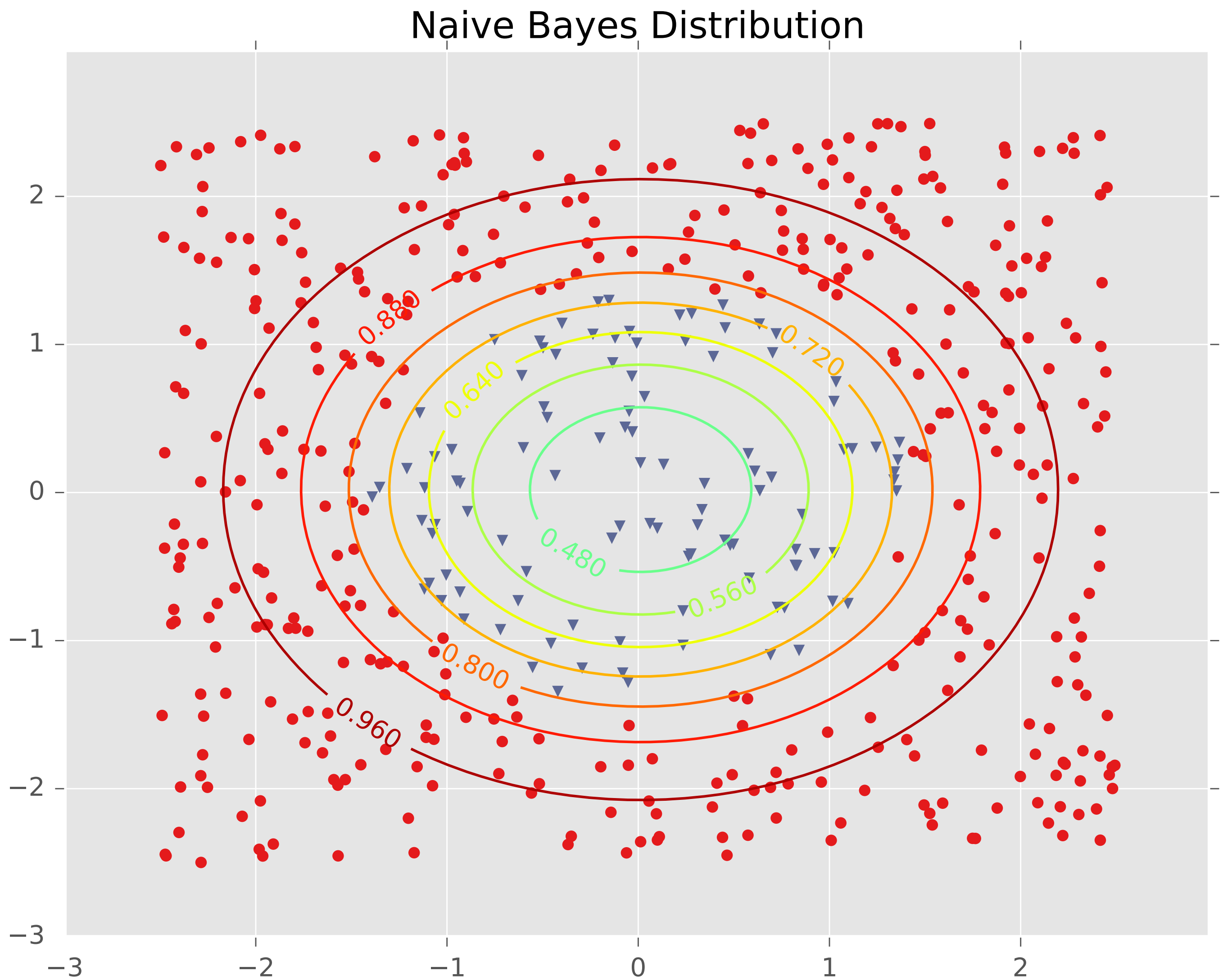


One Pocket



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

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

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

	Gen.	Disc.
MLE	$\prod_i p(\mathbf{x}^{(i)}, y^{(i)} \mid \boldsymbol{\theta})$	$\prod_i p(y^{(i)} \mid \mathbf{x}^{(i)}, \boldsymbol{\theta})$
MAP	$p(\boldsymbol{\theta}) \prod_i p(\mathbf{x}^{(i)}, y^{(i)} \mid \boldsymbol{\theta})$	$p(\boldsymbol{\theta}) \prod_i p(y^{(i)} \mid \mathbf{x}^{(i)}, \boldsymbol{\theta})$

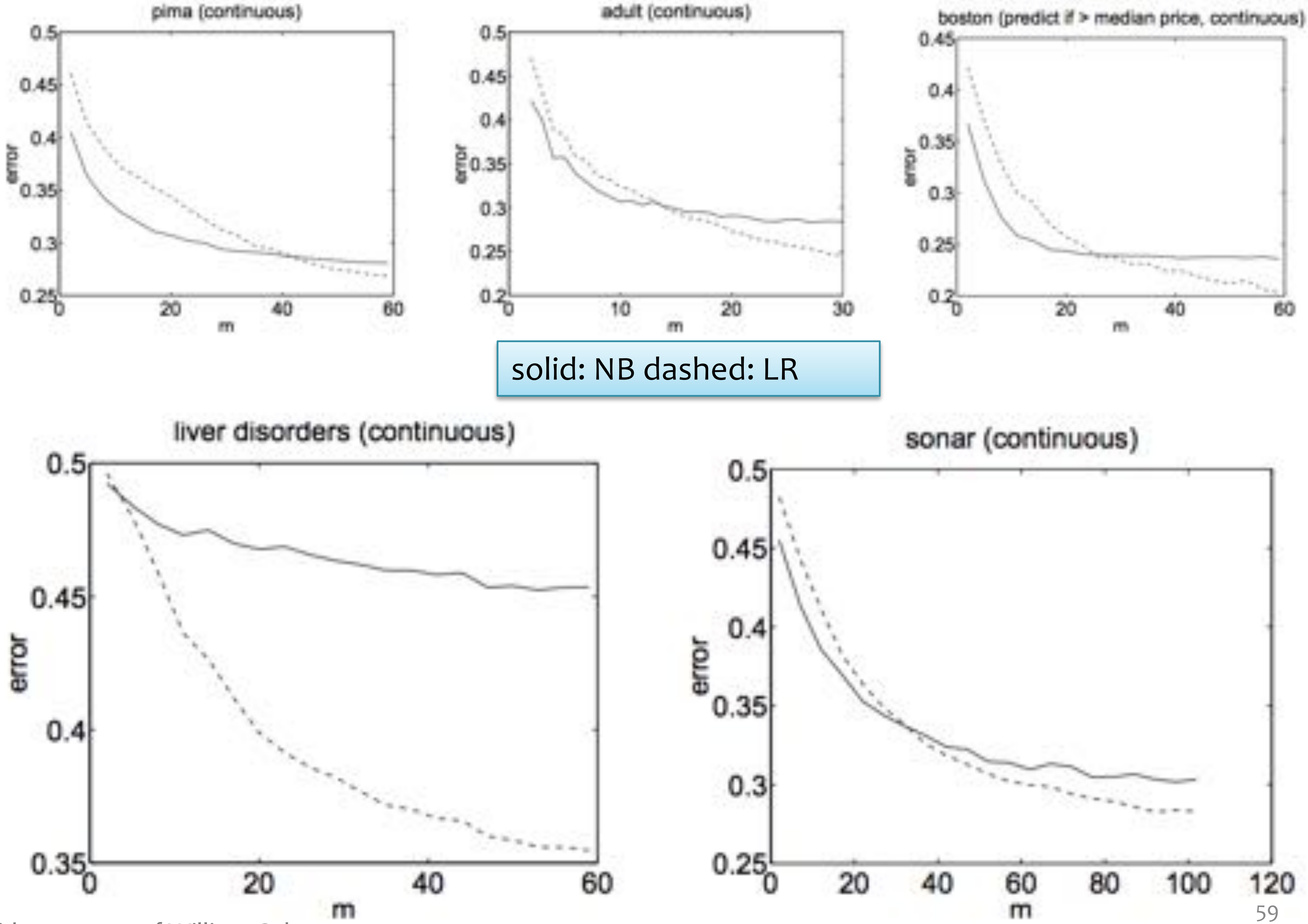
Generative vs. Discriminative

Finite Sample Analysis (Ng & Jordan, 2002)

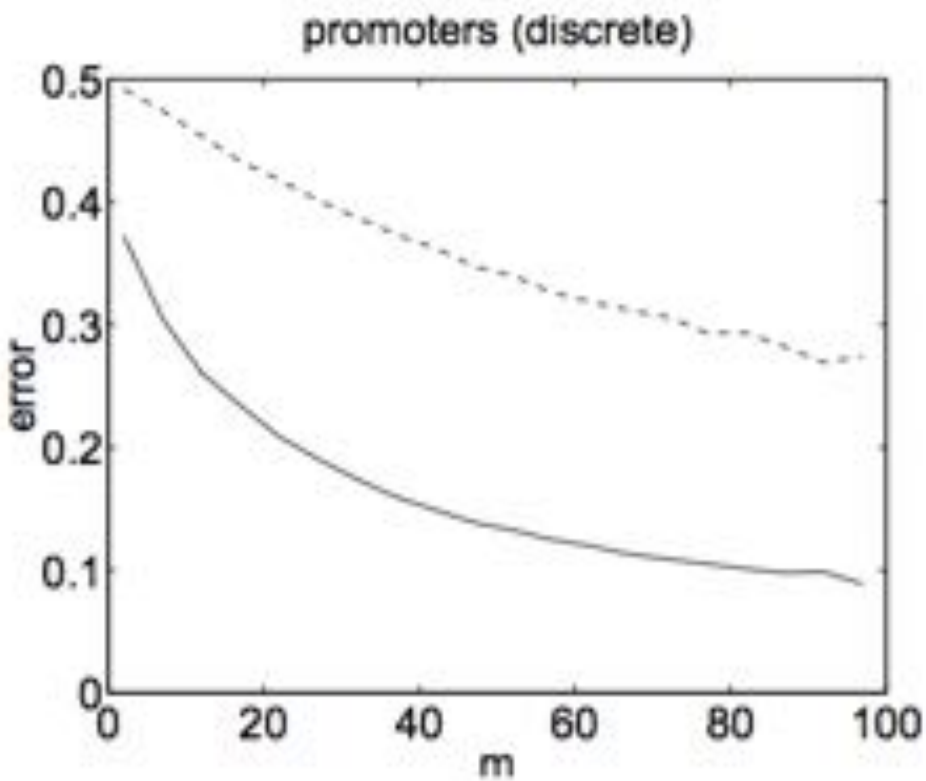
[Assume that we are learning from a finite training dataset]

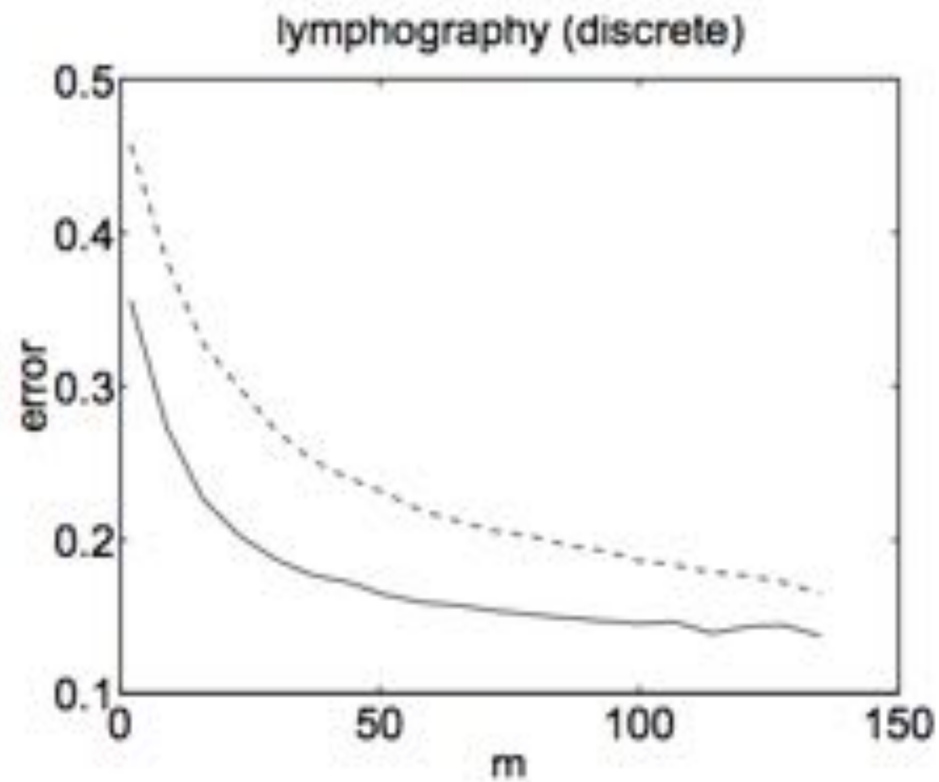
If model assumptions are correct: Naive Bayes is a more efficient learner (requires fewer samples) than Logistic Regression

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


pima (continuous)
adult (continuous)
boston (predict if > median price, continuous)
solid: NB dashed: LR
liver disorders (continuous)
sonar (continuous)
error
m
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0
10
20
30
40
60
80
100
120

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.

Slide courtesy of William Cohen

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.

MOTIVATION: STRUCTURED PREDICTION

Structured Prediction

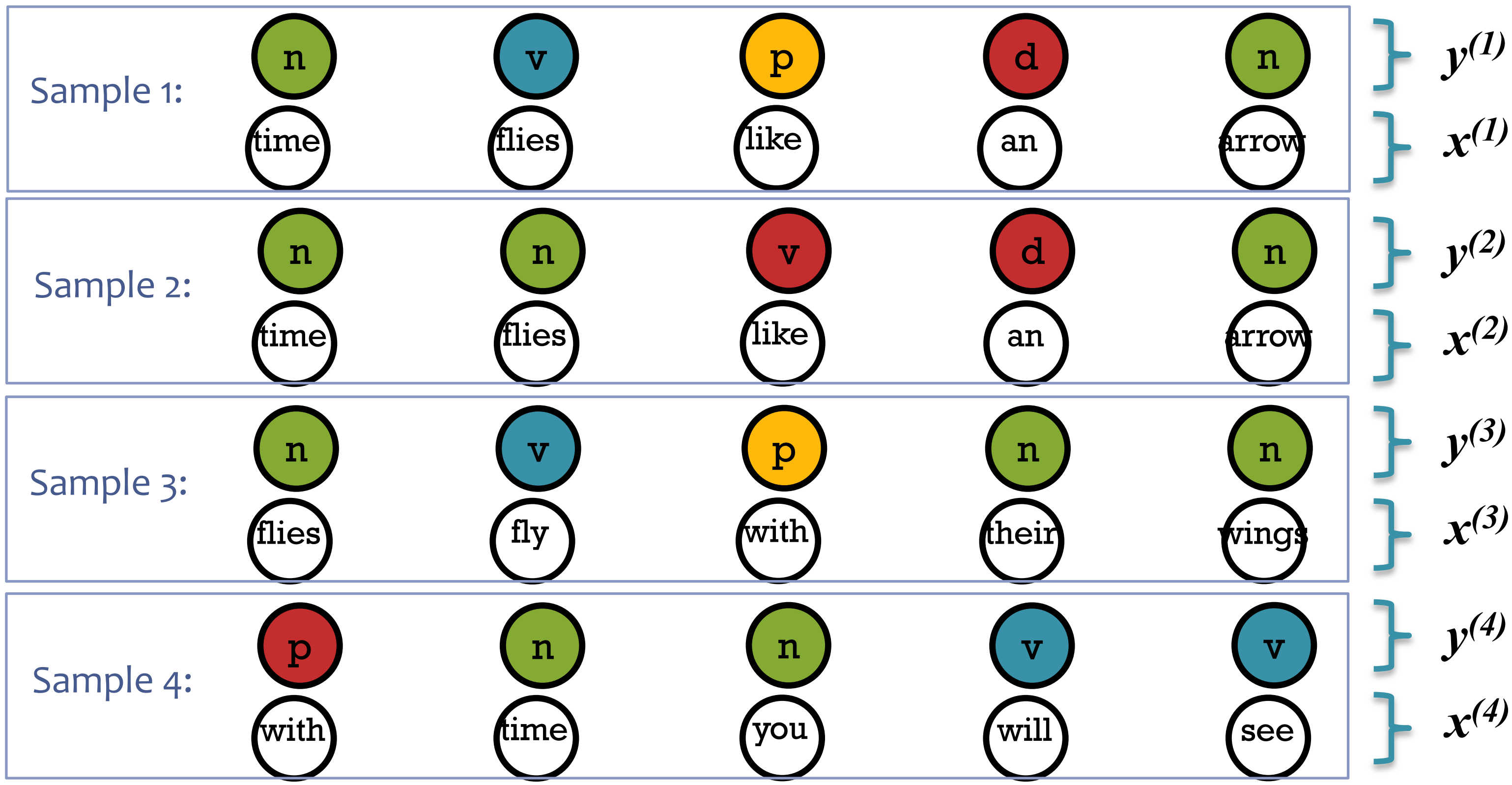
- Most of the models we've seen so far were for **classification**
  - Given observations: $\boldsymbol{x} = (x_1, x_2, \ldots, x_K)$
  - Predict a (binary) **label:** y
- Many real-world problems require **structured prediction**
  - Given observations: $\boldsymbol{x} = (x_1, x_2, \ldots, x_K)$
  - Predict a **structure:** $\boldsymbol{y} = (y_1, y_2, \ldots, y_J)$
- Some *classification* problems benefit from **latent structure**

Structured Prediction Examples

- **Examples of structured prediction**
  - Part-of-speech (POS) tagging
  - Handwriting recognition
  - Speech recognition
  - Word alignment
  - Congressional voting
- **Examples of latent structure**
  - Object recognition

Dataset for Supervised Part-of-Speech (POS) Tagging

Data: $\mathcal{D} = \{\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}\}_{n=1}^{N}$



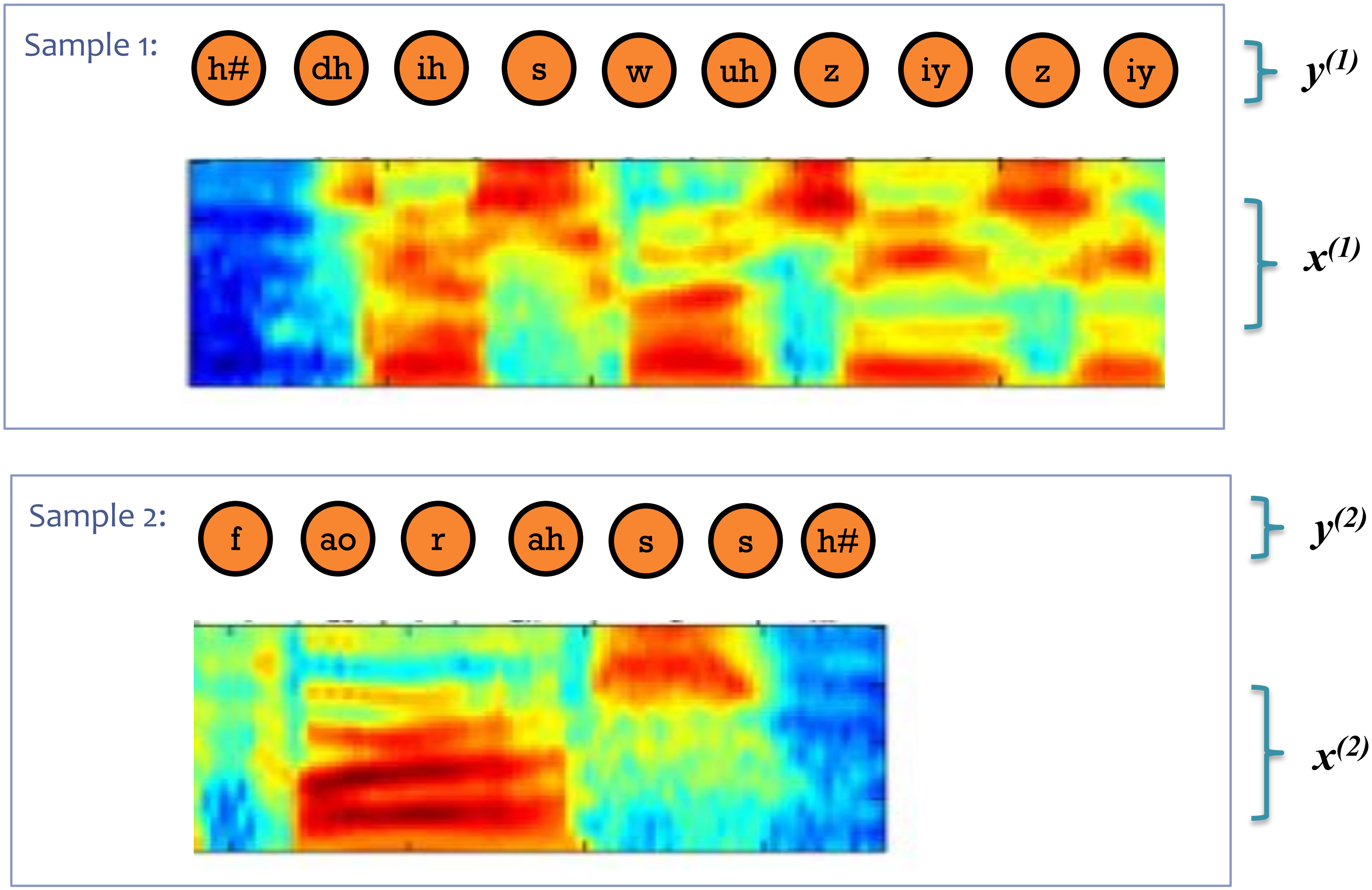
Dataset for Supervised Handwriting Recognition

Data: $\mathcal{D} = \{\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}\}_{n=1}^{N}$

Figures from (Chatzis & Demiris, 2013)

Dataset for Supervised Phoneme (Speech) Recognition

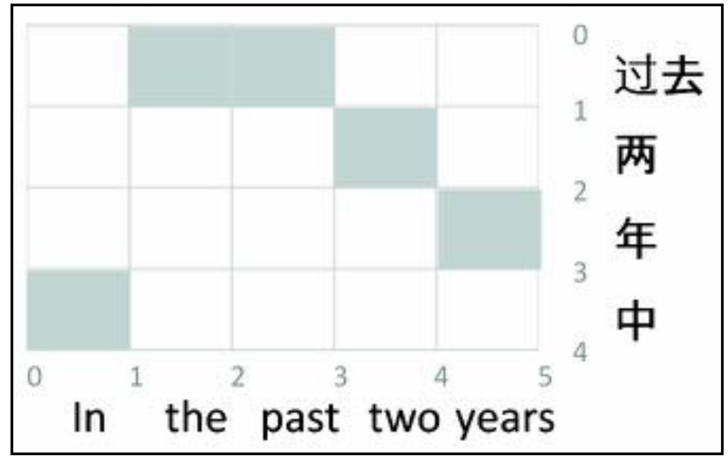
Data: $\mathcal{D} = \{\boldsymbol{x}^{(n)}, \boldsymbol{y}^{(n)}\}_{n=1}^{N}$



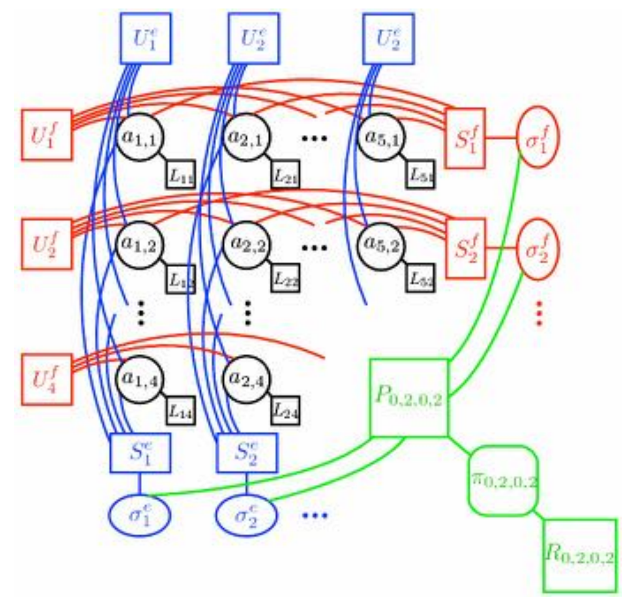
Figures from (Jansen & Niyogi, 2013)

Word Alignment / Phrase Extraction

- **Variables (boolean):**
  - For each (Chinese phrase, English phrase) pair, are they linked?



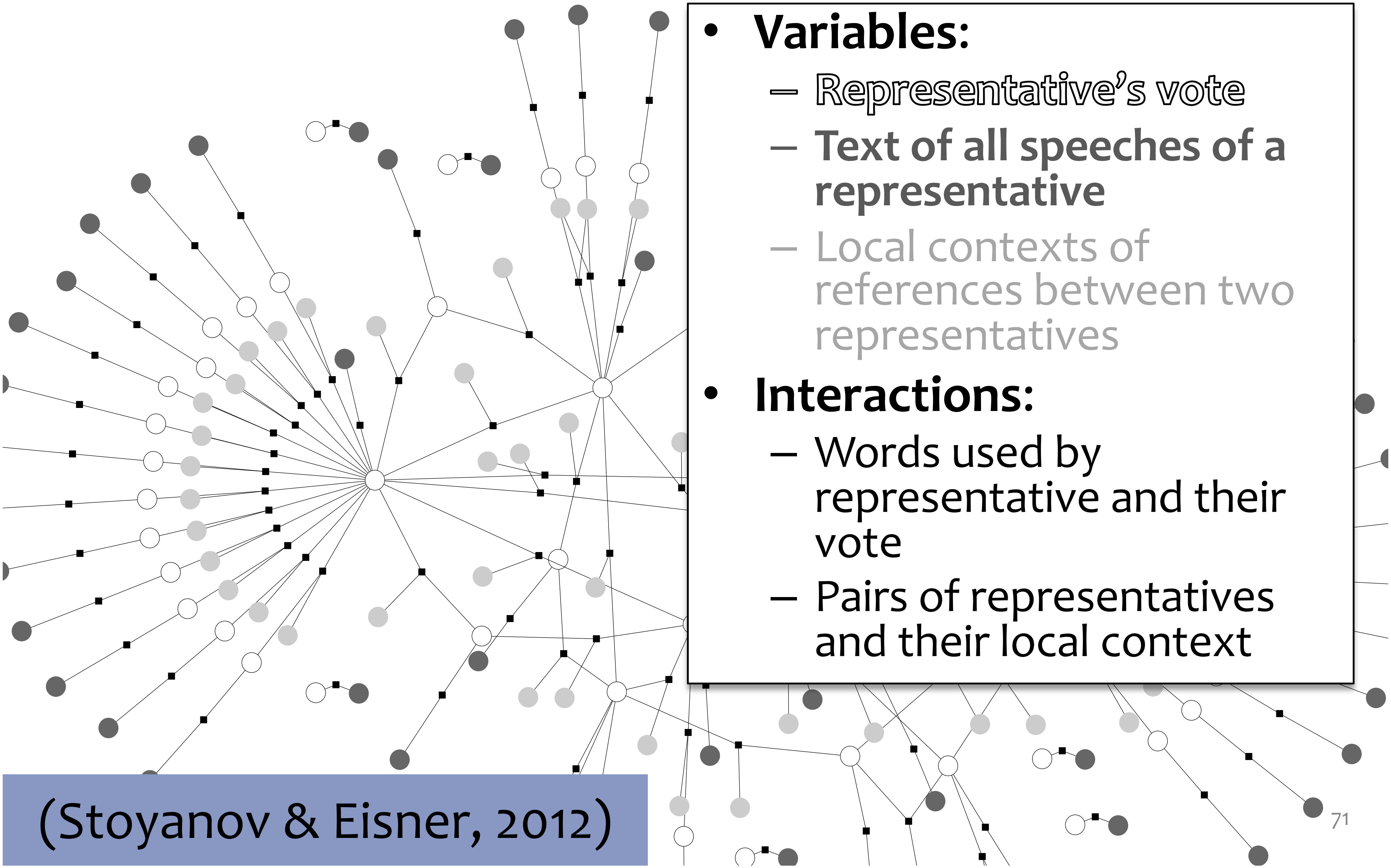
- **Interactions:**
  - Word fertilities
  - Few “jumps” (discontinuities)
  - Syntactic reorderings
  - “ITG contraint” on alignment
  - Phrases are disjoint (?)



(Burkett & Klein, 2012)

Application:

Congressional Voting



- **Variables:**
  - Representative's vote
  - **Text of all speeches of a representative**
  - Local contexts of references between two representatives
- **Interactions:**
  - Words used by representative and their vote
  - Pairs of representatives and their local context

(Stoyanov & Eisner, 2012)

Structured Prediction Examples

- **Examples of structured prediction**
  - Part-of-speech (POS) tagging
  - Handwriting recognition
  - Speech recognition
  - Word alignment
  - Congressional voting
- **Examples of latent structure**
  - Object recognition

Case Study: Object Recognition

Data consists of images $\boldsymbol{x}$ and labels y.

$x^{(1)}$

pigeon

$y^{(1)}$

$x^{(2)}$

rhinoceros

$y^{(2)}$

$x^{(3)}$

leopard

$y^{(3)}$

$x^{(4)}$

llama

$y^{(4)}$

Case Study: Object Recognition

Data consists of images x and labels y.

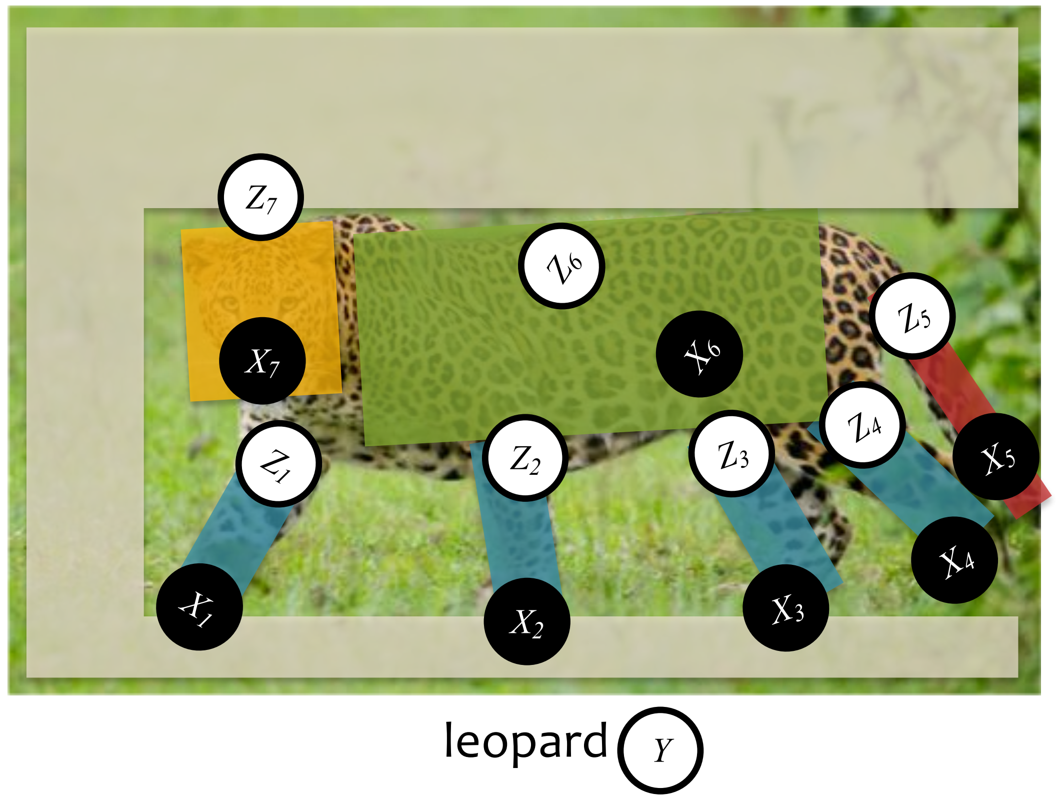
- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- z is not observed at train or test time

leopard

Case Study: Object Recognition

Data consists of images $\boldsymbol{x}$ and labels y.

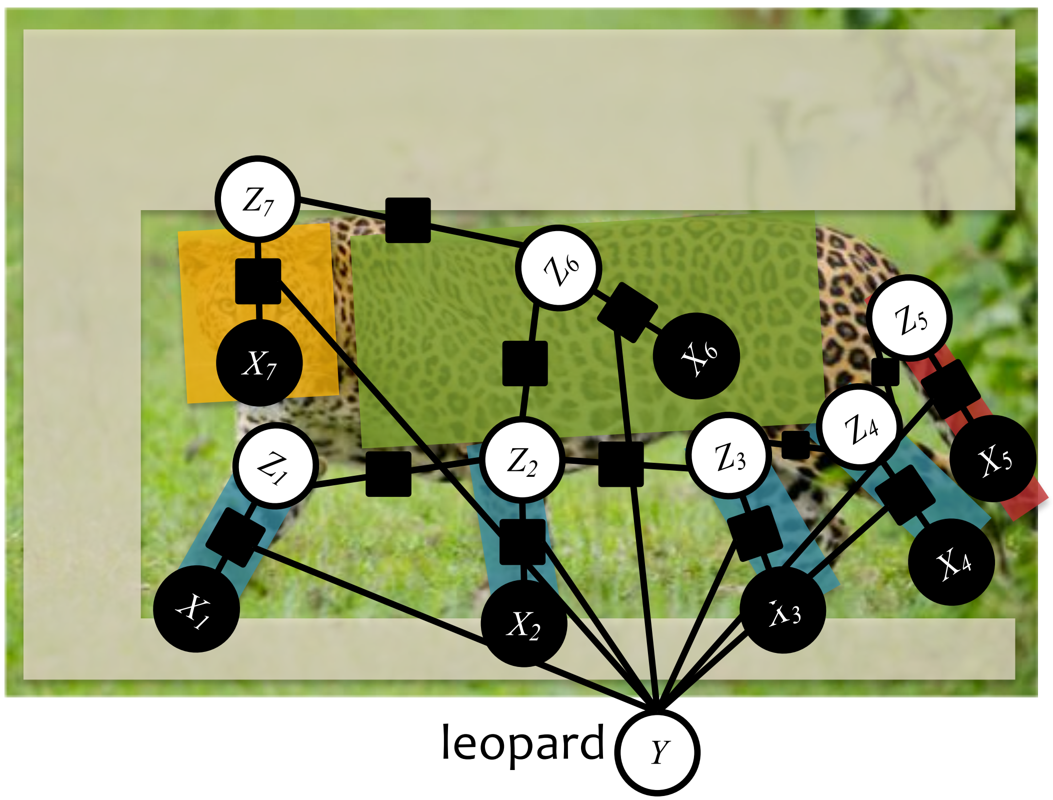
- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- z is not observed at train or test time



Case Study: Object Recognition

Data consists of images x and labels y.

- Preprocess data into "patches"
- Posit a latent labeling z describing the object's parts (e.g. head, leg, tail, torso, grass)
- Define graphical model with these latent variables in mind
- z is not observed at train or test time



Structured Prediction

Preview of challenges to come…

- Consider the task of finding the **most probable assignment** to the output

Classification

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

where $y \in \{+1, -1\}$

Structured Prediction

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

where $\mathbf{y} \in \mathcal{Y}$

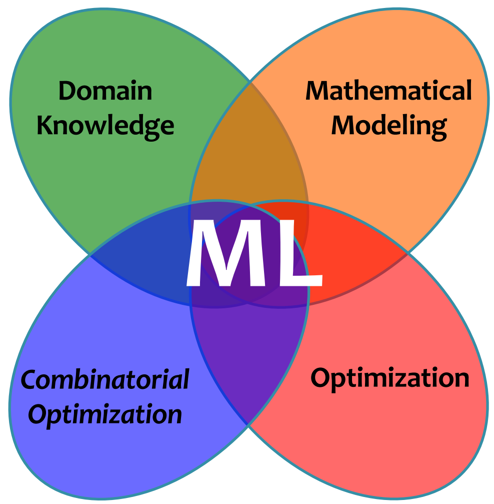
and $|\mathcal{Y}|$ is very large

Machine Learning

The **data** inspires the structures we want to predict

Our **model** defines a score for each structure

It also tells us what to optimize



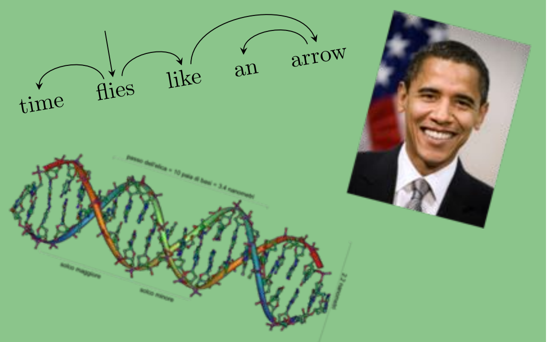
Learning tunes the parameters of the model

Inference finds {best structure, marginals, partition function} for a new observation

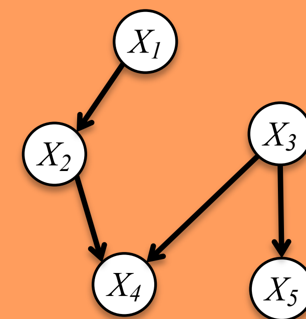
(**Inference** is usually called as a subroutine in learning)

Machine Learning

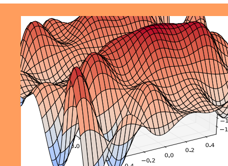
Data



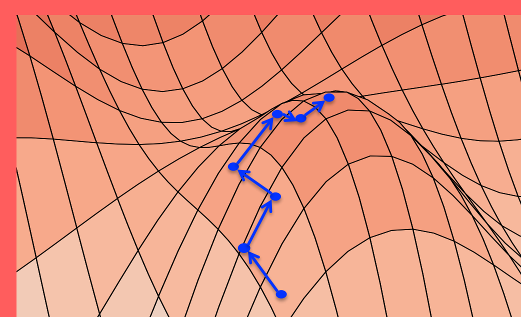
Model



Objective



Learning



Inference

(**Inference** is usually called as a subroutine in learning)

BACKGROUND

Background: Chain Rule of Probability

For random variables A and B:

$$P(A, B) = P(A|B)P(B)$$

For random variables X_1, X_2, X_3, X_4:

$$\begin{aligned} P(X_1, X_2, X_3, X_4) = &P(X_1|X_2, X_3, X_4) \\ &P(X_2|X_3, X_4) \\ &P(X_3|X_4) \\ &P(X_4) \end{aligned}$$

Background: Conditional Independence

Random variables A and B are conditionally independent given C if:

$$P(A, B|C) = P(A|C)P(B|C) \qquad (1)$$

or equivalently:

$$P(A|B, C) = P(A|C) \qquad (2)$$

We write this as:

$$A \perp\!\!\!\perp B|C$$

Later we will also write: $I\langle A, \{C\}, B\rangle$