10-301/601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Naïve Bayes

+

Generative vs. Discriminative

Matt Gormley
Lecture 17
Mar. 21, 2022

Reminders

- Homework 6: Learning Theory / Generative Models
  - Out: Fri, Mar. 18
  - Due: Fri, Mar. 25 at 11:59pm
  - IMPORTANT: only 2 grace/late days permitted
- Exam 2 (Thu, Mar 3rd)
- Exam 3 (Tue, May 3rd)

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:

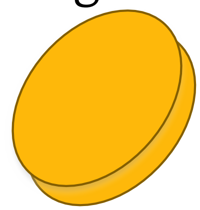
Training is lightning fast!

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

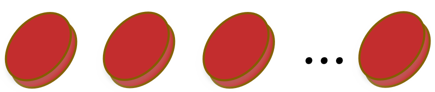
NAÏVE BAYES

Model 1: Bernoulli Naïve Bayes

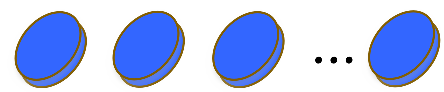
Flip weighted coin



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!

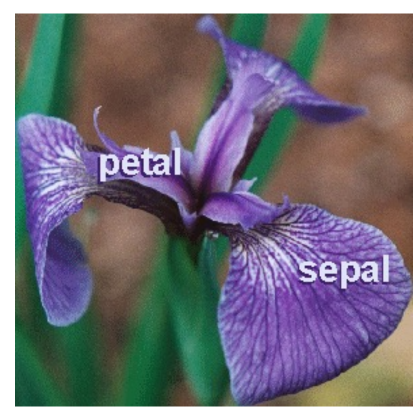
★ ELECTION 2016 ★ MORE ELECTION COVERAGE ▸

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



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: Learning from Data

Whiteboard

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

BERNOULLI NAÏVE BAYES

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:

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

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

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

$$\ldots$$

Maximum Likelihood Estimators:

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

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

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

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

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

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:

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

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

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

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

MLE for Naïve Bayes is a splendid learning algorithm for when you have say billions of training examples and hundreds of millions of features!

You only need one pass through the data to perform some counting.

MAP ESTIMATION FOR BERNOULLI NAÏVE BAYES

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 "`unicorn`" in a real news article.

In this case, what is the MLE of the following quantity?

p($x_{\texttt{unicorn}}$ | y=real) =

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

Now suppose we observe the word "`unicorn`" at test time. What is the posterior probability that the article was a real article?

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

Recipe for Closed-form MAP Estimation

1. Assume data was generated i.i.d. from some model (i.e. write the generative story)
   $\boldsymbol{\theta} \sim p(\boldsymbol{\theta})$ and then for all i: $x^{(i)} \sim p(x|\boldsymbol{\theta})$
2. Write log-likelihood
   $\ell_{MAP}(\boldsymbol{\theta}) = \log p(\boldsymbol{\theta}) + \log p(x^{(1)}|\boldsymbol{\theta}) + \ldots + \log p(x^{(N)}|\boldsymbol{\theta})$
3. Compute partial derivatives (i.e. gradient)
   $\partial\ell_{MAP}(\boldsymbol{\theta})/\partial\theta_1 = \ldots$
   $\partial\ell_{MAP}(\boldsymbol{\theta})/\partial\theta_2 = \ldots$
   …
   $\partial\ell_{MAP}(\boldsymbol{\theta})/\partial\theta_M = \ldots$
4. Set derivatives to zero and solve for $\boldsymbol{\theta}$
   $\partial\ell_{MAP}(\boldsymbol{\theta})/\partial\theta_m = 0$ for all $m \in \{1, \ldots, M\}$
   $\boldsymbol{\theta}^{MAP}$ = solution to system of M equations and M variables
5. Compute the second derivative and check that $\ell(\boldsymbol{\theta})$ is concave down at $\boldsymbol{\theta}^{MAP}$

Model 1: Bernoulli Naïve Bayes

MAP Estimation (Beta Prior)

1. Generative Story:

The parameters are drawn once for the entire dataset.

$\phi \sim \text{Beta}(\alpha', \beta')$

for $m \in \{1, \dots, M\}$**:**

  for $y \in \{0, 1\}$**:**

    $\theta_{m,y} \sim \text{Beta}(\alpha, \beta)$

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

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

  for $m \in \{1, \dots, M\}$**:**

    $x_m^{(i)} \sim \text{Bernoulli}(\theta_{y^{(i)},m})$

2. Likelihood:

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

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

Take derivatives, set to zero and solve…

$$\phi = \frac{(\alpha' - 1) + N_{y=1}}{(\alpha' - 1) + (\beta' - 1) + N}$$

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

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

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

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

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

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

$\dots$

Model 1: Bernoulli Naïve Bayes

MAP Estimation (Beta Prior)

1. Generative Story:

The parameters are drawn once for the entire dataset.

$\phi \sim$ Beta(α', β')

for $m \in \{1, \ldots, M\}$:

for $y \in \{0, 1\}$:

$\theta_{m,y} \sim$ Beta(α, β)

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

$y^{(i)} \sim$ Bernoulli(ϕ)

2. Likelihood:

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

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

Take derivatives, set to zero and solve…

$$\phi = \frac{(\alpha' - 1) + N_{y=1}}{(\alpha' - 1) + (\beta' - 1) + N}$$

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

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

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

A common choice for the class prior:

α' = 1 and β' = 1

Since Beta(1,1) = Uniform(0,1)

THE NAÏVE BAYES FRAMEWORK

Many NB Models

There are many Naïve Bayes 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, $y \sim$ 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

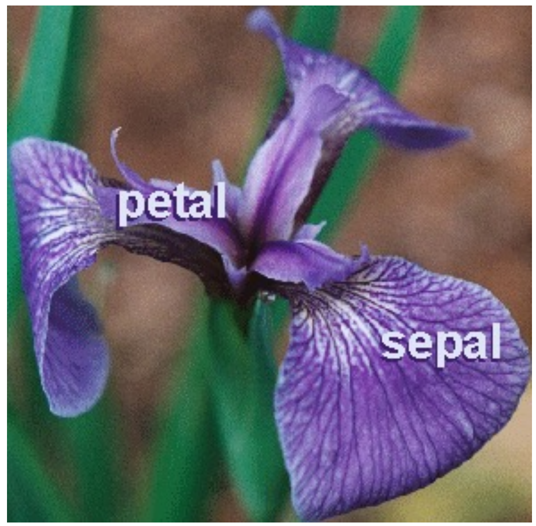
Classification:

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

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

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

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

Iris Data (2 classes)

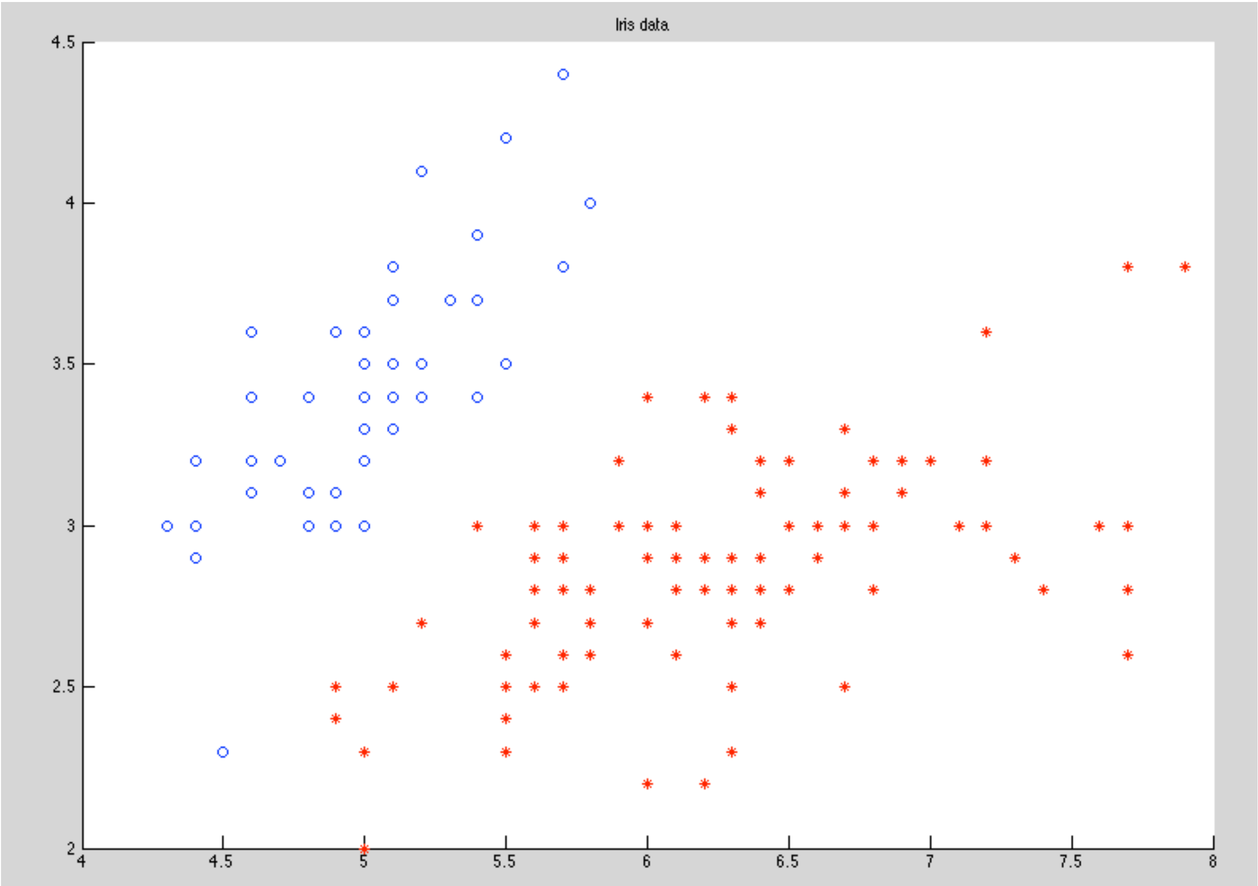


Figure from William Cohen

Iris Data (2 classes)

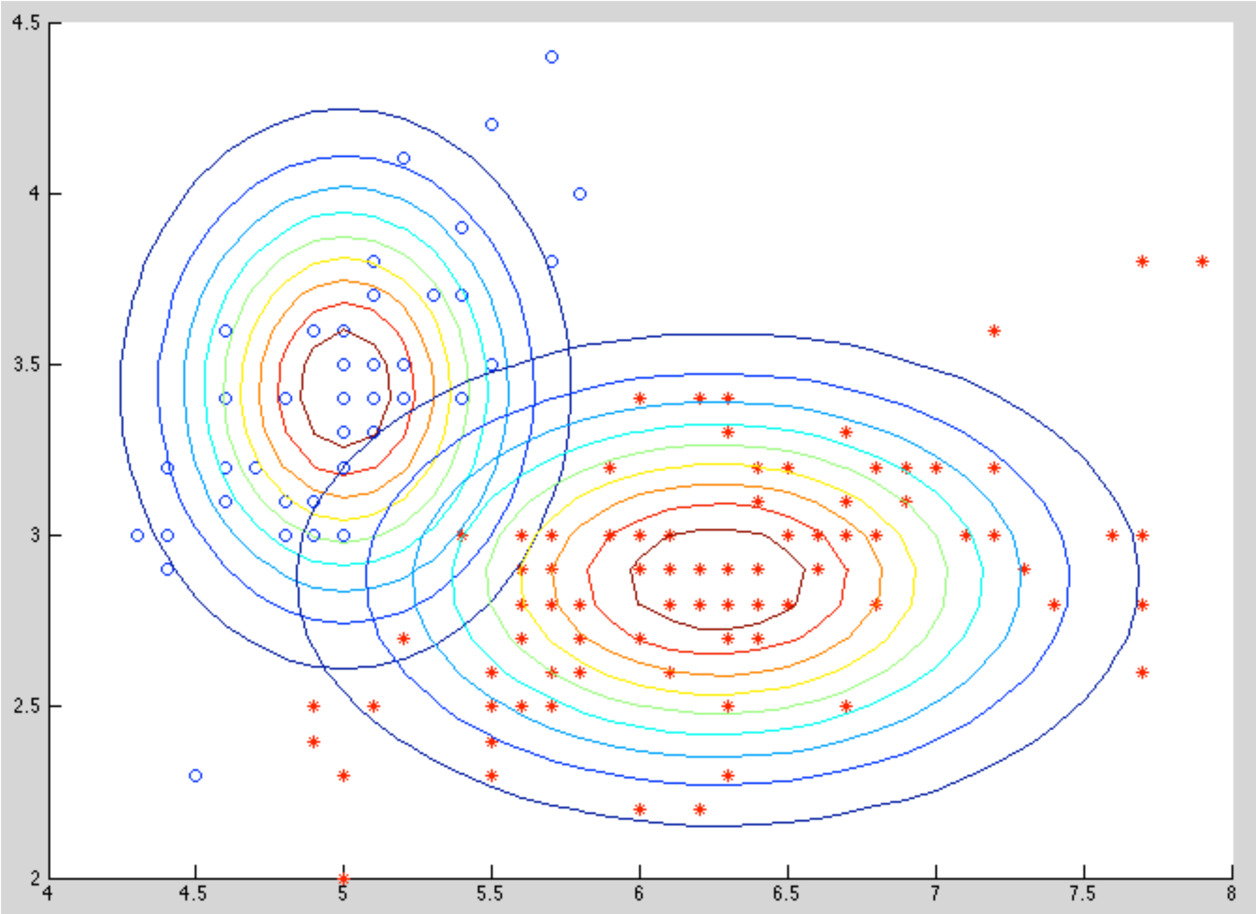
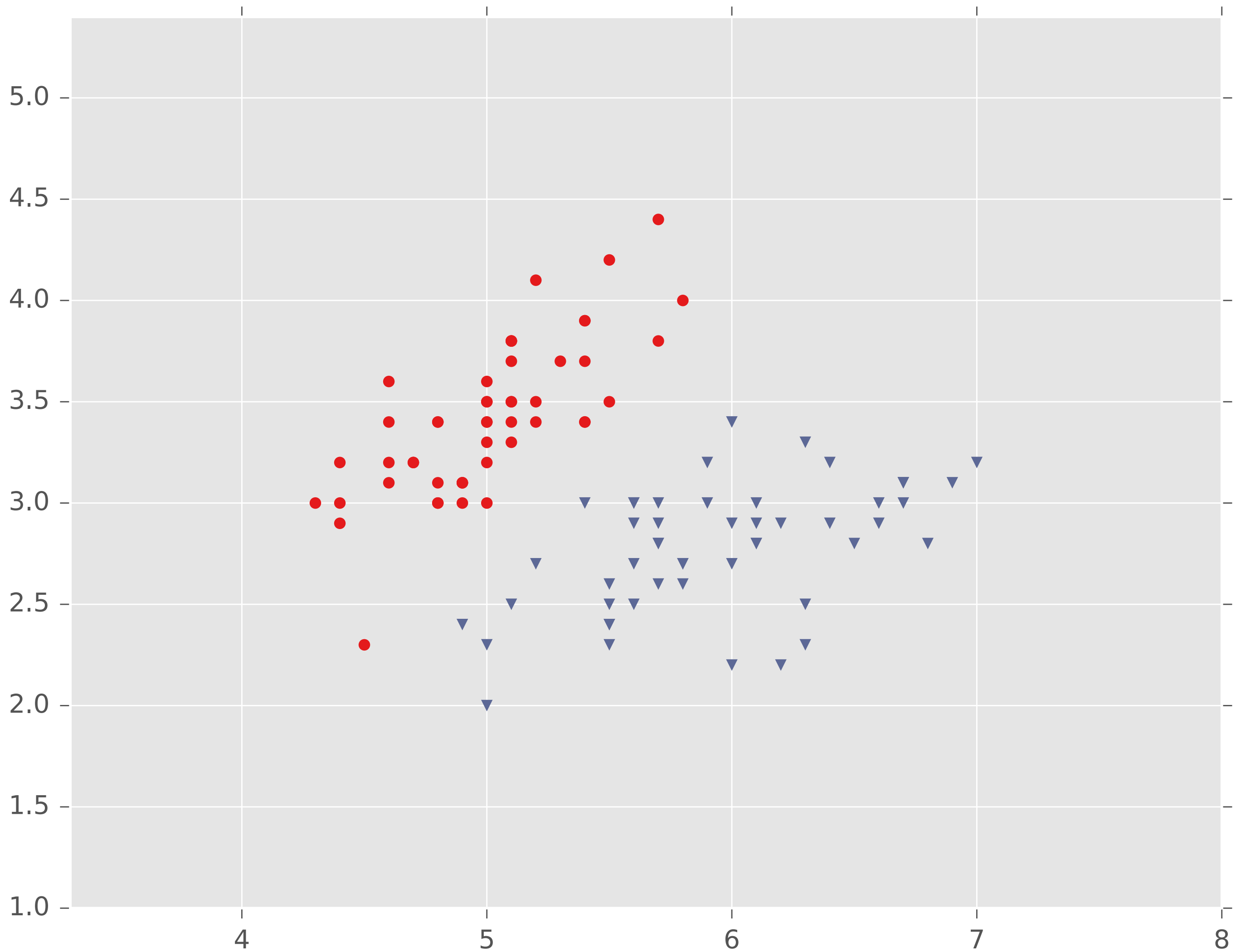


Figure from William Cohen

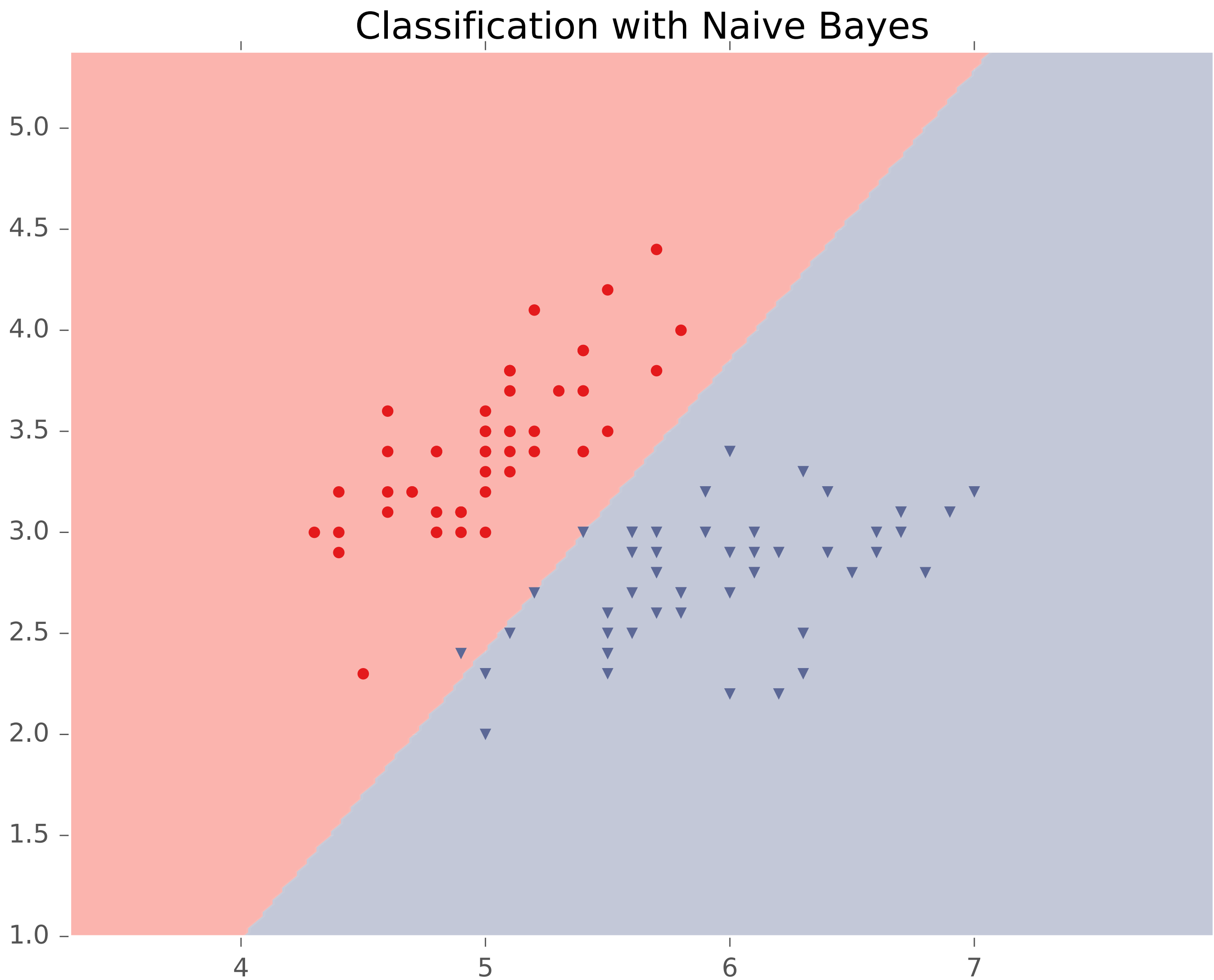
Iris Data (2 classes)

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



Iris Data (2 classes)

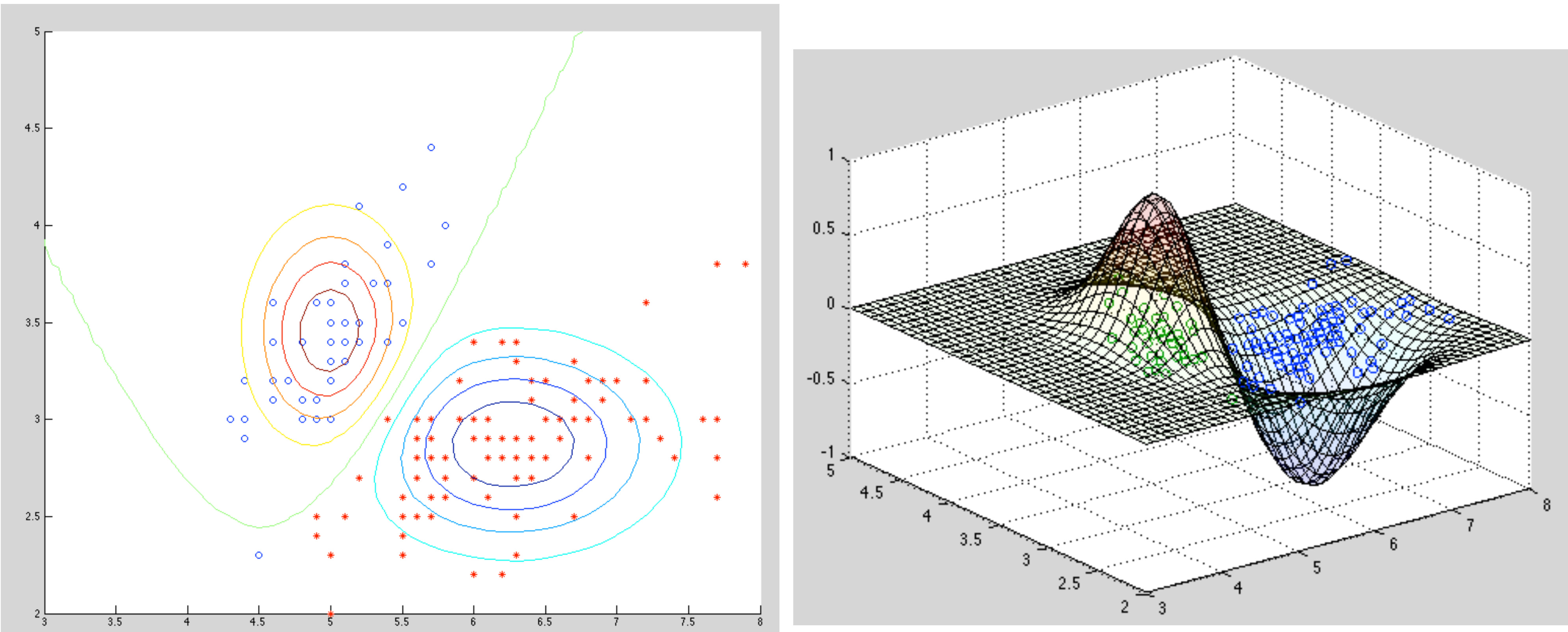
Naïve Bayes has a **linear** decision boundary if variance (sigma) is constant across classes



variance = 1

Iris Data (2 classes)

z-axis is the difference of the posterior probabilities: p(y=1 | **x**) – p(y=0 | **x**)

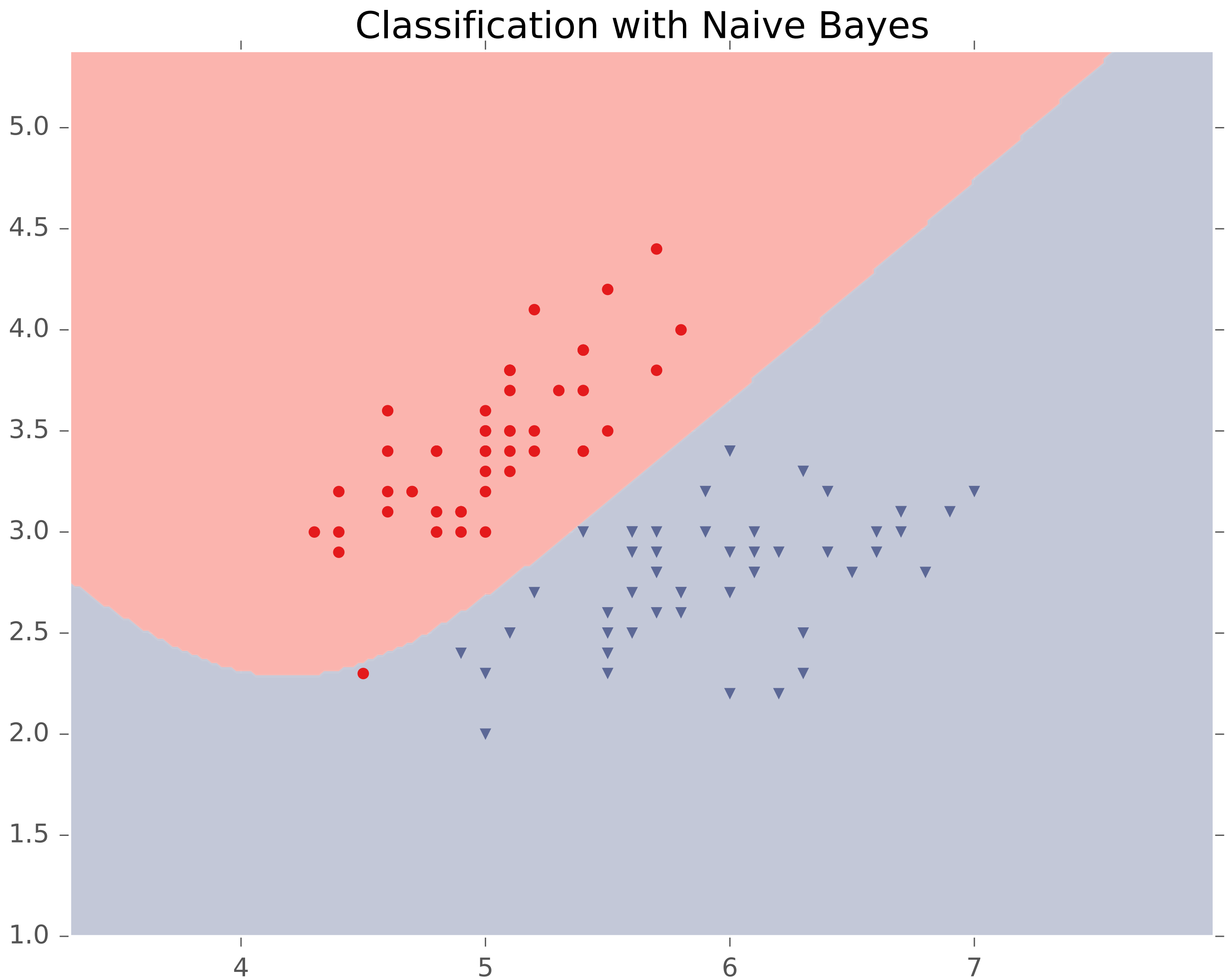


Figures from William Cohen

variance learned for each class

Iris Data (2 classes)

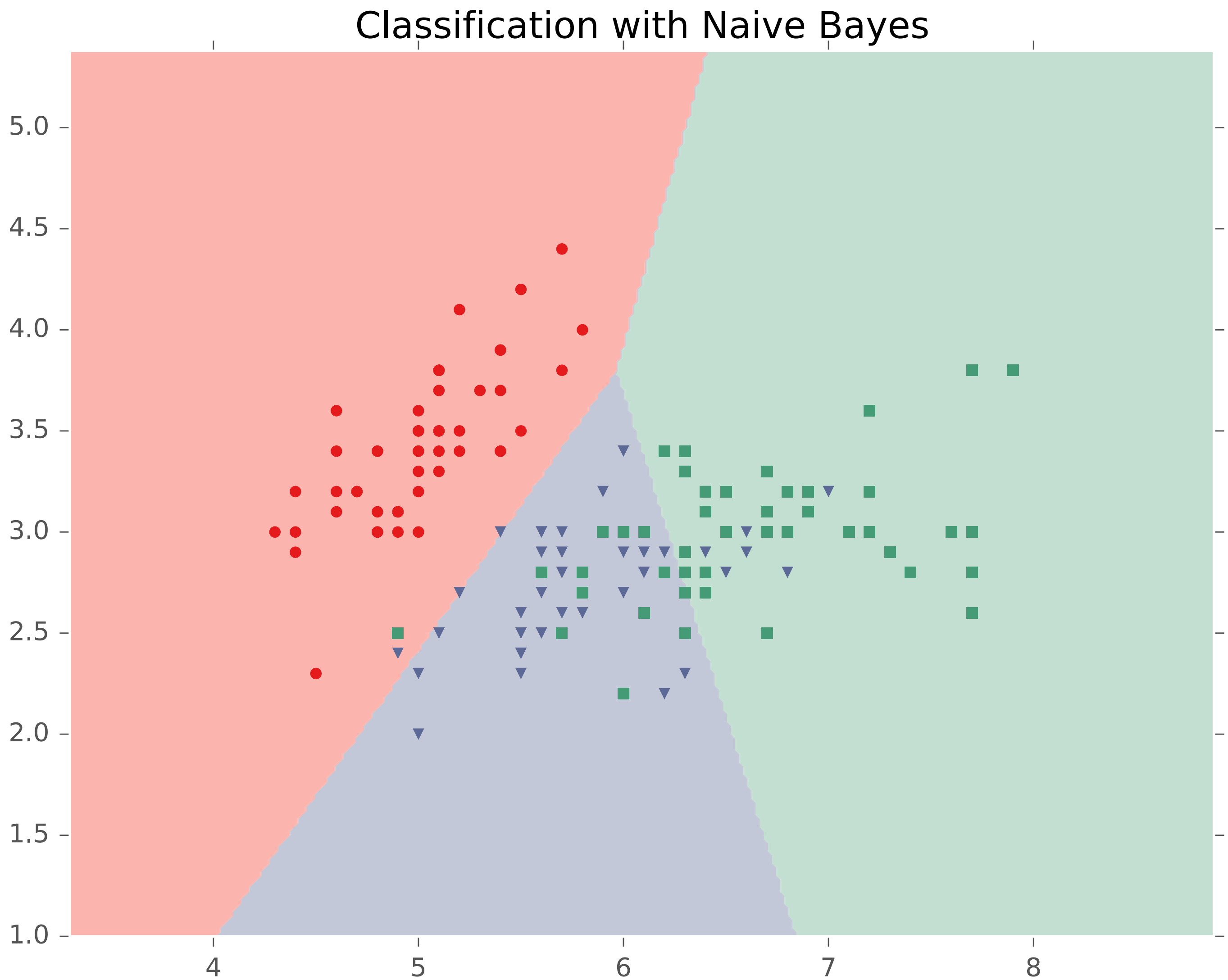
Naïve Bayes can have a **nonlinear** decision boundary if variance (sigma) can vary across classes



variance learned for each class

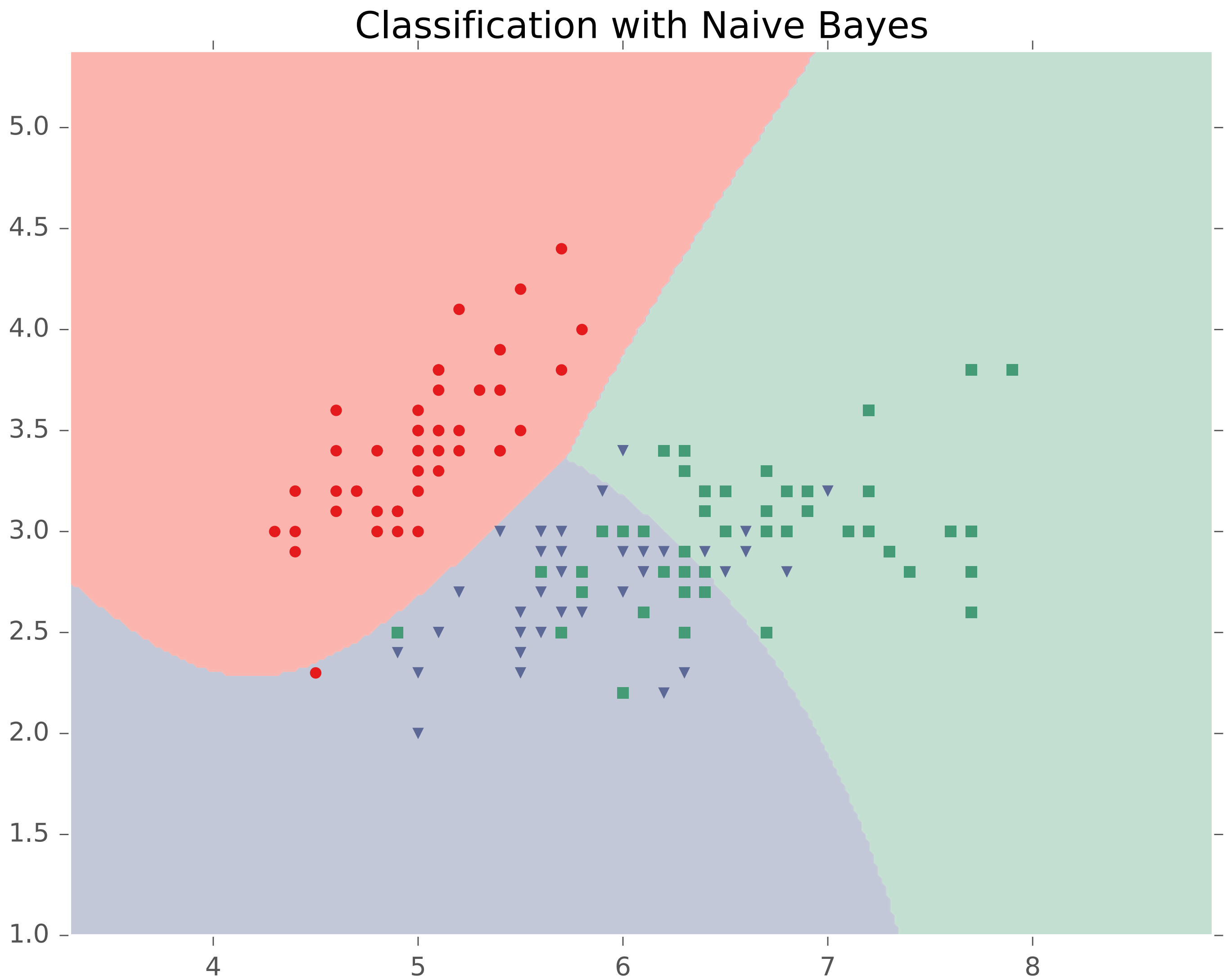
Iris Data (3 classes)

Iris Data (3 classes)



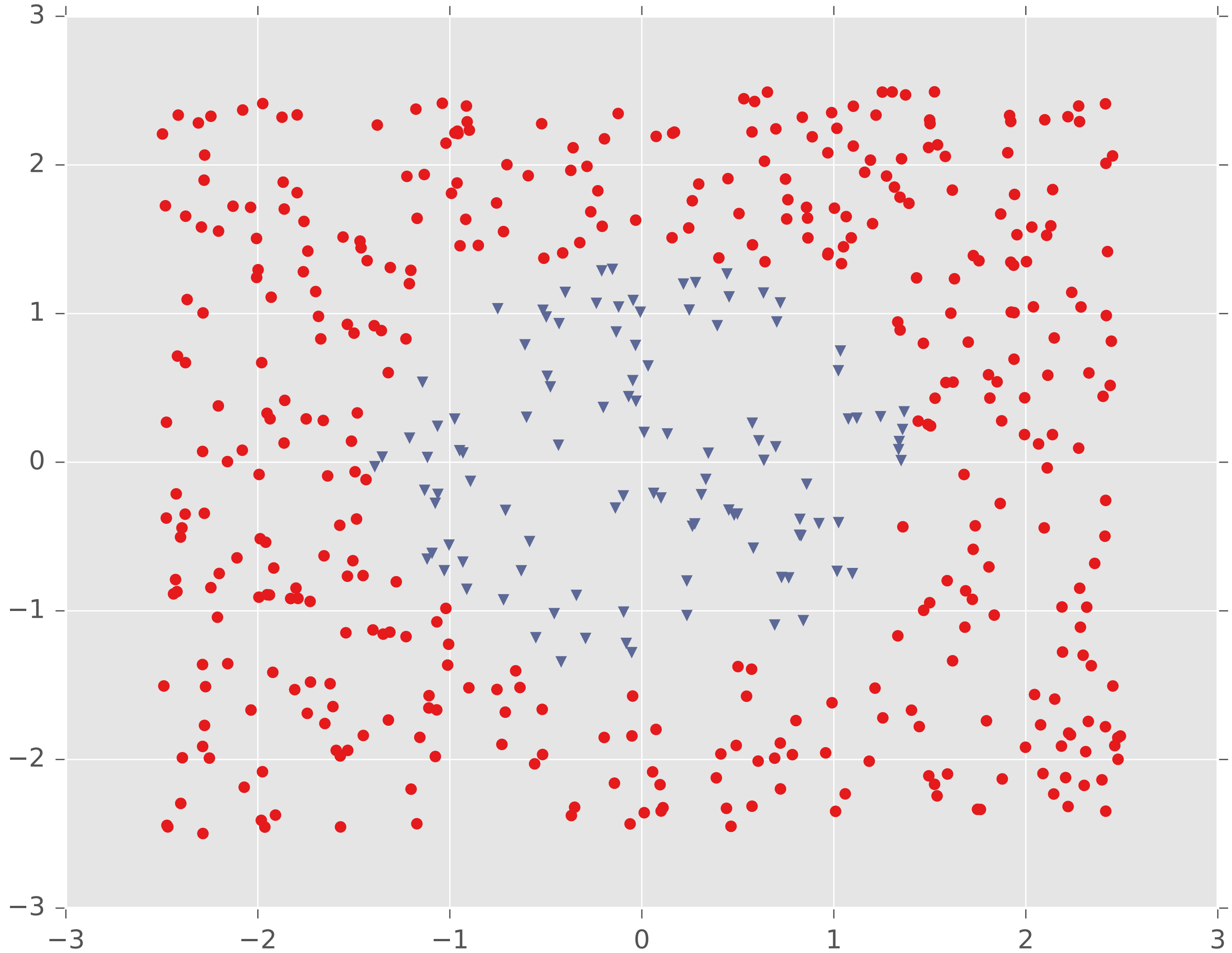
variance = 1

Iris Data (3 classes)

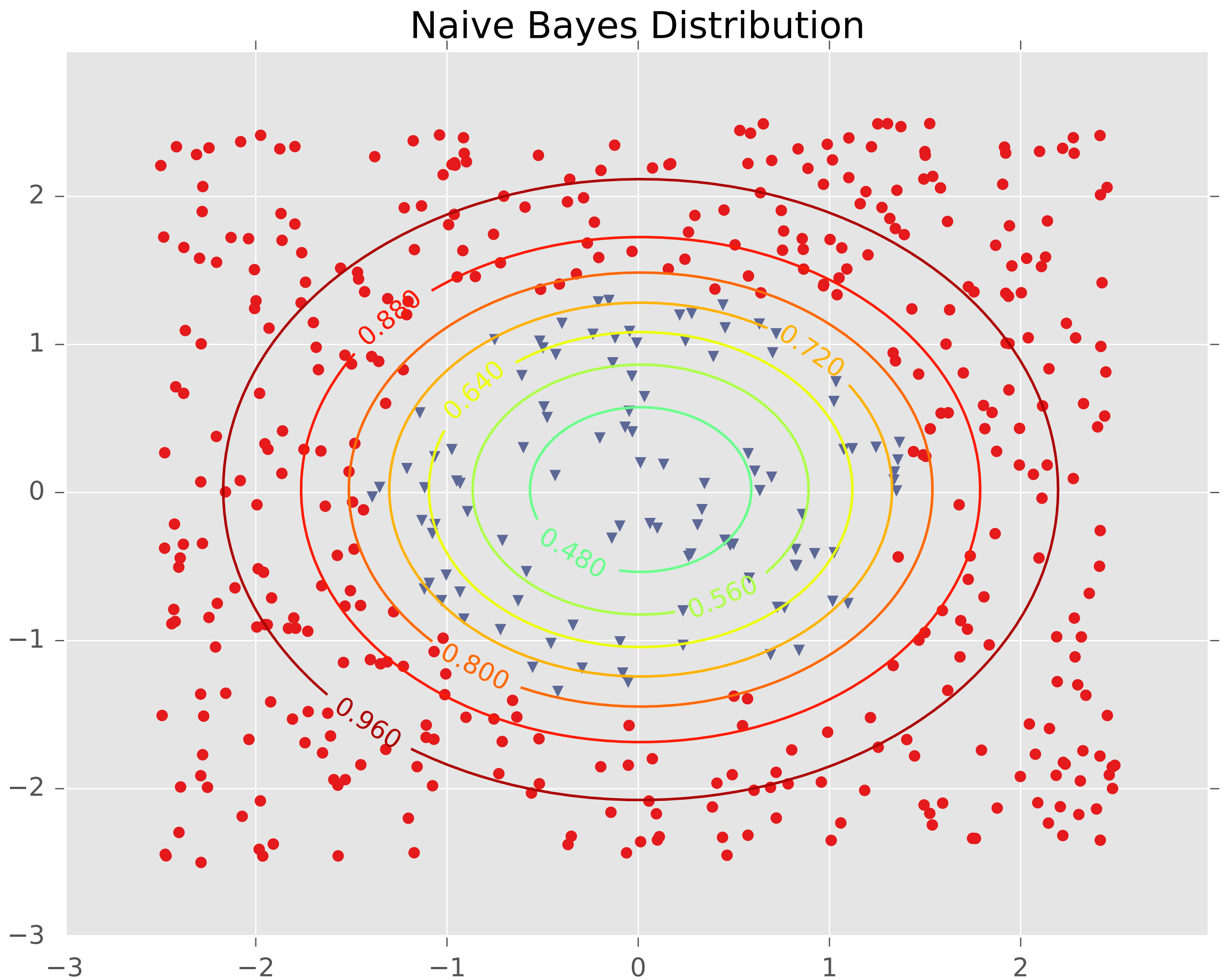


variance learned for each class

One Pocket



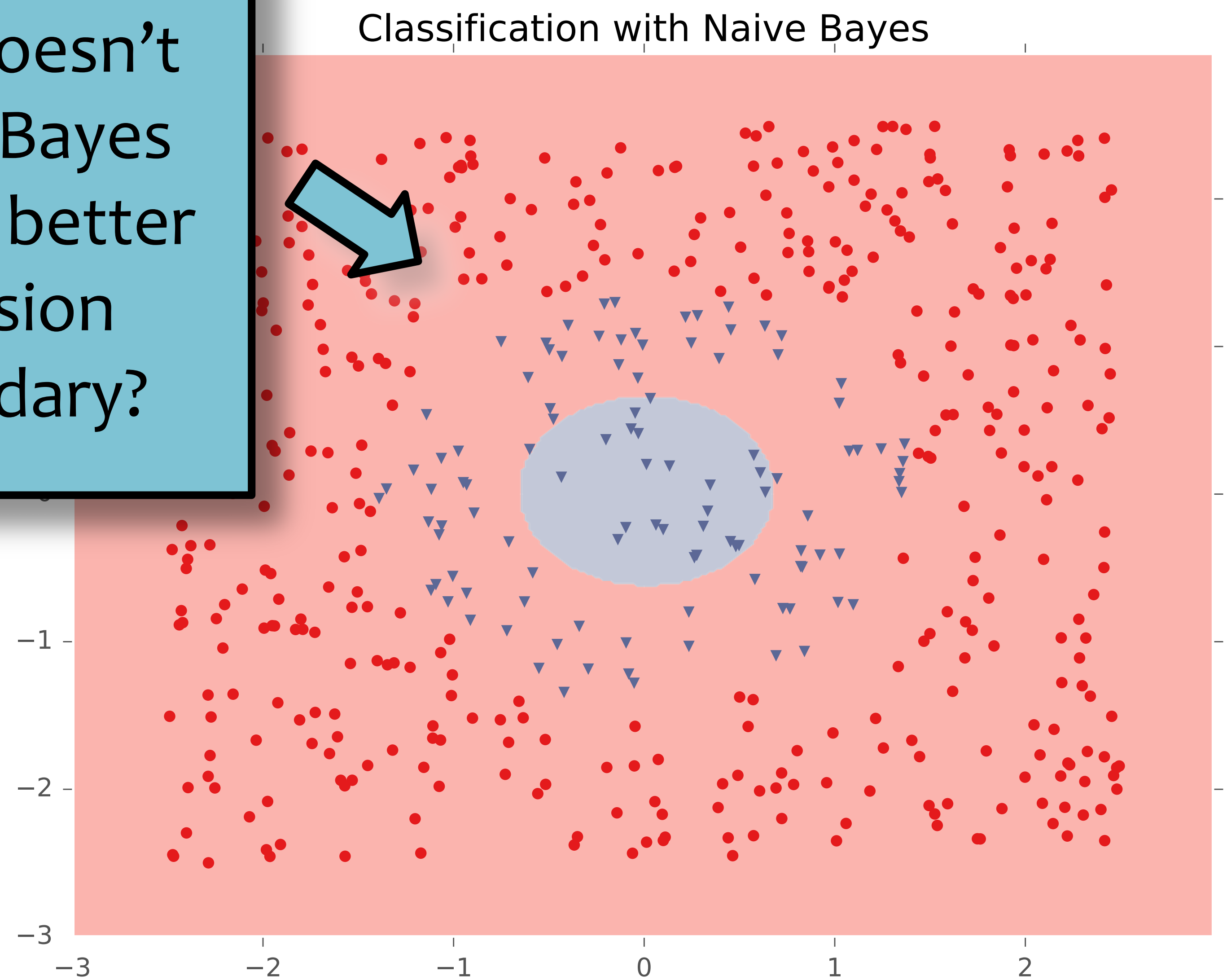
One Pocket



variance learned for each class

One Pocket

Why doesn't Naïve Bayes learn a better decision boundary?



variance learned for each class

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

Whiteboard

– MAP Estimation and Regularization

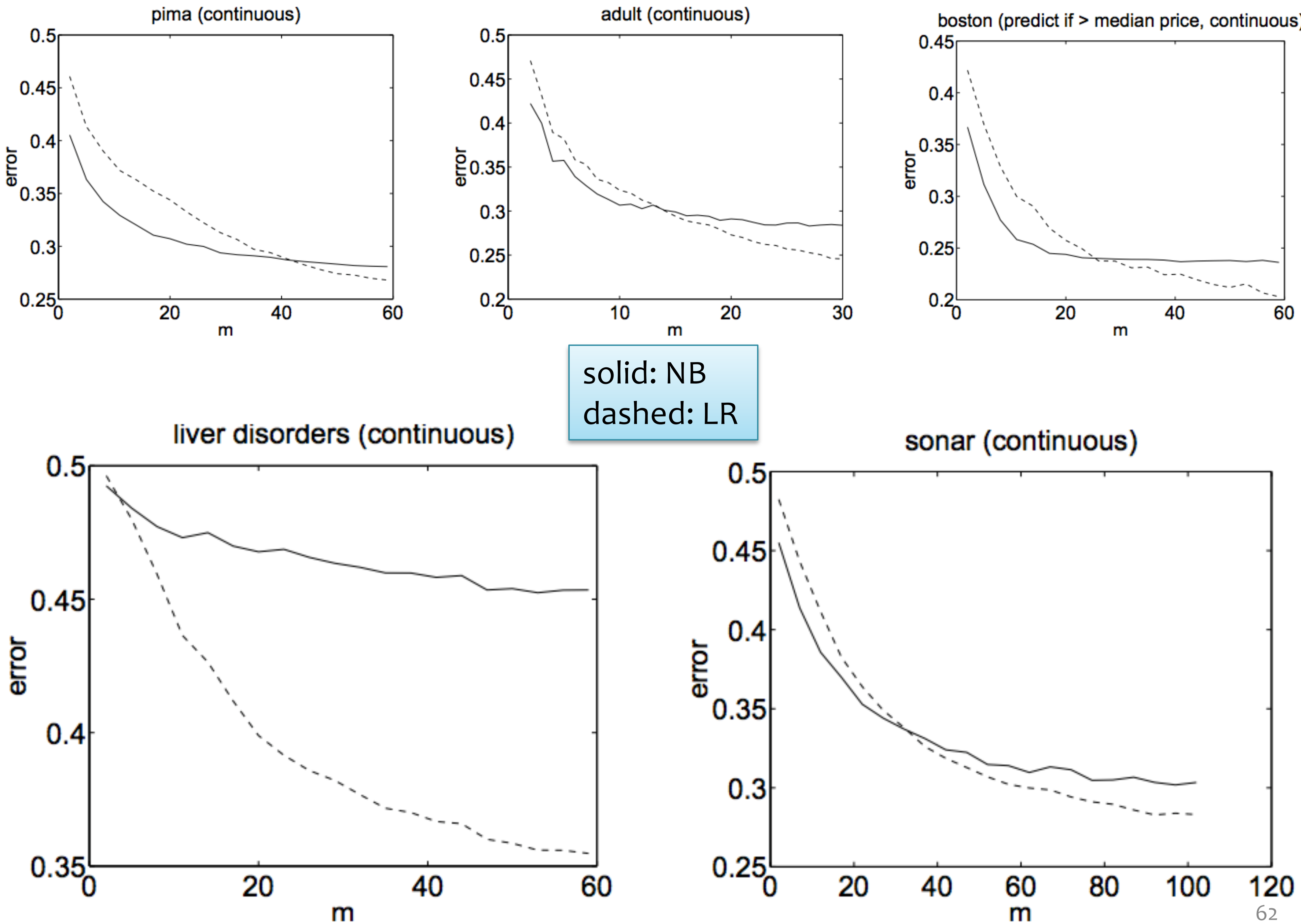
Generative vs. Discriminative

Finite Sample Analysis (Ng & Jordan, 2002)

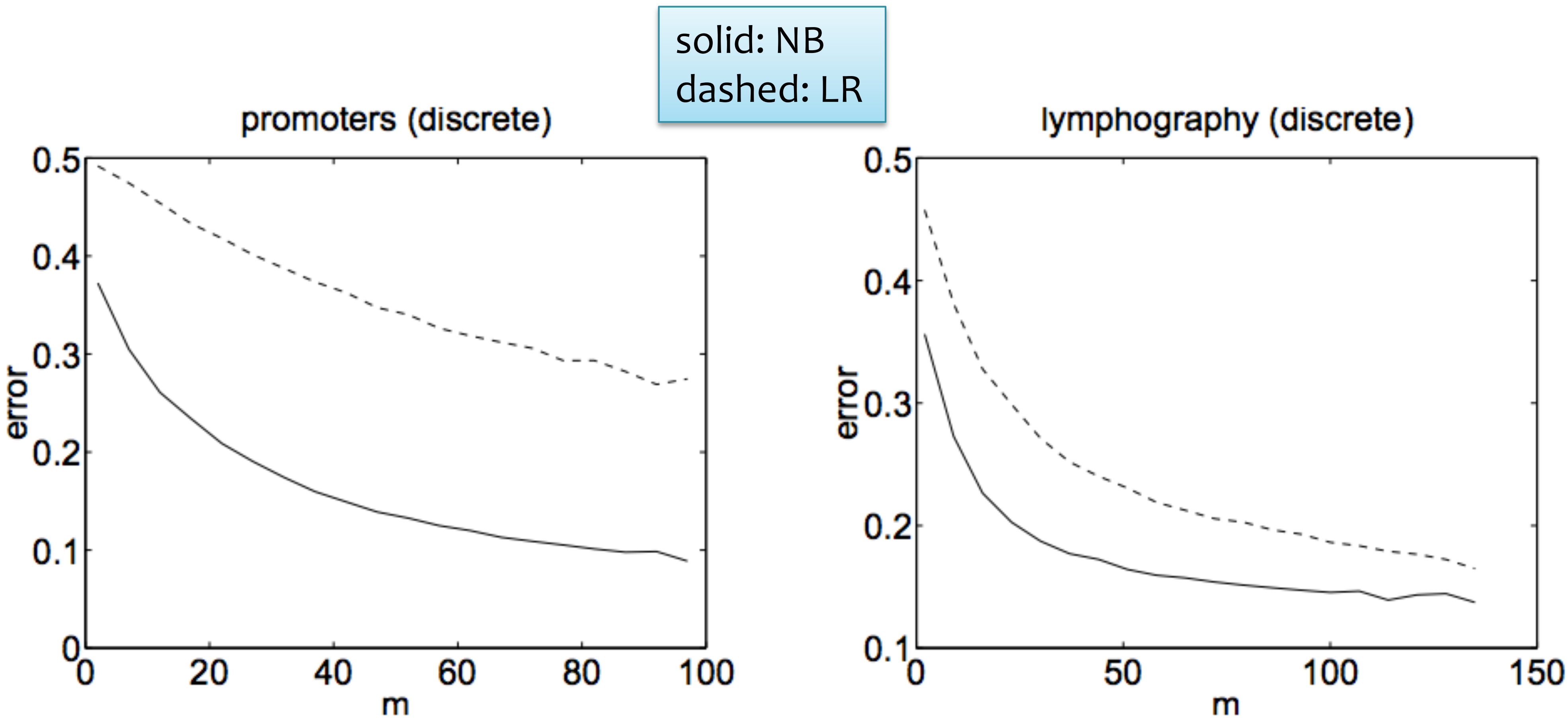
[Assume that we are learning from a finite training dataset]

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 asymptotic error, and does better than Naïve Bayes



Slide courtesy of William Cohen



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

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.

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)

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 Regression

Question:

You just started working at a new company that manufactures comically large pennies. Your manager asks you to build a binary classifier that takes an image of a penny (on the factory assembly line) and predicts whether or not it has a defect.

What follow-up questions would you pose to your manager in order to decide between using a Naïve Bayes classifier and a Logistic Regression classifier?

Answer:

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

THE BIG PICTURE

ML Big Picture

Learning Paradigms:

What data is available and when? What form of prediction?

- supervised learning
- unsupervised learning
- semi-supervised learning
- reinforcement learning
- active learning
- imitation learning
- domain adaptation
- online learning
- density estimation
- recommender systems
- feature learning
- manifold learning
- dimensionality reduction
- ensemble learning
- distant supervision
- hyperparameter optimization

Problem Formulation:

What is the structure of our output prediction?

boolean	Binary Classification
categorical	Multiclass Classification
ordinal	Ordinal Classification
real	Regression
ordering	Ranking
multiple discrete	Structured Prediction
multiple continuous	(e.g. dynamical systems)
both discrete & cont.	(e.g. mixed graphical models)

Application Areas

Key challenges?

NLP, Speech, Computer Vision, Robotics, Medicine, Search

Theoretical Foundations:

What principles guide learning?

- ❑ probabilistic
- ❑ information theoretic
- ❑ evolutionary search
- ❑ ML as optimization

Facets of Building ML Systems:

How to build systems that are robust, efficient, adaptive, effective?

1. *Data prep*
2. *Model selection*
3. *Training (optimization / search)*
4. *Hyperparameter tuning on validation data*
5. *(Blind) Assessment on test data*

Big Ideas in ML:

Which are the ideas driving development of the field?

- *inductive bias*
- *generalization / overfitting*
- *bias-variance decomposition*
- *generative vs. discriminative*
- *deep nets, graphical models*
- *PAC learning*
- *distant rewards*

ML Big Picture

Whiteboard

- Decision Rules / Models
- Objective Functions
- Regularization
- Optimization