CO902
Probabilistic and statistical inference

Lecture 4

Tom Nichols
Department of Statistics &
Warwick Manufacturing Group

t.e.nichols@warwick.ac.uk

Outline for Today

- **Review / Lab Catch Up**
  - Beta-Bernoulli MAP estimator, properties
  - Decision-Theory approach to prediction
- **Supervised Learning (con't)**
  - Class Conditional Models
  - Cross validation
  - Decision boundary

Bayesian Inference Review

- No matter how complicated the problem, Bayesian inference reduces to

$$posterior \propto likelihood \times prior$$

  - Prior – beliefs before seeing data
  - Likelihood – same as frequentist inference
  - Posterior – beliefs after seeing data

- MAP – Maximum A Posteriori estimate
  - Parameter value that maximizes posterior

- Conjugate prior for a likelihood
  - When posterior is in same parametric family as prior

Bernoulli Inference with Beta Prior

- Beta is conjugate for Bernoulli/binomial

$$
\begin{aligned}
X_i|\theta,\alpha,\beta &\sim \mathrm{Ber}(\theta),\ iid \\
\theta|\alpha,\beta &\sim \mathrm{Beta}(\alpha,\beta) \\
\theta|\{X_i\},\alpha,\beta &\sim \mathrm{Beta}(n_1+\alpha, n-n_1+\beta)
\end{aligned}
$$

$$n_1 = \sum_{i=1}^{n} X_i$$

  - α & β are fixed; i.e. are tuning parameters
  - Although, could have "hyperpriors", priors on α & β(!)

- MAP – Maximum A Posteriori estimate

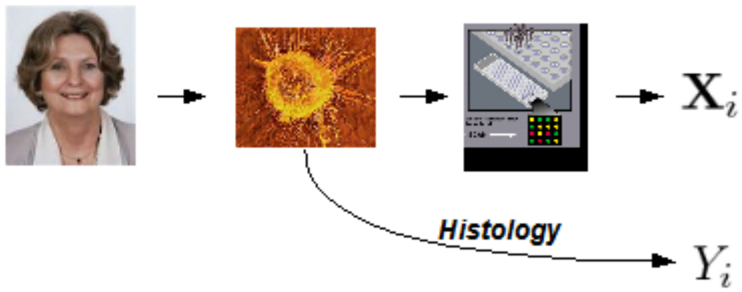
$$\hat{\theta}_{\mathrm{M}AP} = \frac{n_1+\alpha-1}{n+\alpha+\beta-2}$$

  - For (α, β)=(1,1) MAP is MLE; (α, β)=(2,2) is our "modified MLE" from lab

Lab Exercise Observations MLE vs MAP

- When θ is likely under prior
  - Good MSE for small/moderate data
- When θ is far from prior mean
  - Poor MSE for small/moderate data
- With strong prior, dramatic reduction in variance
- When lots of data
  - Little impact of prior

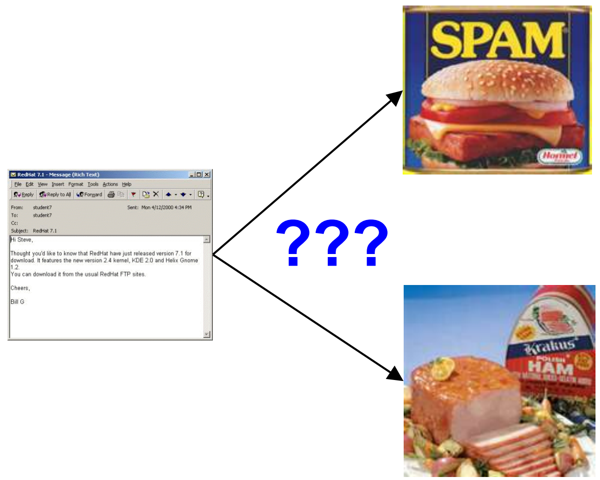
Classification

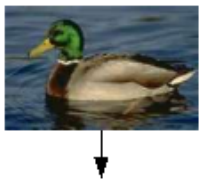


All these problems share a common structure

$$\{\mathbf{X}_i, Y_i\},\ i = 1..n$$

$$\mathbf{X}_i \in \mathbb{R}^d,\ Y_i \in \{1, 2, \ldots k\}$$





"duck"

"tiger"

*Input **X***

Output Y

Classification/Prediction & Decision Theory

- Have done classification in lab, informally justifying the method each time
  - Markov Chain – Choose most likely Xi given Xi-1
  - Spam – Compute probability spam given label

- What are the general principals at work here?
  - Decision Theory… *(boardwork, supported notes)*

Classification/Prediction & Decision Theory: Redux

- For discrete outcome (“class”) prediction
  - Decision Theory gives general principals
  - Leads to vital role class conditional distributions
- *But*
  - Crux of the problem remains estimating the class conditional distributions
  - Many issues rmain

Class-conditional generative model

- Data:

$$\begin{aligned} \{\mathbf{X}_i, Y_i\}, & \quad i = 1..n \\ \mathbf{X}_i & \in \mathbb{R}^d \\ Y_i & \in \{0,1\} \end{aligned}$$

- Two distinct classes
- Use two distributions, one for each class...

$$\begin{aligned} p(\mathbf{X} \mid Y = k) &= p_k(\mathbf{X}) \\ &= p(\mathbf{X} \mid \theta_k) \end{aligned}$$

(same family, different parameters)

- These are called **class-conditional distributions**

Class posterior

- We want to classify a data-vector, i.e. determine it's class
- Using Bayes' rule:

$$P(Y = 1 \mid \mathbf{X}) \quad = \quad \frac{p(\mathbf{X} \mid Y = 1)P(Y = 1)}{p(\mathbf{X} \mid Y = 1)P(Y = 1) + p(\mathbf{X} \mid Y = 0)P(Y = 0)}$$

- If we
  - Assume some prior on class membership and
  - Can estimate the two class-conditional pdfs/pmfs

then we can classify data-points

Inference

- Intuitively
  - We have two groups, labelled by Y=0, Y=1
  - We want the parameters for each group
  - We can just estimate the parameters for all datapoints having Y = k
- This can be described more formally in likelihood terms
- We'll start with a discrete classifier

Discrete data

- Often, the data vectors themselves are **discrete**
- Binary case:

$$\begin{aligned} \{\mathbf{X}_i, Y_i\}, \qquad & i = 1..n \\ \mathbf{X}_i \in & \{0,1\}^d \\ Y_i \in & \{0,1\} \end{aligned}$$

- **Examples:**
  - **spam from presence/absence of *d* words**
  - **Cancer status from presence/absence of *d* genes/proteins**
  - **Drug response from presence/absence of *d* genes/proteins**

Model

- Let's assume **binary inputs** and **two output classes**
- A general class-conditional distribution (for Y=1):

2^d possible configurations

X_1	X_2	. . . . . .	X_d	$P(\mathbf{X} \mid Y=1)$
0	*0*	. . . . . .	*0*	θ_1
1	*1*	. . . . . .	*1*	θ_{2^d}

- **Q: how many parameters does the complete model have?**

“Naïve bayes” assumption

- A common assumption is to allow the class-conditional distribution to factorize over variables:

$$P(X_{i1} \dots X_{id} \mid Y_i) \;=\; \prod_{j=1}^{d} P(X_{ij} \mid Y_i)$$

- That is, assume inputs are independent (given output class)
- Known as the “Naive Bayes” assumption (unfortunate misnomer: actually has nothing intrinsically to do with Bayes)

Bernoulli model

- We want to characterize the chance that input *j* is "on", given *Y*=1, that is given class #1
- Assuming the *N* observations are i.i.d., the natural model is Bernoulli:

$$
\begin{aligned}
P(X_{ij} = 1 \mid Y_i = 1) &= \theta_{j1} \\
P(X_{ij} = 1 \mid Y_i = 0) &= \theta_{j0}
\end{aligned}
$$

Naïve bayes class conditional

- NB assumption with a Bernoulli model gives the following class conditional distribution:

$$P(\mathbf{X}_i \mid Y_i = k) \;=\; \prod_{j=1}^{d} \theta_{jk}^{X_{ij}} (1 - \theta_{jk})^{(1 - X_{ij})}$$

- In summary:
  - What we're talking about is simply having different thetas depending on whether Y is 1 or 0
  - Doing this for each input
  - And then, assuming independence between inputs (given output), multiplying them together to get P(X | Y)

MLEs

- What are the parameters?

θ_{j1} Probability that X_j = 1 when Y = 1

θ_{j0} Probability that X_j = 1 when Y = 0

- What are the MLE's?

MLEs

- What are the parameters?

θ_{j1} Probability that X_j = 1 when Y = 1

θ_{j0} Probability that X_j = 1 when Y = 0

- What are the MLE's?

$$\hat{\theta}_{j1} = \frac{n_{j1}}{n_1} \qquad n_{jk} = \sum_{i\,:\,Y_i=k} X_{ij}$$

$$\hat{\theta}_{j0} = \frac{n_{j0}}{n_0} \qquad n_k = \sum_{i\,:\,Y_i=k} 1$$

Example: Handwritten Digit Classification

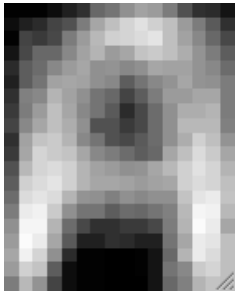
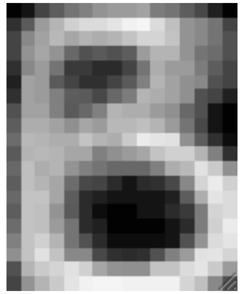
- Classify A vs B in handwritten data?
  - 16×20 pixel images
  - X: d = 16×20 = 320 variables
    - Pixels not independent, but we assume independence as part of “Naïve Bayes”
  - Y: K = 2 (for now), just “A” (Y=0) and “B” (Y=1)
    - n = 78 (39 per class) not much data given d = 320 !

Example: Handwritten Digit Classification

- Estimated Class Conditional Distribution, for k=0 ("A")

$$\log \widehat{P}(\mathbf{X}|Y_i = 0) = \sum_{j=1}^{d} X_j \log(\hat{\theta}_{j0}) + (1 - X_j) \log(1 - \hat{\theta}_{j0})$$

- Ditto for k=1 ("B")

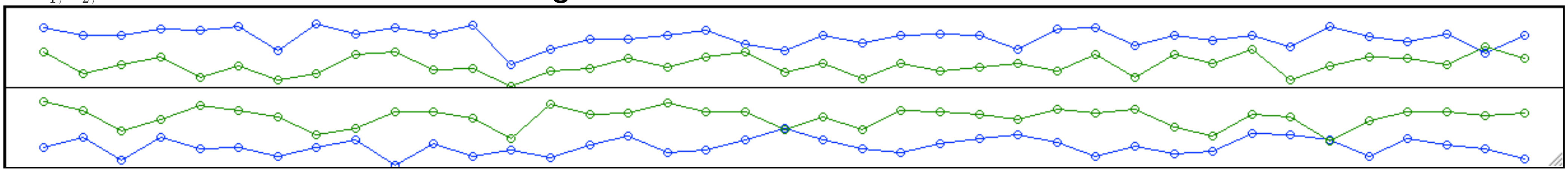
$\hat{\theta}_{j0} =$  $\hat{\theta}_{j1} =$ 

$\mathbf{x}_1, \mathbf{x}_2, \ldots$

$\mathbf{x}_{40}, \mathbf{x}_{41}, \ldots$

Log Class Conditional Distribution

$\mathbf{x}_1, \mathbf{x}_2, \ldots$

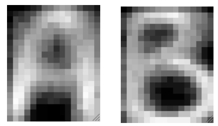


$\mathbf{x}_{40}, \mathbf{x}_{41}, \ldots$

$\log \widehat{P}(\mathbf{X}_i|Y_i = 0 \text{ “A”})$ $\log \widehat{P}(\mathbf{X}_i|Y_i = 1 \text{ “B”})$

Example: Handwritten Digit Classification

- Note nearly every sample correctly classified
  - For A's… $\log \widehat{P}(\mathbf{X}_i|Y_i = 0) > \log \widehat{P}(\mathbf{X}_i|Y_i = 1)$
  - For B's… $\log \widehat{P}(\mathbf{X}_i|Y_i = 0) < \log \widehat{P}(\mathbf{X}_i|Y_i = 1)$
  - However, note the three failures

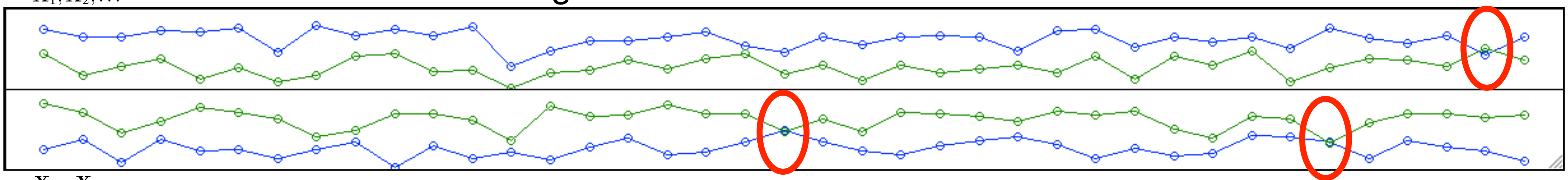


$\mathbf{X}_1, \mathbf{X}_2, \ldots$

$\mathbf{X}_{40}, \mathbf{X}_{41}, \ldots$

Log Class Conditional Distribution

$\mathbf{X}_1, \mathbf{X}_2, \ldots$



$\mathbf{X}_{40}, \mathbf{X}_{41}, \ldots$

$$\log \widehat{P}(\mathbf{X}_i|Y_i = 0 \text{ ``A''}) \qquad \log \widehat{P}(\mathbf{X}_i|Y_i = 1 \text{ ``B''})$$

How good are predictions?

- Once we've chosen a model, estimated required parameters etc., we're ready to classify any given input **X,** i.e. assign it to a class
- But what would the error rate be in such assignment? Let's call our overall classification rule g (i.e. g(X_i) = 0,1, for two classes)
- **In-sample or training error rate**: proportion of training data {X_i} incorrectly assigned under *g*
- **True error rate/risk/generalisation error**: Prob(g(X) \neq Y), i.e. proportion of all possible data incorrectly assigned under *g*
- True error is the real test: does it predict unseen data?

Train and test paradigm

- Idea: since we're interested in **predictive power** on unseen data, why not "train" on a *subset* of the data and "test" on the remainder?
- This would give us some indication of how well we'd be likely to do on new data...
- That is, we want to estimate **risk**

Cross-validation

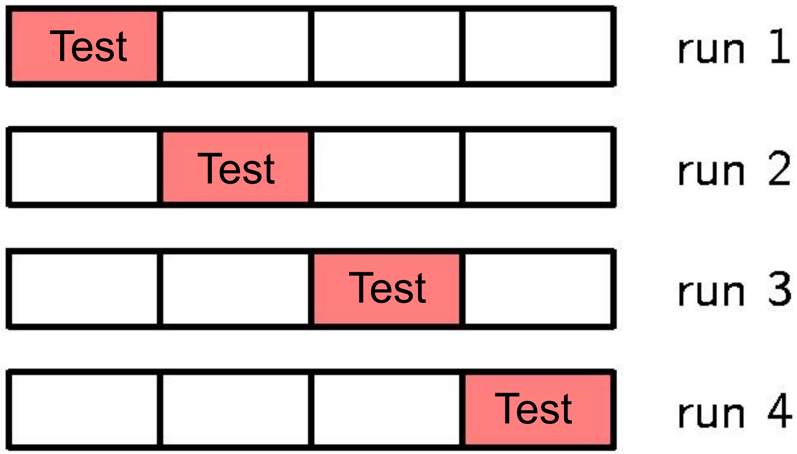
- But what if the dataset is small?
- Training on a subset of a small dataset may well do badly, but does this tell us how things would go in practice, using all of the data for training?

- Idea: use all but one datapoint to train, and test on the one left out, iterating until every datapoint has been used in this way
- This is called (leave-one-out) **cross-validation** (or "LOOCV")

- LOOCV on handwriting sample… $\log \widehat{P}(\mathbf{X}_i | Y_i = 0)$ $\log \widehat{P}(\mathbf{X}_i | Y_i = 1)$
  - Recompute class conditionals 78 times… holding out one sample each time
  - LOOCV gave same result… 3 out of 78 accurately classified

Cross-Validation

- LOOCV great, but computationally expensive
- N-fold cross-validation
  - Split data in to N-folds
  - Hold out 1/N-th as test
  - Use (N-1)/N of data to train
  - Measure accuracy on held-out sample



- Validation in general…
  - It is a simple but immensely useful way to check the behaviour of a model in supervised learning
  - The nice thing about supervised learning is that you have some "correct" answers
    - Train and test and cross validation are about using those data to assess how well your fitted model will generalize to unseen data
    - These can be immensely powerful and can be performed even for complicated models which are not amenable to formal analysis

Prediction with Continuous Response

(1) **Gaussian generative model** and **class-conditional distributions**
(2) **Decision boundary**
(3) **Variable selection**, **Fisher ratio**

Generative model

- Question: given vector-valued **continuous** input data, with each datapoint belonging to one of two classes, can we learn a probability model to automatically classify such observations?

- Data:

$$\{\mathbf{X}_i, Y_i\}, \qquad i = 1..n$$

$$\boxed{\mathbf{X}_i \in \mathbb{R}^d}$$

$$Y_i \in \{0, 1\}$$

- Want to make a generative model for each class of Y_i

Class-conditional generative model

- Data:

$$\begin{aligned} \{\mathbf{X}_i, Y_i\}, & \quad i = 1..n \\ \mathbf{X}_i &\in \mathbb{R}^d \\ Y_i &\in \{0, 1\} \end{aligned}$$

- What kind of model do we want?
- There are two distinct classes, so we certainly don't expect all of the data to come from the same distribution
- We can instead use two distributions, one for each class...

$$\begin{aligned} p(\mathbf{X} \mid Y = k) &= p_k(\mathbf{X}) && \textit{(different distributions)} \\ &= p(\mathbf{X} \mid \theta_k) && \textit{(same family, different parameters)} \end{aligned}$$

- These are called **class-conditional distributions**

Class-conditional Gaussians

- Let the class-conditional densities be **multi-variate Gaussians**
- Assume also that the data are iid *given the class*:

$$
\begin{aligned}
\mathbf{X} \mid Y = k &\overset{iid}{\sim} \mathcal{N}(\mathbf{X} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\
p(\mathbf{X} \mid Y = k) &= \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_k|^{1/2}} e^{-\frac{1}{2}(\mathbf{X}-\boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{X}-\boldsymbol{\mu}_k)}
\end{aligned}
$$

- We have standard estimators for the class-conditional parameters

$\boldsymbol{\mu}_k$ Sample mean of samples in class k

$\boldsymbol{\Sigma}_k$ Sample covariance of (d-dimensional) samples in class k

Class posterior

- We want to classify a data-vector, i.e. determine it's class
- Using Bayes' rule:

$$P(Y = 1 \mid \mathbf{X}) \quad = \quad \frac{p(\mathbf{X} \mid Y = 1)P(Y = 1)}{p(\mathbf{X} \mid Y = 1)P(Y = 1) + p(\mathbf{X} \mid Y = 0)P(Y = 0)}$$

- Same machinery!!
  - That X is continuous doesn’t change the mathematics
- If we can estimate the two class-conditional densities, we can classify data-points

Decision boundary

- Visualize X-space...
- Once we've built our classifier, for any point *X* in this space we can get

P(Y=1 | X)

P(Y=0 | X)

- And thereby assign the point to a class
- **Decision boundary**: set of points {X} for which

P(Y=1 | X) = P(Y=0 | X)

- That is, can't decide which class, in this sense "on the boundary" between regions of the space corresponding to each class

- **Q: For the Gaussian case, what's the equation (in *X*) of the decision boundary? Assume equal covariances Sigma.**
- **What sort of decision boundary do you get?**

Decision boundary

- **Starting with optimal decision rule…**

$$\text{argmax}_k\, P(Y = k|\mathbf{X} = \mathbf{x})$$

$$= \text{argmax}_k\, p(\mathbf{x}|Y = k)P(Y = k)$$

$$= \text{argmax}_k\, \log p(\mathbf{x}|Y = k) + \log P(Y = k)$$

$$= \text{argmax}_k\, -\tfrac{1}{2}\log|\Sigma_k| - \tfrac{1}{2}(\mathbf{x} - \mu_k)'\Sigma_k^{-1}(\mathbf{x} - \mu_k) + \log P(Y = k)$$

$$= \text{argmax}_k\, -\tfrac{1}{2}\log|\Sigma_k| + \mathbf{x}'\Sigma_k^{-1}\mu_k - \tfrac{1}{2}\mu_k'\Sigma_k^{-1}\mu_k - \tfrac{1}{2}\mathbf{x}'\Sigma_k^{-1}\mathbf{x} + \log P(Y = k)$$

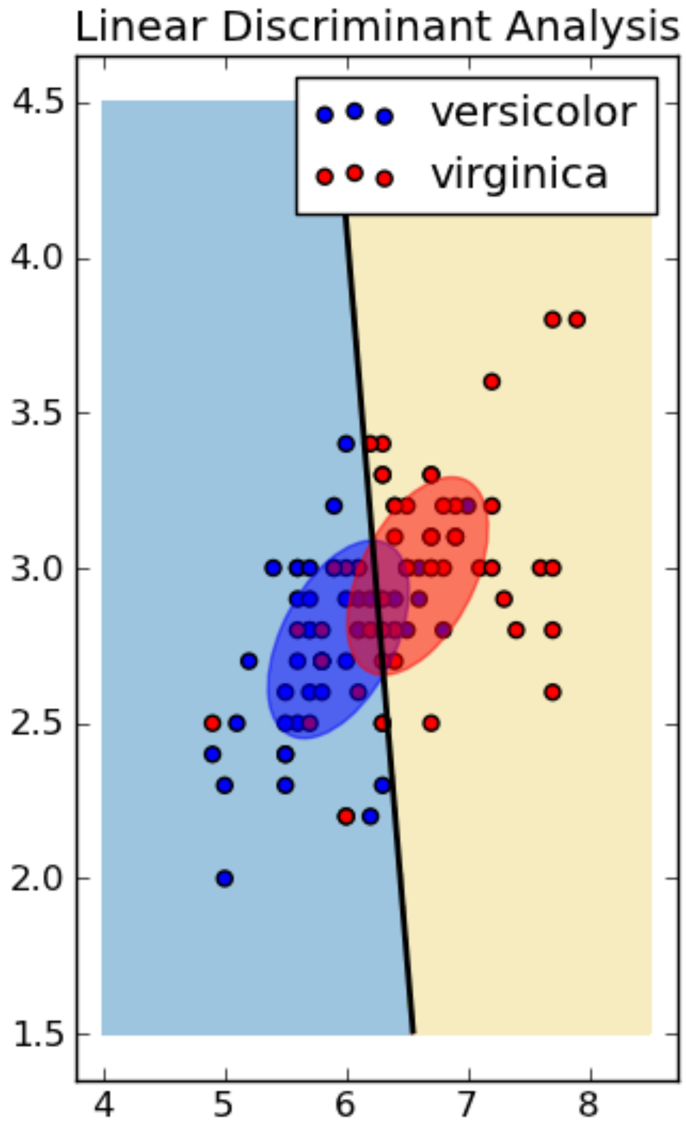
- **If we assume *equal* covariances…** $\Sigma = \Sigma_k$
  - **Then quadratic term becomes** $\mathrm{x}'\Sigma^{-1}\mathbf{x}$ **and is irrelevant for maximizing**
  - **Boundary will depend on** $\mathrm{x}'\Sigma^{-1}\mu_k$ **a linear function in x**

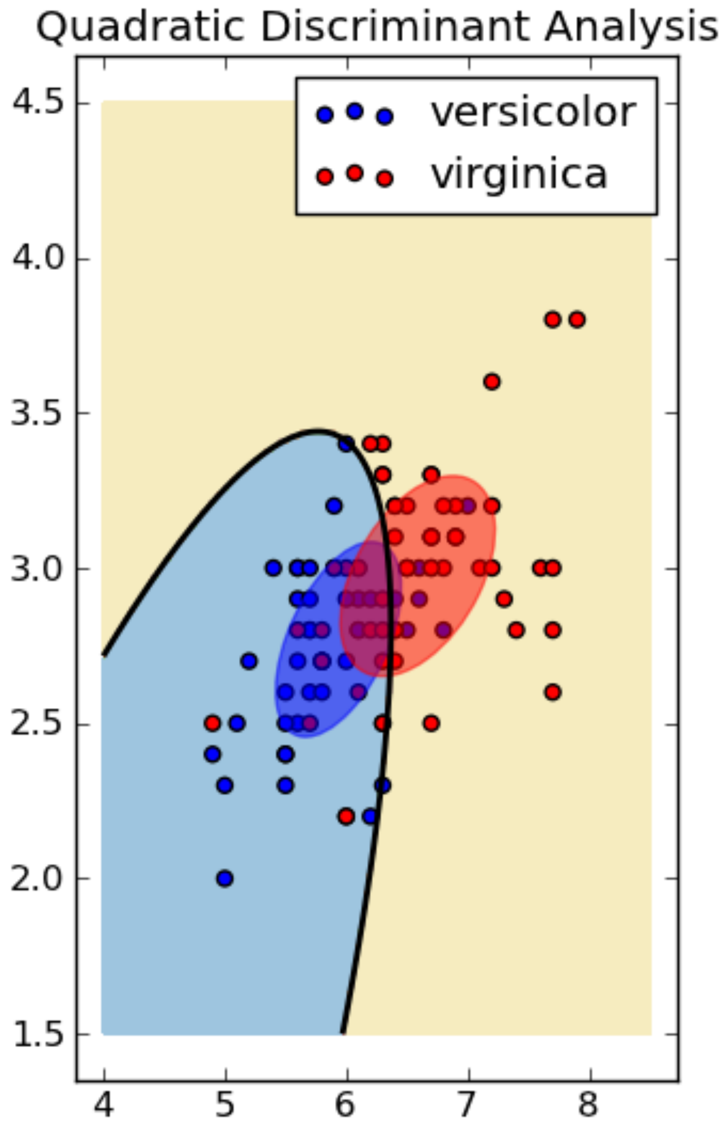
- **Otherwise, for *unequal* covariances, boundary is quadratic**

Linear vs. Quadratic Boundary

"Iris" data

- Based on different types of flowers
- Length and the width of the pedals





Linear and quadratic discriminants

- The corresponding classification algorithms are called
  - **Linear discriminant analysis**, and
  - **Quadratic discriminant analysis**, respectively.
- These are simple, but surprisingly effective classifiers. Hastie *et al.*:

"...LDA and QDA perform well on an amazingly large and diverse set of classification tasks... whatever exotic tools are the rage of the day, we should always have available these two simple tools."