THE UNIVERSITY OF ARIZONA

Classification – Logistic Regression

Agenda

In this session, we will discuss:

- Introduction to classifications
- Linear regression vs. Logistic regression
- Brief about Prediction
- Introduction to Logistic regression

Classification

Supervised learning…class labels!

Classification

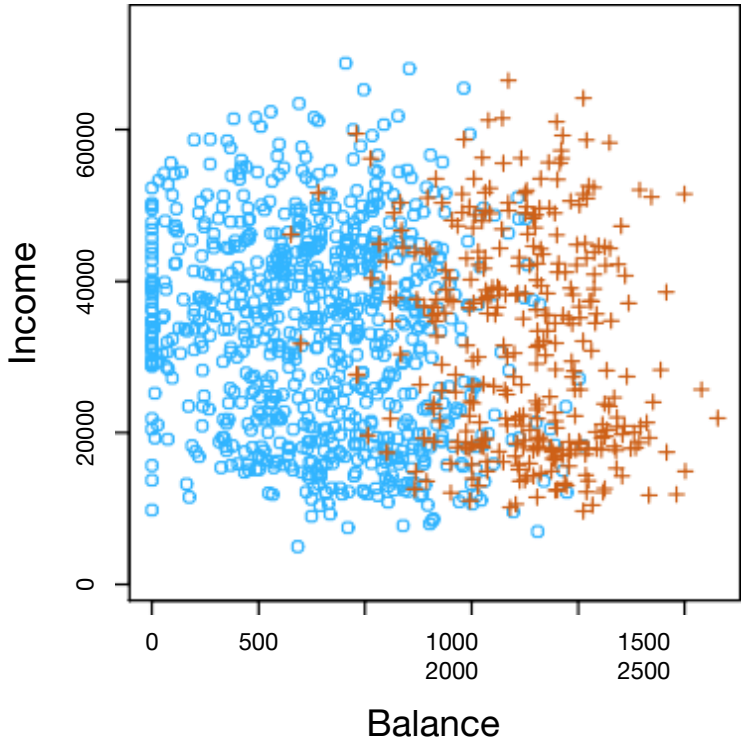
- Qualitative variables (nominal) take values in an unordered set of classes (C), such as:

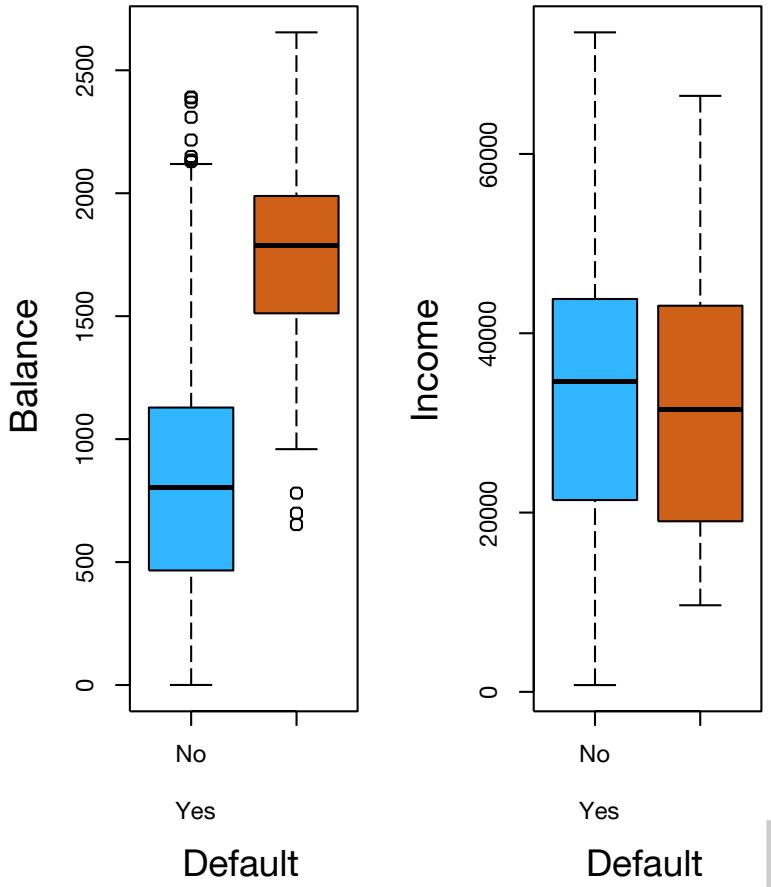
Eye color ∈ {brown, blue, green}
mail∈ {spam, ham}

- Given a feature vector *X* and a qualitative response *Y* taking values in the set C, the classification task is to build a function *C*(*X*) that takes as input the feature vector *X* and predicts its value for *Y*; i.e., *C*(*X*) ∈ C.
- Classifiers – non-probabilistic or probabilistic
- Often, we are more interested in estimating the *probabilities* that *X* belongs to every category in C.

Example: Credit Card Default

Group-level dispersion vs. income and balance





Balance seems to be of more importance than income (default)

Default ~ failing to make CC payments for a given number of days (~180)

Can we use Linear Regression?

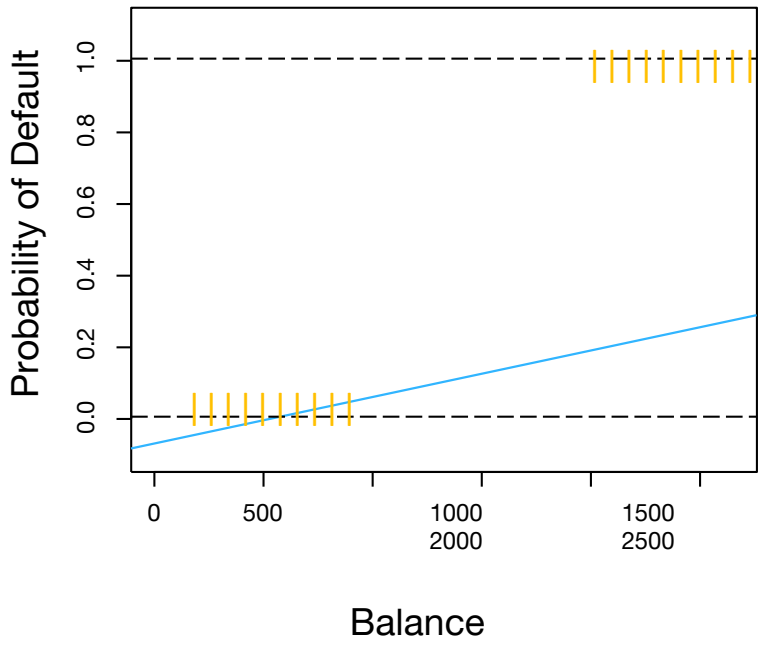
Suppose for the Default classification task that we code:

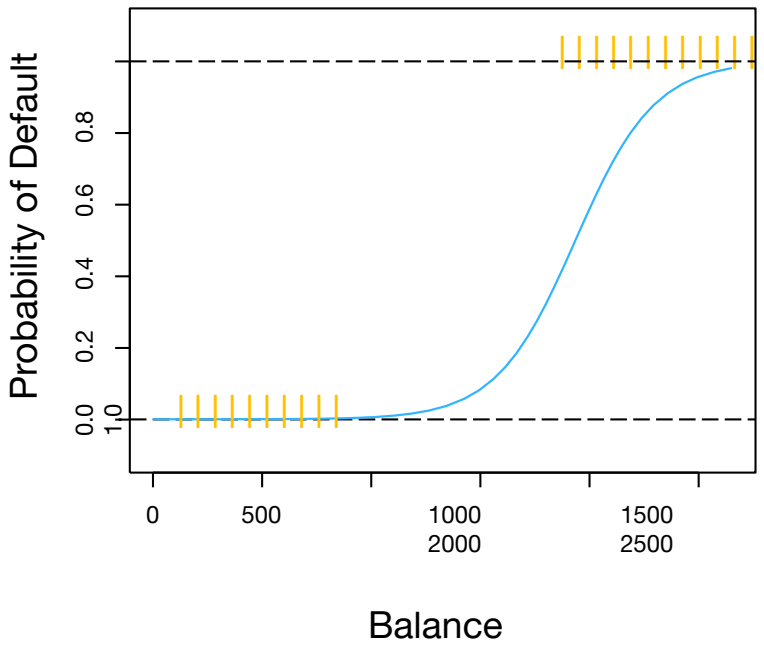
$$Y = \begin{cases} 1 & \text{if No} \\ 2 & \text{if Yes.} \end{cases}$$

Can we simply perform a linear regression of Y on X and classify it as *Yes* if $\hat{y} > 0.5$?

- In this case of a binary outcome, linear regression does a good job as a classifier and is equivalent to *linear discriminant analysis,* which we will discuss later.
- Since in the population $E(Y \mid X = x) = \Pr(Y = 1 \mid X = x)$, we might think that regression is perfect for this task.
- However, *linear* regression might produce probabilities less than zero or bigger than one. *Logistic regression* is more appropriated. ("squeezes" the range of the response).

Linear vs. Logistic Regression

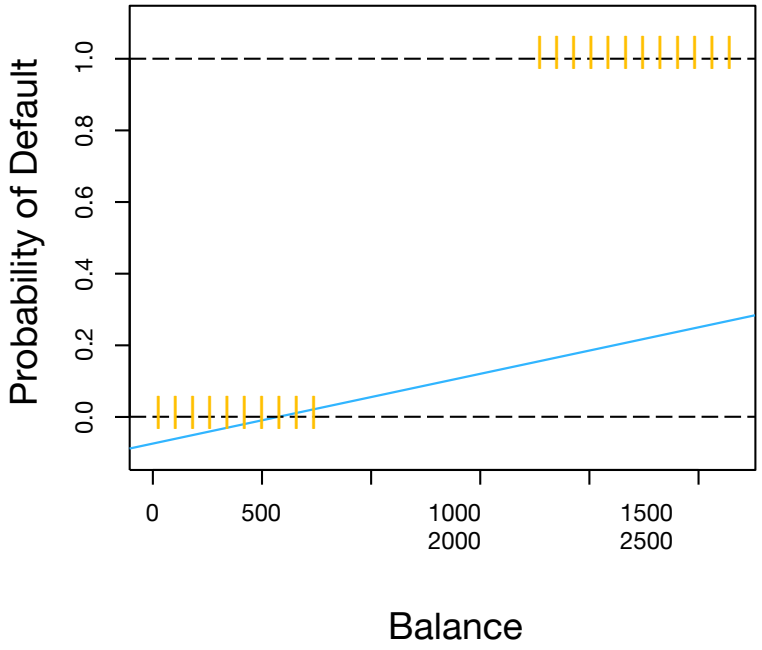




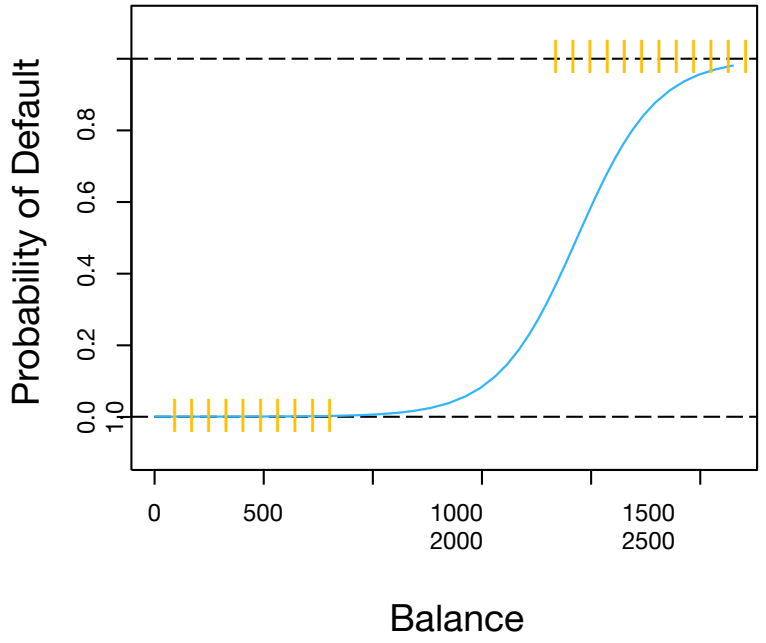
- **Orange marks:** Response *Y* (0 or 1).
- **Left:** Linear regression does not reflect Pr(*Y* =1|*X*) correctly (among others).
- **Right:** Logistic regression seems to reflect the actual change in probabilities across values of X.

Linear vs. Logistic Regression

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$



Sigmoid curve



Logistic regression ensures that our estimate for $p(X)$ lies between 0 and 1.

Logistic Regression

Logistic regression uses the form

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

Remember our lecture on linear models?

It is easy to see that no matter what values β_0, β_1, or X take, $p(X)$ will have values between **0 and 1**.

A bit of rearrangement gives:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$

This monotone transformation is called the *log odds* or *logit* transformation of $p(X)$. (here, log = *natural log*)

Maximum Likelihood

Another approach: Gradient descent

We use maximum likelihood to estimate the parameters.

$$\ell(\beta_0, \beta) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)).$$

"cost function"

Maximum likelihood estimates: differentiate the log-likelihood with respect to the parameters, set the derivatives equal to zero, and solve.

This *likelihood* gives the probability of the observed zeros and ones in the data. We pick β_0 and β_1 to maximize the likelihood of the observed data.

Most statistical packages can fit linear logistic regression models by maximum likelihood.

	Coefficient	Std. error	Z-statistic	P-value
Intercept	-10.7	0.3612	-29.5	< 0.0001
balance	0.006	0.0002	24.9	< 0.0001

Making Predictions

What is our estimated probability of default for someone with a balance of **$1000**?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1000}}{1+e^{-10.6513+0.0055\times 1000}} = 0.006$$

With a balance of **$2000**?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 2000}}{1+e^{-10.6513+0.0055\times 2000}} = 0.586$$

	Coefficient	Std. error	Z-statistic	P-value
Intercept	-10.7	0.3612	-29.5	< 0.0001
balance	0.006	0.0002	24.9	< 0.0001

Making Predictions

Let's do it again, using student as the predictor.

	Coefficient	Std. error	Z-statistic	P-value
Intercept	-3.5	0.0707	-49.6	< 0.0001
student[Yes]	0.405	0.115	3.52	4E-04

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=Yes}) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431,$$

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=No}) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292.$$

Logistic Regression with Several Variables

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

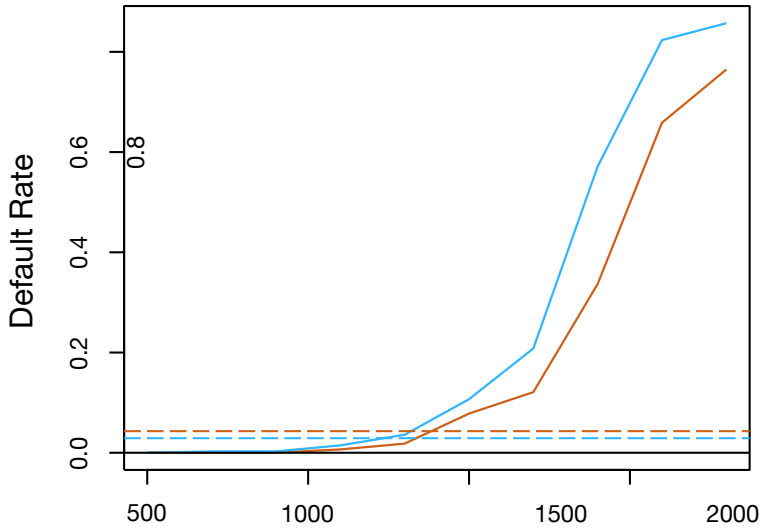
…multiple linear regression models?

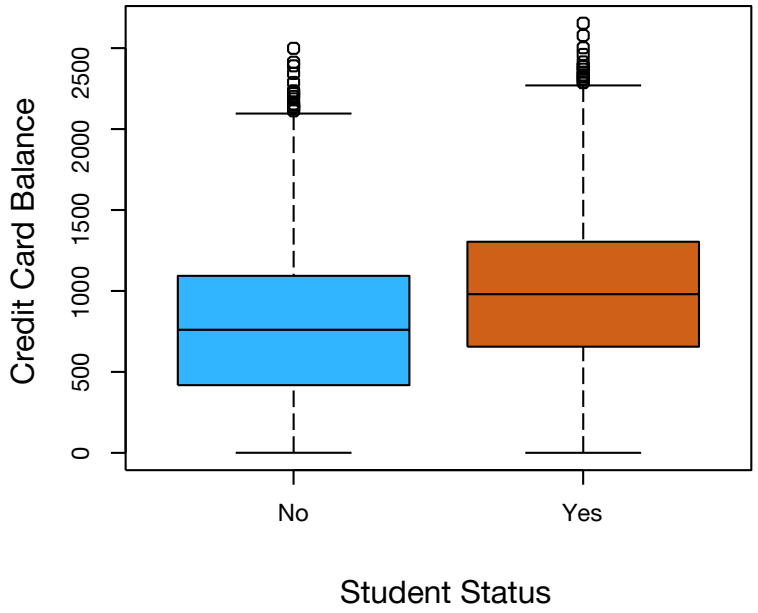
$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

	Coefficient	Std. error	Z-statistic	P-value
Intercept	-10.9	0.4923	-22.1	< 0.0001
balance	0.006	0.0002	24.74	< 0.0001
income	0.003	0.0082	0.37	0.712
student[Yes]	-0.65	0.2362	-2.74	0.006

The coefficient for student is now negative (it was positive before).

Logistic Regressions: Confounding





- Students tend to have higher balances than non-students, so their marginal default rate is higher than for non-students.
- But for each level of balance, students default less than non-students.
- Multiple logistic regression can tease this out.

Logistic Regression with more than Two Classes

Logistic regressions are easily generalized to more than two classes (**multinomial regression**). Two major options:

- **"Classic":** Using a baseline class (*K-1* functions). Parameter estimates are given in respect of the Kth class. Baseline selection is not important.

$$\Pr(Y = k|X = x) = \frac{e^{\beta_{k0}+\beta_{k1}x_1+\cdots+\beta_{kp}x_p}}{1+\sum_{l=1}^{K-1} e^{\beta_{l0}+\beta_{l1}x_1+\cdots+\beta_{lp}x_p}} \qquad \Pr(Y = K|X = x) = \frac{1}{1+\sum_{l=1}^{K-1} e^{\beta_{l0}+\beta_{l1}x_1+\cdots+\beta_{lp}x_p}}.$$

- **Softmax:** No baseline selection. A linear function for each class (implemented in glmnet).

$$\Pr(Y = k|X) = \frac{e^{\beta_{0k}+\beta_{1k}X_1+\ldots+\beta_{pk}X_p}}{\sum_{\ell=1}^{K} e^{\beta_{0\ell}+\beta_{1\ell}X_1+\ldots+\beta_{p\ell}X_p}}$$

Summary

Here is a quick recap:

- We discussed and understood classifications using an example of credit card default.
- We discussed the reason for preferring Logistic regression over Linear regression, as *linear* regression might produce probabilities less than zero or bigger than one, so *Logistic regression* is more appropriate for modeling.
- We talked about making predictions using a simple example of statistical data of a student.
- We discussed Logistic regression which is easily generalized to more than two classes (**multinomial regression**) and ensures that our estimate for the predictor lies between 0 and 1.

Classification – Discriminant Analysis

Agenda

In this session, we will cover:

- Introduction to discriminant analysis
- Conditional Probabilities
- Introduction to Bayes theorem
- Linear Discriminant Analysis with different P value
- Types of errors
- ROC Curve
- Limitations of LDA

Why Discriminant Analysis?

- When the classes are well-separated, parameter estimates for the logistic regression model are unreliable. Linear discriminant analysis does not suffer from this problem.

- If n is small and the distribution of the predictors X is approximately normal in each of the classes, the linear discriminant model is more stable than the logistic regression model.

- Linear discriminant analysis is popular when we have >2 classes because it also provides low-dimensional views of the data.

Discriminant Analysis

Goal:

Model the distribution of X in each of the classes separately.

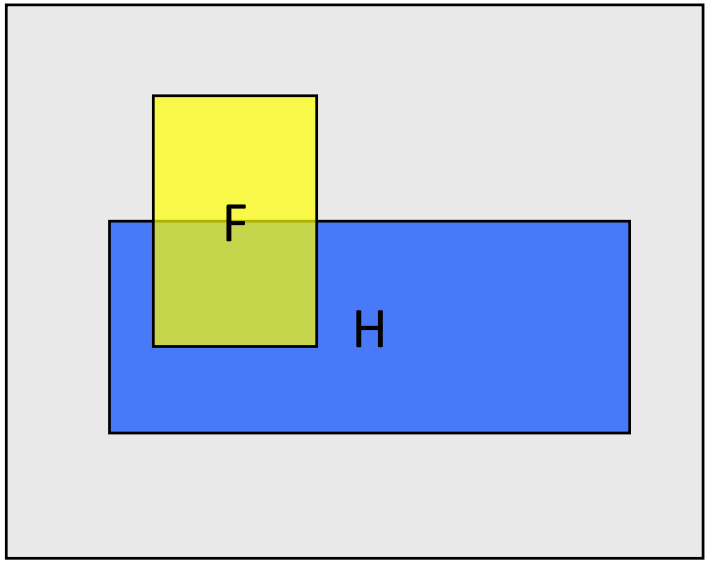
Then, use the *Bayes' theorem* to obtain $\Pr(Y|X)$.

When we use normal (Gaussian) distributions for each class, this leads to linear or quadratic discriminant analysis (other distributions can be used as well).

Conditional Probabilities

- P(A | B) = In worlds where B is true, fraction where A is true

- Example:
  - H: "Have a headache"
  - F: "Have the flu"



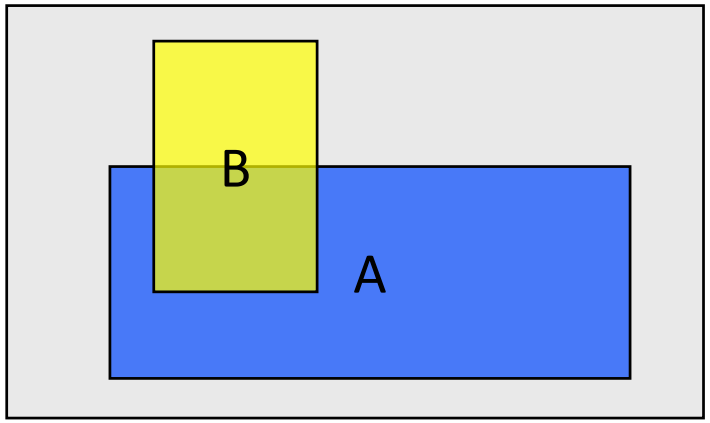
- $P(H) = \frac{1}{10}$
- $P(F) = \frac{1}{40}$
- $P(H \mid F) = \frac{1}{2}$

- Headaches are rare and flu is even more rare, but if you have the flu, there is a 50-50 chance you will have a headache

Bayes' Rule

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} = \frac{P(A \mid B)\, P(B)}{P(A)}$$

- Thomas Bayes "An Essay towards solving a Problem in the Doctrine of Chances" Royal Society, 1763.
- Easy to grasp, if you think of areas:



Bayes' Rule

Concepts:

- Likelihood
  - How much does a certain hypothesis explain the data?

- Prior
  - What do you believe before seeing any data?

- Posterior
  - What do we believe after seeing the data?

Bayes' Theorem for Classification

The Bayes' theorem:

$$\Pr(Y = k|X = x) = \frac{\Pr(X = x|Y = k) \cdot \Pr(Y = k)}{\Pr(X = x)}$$

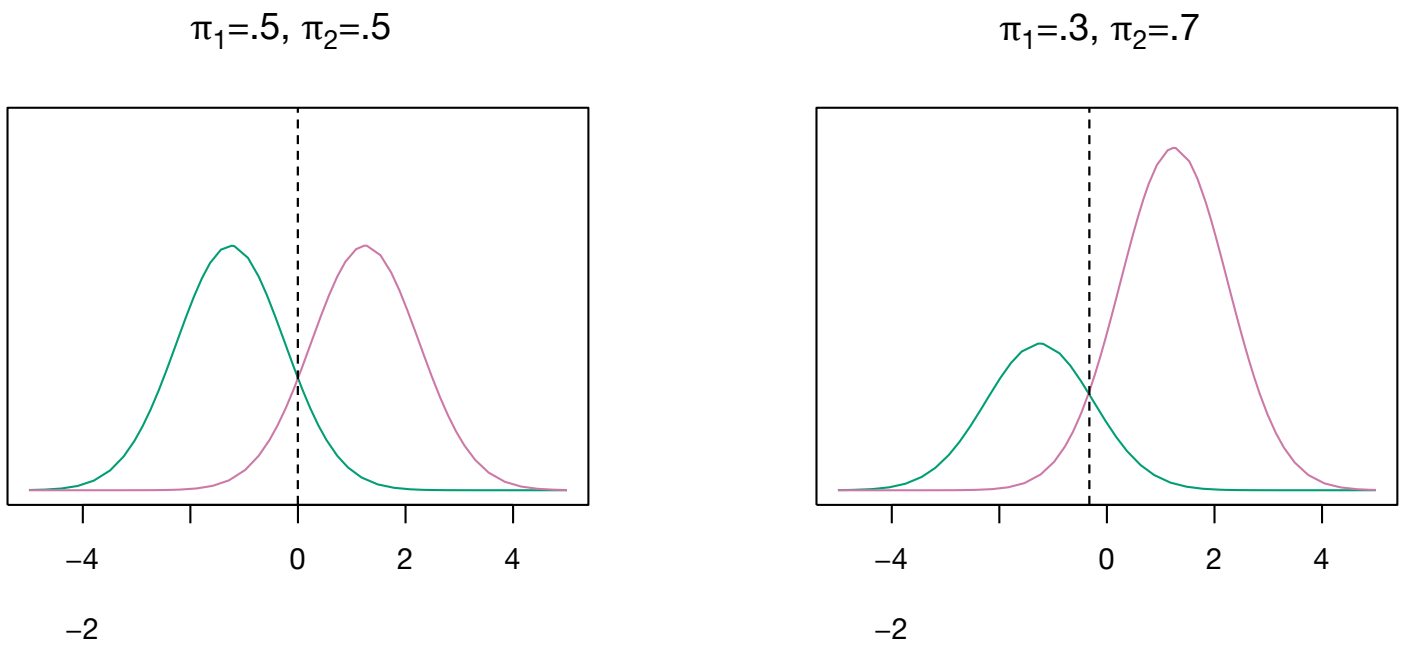
In the context of discriminant analysis,

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}, \quad \textit{where}$$

- $f_k(x)$ =Pr(X=x|Y=k) is the *density* for Xin class k. We will use normal densities (one for each class).
- π_k = Pr(Y =k) is *prior* probability for class k.

Classify to the Highest Density

*Bayes' boundaries…the smallest error**



We classify a new point according to which density is the highest.

When the priors are different, we take them into account as well and compare $\pi_k f_k(x)$. On the right, we favor the pink class — the decision boundary has shifted to the left.

Linear Discriminant Analysis when p = 1

Again, we're assuming normal distributions for each class:

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma_k}\right)^2}$$

- Here μ_k is the mean, and σ_k^2 the variance (in class k). We will assume that all the $\sigma_k = \sigma$ are the same.
- Plugging this into Bayes' formula, we get the following expression for $p_k(x) = \Pr(Y=k|X=x)$:

Description of the k^{th} group

Prior

$$p_k(x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu_k}{\sigma}\right)^2}}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu_l}{\sigma}\right)^2}}$$

Marginal probability

Some simplifications later…

Discriminant Functions (Bayes classifier)

To classify at the value $X=x$, we need to see which of the $p_k(x)$ is the largest. Taking logs, and discarding terms that do not depend on k, we see that this is equivalent to assigning x to the class with the largest *discriminant score*:

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

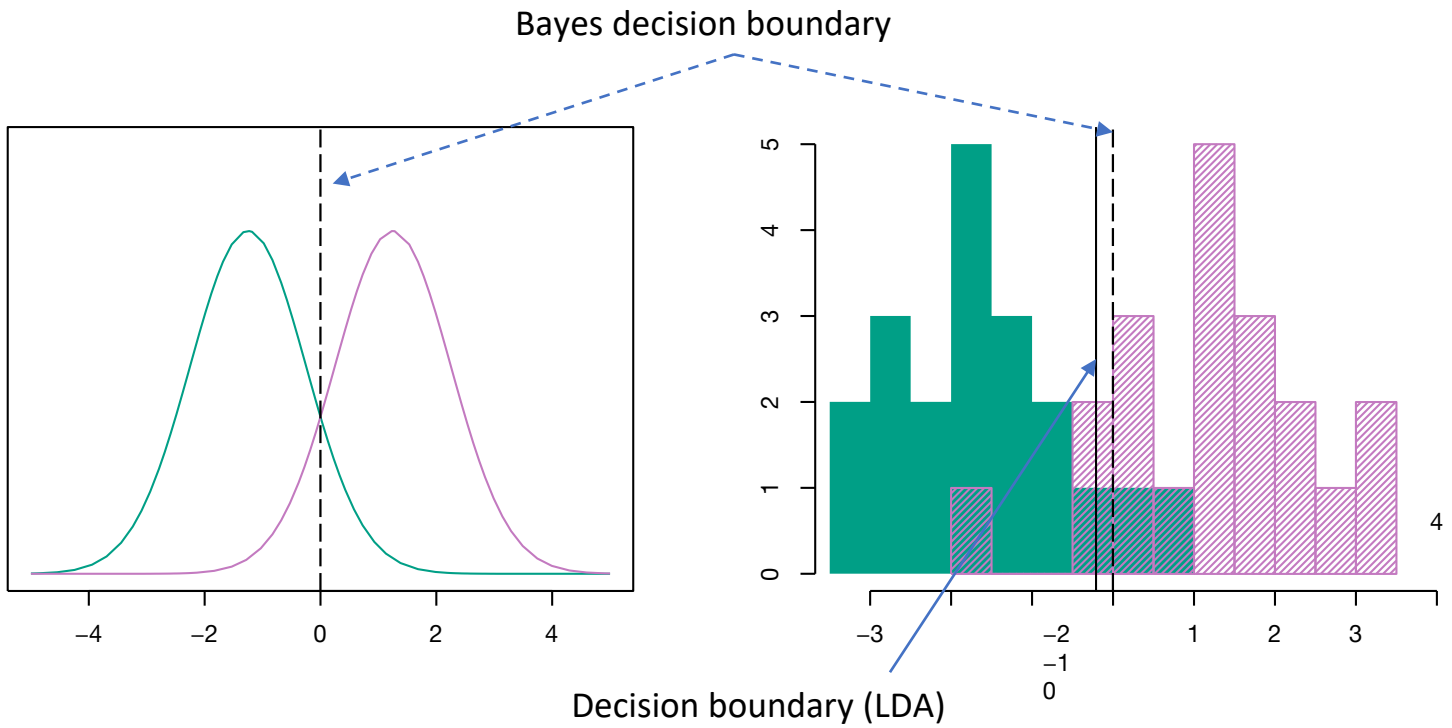
Note that $\delta_k(x)$ is a *linear* function of x.

If there are $K=2$ classes and $\pi_1 = \pi_2 = 0.5$, then one can see that the *decision boundary* is at:

$$x = \frac{\mu_1 + \mu_2}{2}.$$

Discriminant Functions (Bayes' classifier)

$$x = \frac{\mu_1 + \mu_2}{2}.$$



Example with $\mu_1 = -1.5$, $\mu_2 = 1.5$, $\pi_1 = \pi_2 = 0.5$, and $\sigma^2 = 1$.

Typically, we don't know these parameters; we just have the training data. In that case, we simply estimate the parameters (e.g., LDA) and plug them into the rule.

Estimating the parameters (LDA))

$$\hat{\pi}_k = \frac{n_k}{n}$$ An idea for a prior

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:\, y_i = k} x_i$$ An expectation. Average of all the training observations from the k[th] class

LDA:
- Label-specific means
- Shared variance

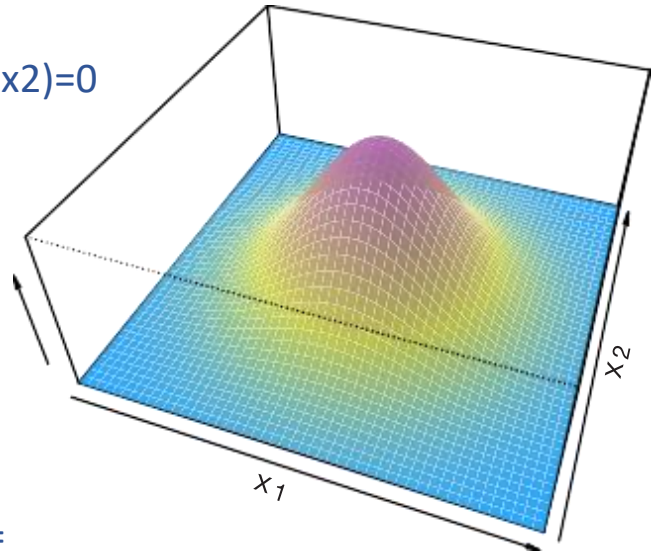
$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:\, y_i = k} (x_i - \hat{\mu}_k)^2$$

$$= \sum_{k=1}^{K} \frac{n_k - 1}{n-K} \cdot \hat{\sigma}_k^2$$ Weighted average of sample variances for each class

where $\hat{\sigma}_k^2 = \frac{1}{n_k - 1} \sum_{i:\, y_i = k} (x_i - \hat{\mu}_k)^2$ is the usual formula for the estimated variance in the *k*th class.

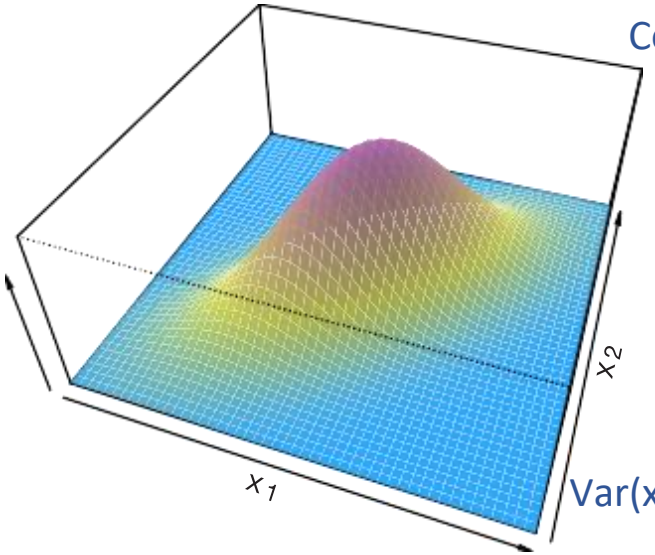
Linear Discriminant Analysis when p > 1

Cor(x1,x2)=0



Var(x1) = Var(x2)

Cor(x1,x2)≠0



Var(x1) ≠ Var(x2)

Density: $f(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \mathbf{\Sigma}^{-1}(x-\mu)}$

Vector of mean values

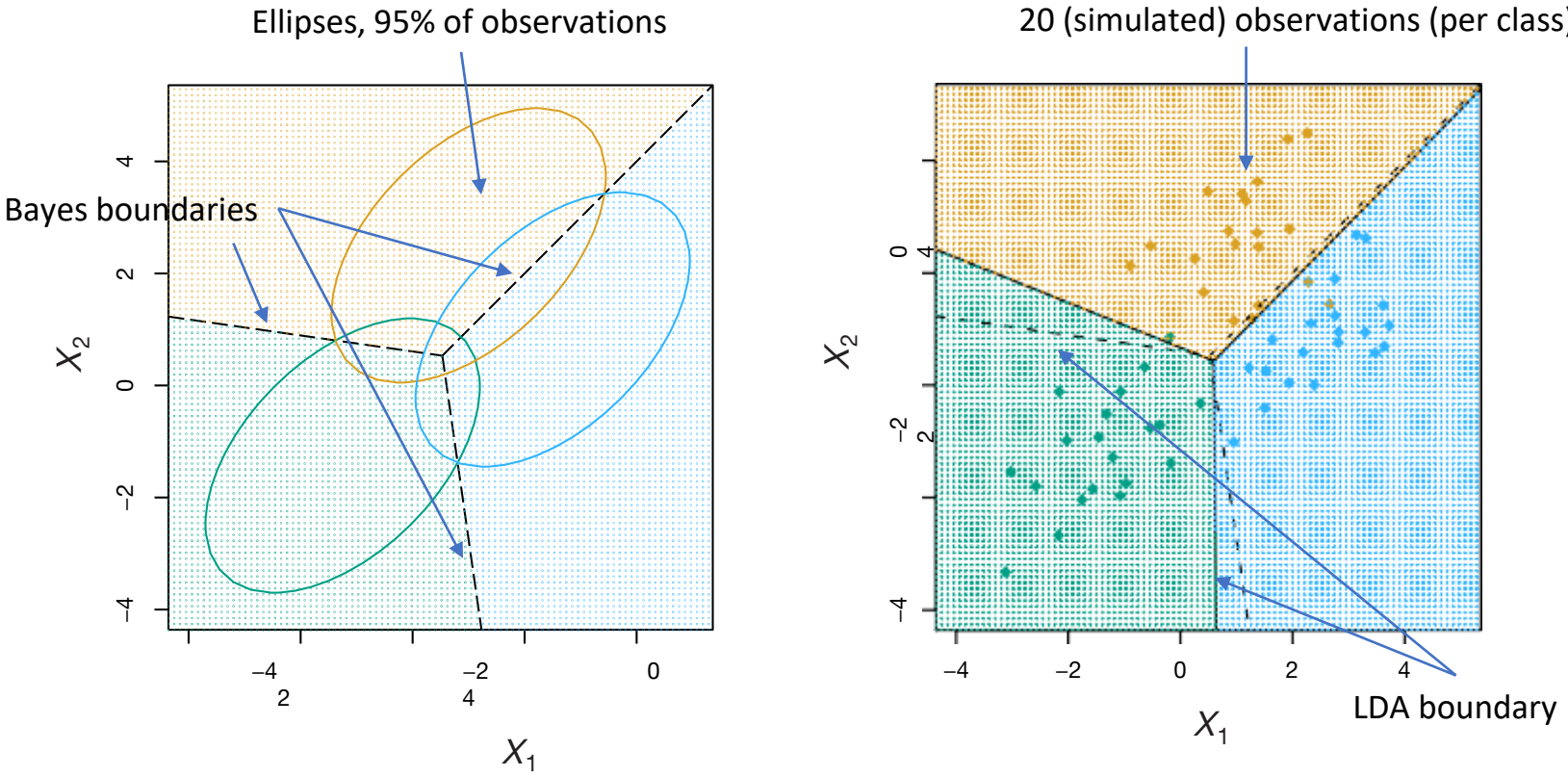
Covariance matrix (shared across labels…)

Discriminant function: $\delta_k(x) = x^T \mathbf{\Sigma}^{-1} \mu_k - \frac{1}{2}\mu_k^T \mathbf{\Sigma}^{-1} \mu_k + \log \pi_k$

Despite its complex form,

$\delta_k(x) = c_{k0} + c_{k1}x_1 + c_{k2}x_2 + \ldots + c_{kp}x_p$ — a linear function.

Illustration: p = 2 and K = 3 classes



Here $\pi_1 = \pi_2 = \pi_3 = 1/3$.

The dashed lines are known as the *Bayes' decision boundaries*. In this case, they yield the fewest misclassification errors, among all possible classifiers.

From $\delta_k(x)$ to Probabilities

Once we have estimates $\hat{\delta}_k(x)$, we can turn these into estimates for class probabilities:

$$\widehat{\Pr}(Y = k|X = x) = \frac{e^{\hat{\delta}_k(x)}}{\sum_{l=1}^{K} e^{\hat{\delta}_l(x)}}.$$

So, classifying to the largest $\hat{\delta}_k(x)$ amounts to classifying to the class for which $\widehat{\Pr}(Y = k|X = x)$ is largest.

When *K* = 2, we classify to class 2 if $\widehat{\Pr}(Y = 2|X = x) \geq 0.5$, else to class 1.

LDA on Credit Data

		True Default Status		
		No	Yes	Total
Predicted Default Status	No	9644	252	9896
	Yes	23	81	104
	Total	9667	333	10000

(23 + 252)/10000 errors — a 2.75% misclassification rate! Some caveats:

- This is a *training* error, and we may be overfitting.
- Of the true No's, we make 23/9667 = 0.2% errors; of the true Yes's, we make 252/333 = 75.7% errors!

A Summary of the Types of Errors (Confusion Matrix)

		True class		
		− or Null	+ or Non-null	Total
Predicted class	− or Null	True Neg. (TN)	False Neg. (FN)	N*
	+ or Non-null	False Pos. (FP)	True Pos. (TP)	P*
	Total	N	P	

Name	Definition	Synonyms
False Pos. rate	FP/N	Type I error, 1−Specificity
True Pos. rate	TP/P	1−Type II error, power, sensitivity, recall
Pos. Pred. value	TP/P*	Precision, 1−false discovery proportion
Neg. Pred. value	TN/N*	

Types of Errors

False positive rate: The fraction of true negative examples that are classified as positive — 0.2% in example.

False negative rate: The fraction of true positives that are classified as negative — 75.7% in example.

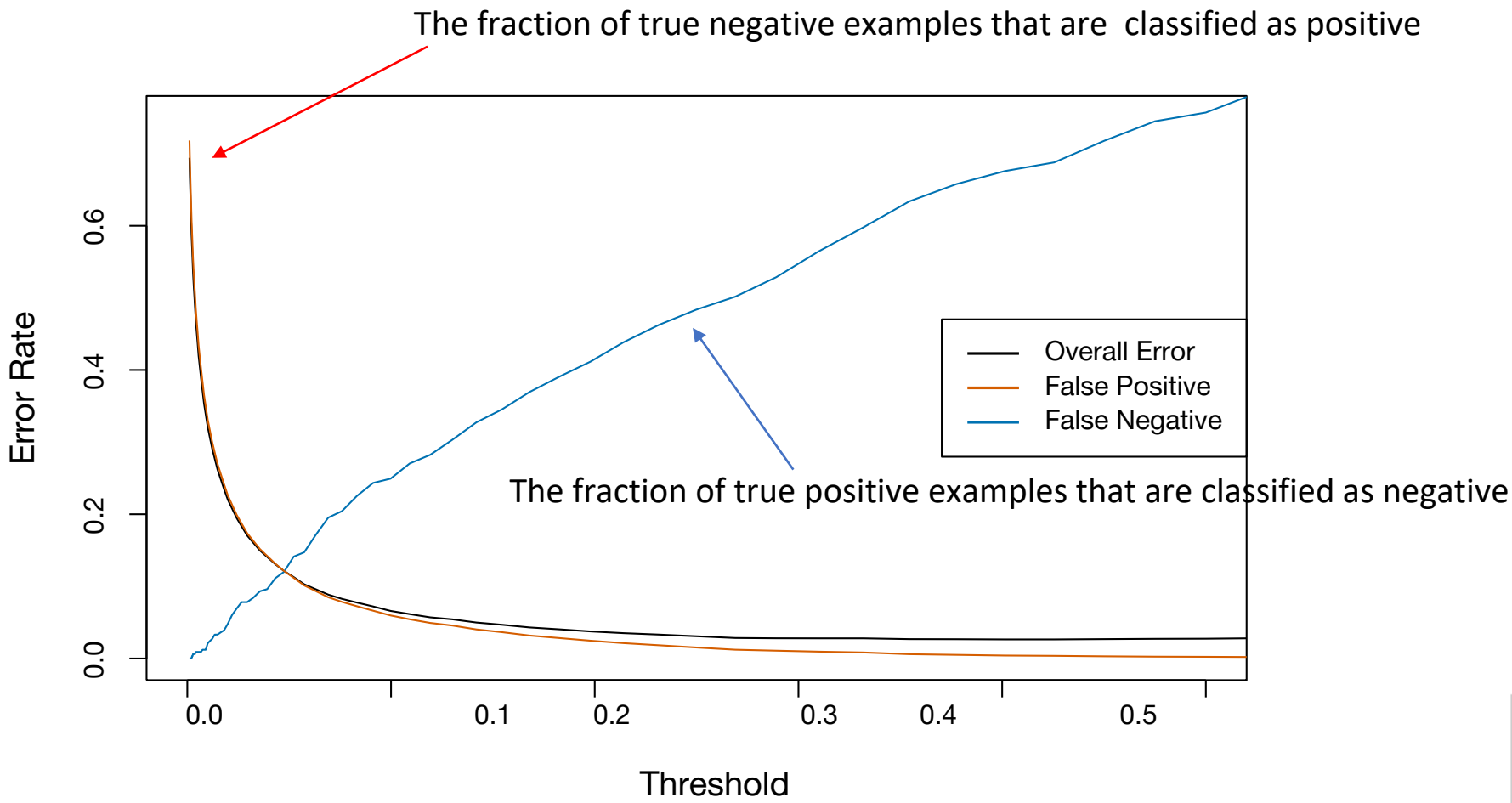
We produced this table by classifying to class Yes if

$\hat{\Pr}(\text{Default} = \text{Yes} | \text{Balance}, \text{Student}) \geq 0.5$

We can change the two error rates by changing the threshold from 0.5 to some other value in [0, 1]:

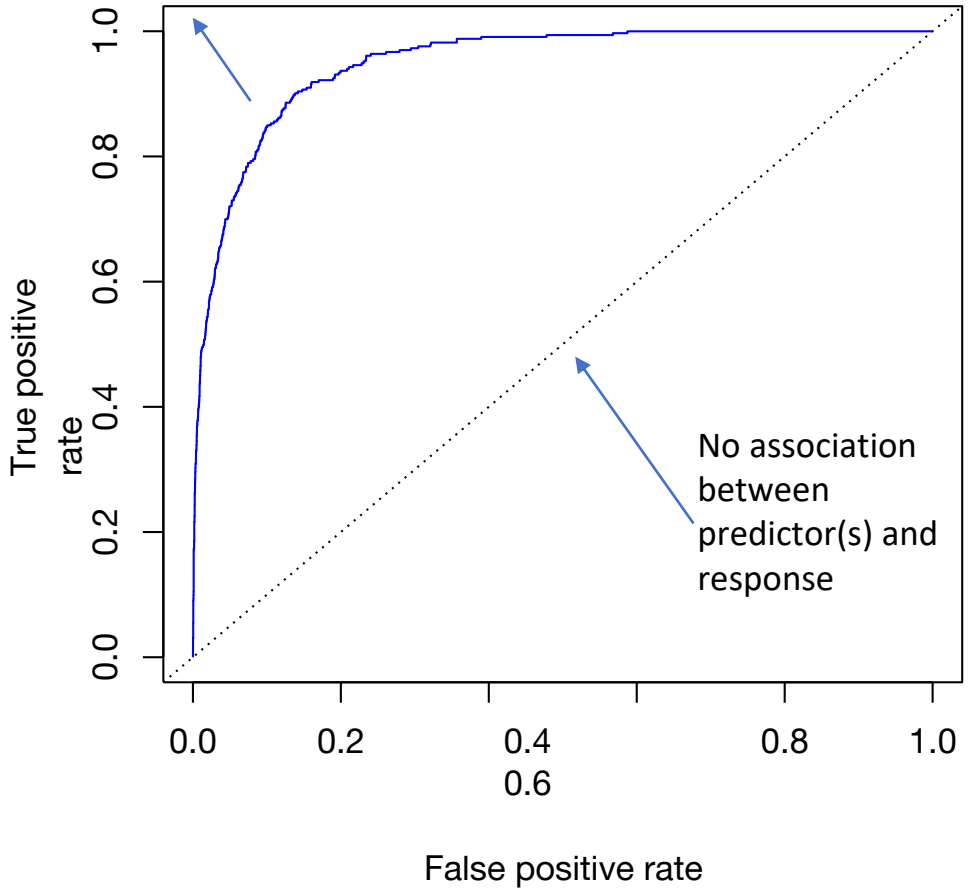
$\hat{\Pr}(\text{Default} = \text{Yes} | \text{Balance}, \text{Student}) \geq \textit{threshold}$, and vary *threshold*.

Varying the threshold



In order to reduce the false negative rate, we may want to reduce the threshold to 0.1 or less.

ROC Curve

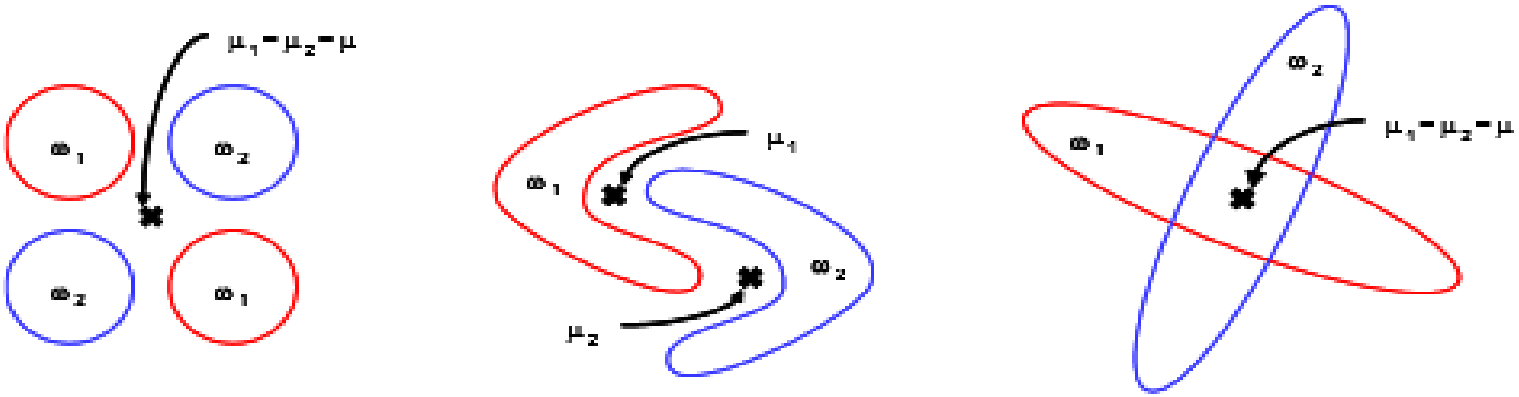


The *ROC plot* displays both simultaneously.

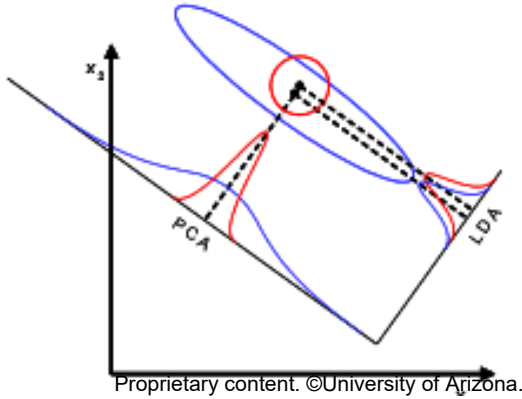
Sometimes we use the *AUC* or *area under the curve* to summarize the overall performance. Higher *AUC* is good.

Limitations of LDA

If the distributions are significantly non-Gaussian, the LDA projections may not preserve complex structure in the data needed for classification.



LDA will also fail if discriminatory information is not in the mean but in the variance of the data.



Summary

Here is a quick recap:

- We discussed that we use discriminant analysis when the classes are well-separated, and parameter estimates for the logistic regression model are unreliable.
- We learned conditional probabilities using an example.
- We discussed Bayes' theorem mathematically and understood how it works.
- We talked about Linear Discriminant Analysis with different p and K values to yield the fewest misclassification errors, among all possible classifiers.
- We talked about various types of errors, such as false positive rate, true positive rate, etc.
- We discussed the threshold rate graphically in order to reduce the false negative rate and understood ROC Curve between true and false positive rates.
- We talked about the limitations of LDA, such as it failed to predict if discriminatory information is not in the mean but in the variance of the data.

Extensions of the LDA

Agenda

In this session, we will discuss:

- Other forms of Discriminant Analysis

Other forms of Discriminant Analysis

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

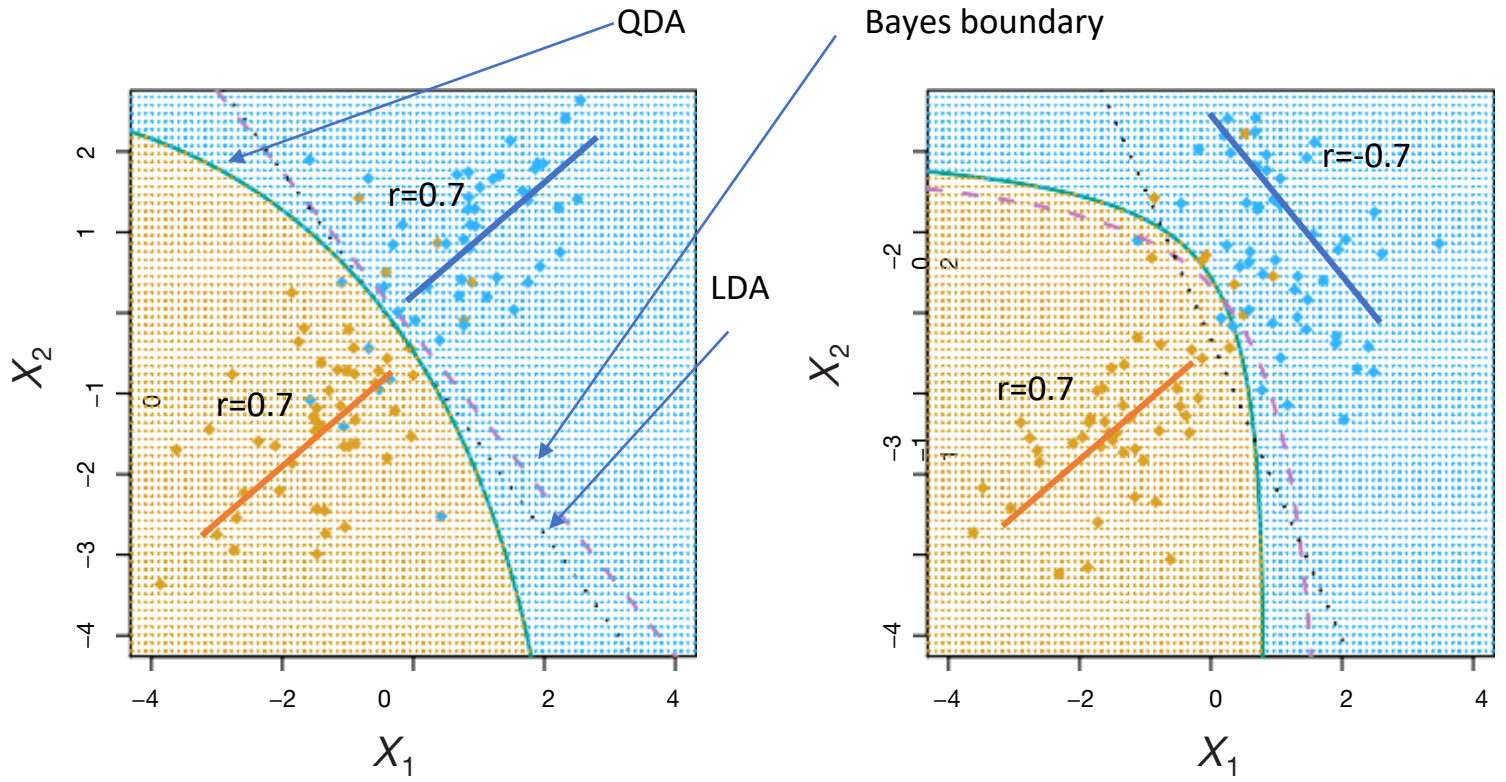
LDA: When $f_k(x)$ are Gaussian densities, with the same covariance matrix $\mathbf{\Sigma}$ in each class.
By altering the forms for $f_k(x)$, we get different classifiers.

- With Gaussians but different $\mathbf{\Sigma}_k$ in each class, we get ***quadratic discriminant analysis***.

- By assuming independency between features, we get a **näive Bayes classifier**.

$$f_k(x) = \prod_{j=1}^{p} f_{jk}(x_j)$$

Quadratic Discriminant Analysis



$$\delta_k(x) = -\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}_k^{-1}(x-\mu_k) + \log\pi_k - \frac{1}{2}\log|\Sigma_k|$$

Because the $\mathbf{\Sigma}_k$ are different, the quadratic terms matter.

Number of parameters

QDA = Kp(p+1)/2 vs LDA=Kp

Näive Bayes

Assumes features are independent in each class.
Useful when ***p* is large**, and so multivariate methods like QDA and even LDA break down (**curse of dimensionality**!).

- Gaussian näive Bayes assumes each $\mathbf{\Sigma}_k$ is diagonal:

$$\begin{aligned}\delta_k(x) &\propto \log\left[\pi_k \prod_{j=1}^{p} f_{kj}(x_j)\right] \\ &= -\frac{1}{2}\sum_{j=1}^{p}\left[\frac{(x_j-\mu_{kj})^2}{\sigma_{kj}^2} + \log\sigma_{kj}^2\right] + \log\pi_k\end{aligned}$$

- Can use for *mixed* feature vectors (qualitative and quantitative). If X_j is qualitative, replace $f_{kj}(x_j)$ with probability mass function (histogram) over discrete categories.

Despite strong assumptions, näive Bayes often produces good classification results.

Summary

Here is a quick recap:

- We discussed other types of Discriminant Analysis, such as *quadratic discriminant analysis, and* näive Bayes classifier, and understood both the forms graphically and mathematically in detail.

THE UNIVERSITY OF ARIZONA

KNN

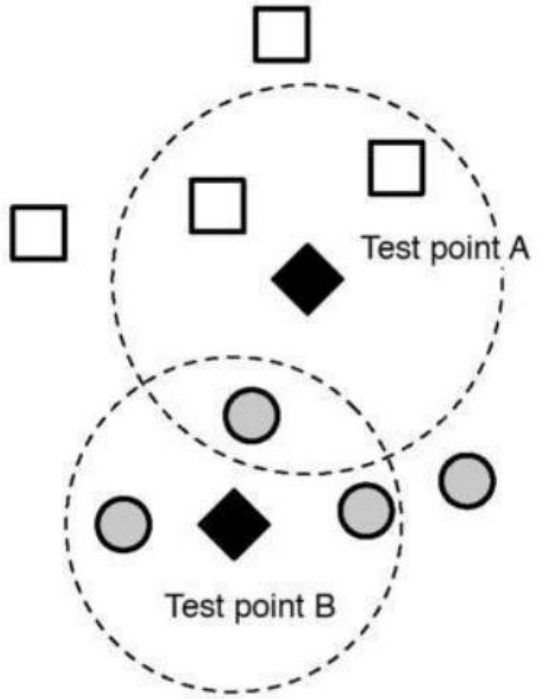
Agenda

In this session, we will discuss:

- Introduction of K-nearest Neighbor
- Distance metrics
- Advantages and limitations of KNN

KNN

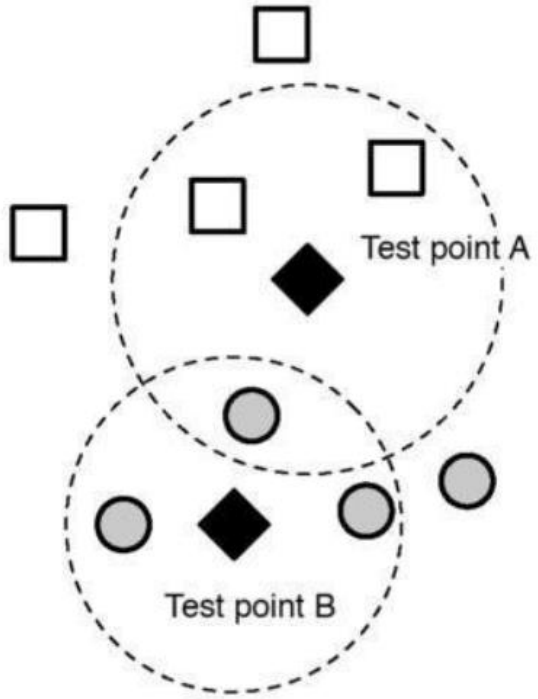
- **K-nearest neighbors** (KNN)
- Very popular because very simple *and* excellent empirical performance
- Handles both binary and multi-class data
- Makes no assumptions about the parametric form of the decision boundary:
  - A **non-parametric** method



KNN

- **Does not have a training phase** – just store the training data and do computation when time to classify.

- Find the K "training points" that are closest to x_{new} .
- Select the **majority** class amongst these K neighbors
  (or for regression: **average**)



K-nearest Neighbor

What value of k should we use?

- Using only the closest example (1NN) to determine the class is subject to errors due to:
  - A single atypical example
  - Noise

- Pick k too large and you end up with looking at neighbors that are not that close.

- Value of *k* is typically odd to avoid ties; 3 and 5 are most common.

Similarity Metrics

Nearest neighbor methods depends on a similarity (or distance) metric.

Ideas?

Euclidean distance

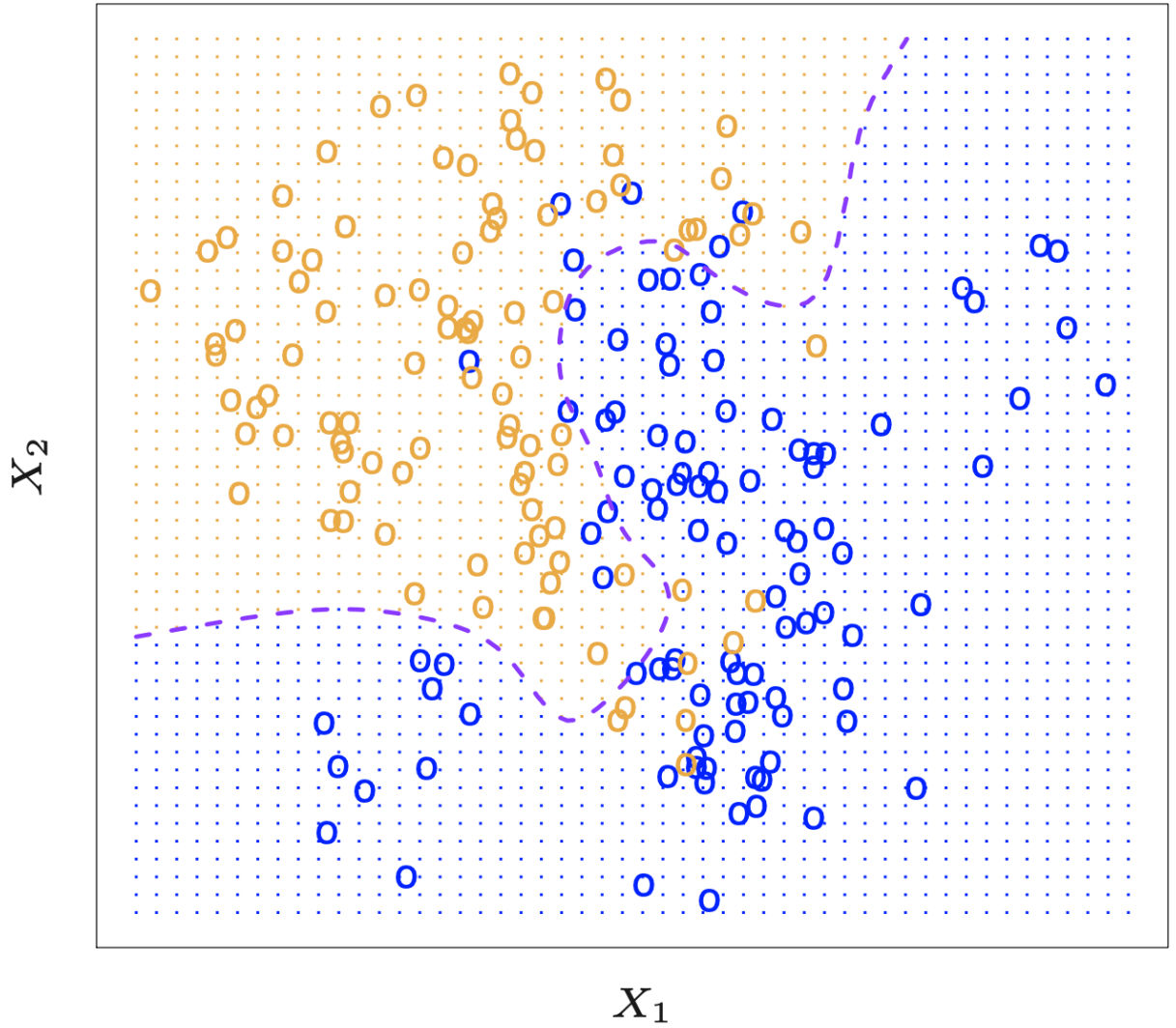
Binary instance space is *Hamming distance* (number of feature values that differ).

For text, cosine similarity of tf.idf weighted vectors is typically most effective.

Advantages and limitations of KNN

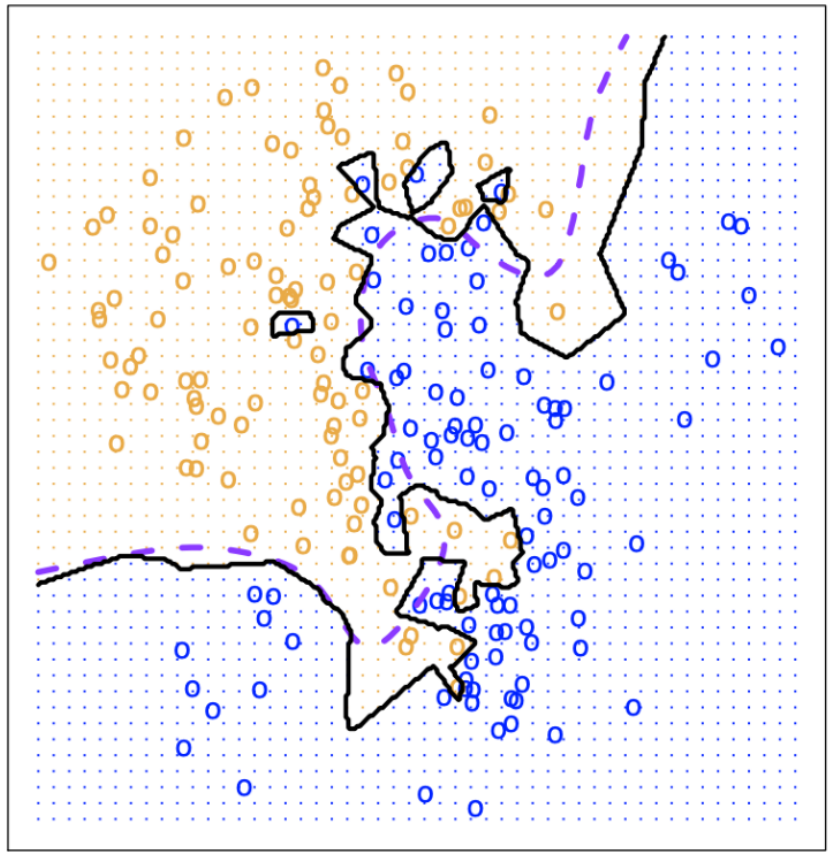
- **Good**
  - No training is necessary.
  - No feature selection necessary.
  - Scales well with large number of classes.
    - Don’t need to train *n* classifiers for *n* classes.
- **Bad**
  - Classes can influence each other.
    - Small changes to one class can have ripple effect.
  - Scores can be hard to convert to probabilities.
  - Can be more expensive at test time.
  - “Model” is all of your training examples which can be large.

Example: K-nearest neighbors in two dimensions

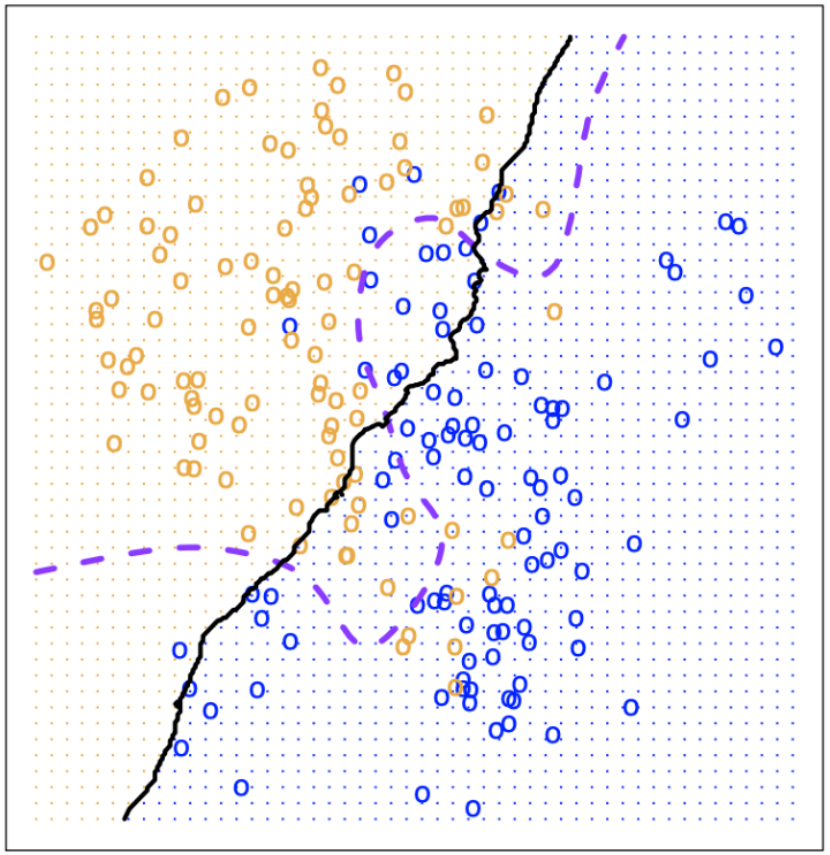


For different values of K

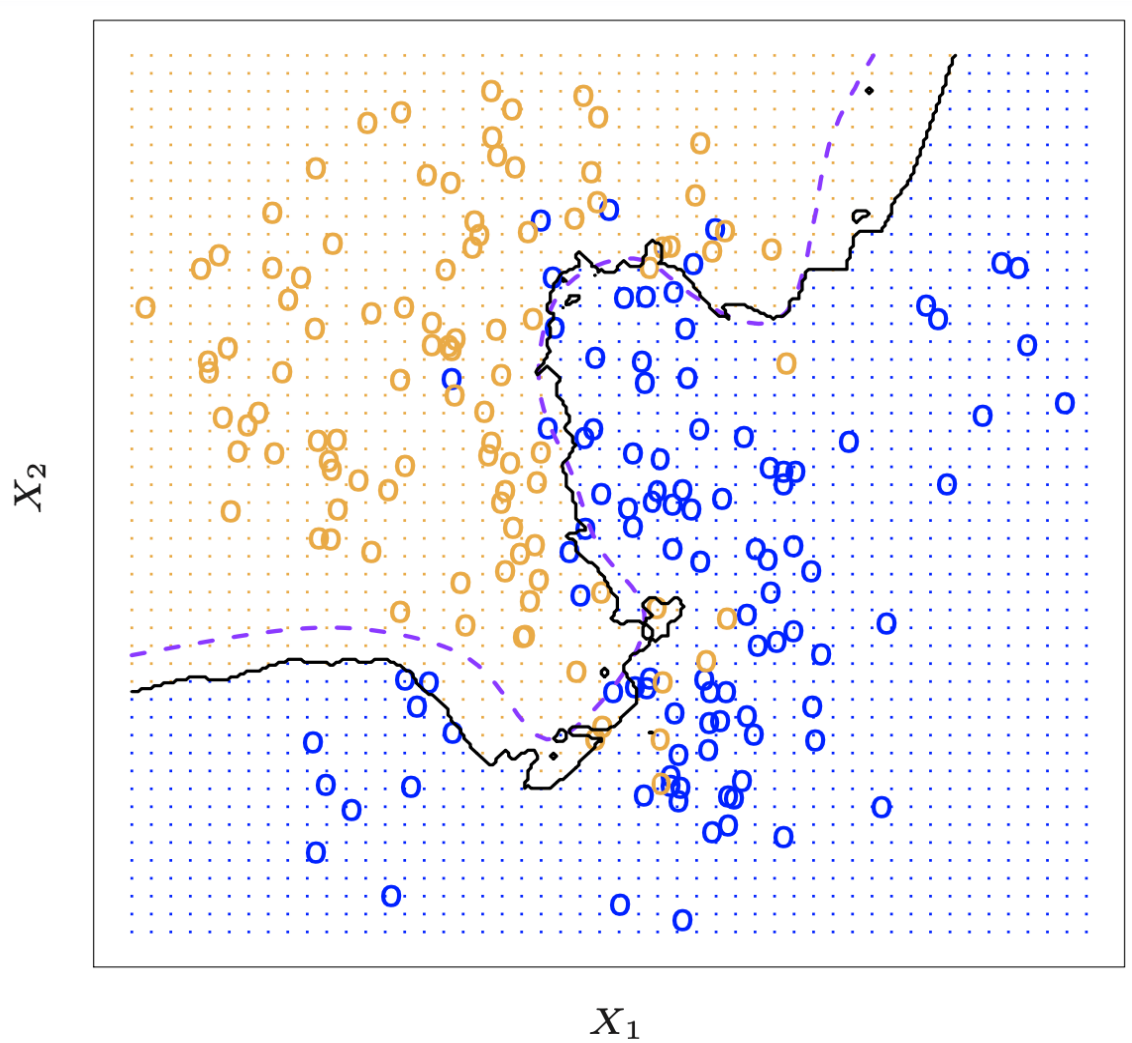
KNN: K=1



KNN: K=100



KNN: K=10



Summary

Here is a quick recap:

- We discussed K-nearest Neighbor is a simple method that has empirical performance and handles both binary and multi-class data.
- We talked about in brief on various distance metrics, such as Euclidean and *Hamming distance*
- We discussed the advantages and limitations of KNN, such that no training is required but can be more expensive during testing the model.
- We talked about KNN with different k-values and understood the effect on test data graphically.