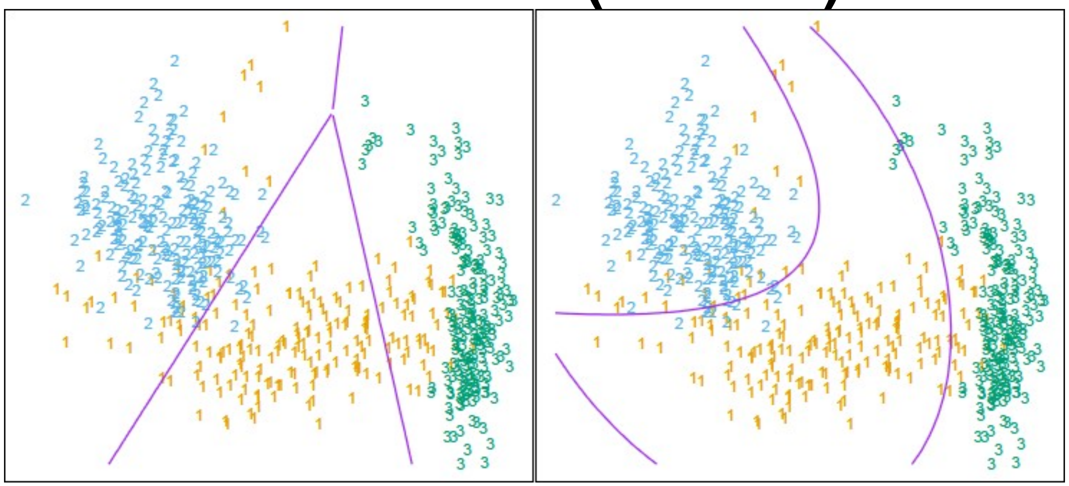
DSCI 552, Machine Learning for Data Science

University of Southern California

M. R. Rajati, PhD

Lesson 3
Classification

Linear and Logistic Regression:
LDA, QDA
k-NN (k Nearest Neighbors)
optimal separating hyperplane – will be discussed later (SVM)



Classification

- Qualitative variables take values in an *unordered* set C, such as:
  eye color $\in$ {brown, blue, green}
  digit $\in$ {0, 1, ..., 9}
  email $\in$ {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. $C(X) \in C$.

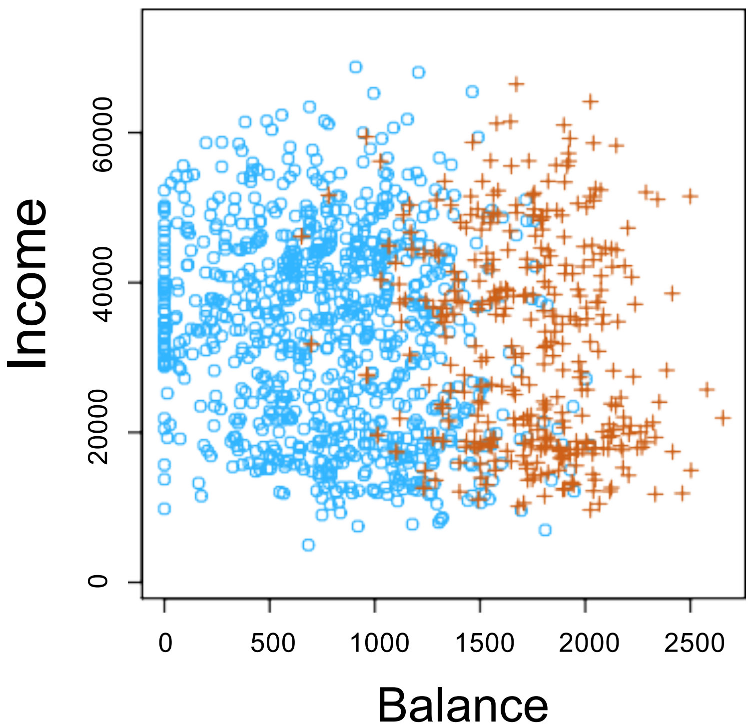
Classification

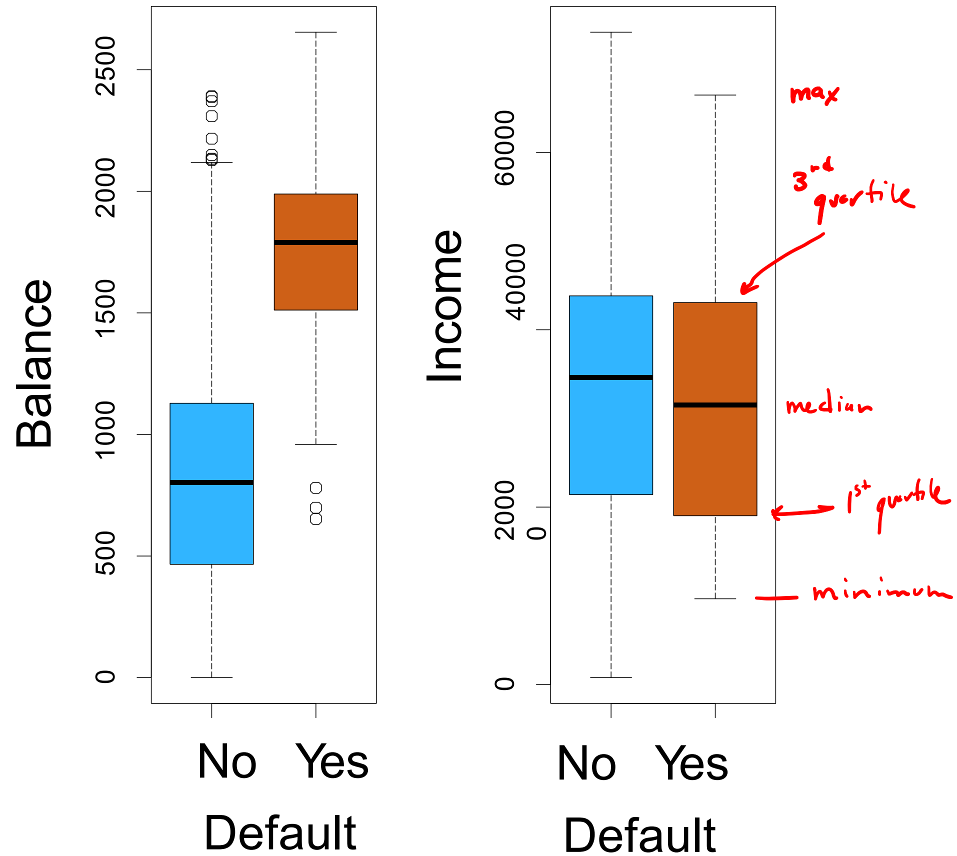
- Often we are more interested in estimating the *probabilities* that X belongs to each category in C.

Case: Credit Card Default Data

- To predict customers that are likely to default
- Possible *X* variables are:
  - Annual Income
  - Monthly credit card balance
- The *Y* variable (Default) is <u>categorical</u>: Yes or No
- How do we check the relationship between *Y* and *X*?

Example: Credit Card Defualt





Can we use Linear Regression?

Suppose for the Default classification task that we code

$$Y = \begin{cases} 0 & \text{if No} \\ 1 & \text{if Yes.} \end{cases}$$

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

Can we use Linear Regression?

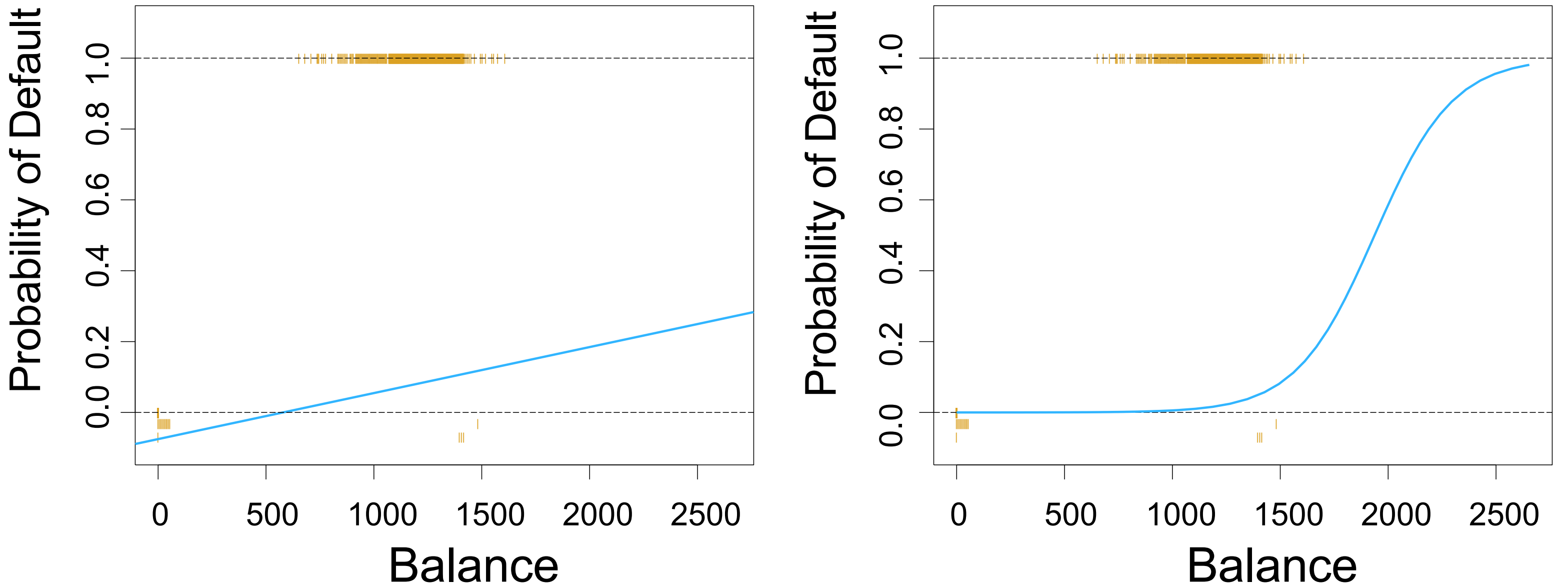
Can we simply perform a linear regression of Y on X and classify as Yes if $\hat{Y} > 0.5$?

- 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 discuss later.
- Since in the population $E(Y|X = x) = \Pr(Y = 1|X = x)$, we might think that regression is perfect for this task.

Can we use Linear Regression?

- However, *linear* regression might produce probabilities less than zero or bigger than one. *Logistic regression* is more appropriate.

Linear versus Logistic Regression



The orange marks indicate the response Y, either 0 or 1.

Linear regression <u>does not</u> estimate $\Pr(Y = 1|X)$ well. Logistic regression seems well suited to the task.

Linear Regression continued

Now suppose we have a response variable with three possible values. A patient presents at the emergency room, and we must classify them according to their symptoms.

$$Y = \begin{cases} 1 & \text{if stroke;} \\ 2 & \text{if drug overdose;} \\ 3 & \text{if epileptic seizure} \end{cases}$$

This coding suggests an ordering, and in fact implies that the difference between stroke and drug overdose is the same as between drug overdose and epileptic seizure.

Linear Regression continued

- This coding suggests an ordering, and in fact implies that the difference between stroke and drug overdose is the same as between drug overdose and epileptic seizure.
- Linear regression is not appropriate here.
- *Multiclass Logistic Regression* or *Discriminant Analysis* are more appropriate.

Multi-Class and Multi-Label Problems

Multiclass classification means a classification task with more than two classes; e.g., classify a set of images of animals which may be horses, birds, or fish.

Multiclass classification makes the assumption that each sample is assigned to one and only one label: an animal can be either a horse or a bird but not both at the same time.

Multi-Class and Multi-Label Problems

Multilabel classification assigns to each sample a set of target labels. This can be thought as predicting properties of a data-point that are not mutually exclusive, such as topics that are relevant for a document.

A text might be about any of religion, politics, finance or education at the same time or none of these.

Binary Classification

A binary classification task assigns only one of the two possible classes to each observation.

Because multi-class and multi-label classification tasks can be performed using binary classification techniques, many times we focus on binary classification.

Logistic Regression

Let's write $p(X) = \Pr(Y = 1|X)$ for short and consider using balance to predict default. Logistic regression uses the form

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

($e \approx 2.71828$ is a mathematical constant [Euler's number.])

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.

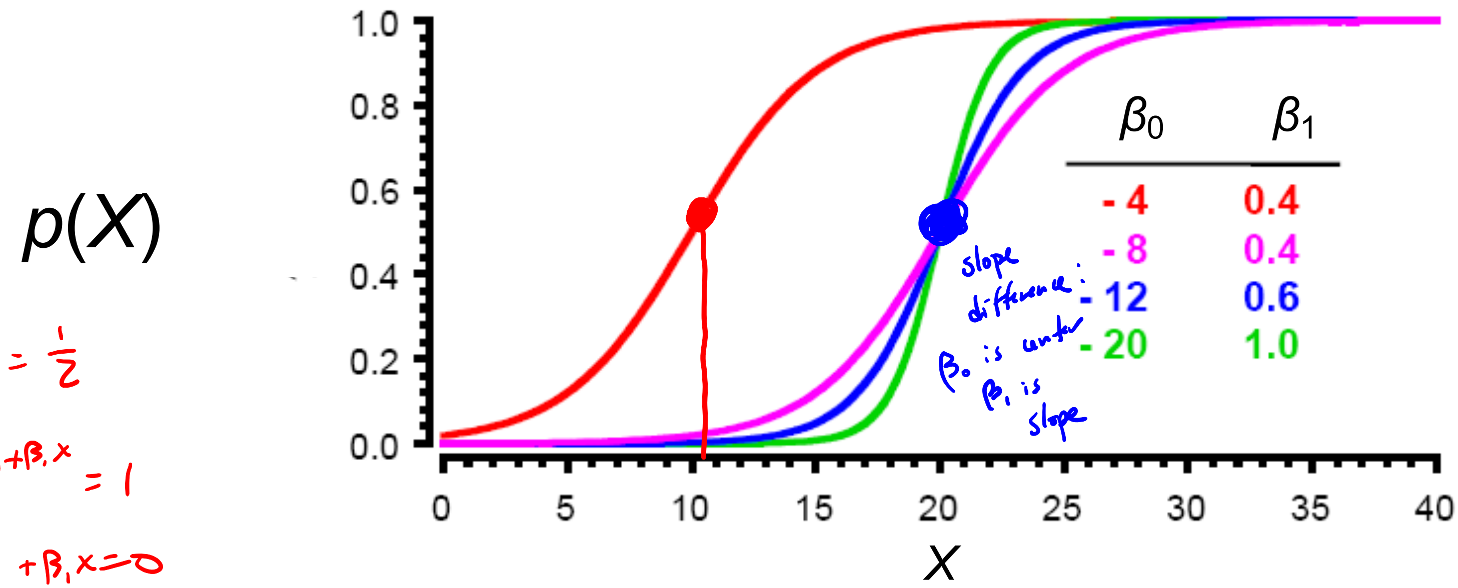
Sigmoid Function

Parameters control shape and location of sigmoid curve

β_0 controls location of midpoint

β_1 controls slope of rise

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



$p(x) = \frac{1}{2}$

$e^{\beta_0+\beta_1 x} = 1$

$\beta_0 + \beta_1 x = 0$

$x = -\frac{\beta_0}{\beta_1}$

When x=- β_0/ β_1 , β_0+ β_1x=0; thus $p(X) = 0.5$

Logistic Regression

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

Math

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*).

Definition of Odds

- The probability of an event divided by the probability of its complement is called its odds.
- Example: The probability of winning in a casino is 1%. What is the odds of winning in that casino?

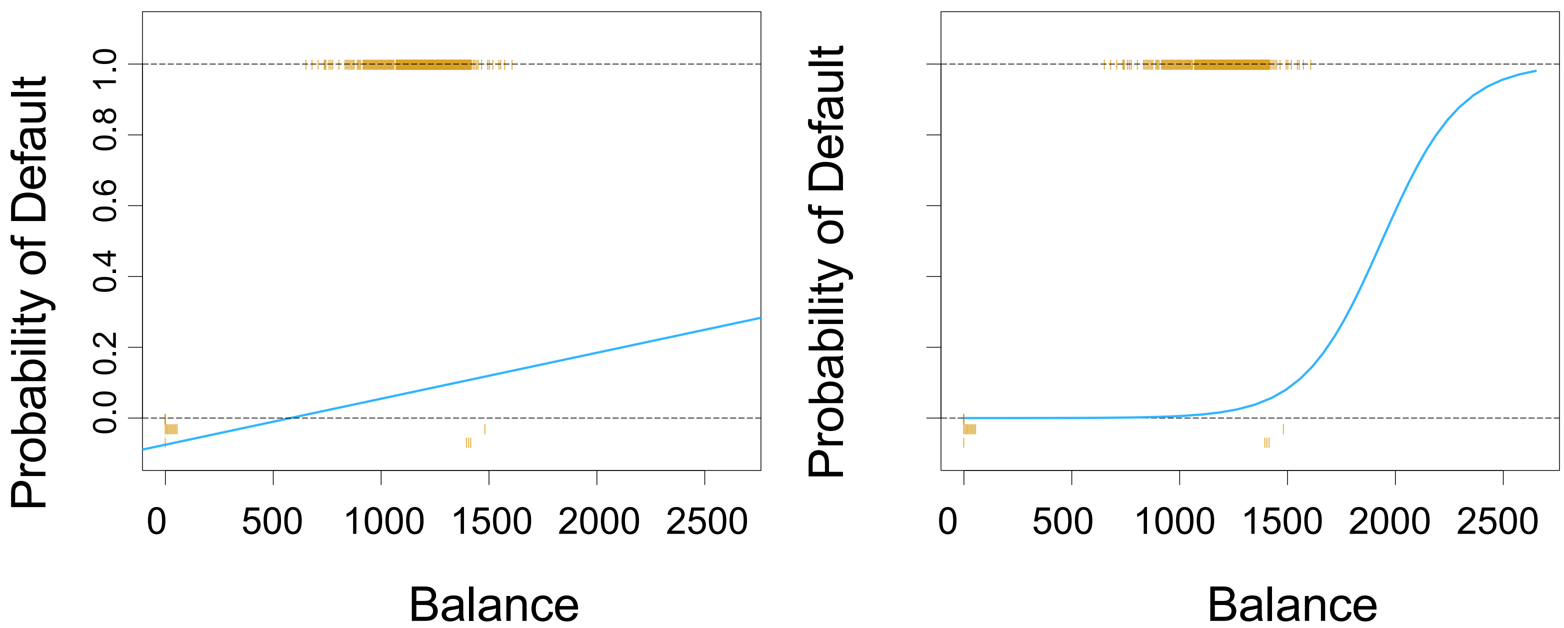
$$O(W)=Pr(W)/(1-Pr(W))$$
$$=0.01/0.99=1/99$$

Logit Model

Therefore, the logit model is trying to predict the log of odds of a model as a linear combination of the predictor(s):

$\log(O(Y=1|X=x))=\beta_0+\beta_1 X$

Linear versus Logistic Regression



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

Class as A Bernoulli Random Variable

One can see class in a binary classification problem as a Bernoulli random variable that can take two values 0 and 1:

$$\Pr(Y=1|\mathrm{X}=\mathrm{x}) = p(x)$$

$$\Pr(Y=0|\mathrm{X}=\mathrm{x}) = 1-p(x)$$

This can be rewritten as:

$$p_{Y|X}(y\,|\,x) = \Pr(Y = y\,|\,X = x) = [\,p(x)]^{y}[1 - p(x)]^{1-y}, y = 0,1$$

Independent Sample of Bernoulli Variables

Assume that we have an *independent* sample whose classes are $Y^{(1)} = y_1$, $Y^{(2)} = y_2$,…, $Y^{(N)} = y_N$

y_i is 0 or 1.

The *joint probability mass* function of this independent sample is:

$p_{Y|X}(y_1, y_2,\ldots, y_N|\text{Data})$
$= \Pr(Y^{(1)} = y_1, Y^{(2)} = y_2,\ldots, Y^{(N)} = y_N|\text{Data})$
$= \Pr(Y^{(1)} = y_1|\text{Data}) \Pr(Y^{(2)} = y_2|\text{Data})\ldots\Pr(Y^{(N)} = y_N|\text{Data})$

Data: $X^{(1)}= x_1,\ldots, X^{(N)}= x_N$

Independent Sample of Bernoulli Variables

The assumption is that the probability that $Y^{(i)} = y_i$ is a function of the features $X^{(i)} = x_i$

So

Independent Sample of Bernoulli Variables

The joint probability mass function is a function of β_0, β_1 and is called the likelihood function, given the data samples.

Maximum Likelihood: Simple Example

Your friend gives you a biased coin and tells you that he is sure that the probability of Heads is either 0.1 or 0.5. You flip the coin 100 times and see 90 Heads. What would be your best estimate of the probability of heads, given that you are sure that your friend is right?

Maximum Likelihood: Simple Example

What would be your best estimate of the probability of heads, given that you are sure that your friend is right?

0.5, it's more likely intuitively

p (90 heads from 100 flips | $p(H) = 0.5$) (this is binomial)

$\binom{100}{90}(0.5)^{90}(1-0.5)^{100-90} = \binom{100}{90}(0.5)^{100}$; vs $\binom{100}{90}(0.1)^{90}(1-0.1)^{100-90}$

Maximum Likelihood: Simple Example

What would be your best estimate of the probability of heads, given that you are sure that your friend is right?

$$\frac{\binom{100}{90}(0.5)^{100}}{\binom{100}{90}(0.1)^{90}(0.9)^{10}} = \frac{(0.5)^{100}}{(0.1)^{100}(0.1)^{-10}(0.9)^{10}} = \frac{5^{100}}{9^{10}} >>>> 1$$

Maximum Likelihood: Simple Example

What would be your best estimate of the probability of heads, given that you are sure that your friend is right?

This simple example motivates us to estimate parameters from data samples by maximizing their likelihood.

Maximum Likelihood

We use maximum likelihood to estimate the parameters β_0, β_1.

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

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.

Maximum Likelihood

Most statistical packages can fit linear logistic regression models by maximum likelihood. In R we use the glm function. In Python, **LogisticRegression** is used.

		Coefficient	Std. Error	Z-statistic	P-value
β_0	`Intercept`	-10.6513	0.3612	-29.5	< 0.0001
β_1	`balance`	0.0055	0.0002	24.9	< 0.0001

Maximum Likelihood

Methods of Maximizing Likelihood:
1- Gradient Methods (to be seen in the Lesson on Neural Networks)
2- Expectation Maximization

Are the coefficients significant?

- We perform a hypothesis test to see whether β_0 and β_1 are significantly different from zero.
- A Z test is used instead of a T test, but the p-value is interpreted similarly.

$$H_0: \beta_i = 0$$
$$H_A: \beta_i \neq 0$$

Are the coefficients significant?

- Because the estimates of β_i 's are Maximum Likelihood Estimates, one can show that assuming that the null hypothesis

$$H_0: \beta_i = \beta$$

is true, the quantity

Usually zero

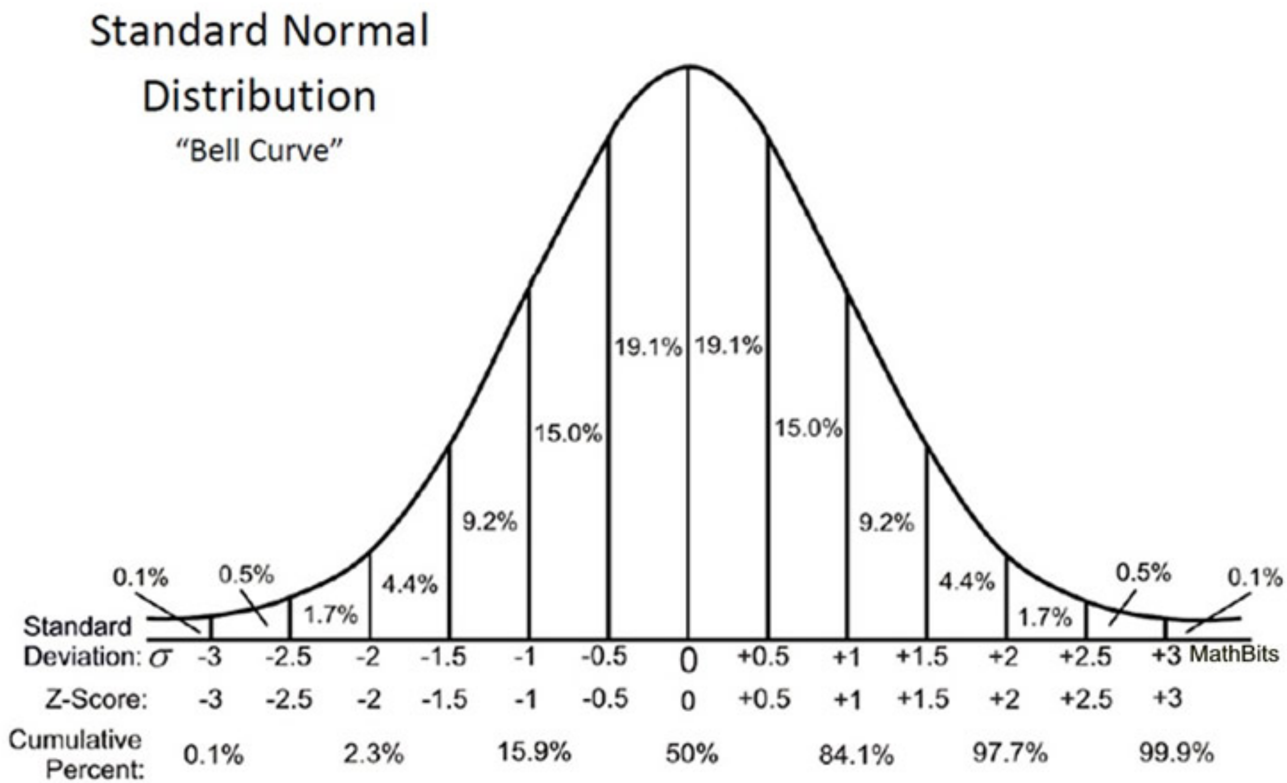
$$z = \frac{\hat{\beta}_i - \beta}{\mathrm{SE}(\hat{\beta}_i)}$$

follows a *standard normal distribution* $N(0,1)$.

Are the coefficients significant?

The quantity $$z = \frac{\hat{\beta}_i - \beta}{\mathrm{SE}(\hat{\beta}_i)}$$

follows a *standard normal distribution* $N(0,1)$.



Are the coefficients significant?

Note: This statement is true for any maximum likelihood estimate, provided that the number of samples is large:

The quantity

$$z = \frac{\hat{\beta}_i - \beta}{\mathrm{SE}(\hat{\beta}_i)}$$

follows a *standard normal distribution* $N(0,1)$ when the null hypothesis is true.

Are the coefficients significant?

The quantity $$z = \frac{\hat{\beta}_i - \beta}{\mathrm{SE}(\hat{\beta}_i)}$$

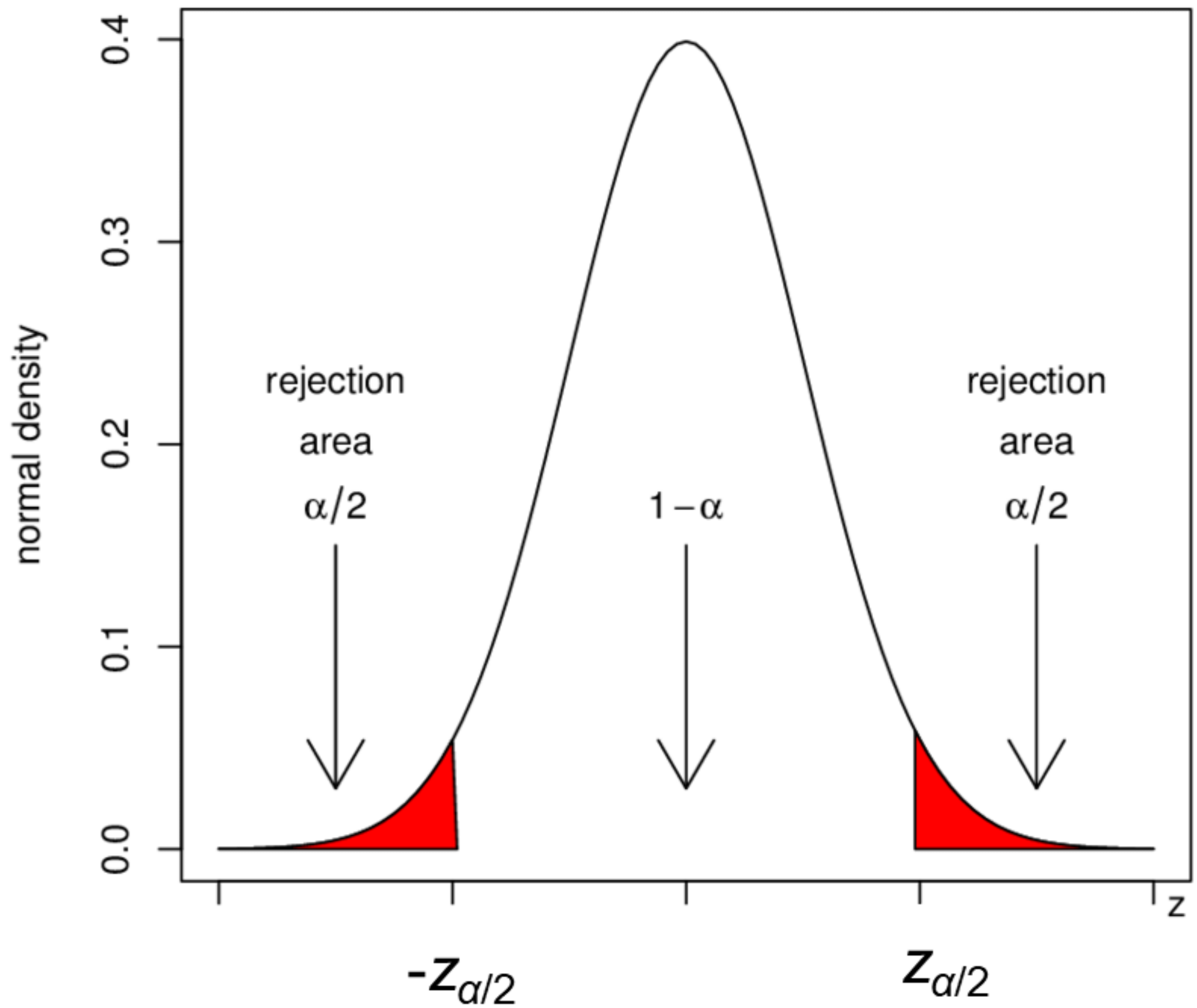
follows a *standard normal distribution* $N(0,1)$.

The rejection region is:

$$z > z_{\alpha/2}$$

$$z < -z_{\alpha/2}$$

Are the coefficients significant?



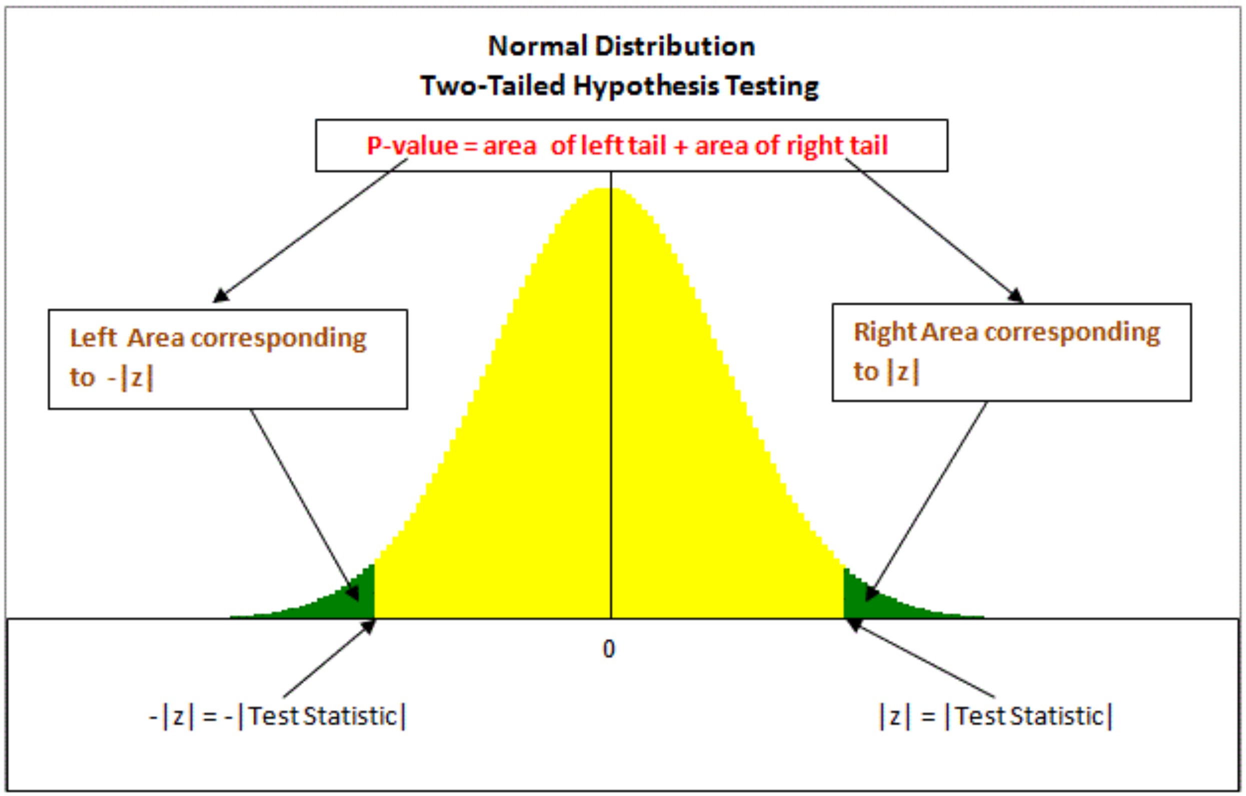
The rejection region is:

$z > z_{\alpha/2}$

$z < -z_{\alpha/2}$

Are the coefficients significant?

The p-value is the probability of observing something whose magnitude is bigger than |z|:



Are the coefficients significant?

- The p-value for balance is very small, and estimate of β_1 is positive, so we are sure that if the balance increase, then the probability of default will increase as well.

	Coefficient	Std. error	Z-statistic	P-value
`Intercept`	-10.6513	0.3612	-29.5	<0.0001
`balance`	0.0055	0.0002	24.9	<0.0001

Predictions

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

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

Predictions

With a balance of $2000?

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

Lets do it again, using student as the predictor.

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

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

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

Logistic regression with several variables

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

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

Logistic regression is a linear classifier?!

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

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

Logistic regression with several variables

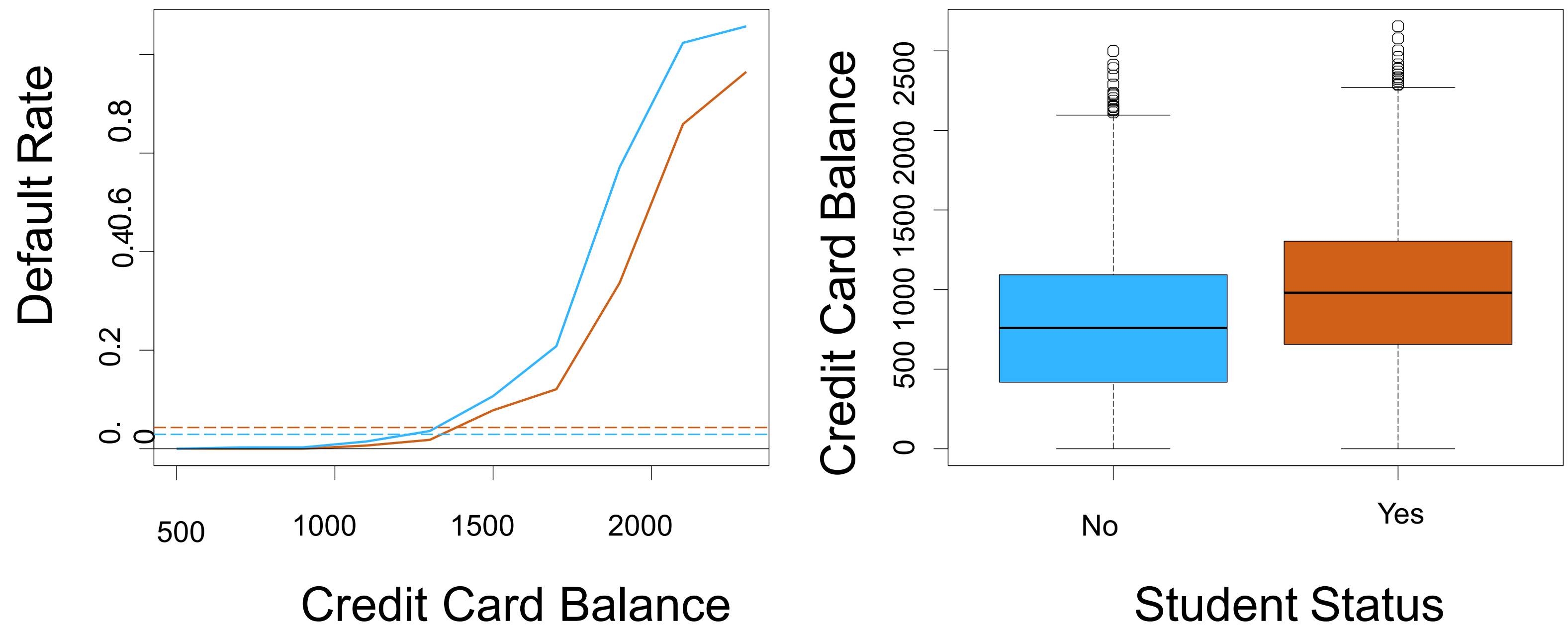
$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

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

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	-0.6468	0.2362	-2.74	0.0062

Why is coefficient for student negative, while it was positive before?

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.

To whom should credit be offered?

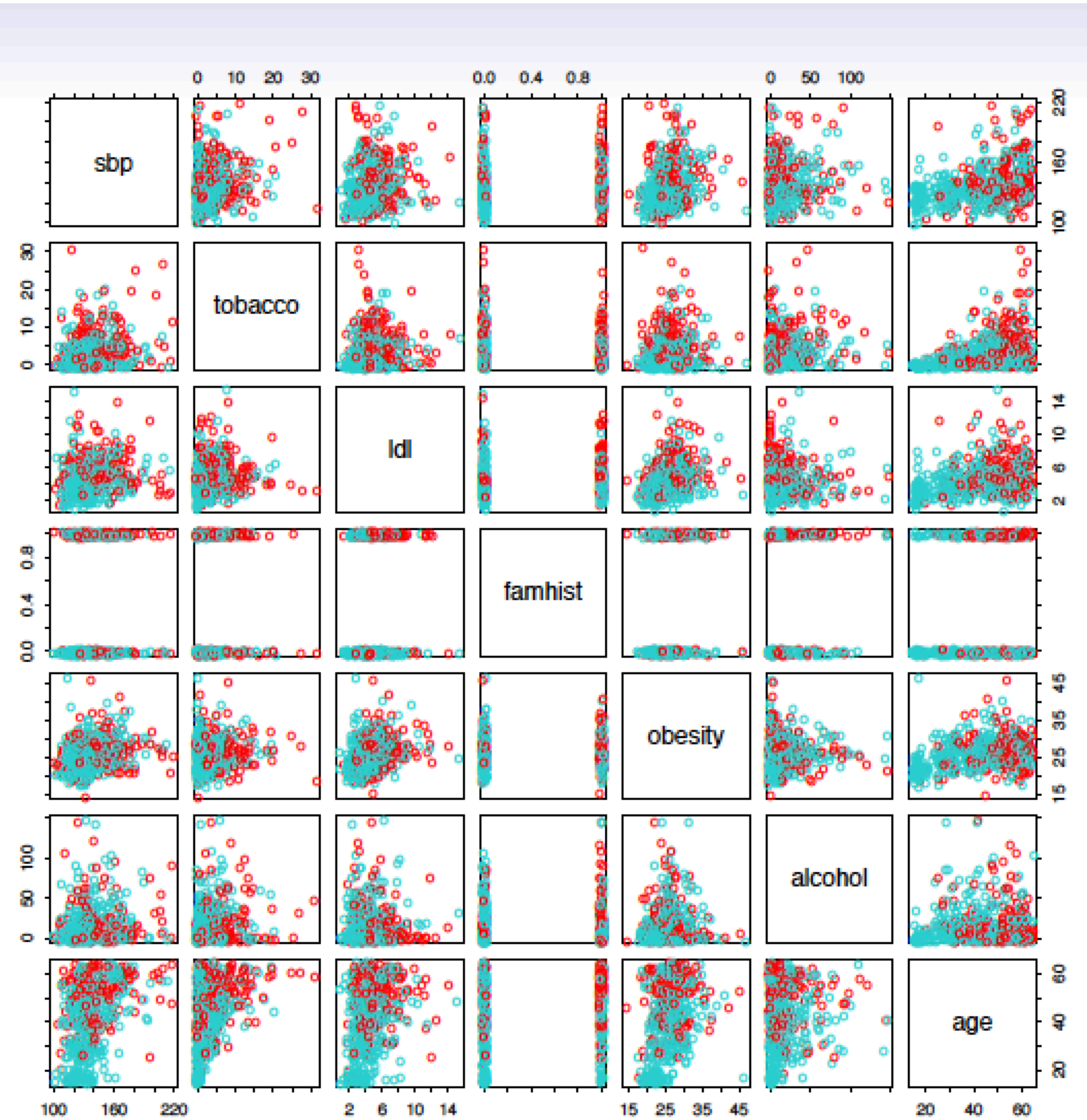
- A student is riskier than non students if no information about the credit card balance is available
- However, that student is less risky than a non student with the same credit card balance!
- Example of data-driven decision-making

Example: South African Heart Disease

- 160 cases of MI (myocardial infarction) and 302 controls (all male in age range 15-64), from Western Cape, South Africa in early 80s.
- Overall prevalence very high in this region: 5.1%.

Example: South African Heart Disease

- Measurements on seven predictors (risk factors), shown in scatterplot matrix.
- Goal is to identify relative strengths and directions of risk factors.
- This was part of an intervention study aimed at educating the public on healthier diets.



Scatterplot matrix of the *South African Heart Disease* data. The response is color coded — The cases (MI) are red, the controls turquoise.

famhist is a binary variable, with 1 indicating family history of MI.

```
> heartfit<-glm(chd~.,data=heart,family=binomial)
> summary(heartfit)

Call:
glm(formula = chd ~ ., family = binomial, data = heart)

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)   -4.1295997  0.9641558  -4.283 1.84e-05 ***
sbp            0.0057607  0.0056326   1.023  0.30643
tobacco        0.0795256  0.0262150   3.034  0.00242 **
ldl            0.1847793  0.0574115   3.219  0.00129 **
famhistPresent 0.9391855  0.2248691   4.177 2.96e-05 ***
obesity       -0.0345434  0.0291053  -1.187  0.23529
alcohol        0.0006065  0.0044550   0.136  0.89171
age            0.0425412  0.0101749   4.181 2.90e-05 ***

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 596.11  on 461  degrees of freedom
Residual deviance: 483.17  on 454  degrees of freedom
AIC: 499.17
```

Class Imbalance

- Intuitively, a dataset is imbalanced when members of certain class(es) are rare.
- The lack of observations of certain classes does not always imply their irrelevance.
- For example, in medical studies of rare diseases, the small number of infected patients (cases) conveys the most valuable information for diagnosis and treatments.

Types of Imbalance

- Formally, an imbalanced dataset exhibits one or more of the following properties:
  - **Marginal Imbalance**. A dataset is marginally imbalanced if one class is rare compared to the other class. In other words, $\Pr(Y=1)\approx 0$.

Types of Imbalance

Marginal Imbalance.

Such imbalance typically occurs in data sets for predicting click-through rates in online advertising, detecting fraud or diagnosing rare diseases.

Start 09.15.20

Types of Imbalance

- Formally, an imbalanced dataset exhibits one or more of the following properties:
  - ***Conditional Imbalance***. A dataset is conditionally imbalanced when it is easy to predict the correct labels in most cases. For example, if $X \in \{0,1\}$, the dataset is conditionally imbalanced if $\Pr(Y=1|X=0) \approx 0$ and $\Pr(Y=1|X=1) \approx 1$.

1) insufficient data

2) understand the rare $X=0, Y=1$

How to handle imbalance:

Subsampling (Down-sampling)

- Typically in such problems, the statistical noise is primarily driven by the number of representatives of the rare class
- We might hope to remedy the problem by subsampling the training set in a way that enriches for the rare class.

Subsampling (Down-sampling)

However, if the training set is randomly sampled to be balanced, the test set should be sampled to be more consistent with the state of nature and should reflect the imbalance so that honest estimates of future performance can be computed.

Don't mess with test set!

Upsampling

- Ling and Li (1998) provide an approach to up-sampling in which cases from the minority classes are sampled with replacement until each class has approximately the same size.
- Some minority class samples may show up in the training set with a fairly high frequency

Ling C, Li C (1998). "Data Mining for Direct Marketing: Problems and solutions." In "Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining," pp. 73–79.

Dealing w/ imbalance . . .

SMOTE

- The *synthetic minority over-sampling technique* (SMOTE) is a data sampling procedure that uses both up-sampling and down-sampling, depending on the class
- To up-sample for the minority class, SMOTE synthesizes new cases. To do this, a data point is randomly selected from the minority class and its K -nearest neighbors (KNNs) are determined.

Chawla N, Bowyer K, Hall L, Kegelmeyer W (2002). "SMOTE: Synthetic Minority Over–Sampling Technique." Journal of Artificial Intelligence Research,16 (1), 321–357.

only k neighbors of same class

SMOTE

- The new synthetic data point is a random combination of the predictors of the randomly selected data point and its neighbors.
- SMOTE can down-sample cases from the majority class via random sampling.
- Three operational parameters:
  - the amount of up-sampling,
  - the amount of down-sampling,
  - and the number of neighbors

in python

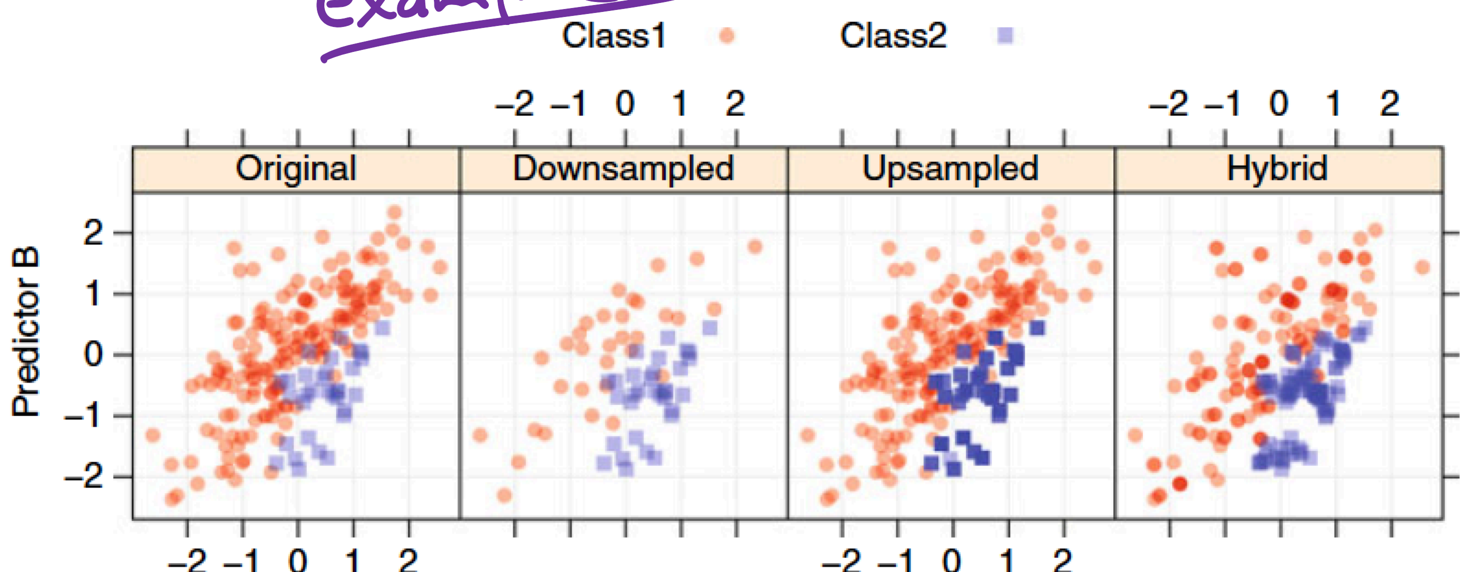
Chawla N, Bowyer K, Hall L, Kegelmeyer W (2002). “SMOTE: Synthetic Minority Over–Sampling Technique.” Journal of Artificial Intelligence Research,16 (1), 321–357.

Problem: synthetic data might choose the wrong class

SMOTE

From left to right: The original simulated data set and realizations of a down-sampled version, an up-sampled version, and sampling using SMOTE

example



less red

more blue

Smote

heart disease

Case-control sampling and logistic regression

downsampled majority

- In South African data, there are 160 cases, 302 controls — $\tilde{\pi} = 0.35$ are cases. Yet the prevalence of MI in this region is $\pi = 0.05$.
- With case-control samples, we can estimate the regression parameters β_j accurately (if our model is correct); the constant term β_0 is incorrect.

Case-control sampling and logistic regression

$\hat{\beta}_0$

- Can correct the estimated intercept by a simple transformation

artificial

$$\hat{\beta}_0^* = \hat{\beta}_0 + \log\frac{\pi}{1-\pi} - \log\frac{\tilde{\pi}}{1-\tilde{\pi}}$$

- Why?

$$\frac{e^{\hat{\beta}_0}}{1-e^{\hat{\beta}_0}} = \tilde{\pi} \quad \text{so} \quad \frac{e^{\hat{\beta}_0^*}}{1-e^{\hat{\beta}_0^*}} = \pi$$

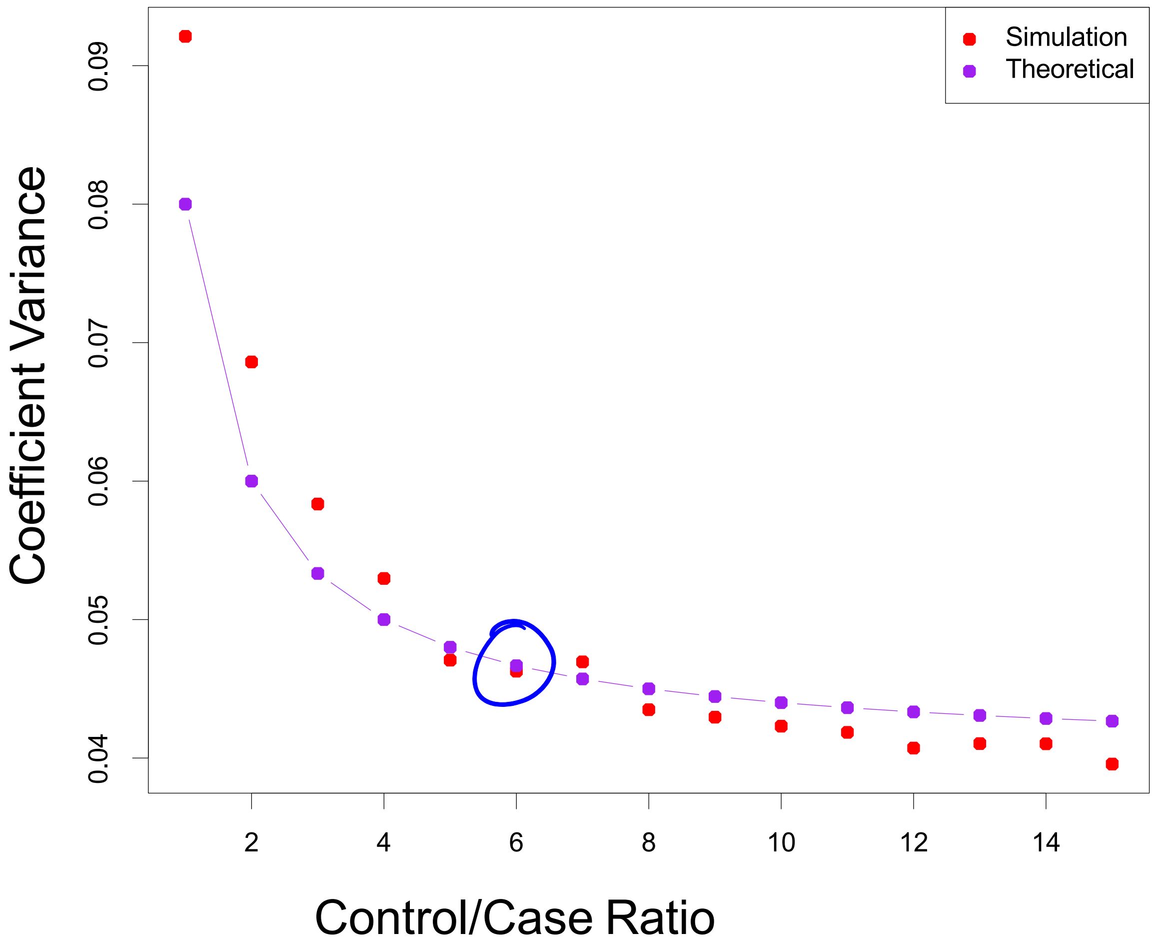
Case-control sampling and logistic regression

- Can correct the estimated intercept by a simple transformation

$$\hat{\beta}_0^* = \hat{\beta}_0 + \log \frac{\pi}{1-\pi} - \log \frac{\tilde{\pi}}{1-\tilde{\pi}}$$

- Often cases are rare and we take them all; up to five times that number of controls is sufficient.

Diminishing returns in unbalanced binary data



Sampling more controls than cases reduces the variance of the parameter estimates. But after a ratio of about 5 to 1 the variance reduction flattens out.

Control: new values

case: original values

Logistic regression with more than two classes

So far we have discussed logistic regression with two classes. It is easily generalized to more than two classes. One version (used in the R package glmnet) has the symmetric form

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

Logistic regression with more than two classes

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

Here there is a linear function for *each* class. (Examine the logit model)
Multiclass logistic regression is also referred to as *multinomial regression*.

Logistic regression with more than two classes

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

The students will recognize that some cancellation is possible, and only $K - 1$ linear functions are needed as in 2-class logistic regression:

(Bayesian) Discriminant Analysis

- Here the approach is to model the distribution of X in each of the classes separately, and then use *Bayes theorem* to flip things around and obtain $\Pr(Y|X)$.
- When we use normal (Gaussian) distributions for each class, this leads to linear or quadratic discriminant analysis.

Discriminant Analysis

- However, this approach is quite general, and other distributions can be used as well. We will focus on normal distributions.

Bayes theorem for classification

Thomas Bayes was a famous mathematician whose name represents a big subfield of statistical and probablistic modeling. Here we focus on a simple result, known as Bayes theorem:

likelihood function

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

Bayes theorem for classification

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

One writes this slightly differently for discriminant analysis:

Pr(y=k)

Pr(X=x | Y=k)

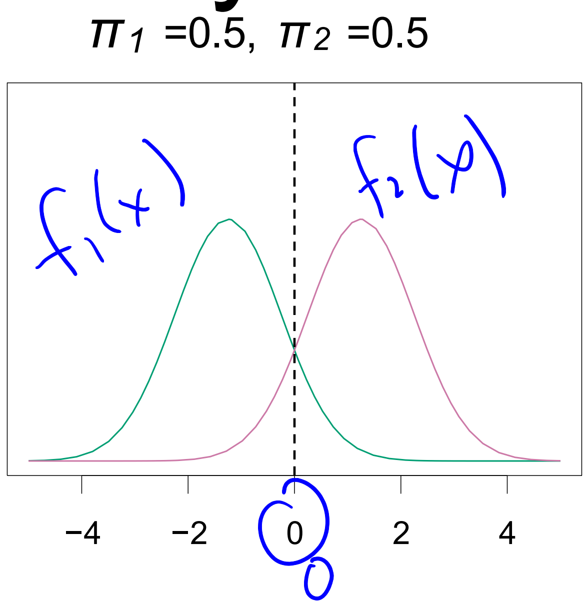
$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

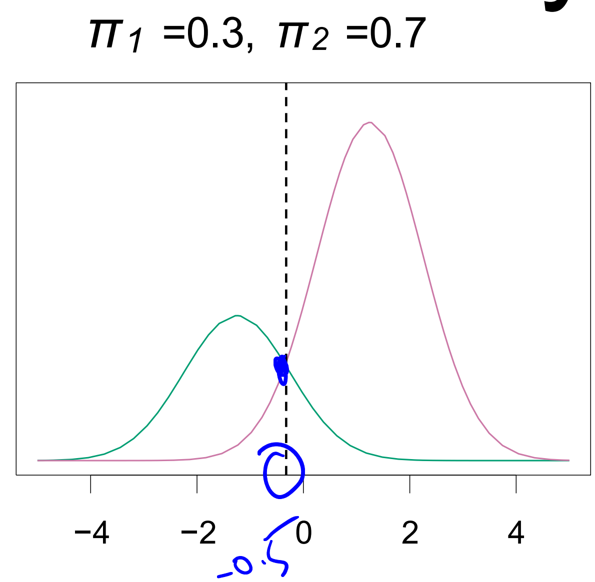
joint distribution of x in class k

where

- $f_k(x) = \Pr(X = x|Y = k)$ is the *density* for X in class k. Here we will use normal densities for these, separately in each class.
- $\pi_k = \Pr(Y = k)$ is the marginal or *prior* probability for class k.

Classify to the highest density





- We classify a new point according to which density is 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.

Why discriminant analysis?

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

Why discriminant analysis?

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

$n << p$ (predictors) $\rightarrow$ data is close to linearly separable

Why discriminant analysis?

- Linear discriminant analysis is popular when we have more than two response classes, because it also provides low-dimensional views of the data.

Linear Discriminant Analysis when $p = 1$

The Gaussian density has the form

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

Here μ_k is the mean, and $\sigma_k{}^2$ is the variance (in class k). We will assume that all the $\sigma_k = \sigma$ are the same.

Linear Discriminant Analysis when $p = 1$

Plugging this into Bayes formula, we get a rather complex expression for $p_k(x) = \Pr(Y = k|X = x)$:

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

Happily, there are simplifications and cancellations.

Linear Discriminant Analysis when $p = 1$

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

Happily, there are simplifications and cancellations.

Linear Discriminant Analysis when *p* = 1

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

Happily, there are simplifications and cancellations.

Discriminant functions

To classify at the value $X = x$, we need to see which of the $p_k(x)$ is 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.

Discriminant functions

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

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

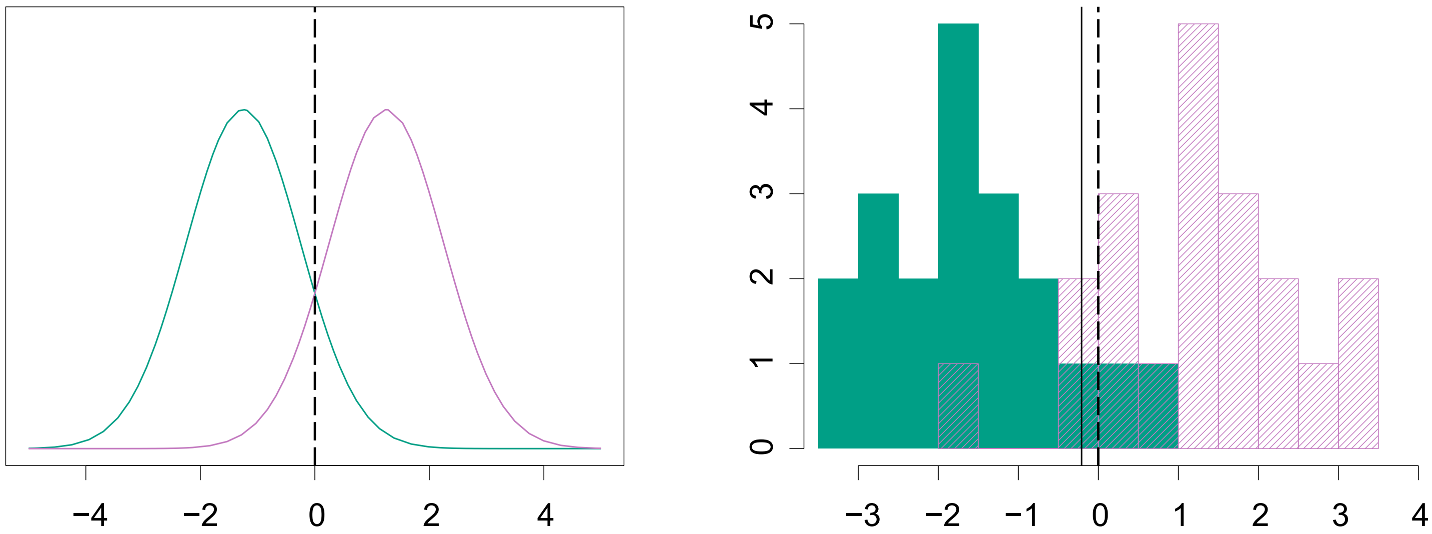
$$x = \frac{\mu_1 + \mu_2}{2}$$

(show this)

Discriminant functions

(show this)

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



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

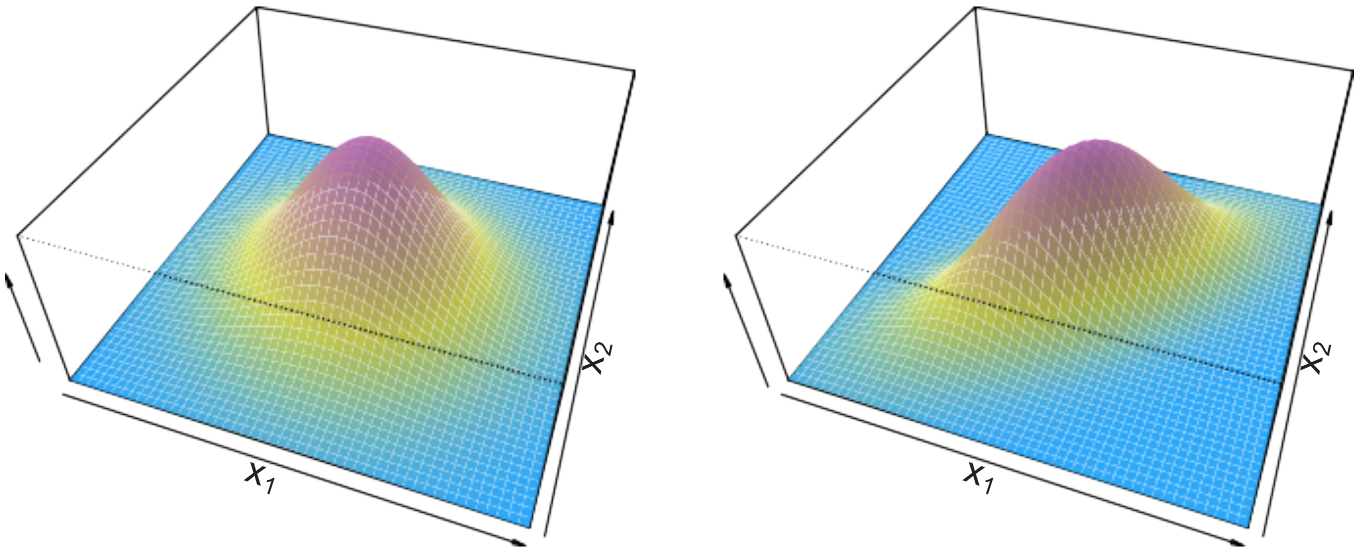
Typically we don't know these parameters; we just have the training data. In that case we simply estimate the parameters and plug them into the rule.

Estimating the parameters

$$\begin{aligned}
\hat{\pi}_k &= \frac{n_k}{n} \\
\hat{\mu}_k &= \frac{1}{n_k} \sum_{i:\, y_i = k} x_i \\
\hat{\sigma}^2 &= \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:\, y_i = k} (x_i - \hat{\mu}_k)^2 \\
&= \sum_{k=1}^{K} \frac{n_k - 1}{n - K} \cdot \hat{\sigma}_k^2
\end{aligned}$$

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 $p > 1$



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

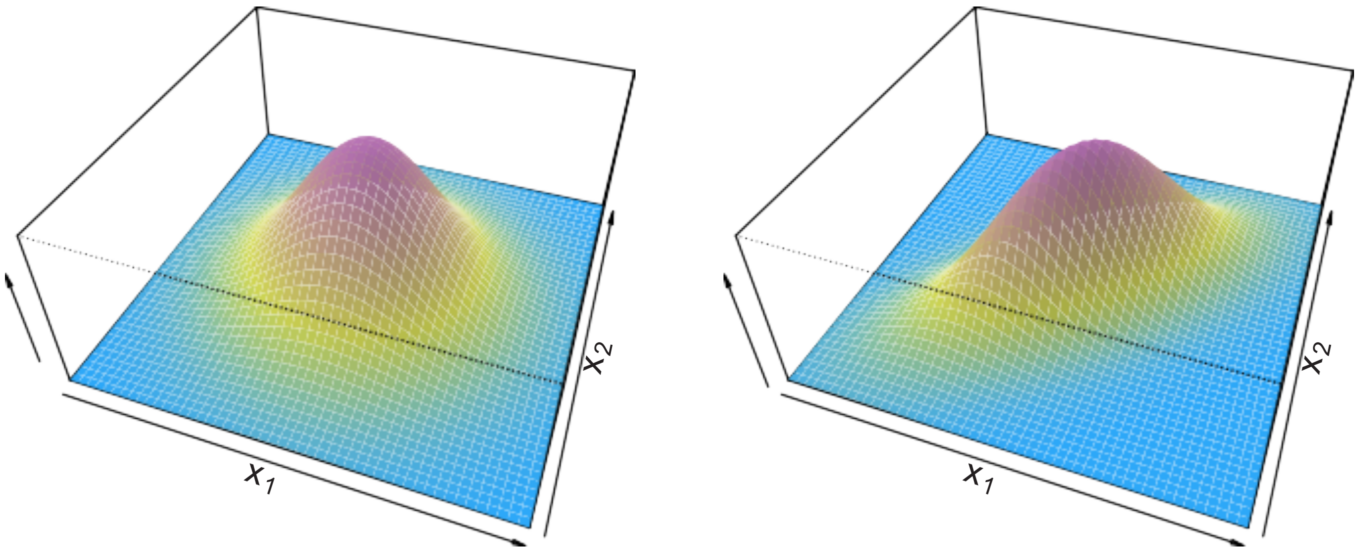
Discriminant function: $$\delta_k(x) = x^T\mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}^{-1}\mu_k + \log\pi_k$$

Linear Discriminant Analysis when *p* > 1

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

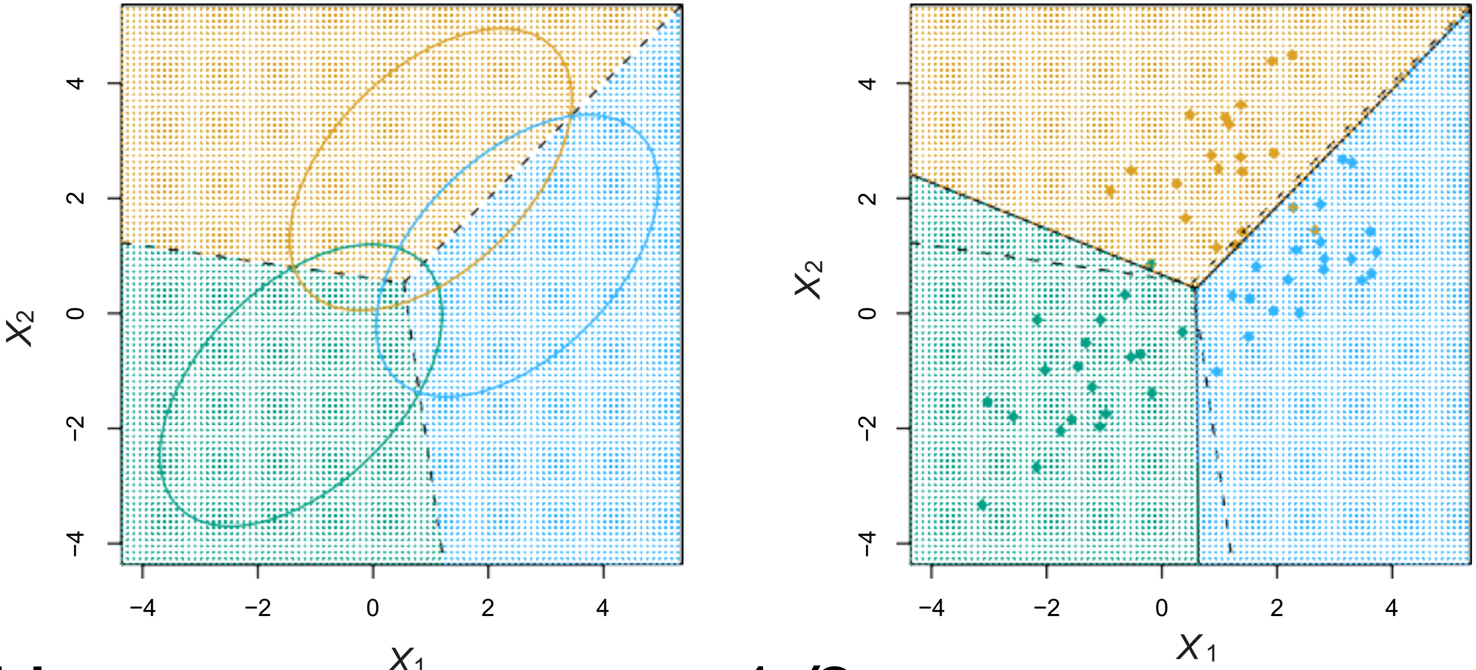
Discriminant function: $\delta_k(x) = x^T \mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T \mathbf{\Sigma}^{-1}\mu_k + \log \pi_k$

Linear Discriminant Analysis when $p > 1$



Despite its complex form,
$\delta_k(x) = c_{k0} + c_{k1}x_1 + c_{k2}x_2 + \ldots + c_{kp}x_p$
is 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*. Were they known, they would yield the fewest misclassification errors, among all possible classifiers.

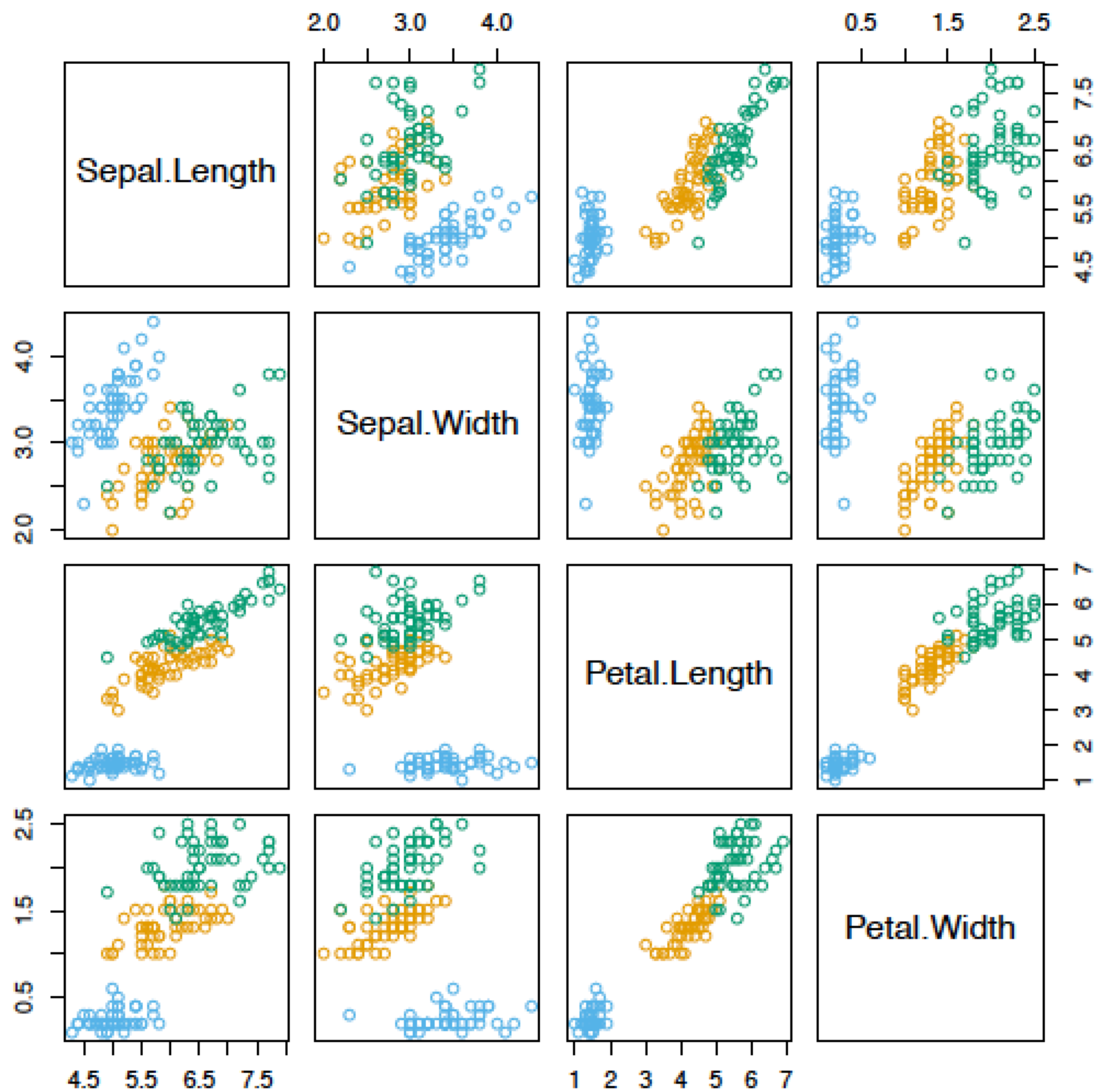
iris flower

Fisher's Iris Data

4 variables
3 species
50 samples/class

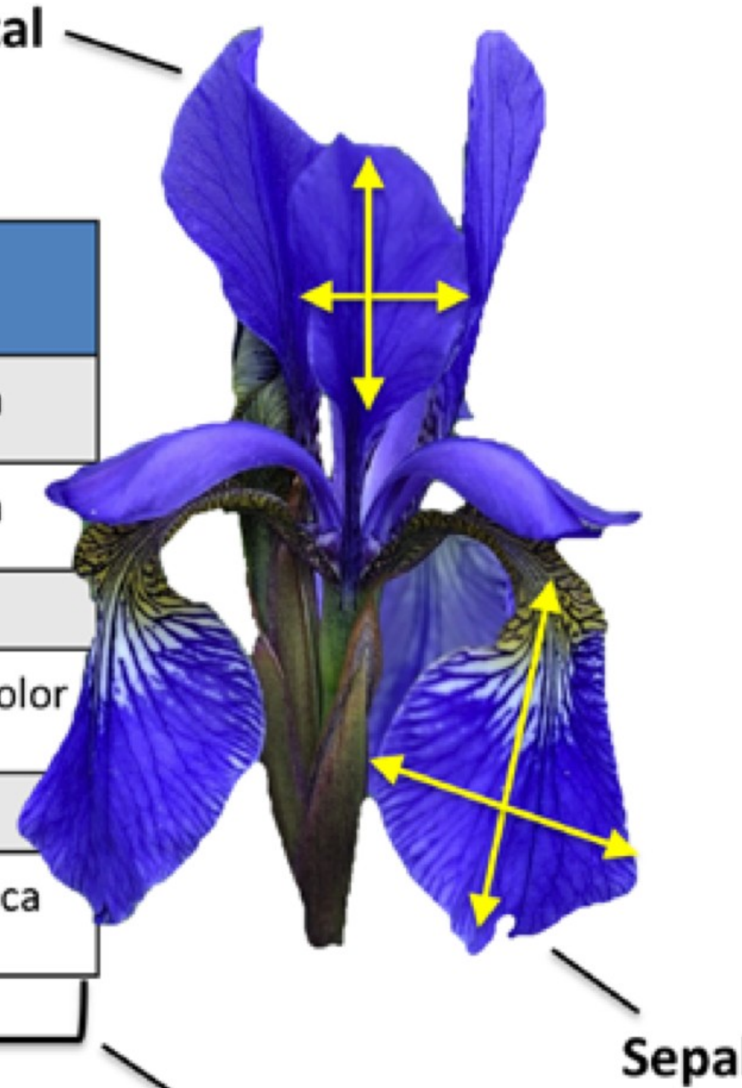
- Setosa
- Versicolor
- Virginica

LDA classifies all but 3 of the 150 training samples correctly.



Samples
(instances, observations)

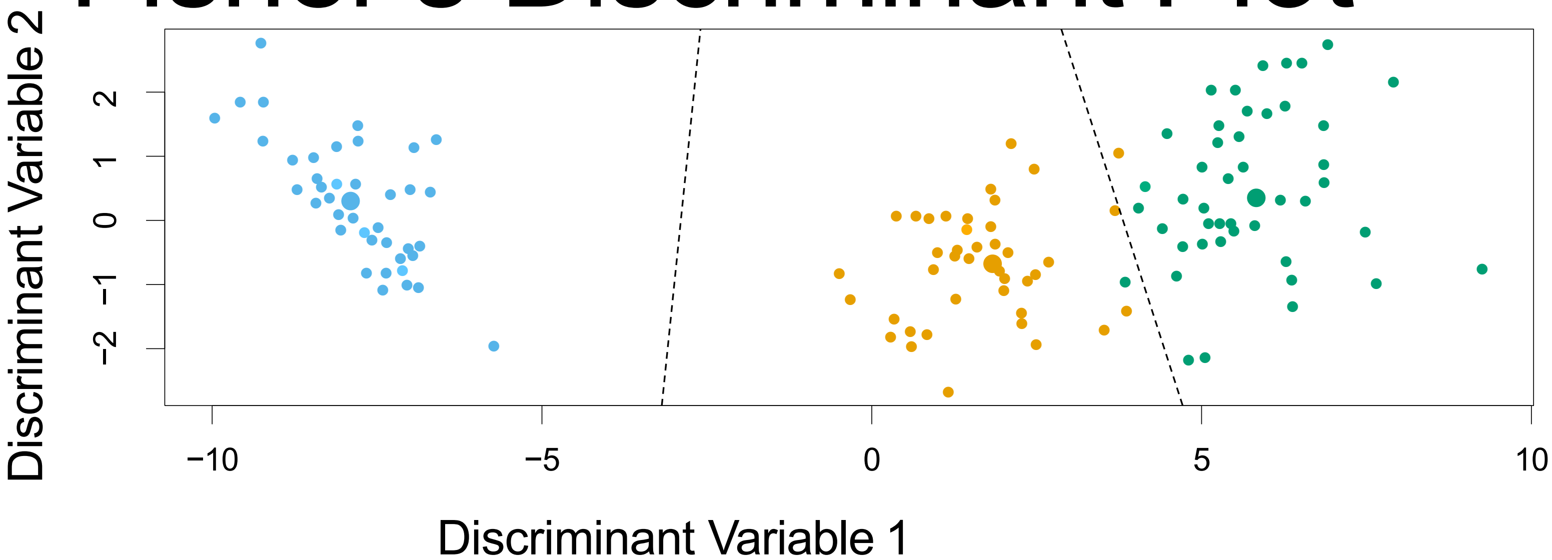
	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica



Class labels
(targets)

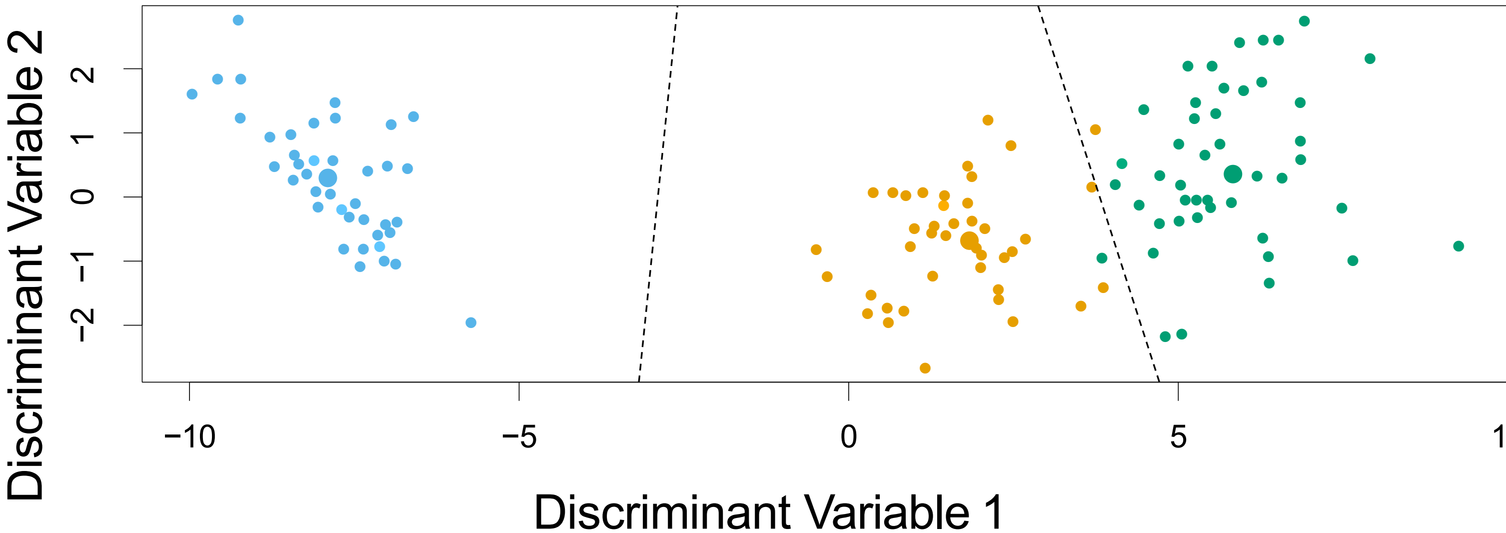
Features
(attributes, measurements, dimensions)

Fisher's Discriminant Plot



When there are K classes, linear discriminant analysis can be viewed exactly in a $K - 1$ dimensional plot.

Fisher's Discriminant Plot



Why? Because it essentially classifies to the closest centroid, and they span a $K - 1$ dimensional plane.

Even when $K > 3$, we can find the “best” 2-dimensional plane for vizualizing the discriminant rule.

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 $\hat{\Pr}(Y = k|X = x)$ is largest.

From $\delta_k(x)$ to probabilities

- So classifying to the largest $\hat{\delta}_k(x)$ amounts to classifying to the class for which $\hat{\Pr}(Y = k|X = x)$ is largest.
- When $K = 2$, we classify to class 2 if $\hat{\Pr}(Y = 2|X = x) \geq 0.5$, else to class 1.

LDA on Credit Data: Confusion Matrix

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

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

LDA on Credit Data

Some caveats:

- This is *training* error, and we may be overfitting. Not a big concern here since $n = 10000$ and $p = 4$!

LDA on Credit Data

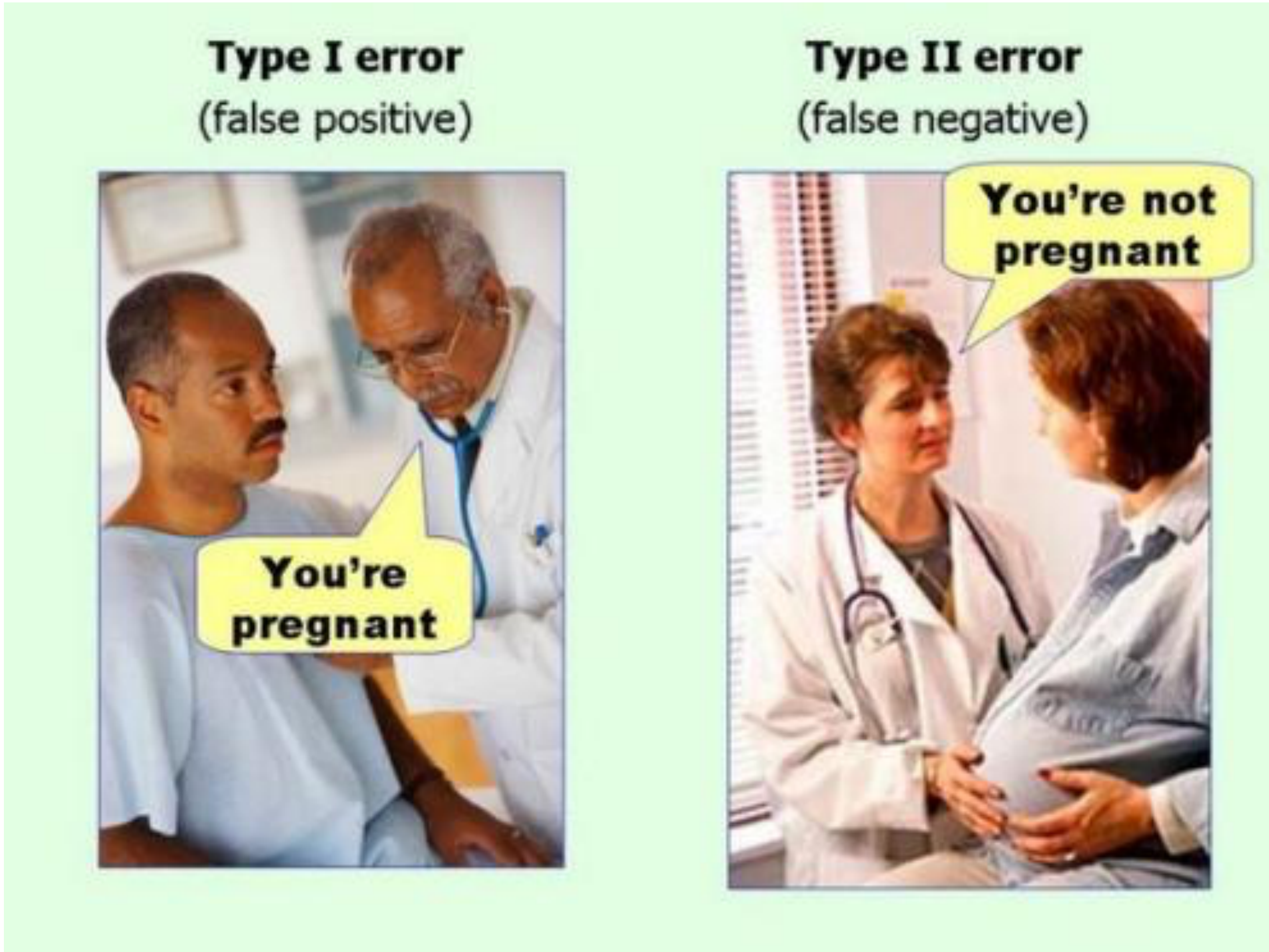
Some caveats:

- If we classified to the prior — always to class No in this case — we would make 333/10000 errors, or only 3.33%.
- Of the true No's, we make 23/9667 = 0.2% errors; of the true Yes's, we make 252/333 = 75.7% errors!
- This is because of class imbalance!

Types of errors

- False positive (type I error) rate: The fraction of negative examples that are classified as positive — 0.2% in example.
- False negative (type II error) rate: The fraction of positive examples that are classified as negative — 75.7% in example.

Types of errors



https://chemicalstatistician.files.wordpress.com/2014/05/pregnant.jpg?w=500

Measures for Different Types of Error

The sensitivity or recall of a binary classifier is the rate that the event of interest is predicted correctly for all samples having the event, or

TP/P=TP/(TP+FN)

It is also called the True Positive Rate (TPR) or Hit Rate (HR).

- What proportion of credit card defaults did we detect?

Measures for Different Types of Error

The specificity of a binary classifier is the rate that non-events are predicted correctly for all non-event samples or

TN/N=TN/(TN+FP)

It is also called the True Negative Rate (TNR).

- What proportion of credit card non-defaults did we detect?

Types of errors

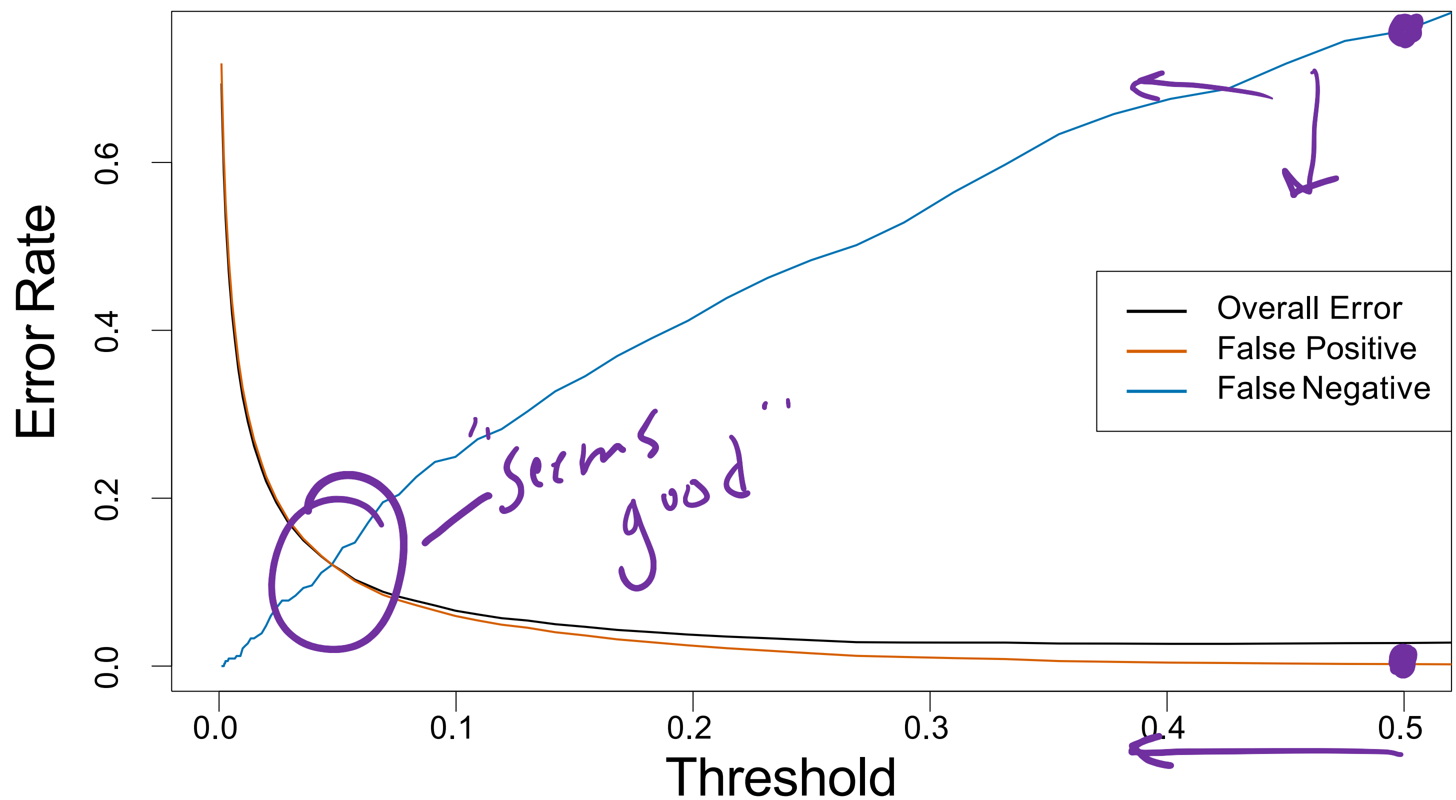
We produced the confusion matrix for credit data by classifying to class Yes if

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

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

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

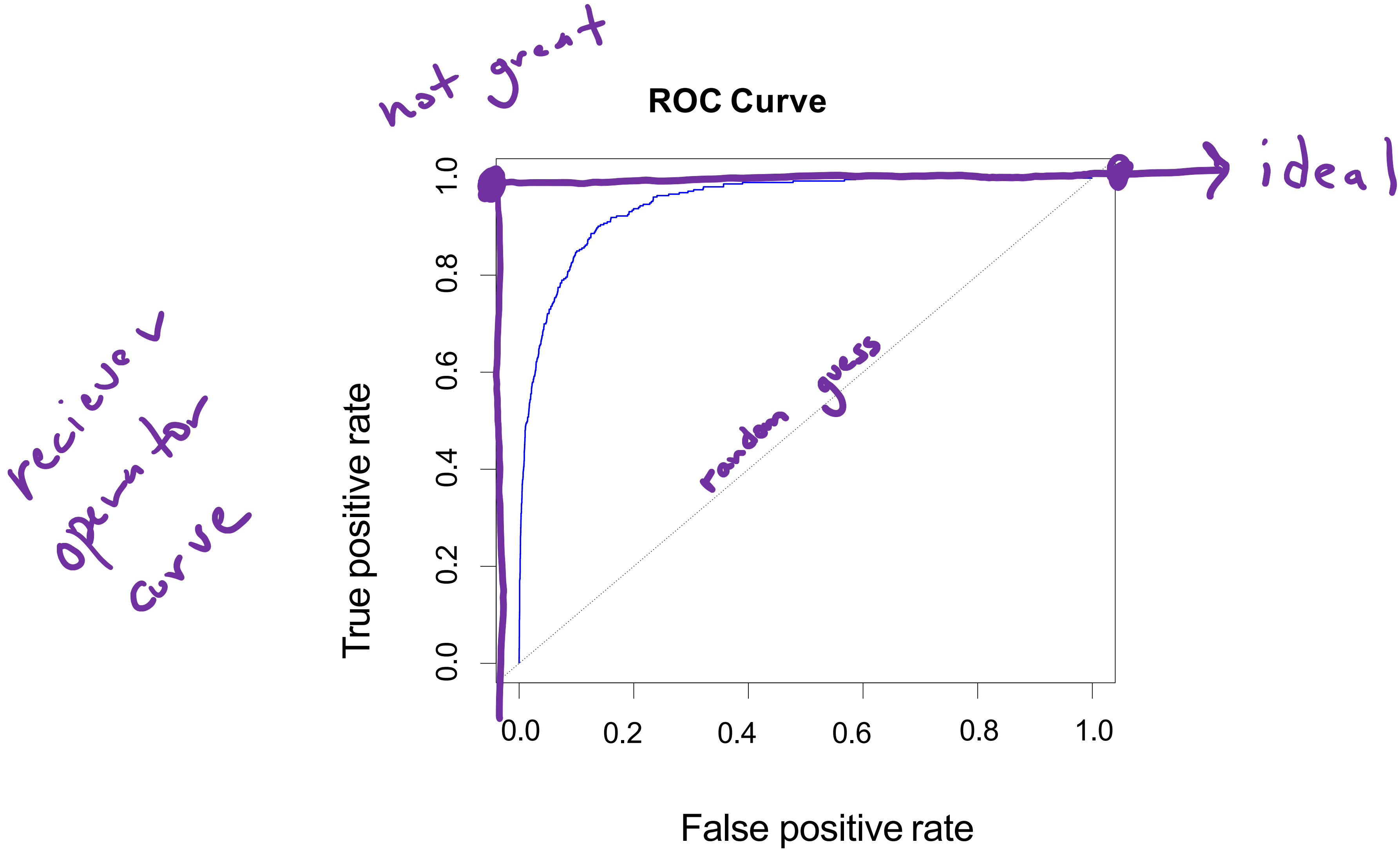
Varying the *threshold*



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

false positives are n bad

false negatives are bad



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.

Measures for Different Types of Error

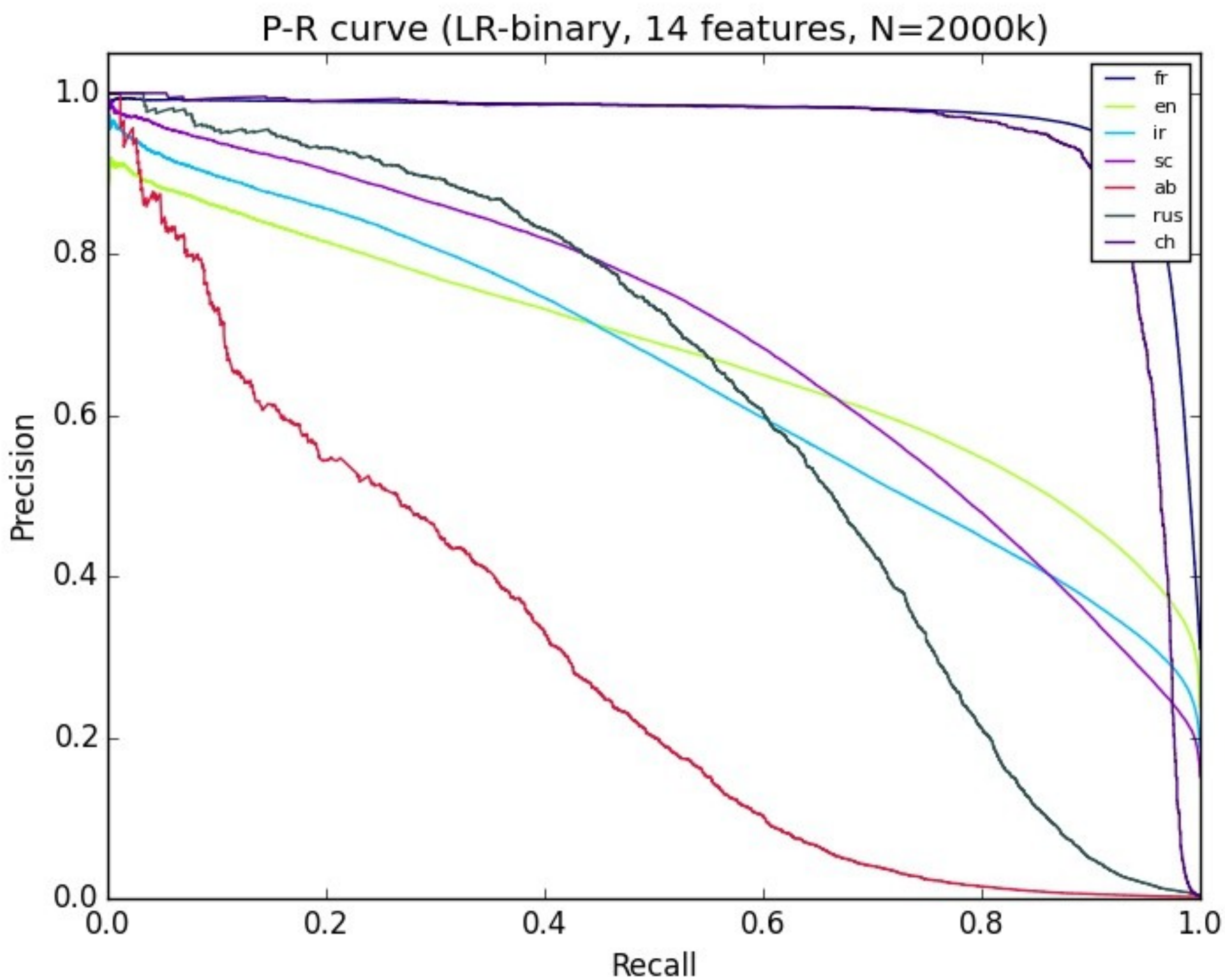
The precision or the positive predictive value of a binary classifier is the ratio of true positives with respected to all detected positives.

Precision= PPV=TP/(TP+FP)

Of those defaults that we detected, what proportion actually defaulted?

AUPRC (area under)

Precision-Recall Curve



From: https://stackoverflow.com/questions/33294574/good--roc--curve--but--poor--precision--recall--curve

Measures for Different Types of Error

The negative predictive value of a binary classifier is the ratio of true negatives with respect to all detected negatives.

NPV=TN/(TN+FN)

Of those non-defaults that we detected, what proportion actually did not default?

Measures for Different Types of Error

The F1 score or F measure of a binary classifier is the harmonic mean of precision and recall:

$$F = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Measures for Different Types of Error

F1 Score: seeks a balance between Precision (ratio of true positives to all detected positives) and Recall (True Positive Rate).
F1 Score might be better than accuracy if we seek a balance between Precision and Recall AND there is a class imbalance (large number of Actual Negatives).

Measures for Different Types of Error

F_β Score: seeks a *skewed* balance between Precision (ratio of true positives to all detected positives) and Recall (True Positive Rate).

$$F_\beta = \frac{\beta^2 + 1}{\frac{\beta^2}{recall} + \frac{1}{precision}}$$

$\beta \in [0, 1]$ → emphasis precision (spam detection)

$\beta \in [1, \infty]$ → emphasis recall (medical diagnosis)

Multiple Classes?

Precision/recall/F_β are specific to one class

- How to summarize for multiple classes?

Multiple Classes?

Two different ways of averaging:

- A **macro** average just averages the individually calculated scores of each class
  - Weights each *class* equally

$$PRE_{macro} = \frac{PRE_1 + \cdots + PRE_k}{k}$$

Multiple Classes?

Two different ways of averaging:

- A **micro** average calculates the metric by first pooling all instances of each class
  - Weights each *instance* equally

$$PRE_{micro} = \frac{TP_1 + \cdots + TP_k}{TP_1 + \cdots + TP_k + FP_1 + \cdots + FP_k}$$

Measures: Training or Testing?

Note that all of the measures used to evaluate types of error can be computed over both training and test sets.

Other forms of Discriminant Analysis

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

When $f_k(x)$ are Gaussian densities, with the same covariance matrix $\mathbf{\Sigma}$ in each class, this leads to linear discriminant analysis. 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*.

Other forms of Discriminant Analysis

- With Gaussians but different $\boldsymbol{\Sigma}_k$ in each class, we get *quadratic discriminant analysis*. Let's show it for p=1.

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

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

Other forms of Discriminant Analysis

- With Gaussians but different $\mathbf{\Sigma}_k$ in each class, we get *quadratic discriminant analysis*. Let's show it for p=1.

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

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

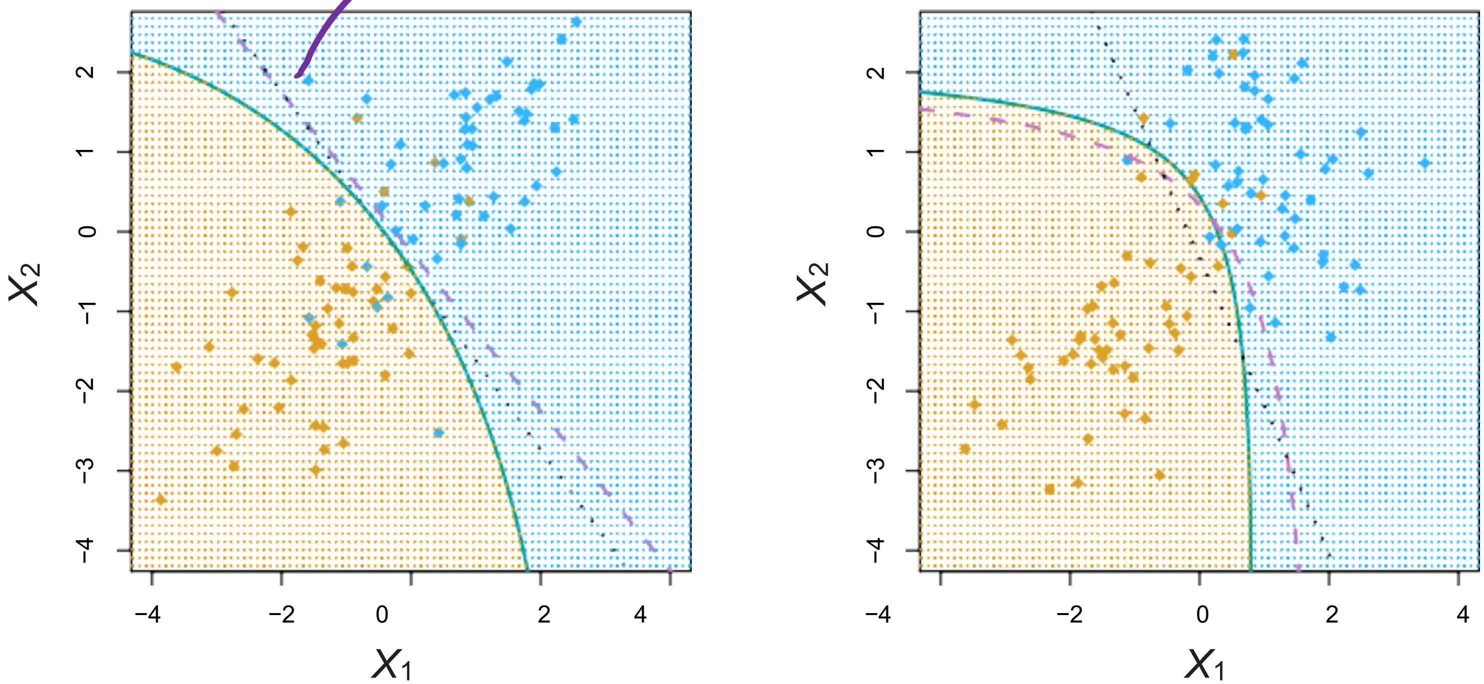
Other forms of Discriminant Analysis

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

- With $f_k(x) = \prod_{j=1}^{p} f_{jk}(x_j)$ (conditional independence model) in each class we get *naïve Bayes*. For Gaussian this means the $\mathbf{\Sigma}_k$ are diagonal.
- Many other forms, by proposing specific density models for $f_k(x)$, including nonparametric approaches.

theoretical & empirical LDA

Quadratic Discriminant Analysis



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

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

Logistic Regression versus LDA

For a two-class problem, one can show that for LDA .

$$\log\left(\frac{p_1(x)}{1-p_1(x)}\right) = \log\left(\frac{p_1(x)}{p_2(x)}\right) = c_0 + c_1 x_1 + \ldots + c_p x_p$$

So it has the same form as logistic regression.

The difference is in how the parameters are estimated.

Logistic Regression versus LDA

.

The difference is in how the parameters are estimated.

- Logistic regression uses the conditional likelihood based on $\Pr(Y|X)$ (known as *discriminative learning*).
- LDA uses the full likelihood based on $\Pr(X, Y)$ (known as *generative learning*).
- Despite these differences, in practice the results are often very similar.

Logistic regression can also fit quadratic boundaries like QDA, by explicitly including quadratic terms in the model.

Summary

- Logistic regression is very popular for classification, especially when $K = 2$.
- LDA is useful when n is small, or the classes are well separated, and Gaussian assumptions are reasonable. Also when $K > 2$.
- Naïve Bayes is useful when p is very large.
- See Section 4.5 for some comparisons of logistic regression, LDA and KNN.

Naïve Bayes

Assumes features are independent in each class.
Useful when p is large, and so multivariate methods like QDA and even LDA break down.

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

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

- 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, naïve Bayes often produces good classification results.

Bayesian: Classify to the highest density

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

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

- $f_k(x) = \Pr(X = x|Y = k)$ is the *density* for X in class k. Here we will use normal densities for these, separately in each class.
- $\pi_k = \Pr(Y = k)$ is the marginal or *prior* probability for class k.

Bayesian: Classify to the highest density

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

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

- We classify a new point according to which density is highest.
- When the priors are different, we take them into account as well, and compare

$$\pi_k f_k(x) = \Pr(Y = k) \Pr(X = x|Y = k) .$$

Bayesian classifiers

Easy to estimate priors π_k from data. (*How?*)
The real challenge: how to estimate
$f_k(x) = \Pr(X = x | Y = k)$
$= \Pr(X = (x_1, x_2, \ldots, x_p) | Y = k)$

Bayesian classifiers

How to estimate
$f_k(x_{1,}x_{2,}\ldots,x_p) = \Pr(X = (x_{1,}x_{2,}\ldots,x_p) | Y = k)$

- In the general case, where the attributes x_j have dependencies, this requires estimating the full joint distribution $f_k(x_{1,}x_{2,}\ldots,x_p)$ for each class k in C.
- There is almost never enough data to confidently make such estimates.

Naïve Bayes classifier

Assume independence among attributes x_j when class is given:

$$f_k(x_1, x_2, \ldots, x_p) = f_k(x_1)\ f_k(x_2)\ \ldots\ f_k(x_n)$$

Usually straightforward and practical to estimate $f_k(x_\mathrm{i}) = \Pr(X_i = x_i | Y = k)$ for all x_j and k.

New sample is classified to Y=k if

$$\pi_k \Pi_i\ f_k(x_\mathrm{i}) \text{ is maximal.}$$

How to estimate $f_k(x_i) = \Pr(X_i = x_i | Y = k)$ from data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

start 9/22

Class priors:
$\pi\hat{}_k = N_k / N$

π (No) = 7/10

π (Yes) = 3/10

For discrete attributes:

$\Pr\hat{}\ (X_i = x_i \mid Y = k) = | x_{ik} | / N_k$

where $| x_{ik} |$ is number of instances in class k having attribute value x_i

Examples:

$\Pr\hat{}$(Status = Married | No) = 4/7

$\Pr\hat{}$ (Refund = Yes | Yes) = 0

How to estimate $f_k(x_i)$ from data?

For continuous attributes:

Discretize the range into bins

replace with an ordinal attribute

Two-way split: $(x_i < v)$ or $(x_i > v)$

replace with a binary attribute

Probability density estimation:

- assume attribute follows some standard parametric probability distribution (usually a Gaussian)
- use data to estimate parameters of distribution (e.g. mean and variance)
- once distribution is known, can use it to estimate the conditional probability $\Pr(X_i = x_i | Y = k)$

How to estimate $f_k(x_i)$ from data?

Tid	Refund	Marital Status	Taxable Income	Evade
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Gaussian distribution:

$$f_k(x_i) = \Pr(X_i = x_i | Y = k) = \frac{1}{\sqrt{2\pi\sigma_{ik}}} e^{-\frac{(x_i-\mu_{ik})^2}{2\sigma_{ik}}}$$

one for each (x_i, k) pair

For (Income | Class = No):

sample mean = 110

sample variance = 2975

$$\Pr(\text{Income} = 120 | \text{No}) = \frac{1}{\sqrt{2\pi}(54.54)} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072$$

Naïve Bayes classifier

Problem: if one of the conditional probabilities is zero, then the entire expression becomes zero.
This is a significant practical problem, especially when training samples are limited.
Ways to improve probability estimation:

$$\text{Original}: p(x_j \mid C_i) = \frac{N_{ji}}{N_i}$$

$$\text{Laplace}: p(x_j \mid C_i) = \frac{N_{ji} + 1}{N_i + c}$$

$$\text{m-estimate}: p(x_j \mid C_i) = \frac{N_{ji} + mp}{N_i + m}$$

c: number of levels variable x_j can take.

p: prior probability

m: parameter

Summary of Naïve Bayes

- Robust to isolated noise samples.
- Handles missing values by ignoring the sample during probability estimate calculations.
- Robust to irrelevant attributes.
- *NOT* robust to redundant attributes.
  - Independence assumption does not hold in this case.
  - Use other techniques such as Bayesian Belief Networks (BBN).

Appendix:

Text Classification

Is this spam?

Subject: **Important notice!**
From: Stanford University <newsforum@stanford.edu>
Date: October 28, 2011 12:34:16 PM PDT
To: undisclosed-recipients:;

Greats News!

You can now access the latest news by using the link below to login to Stanford University News Forum.

http://www.123contactform.com/contact-form-StanfordNew1-236335.html

Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information
about the new services.

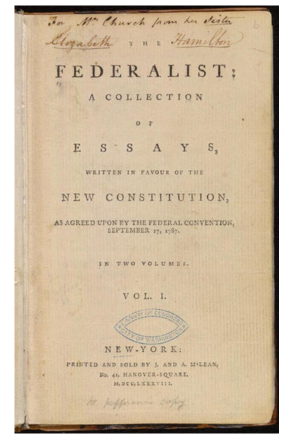
© Stanford University. All Rights Reserved.

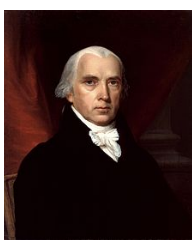
Who wrote which Federalist papers?

1787-8: anonymous essays try to convince New York to ratify U.S Constitution: Jay, Madison, Hamilton.

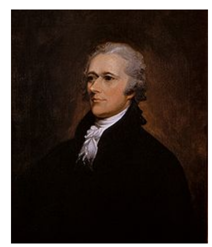
Authorship of 12 of the letters in dispute

1963: solved by Mosteller and Wallace using Bayesian methods





James Madison



Alexander Hamilton

Male or female author?

1. By 1925 present-day Vietnam was divided into three parts under French colonial rule. The southern region embracing Saigon and the Mekong delta was the colony of Cochin-China; the central area with its imperial capital at Hue was the protectorate of Annam…
2. Clara never failed to be astonished by the extraordinary felicity of her own name. She found it hard to trust herself to the mercy of fate, which had managed over the years to convert her greatest shame into one of her greatest assets…

S. Argamon, M. Koppel, J. Fine, A. R. Shimoni, 2003. “Gender, Genre, and Writing Style in Formal Written Texts,” Text, volume 23, number 3, pp. 321–346

Positive or negative movie review?

unbelievably disappointing

Full of zany characters and richly applied satire, and some great plot twists

this is the greatest screwball comedy ever filmed

It was pathetic. The worst part about it was the boxing scenes.

What is the subject of this article?

MEDLINE Article

?

MeSH Subject Category Hierarchy

Antogonists and Inhibitors
Blood Supply
Chemistry
Drug Therapy
Embryology
Epidemiology
…

Text Classification

- Assigning subject categories, topics, or genres
- Spam detection
- Authorship identification
- Age/gender identification
- Language Identification
- Sentiment analysis
- …

Classification Methods: Hand-coded rules

Rules based on combinations of words or other features

spam: black-list-address OR (“dollars” AND“have been selected”)

Accuracy can be high

If rules carefully refined by expert

But building and maintaining these rules is expensive

Classification Methods: Supervised Machine Learning

Any kind of classifier

- Naïve Bayes
- Logistic regression
- Support-vector machines
- k-Nearest Neighbors

…

The bag of words representation

$\gamma($

I love this movie! It's sweet,
but with satirical humor. The
dialogue is great and the
adventure scenes are fun… It
manages to be whimsical and
romantic while laughing at the
conventions of the fairy tale
genre. I would recommend it to
just about anyone. I've seen
it several times, and I'm
always happy to see it again
whenever I have a friend who
hasn't seen it yet.

$) = c$

The bag of words representation

$\gamma($

I **love** this movie! It's **sweet,** but with **satirical** humor. The dialogue is **great** and the adventure scenes are **fun…** It manages to be **whimsical** and **romantic** while **laughing** at the conventions of the fairy tale genre. I would **recommend** it to just about anyone. I've seen it **several** times, and I'm always **happy** to see it **again** whenever I have a friend who hasn't seen it yet.

$)=c$

The bag of words representation: using a subset of words

$\gamma($

```
x love xxxxxxxxxxxxxxxx sweet
xxxxxxx satirical xxxxxxxxxx
xxxxxxxxxxx great xxxxxxx
xxxxxxxxxxxxxxxxxxx fun  xxxx
xxxxxxxxxxxxx whimsical xxxx
romantic xxxx  laughing
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxx recommend xxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xx several xxxxxxxxxxxxxxxxx
xxxxx  happy xxxxxxxxx again
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxx
```

$)=c$

The bag of words representation

$\gamma($

great	2
love	2
recommend	1
laugh	1
happy	1
...	...

$)=c$

Bag of Words

The bag-of-words model is a simplifying representation used in natural language processing and information retrieval (IR).
In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity.

Example

(1) John likes to watch movies. Mary likes movies too.
(2) John also likes to watch football games.

Parsed Model:

```
"John","likes","to","watch","movies","M
ary","likes","movies","too"

"John","also","likes","to","watch","foo
tball","games"
```

Example

(1) John likes to watch movies. Mary likes movies too.
(2) John also likes to watch football games.

Bag of Words Model:

```
BoW1 =
{"John":1,"likes":2,"to":1,"watch":1,"movies":2,"Mary":1,"too":1};
BoW2 =
{"John":1,"also":1,"likes":1,"to":1,"watch":1,"football":1,"games":1};
```

Example

(1) John likes to watch movies. Mary likes movies too.
John also likes to watch football games.

Combined Documents:

```
BoW3 =
{"John":2,"likes":3,"to":2,"watch":2,"mov
ies":2,"Mary":1,"too":1,"also":1,"footbal
l":1,"games":1};
```

Application

The Bag-of-words model is mainly used as a tool of feature generation.
After transforming the text into a "bag of words", we calculate various measures to characterize the text.

The most common type of characteristics, or features calculated from the Bag-of-words model is term frequency, namely, the number of times a term appears in the text.

Application

For the example above, we can construct the following two lists to record the term frequencies of all the distinct words (BoW1 and BoW2 ordered as in BoW3):

```
(1)[1, 2, 1, 1, 2, 1, 1, 0, 0, 0]
(2)[1, 1, 1, 1, 0, 0, 0, 1, 1, 1]
```

Term Frequency

The simplest choice to calculate Term Frequency to measure the importance of a word in a document is to use the *raw count* of a term in a document.

tf(t,d)=number of occurrences of term t in document d

Term Frequency

Other choices:

Boolean Frequencies tf(t,d)=1 if t occurs in d otherwise 0

Adjustment for document length:

tf(t,d)=[number of occurrences of term t in document d]/number of words in document d

Inverse Document Frequency

A measure of information each word provides, i.e. if a word is common or rare among documents

IDF(t,D) = log(total number of documents in the corpus D/number of documents where the term t appears)

TFIDF

A combined measure for each word is the TF-IDF which is a product of TF and IDF:

$$TFIDF(t,d,D)=TF(t,d).IDF(t,D)$$

Therefore, each document can be represented as a vector representing the bag of words, but instead of simple frequencies, the TFIDF of each word can be given in the vector.

Naïve Bayes and Language Modeling

- Naïve bayes classifiers can use any sort of feature
  - URL, email address, dictionaries, network features
- But if, as in the previous slides
  - We use **only** word features
  - we use **all** of the words in the text (not a subset)
- Then
  - Naïve bayes has an important similarity to language modeling.

Each class = a unigram language model

Assigning each word: P(word | c)
Assigning each sentence: $P(s|c)=\Pi\ P(word|c)$

Class *pos*

0.1	I
0.1	love
0.01	this
0.05	fun
0.1	film
…	

I	love	this	fun	film
0.1	0.1	.05	0.01	0.1

P(s | pos) = 0.0000005

Naïve Bayes as a Language Model

Sec.13.2.1

Which class assigns the higher probability to s?

Model pos

0.1	I
0.1	love
0.01	this
0.05	fun
0.1	film

Model neg

0.2	I
0.001	love
0.01	this
0.005	fun
0.1	film

I	love	this	fun	film
0.1	0.1	0.01	0.05	0.1
0.2	0.001	0.01	0.005	0.1

P(s|pos) > P(s|neg)

Appendix

How to Deal with Missing Values?

From:
https://machinelearningmastery.com/handle-missing-data-python/

A Simple Strategy

- The simplest strategy for handling missing data is to remove samples that contain a missing value (feature).

- Advantage:
  - Simplicity
- Disadvantage:
  - Missing a lot of information
  - Works poorly if the percentage of missing values is high (say 30%), compared to the whole dataset

Data Imputation

- Imputing refers to using a model to replace missing values.
  - A constant value that has meaning within the domain, such as 0, distinct from all other values. This works for categorical features.

Data Imputation

- Imputing refers to using a model to replace missing values.
    - A feature value from another randomly selected observation.

Data Imputation

- Imputing refers to using a model to replace missing values.
  - A statistics such as mean, median or mode value of the feature/column.

Data Imputation

- Imputing refers to using a model to replace missing values.
  - A value estimated by another predictive model:
    - Treat the missing value as the output of your predictive model
    - Predict it based on other data points that do not have missing values
      - Examples: KNN regression and classification and Linear Regression

Data Imputation

- Imputing refers to using a model to replace missing values.
  - A value estimated by another predictive model:
    - What if a lot of data has missing values?
    - There will not be enough data without missing values for prediction of other missing values
    - An iterative method based on Expectation Maximization can be used

Data Imputation

- Imputing refers to using a model to replace missing values.
  - A value estimated by another predictive model:
    - An iterative method based on Expectation Maximization can be used:
    - First, impute all missing values randomly or using mean or median
    - Predict missing values using other data points
    - Update all missing values and iterate

Algorithms That Support Missing Values

- k-Nearest Neighbors can ignore a feature from a distance measure when a value is missing.
  - Calculate distance measure without the missing feature

Algorithms That Support Missing Values

- There are also algorithms that can use the missing value as a unique and different value (say NaN) when building the predictive model, such as classification and regression trees.

Algorithms That Support Missing Values

- The scikit-learn implementations of decision trees and k-NN are not robust to missing values. Although it is being considered.
- This remains as an option if you consider using another algorithm implementation (such as xgboost) or developing your own implementation.