START Tuesday 1/27/22
(Week 2/Lecture 4)

Classification[1]

[1] Based on materials in ISLR Ch 4, Ch 9 ☆

Classification

- Qualitative variables take values in an unordered set $\mathcal{C}$, such as:

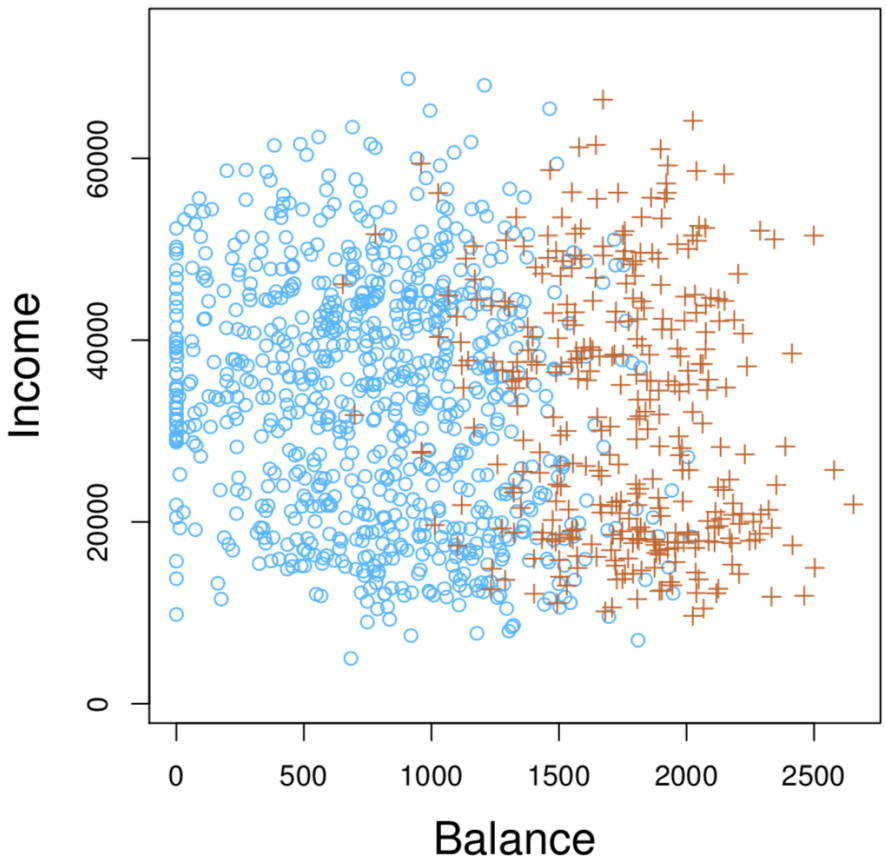
$$\text{eye color} \in \{\text{brown}, \text{blue}, \text{green}\},$$
$$\text{email} \in \{\text{spam}, \text{ham}\}.$$

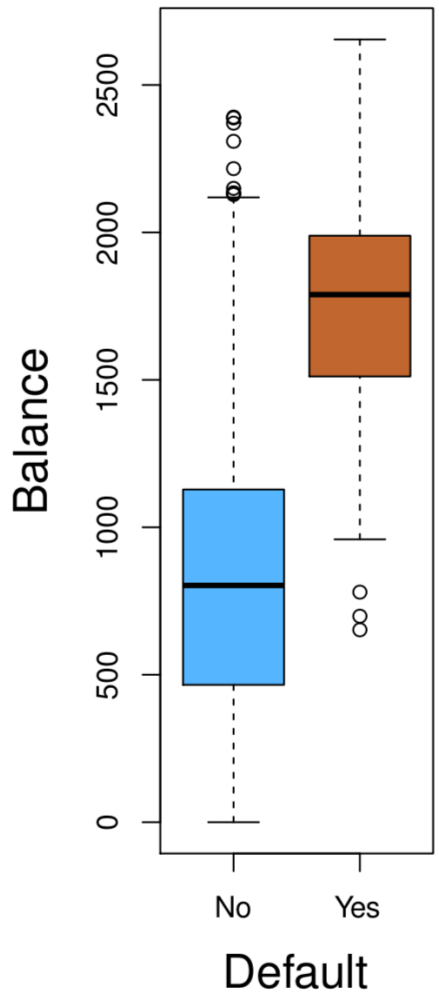
- Given a feature vector X and a qualitative response Y taking values in the set $\mathcal{C}$, the classification task is to build a function $C(X)$ that takes as input the feature vector X and predicts its value for Y ; i.e. $C(X) \in \mathcal{C}$. — Doesn't give you associated probabilities —
- Often we are more interested in estimating the **probabilities** that X belongs to each category in $\mathcal{C}$.

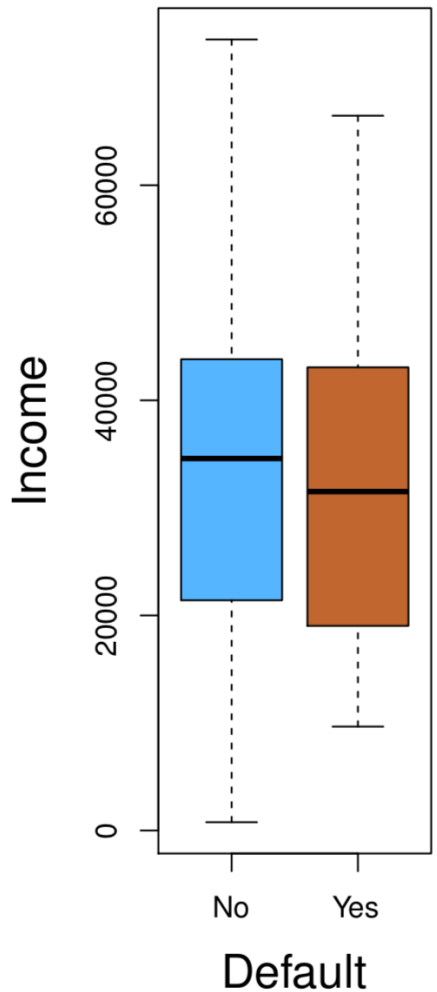
For example, it is more valuable to have an estimate of the probability that an insurance claim is fraudulent, than a classification of fraudulent or not.

Example: Credit Card Defualt

- As balance increases P(Default) ↑
- income is independent of default







Can we use Linear Regression?

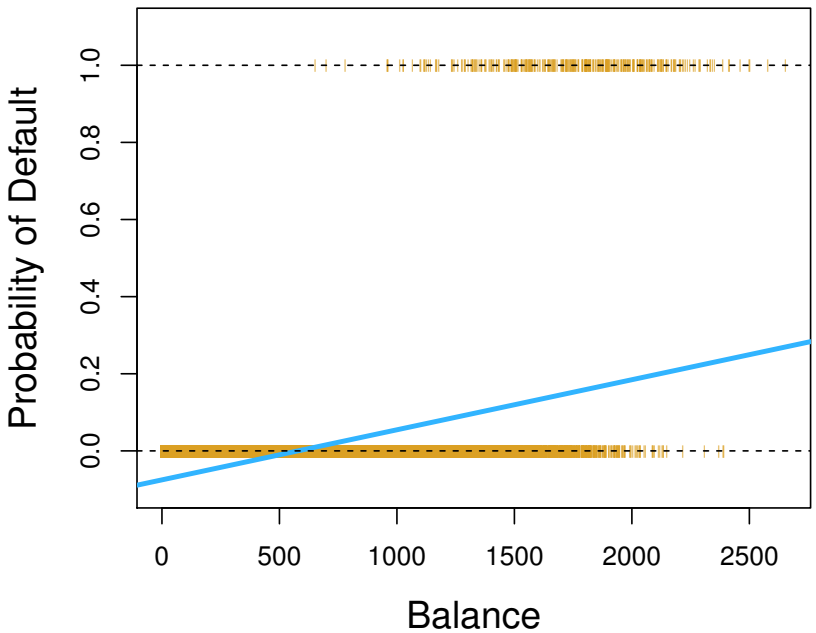
Suppose for the Default classification task that we code

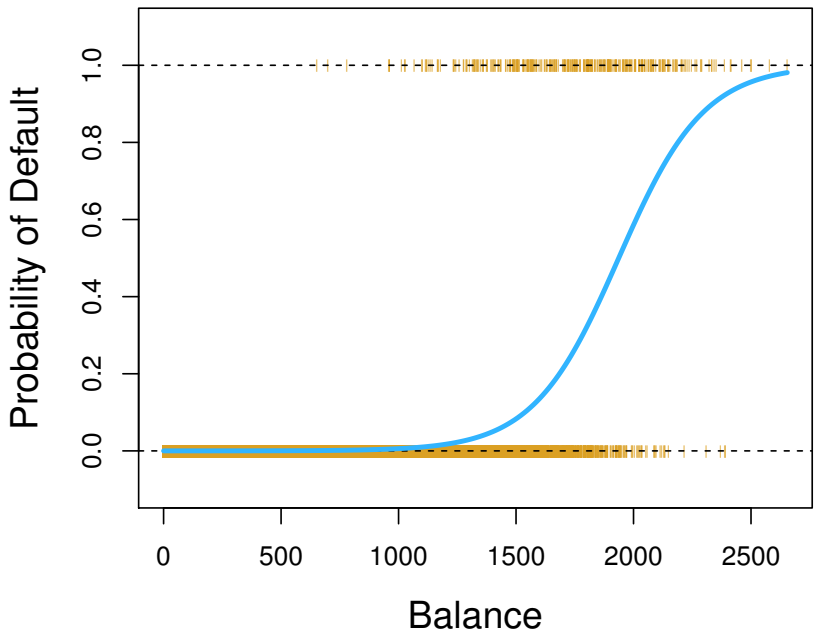
$$Y = \begin{cases} 0, & \text{if No} \\ 1, & \text{if Yes.} \end{cases}$$

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

- 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.
- 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 does not 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 is not appropriate here. **Multiclass Logistic Regression** or **Discriminant Analysis** are more appropriate.

Logistic Regression

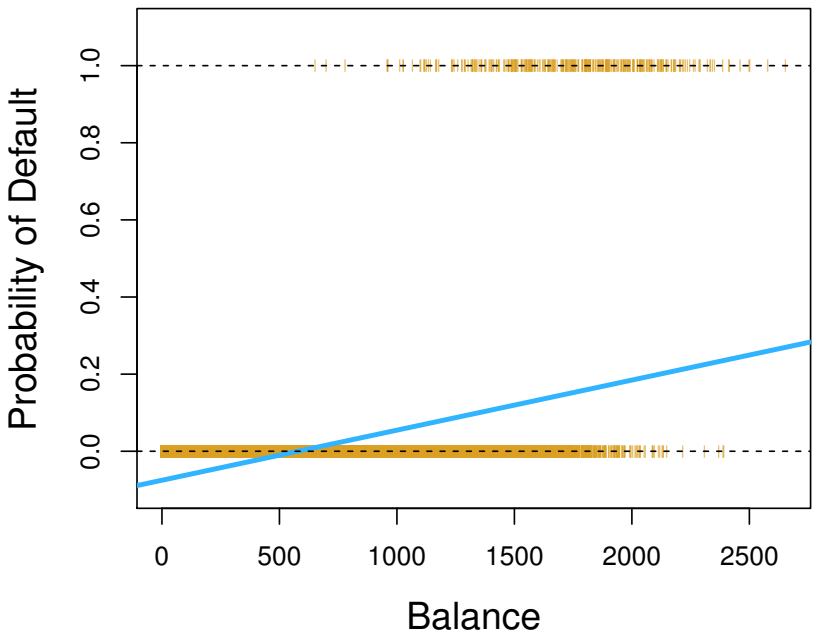
Let's write $p(X) = Pr(Y = 1|X)$ for short and consider using balance to predict default. Logistic regression uses the form

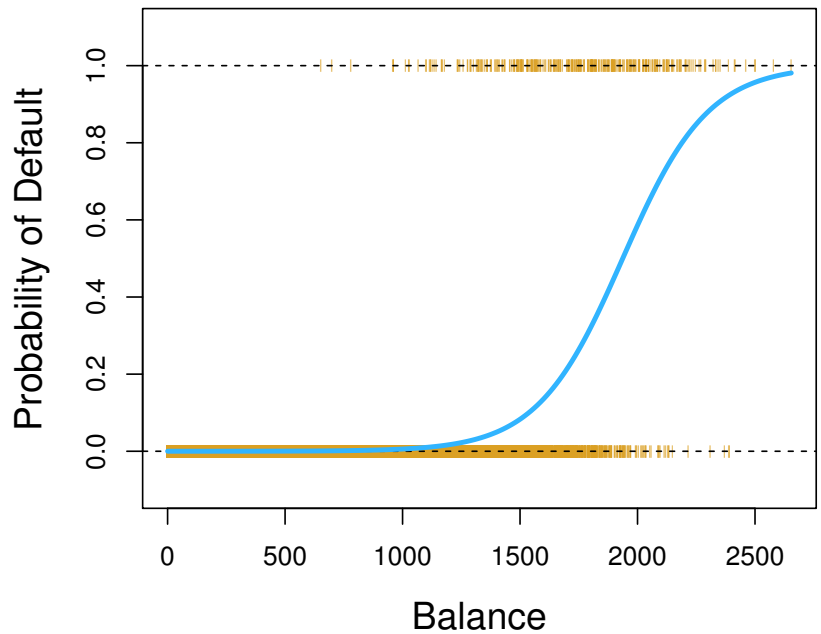
$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

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 $p(X)$.





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

Maximum Likelihood

We use maximum likelihood to estimate the parameters.

BC y can only take 2 values $y=1$ or 0.

$$l(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} (1 - p(x_i)).$$

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. Most statistical packages can fit linear logistic regression models by maximum likelihood. In R we use the glm function.

	Coefficient	Std. Error	t-statistic	p-value
Intercept	-10.6513	0.3612	-29.5	$<$0.0001
balance	0.0055	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?

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

Lets do it again, using student as the predictor.

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

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

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

Logistic regression with several variables

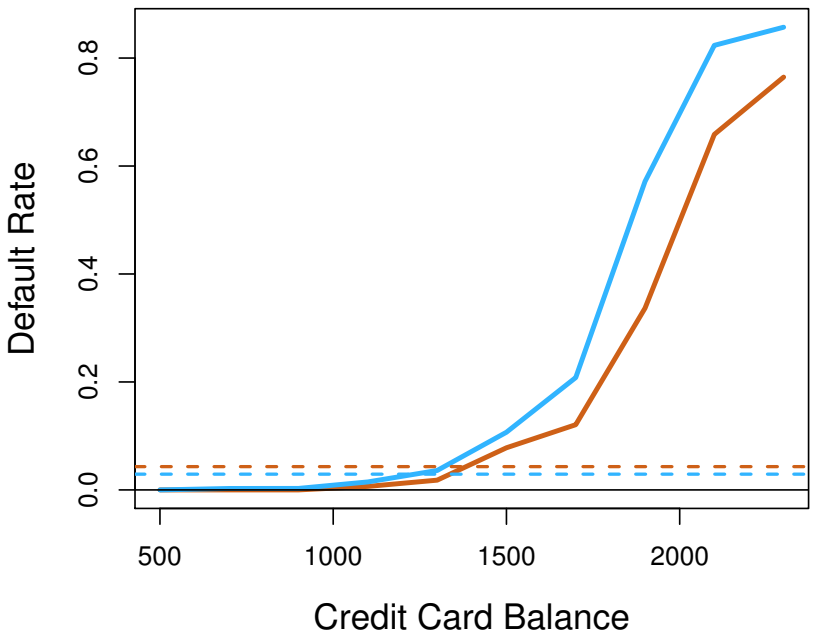
$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

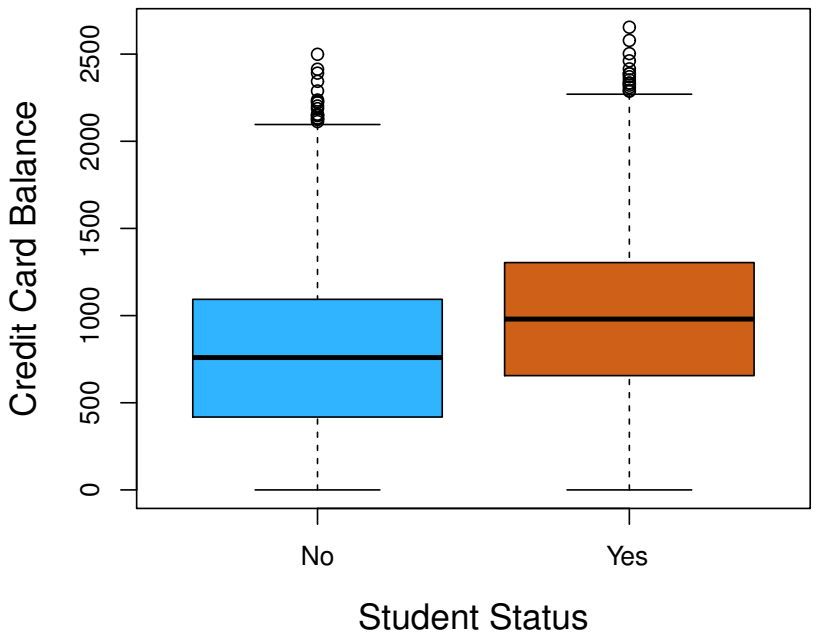
$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

	Coefficient	Std. Error	t-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

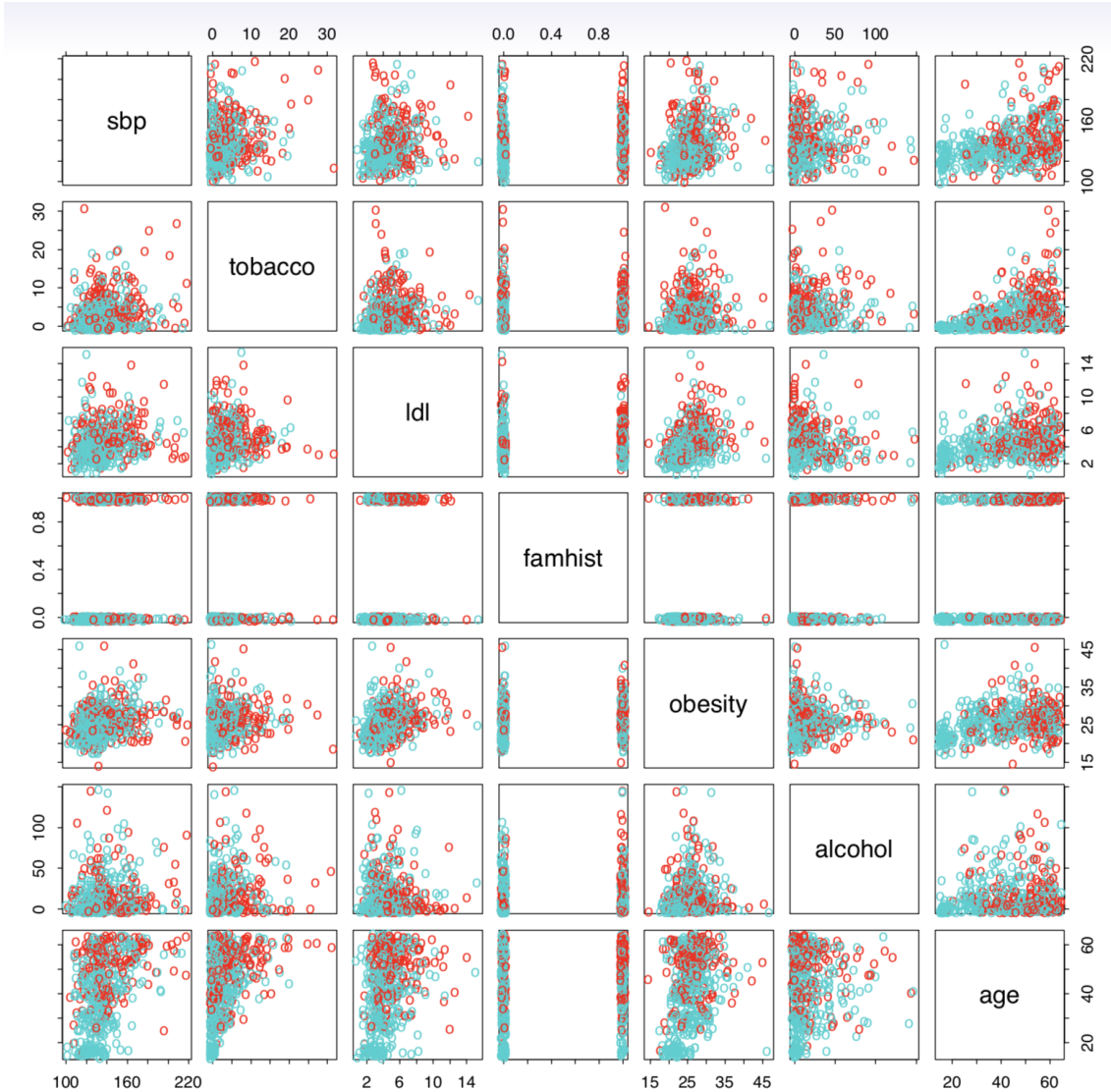




- Students tend to have higher balances than non-students, so their marginal default rate is higher than for non-students.
- But for each level of balance, students default less than non-students.
- Multiple logistic regression can tease this out.

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%.
- 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
```

=> we are doing logistic regression.

If we look at diff models, e.g. only w/ intercept, we'll model w/ lowest AIC.

$X \in \{1, 2, 3\}$ $x_1 = \begin{cases} 1 & \text{if } x = 2 \\ 0 & \text{o.w.} \end{cases}$, $x_2 = \begin{cases} 1 & \text{if } x = 3 \\ 0 & \text{o.w.} \end{cases}$

$x_1 = x_2 = 0$ if $x = 1$

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

$$Pr(Y = k|X) = \frac{e^{\beta_{ok}+\beta_{1k}X_1+\cdots+\beta_{pk}X_p}}{\sum_{l=1}^{K} e^{\beta_{0l}+\beta_{1l}X_1+\cdots+\beta_{pl}X_p}}.$$

Here there is a linear function for **each** class.

(Note, some cancellation is possible, and only $K-1$ linear functions are needed as in 2-class logistic regression.)

Multiclass logistic regression is also referred to as **multinomial regression**.

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

Using Baye's 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:

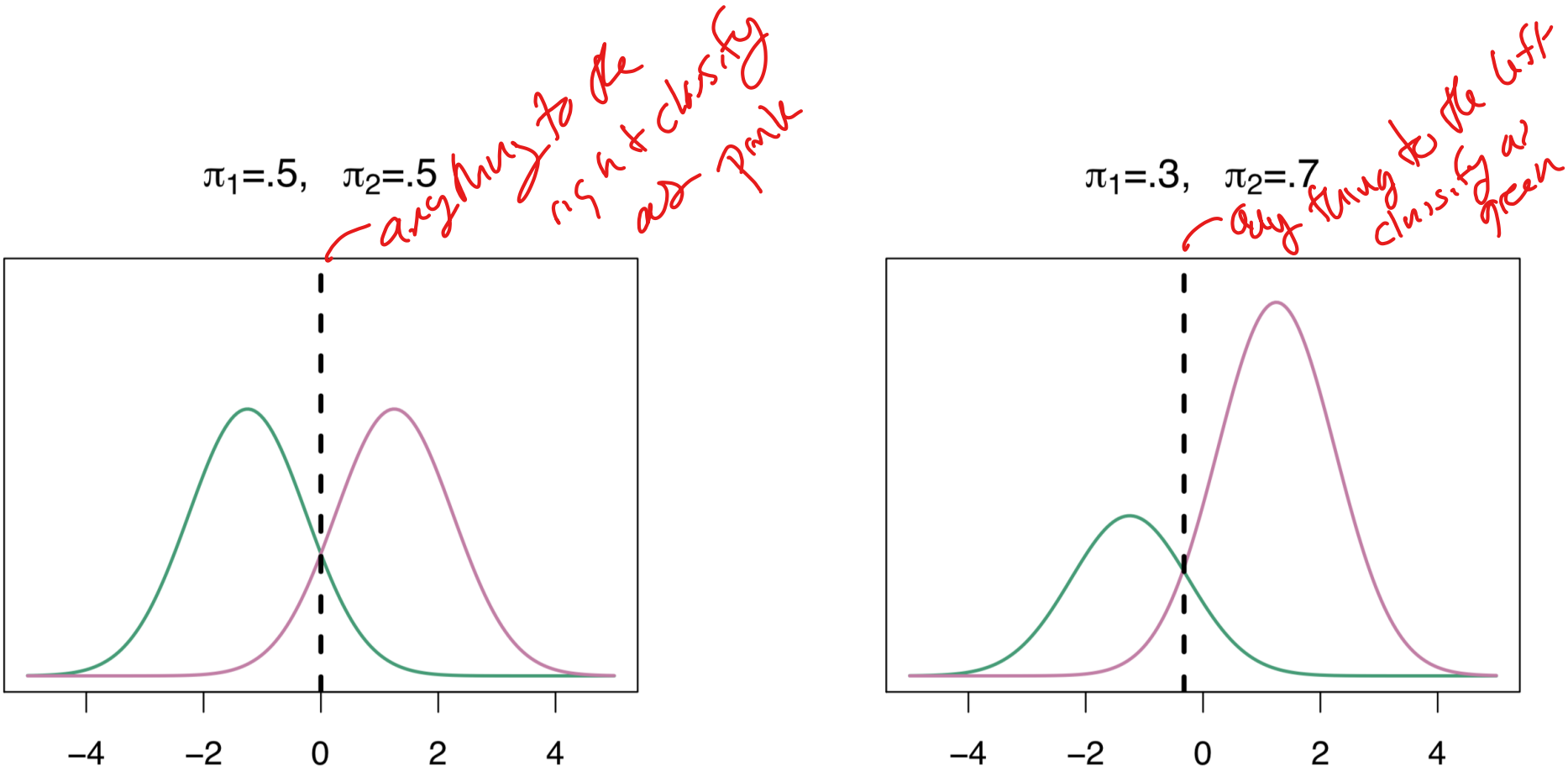
$$Pr(Y = k|X = x) = \frac{Pr(X = x|Y = k) \cdot Pr(Y = k)}{Pr(X = x)}.$$

One writes this slightly differently for discriminant analysis:

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

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

Linear Discriminant Analysis for p = 1

START Tuesday 2/1/22 (week 3, lecture 5)

The Gaussian density has the form

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

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

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

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

LDA is assumption alone.

constant wrt k

different scan k

Happily, there are simplifications and cancellations.

Discriminant functions

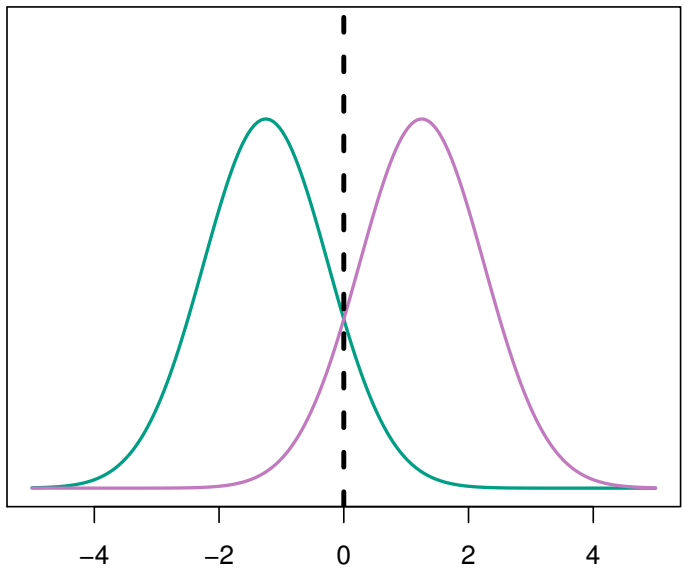
To classify at the value $X = x$, we need to see which of the $p_k(x)$ is largest. Taking logs, and discarding terms that do not depend on k, we see that this is equivalent to assigning x to the class with the largest **discriminant score**:

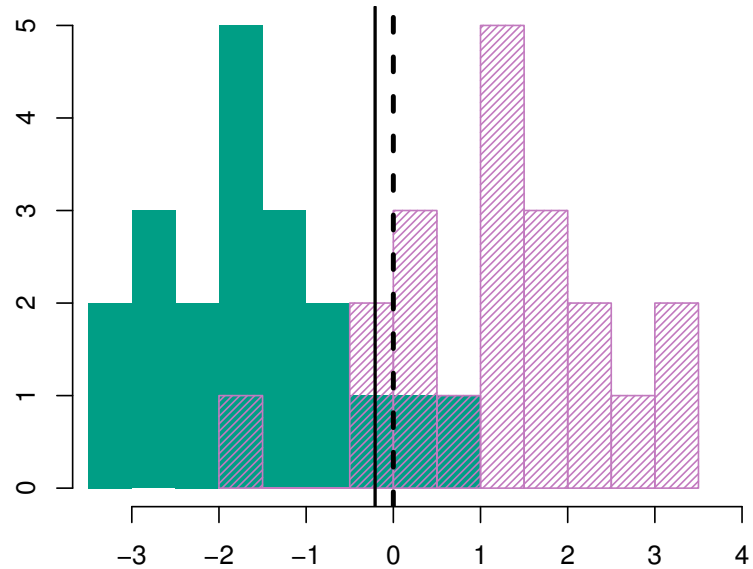
$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k).$$

Note that $\delta(x)$ is a **linear** function of x. - Therefore this is called linear discriminant analysis -

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

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





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

$$\hat{\pi}_k = \frac{n_k}{n}$$

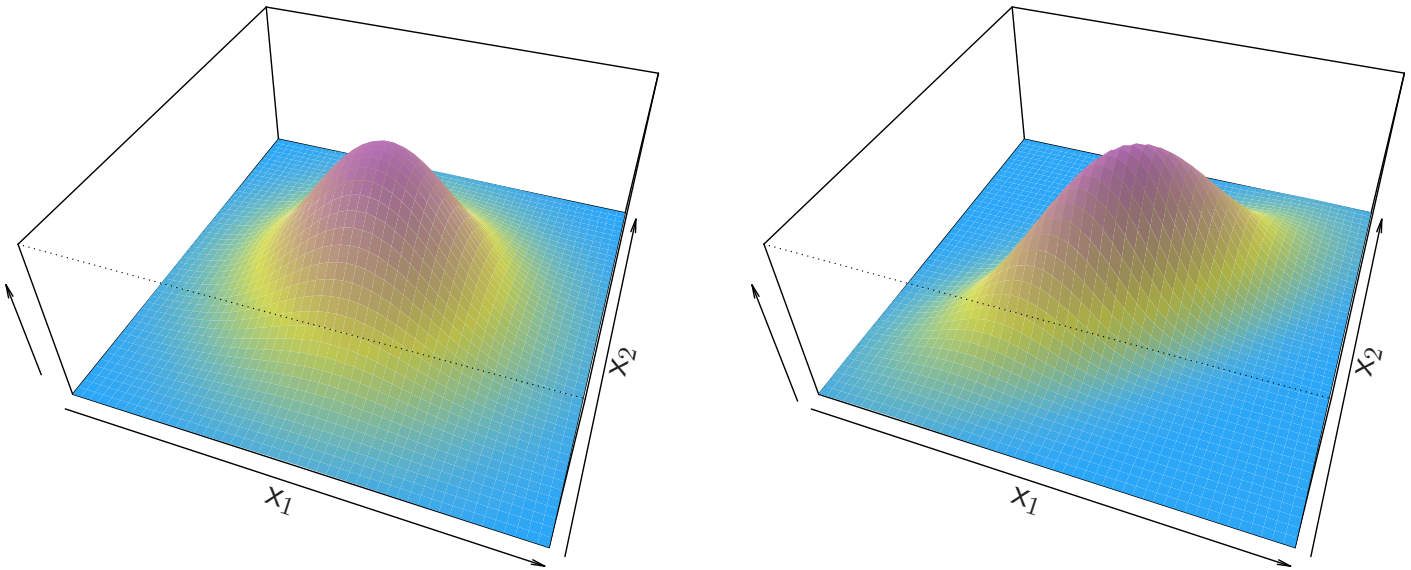
$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

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

$$= \sum_{k=1}^{K} \frac{n_K - 1}{n - K} \cdot \hat{\sigma}_k^2,$$

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

Linear Discriminant Analysis when $p > 1$

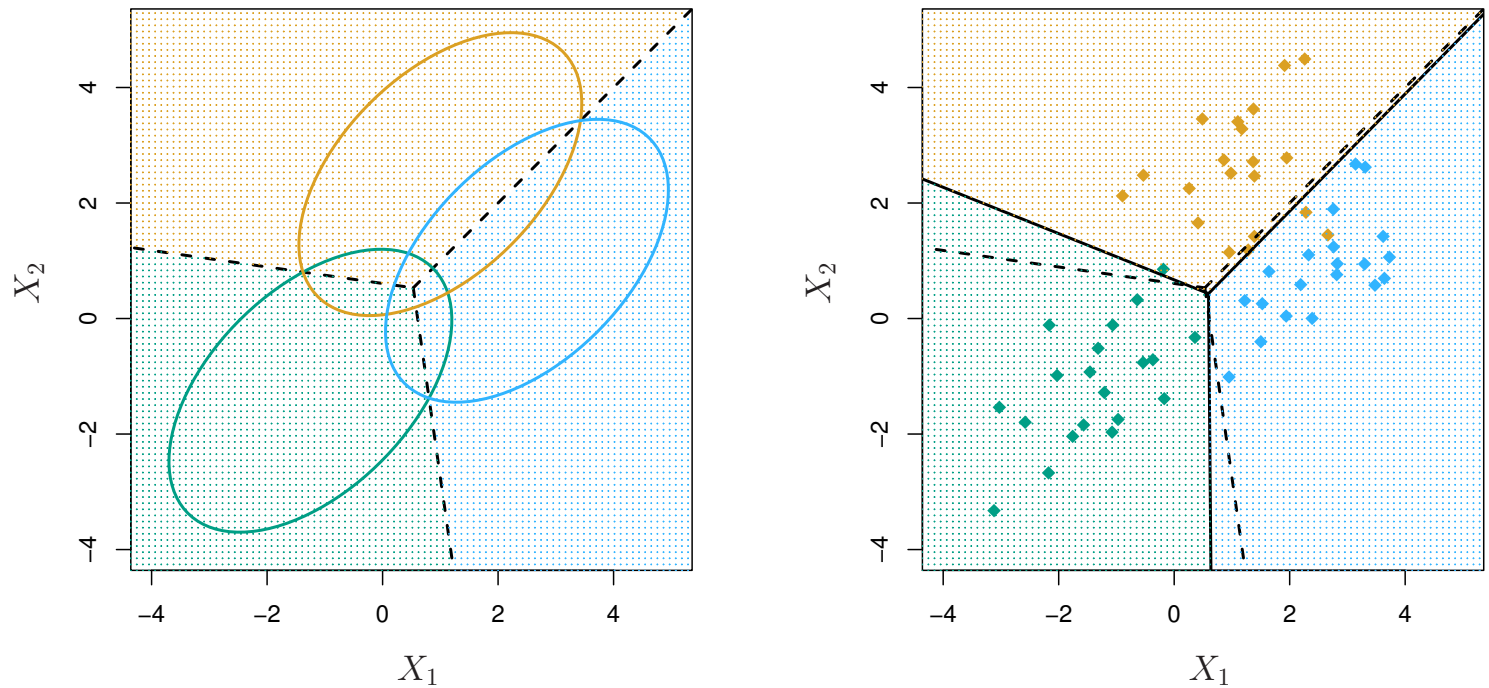


$$\text{Density: } f(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}.$$

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

This can be written as:
$\delta_k(x) = c_{k0} + c_{k1}x_1 + c_{k2}x_2 + \cdots + c_{kp}x_p$—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**. Were they known, they would 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:

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

So classifying to the largest $\delta_k(x)$ amounts to classifying to the class for which $Pr(Y = k|X = x)$ 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

STOP Tuesday 2/1/22 (week 3, lecture 5)

		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!

Some caveats:

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

Types of errors

START Tuesday 2/8/22 (week 4, lecture 6)

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

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

We produced this table by classifying to class Yes if

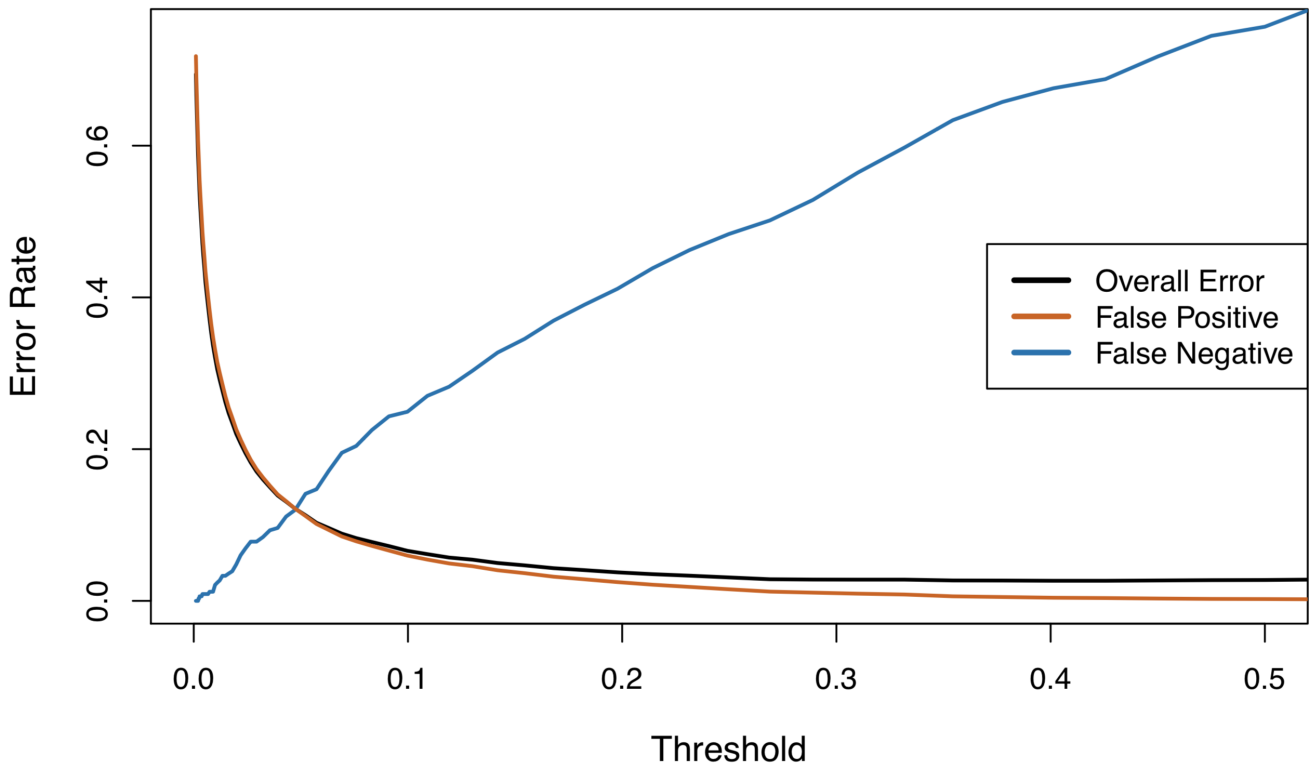
$$\widehat{\Pr}(\text{Default} = \text{Yes}|\text{Balance}, \text{Student}) \geq 0.5.$$

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

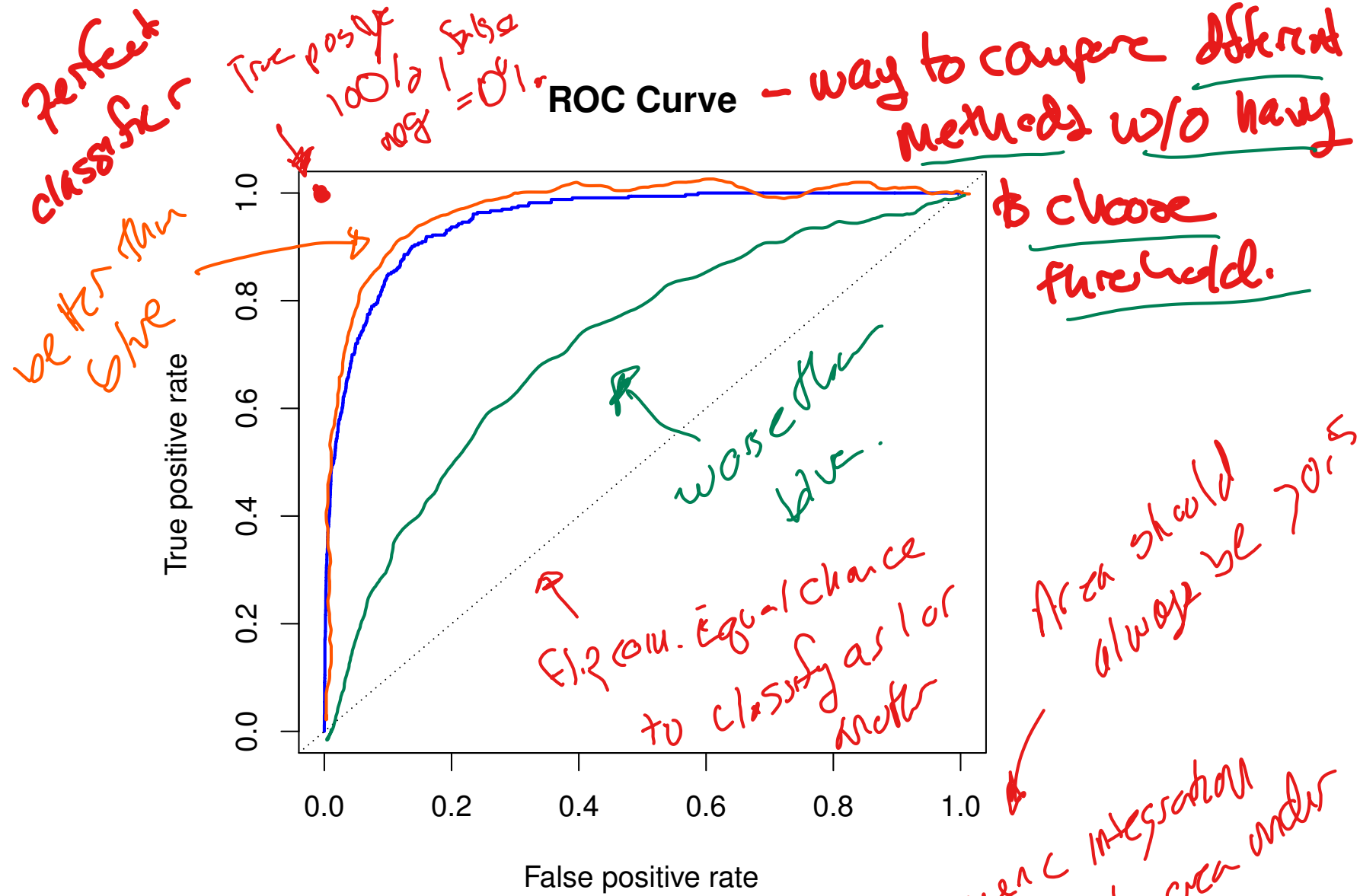
$$\widehat{\Pr}(\text{Default} = \text{Yes}|\text{Balance}, \text{Student}) \geq \textbf{threshold},$$

and vary **threshold**. * - way to control false positive & false negative.

Varying the threshold



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



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.

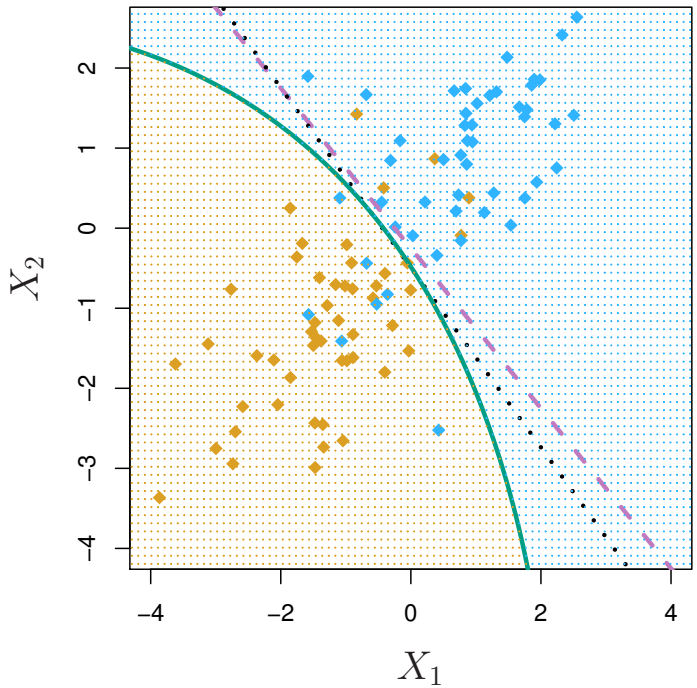
Other forms of Discriminant Analysis

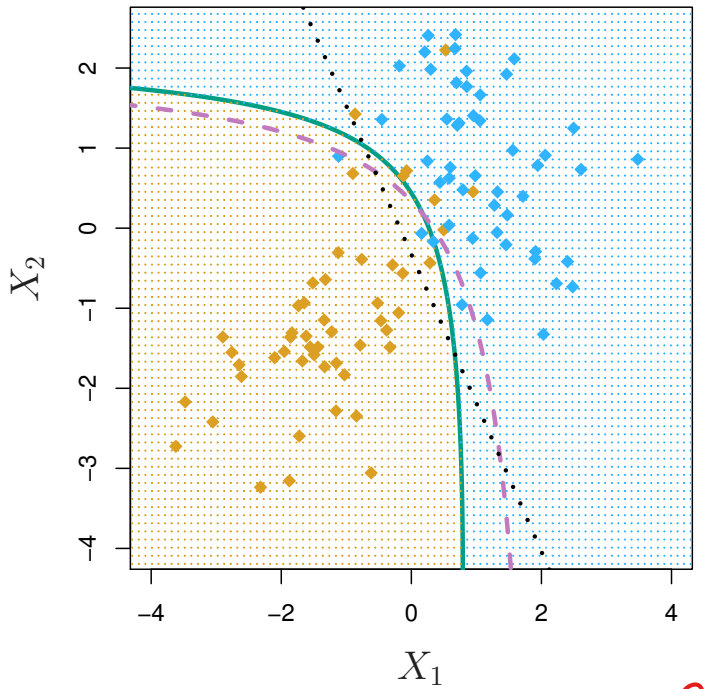
$$Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}.$$

When $f_k(x)$ are Gaussian densities, with the same covariance matrix Σ 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 Σ_k in each class, we get **quadratic discriminant analysis**.
- With $f_k(x) = \prod_{j=1}^{p} f_{jk}(x_j)$ (conditional independence model) in each class we get **naive Bayes**. For Gaussian this means the Σ_k are diagonal.
- Many other forms, by proposing specific density models for $f_k(x)$, including nonparametric approaches.

Quadratic Discriminant Analysis





can be hard to compute if p is large.

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

Because the Σ_k are different, the quadratic terms cannot be canceled out.

Naive 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 naive Bayes assumes each Σ_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, naive Bayes often produces good classification results.

Logistic Regression versus LDA

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

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

So it has the same form as logistic regression.
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.

Footnote: 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 $K > 2$.
- Naive Bayes is useful when p is very large.
- See ISLR chapter 4.5 "A Comparison of Classification Methods" for some comparisons of logistic regression, LDA and KNN.

Support Vector Machines

↳ don't give probabilities, only give you decision.

Popular b/c very flexible, uses kernel methods.

Here we approach the two-class classification problem in a direct way:

We try and find a plane that separates the classes in feature space.

If we cannot, we get creative in two ways:

- We soften what we mean by "separates", and
- We enrich and enlarge the feature space so that separation is possible.

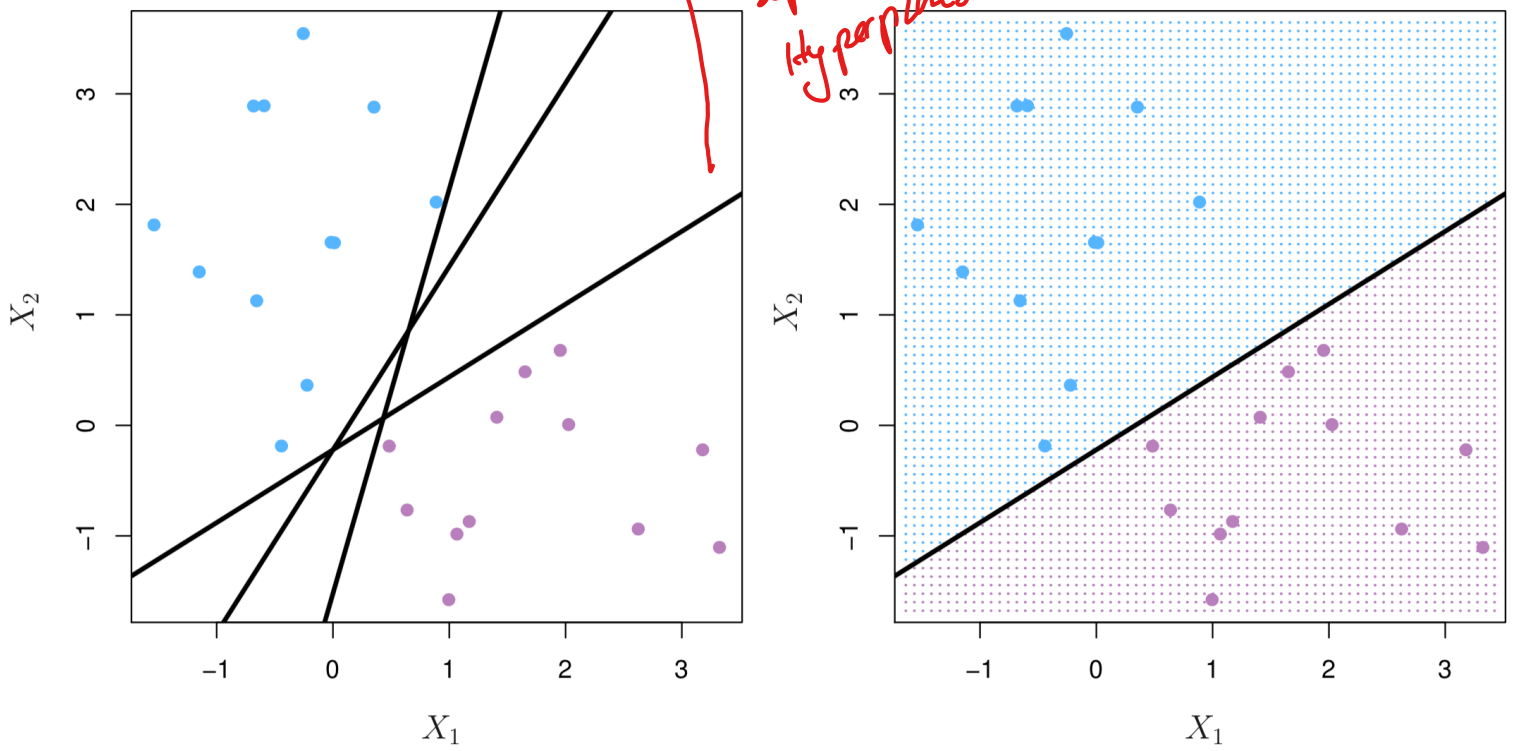
What is a Hyperplane?

- A hyperplane in p dimensions is a flat subspace of dimension $p - 1$.
- In general the equation for a hyperplane has the form

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p = 0.$$

- In $p = 2$ dimensions a hyperplane is a line.
- If $\beta_0 = 0$, the hyperplane goes through the origin, otherwise not.
- The vector $\beta = (\beta_1, \beta_2, \cdots, \beta_p)$ is called the normal vector — it points in a direction orthogonal to the surface of a hyperplane.

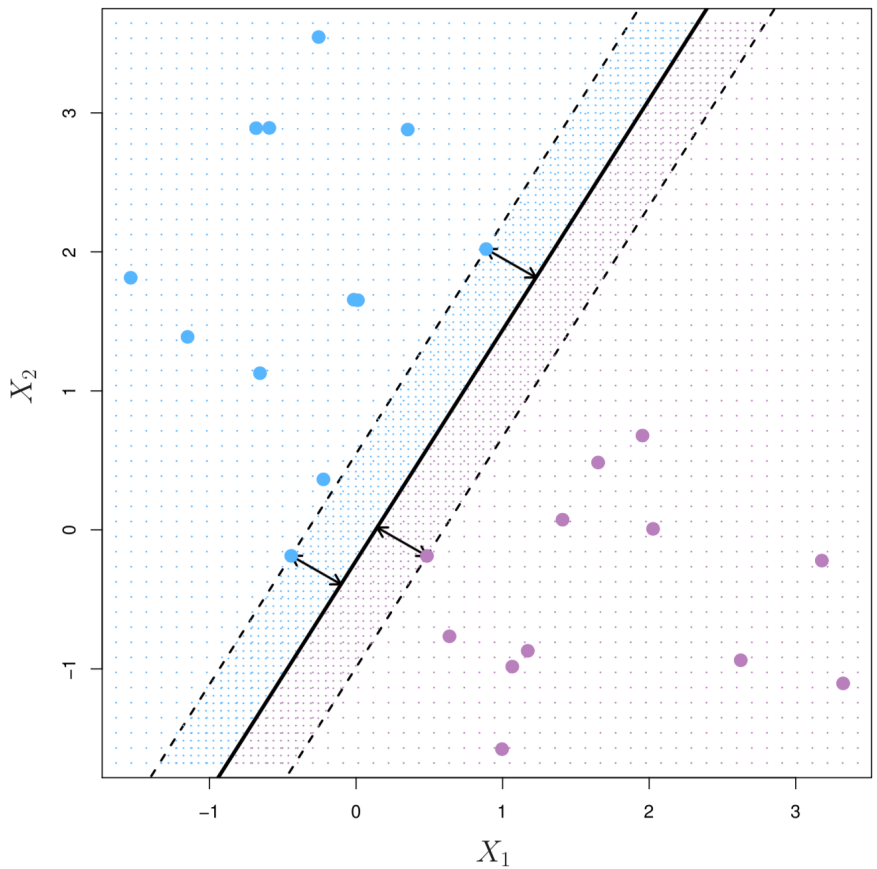
Separating Hyperplanes



- If $f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$, then $f(X) > 0$ for points on one side of the hyperplane, and $f(X) < 0$ for points on the other.
- If we code the colored points as $Y_i = +1$ for blue, say, and $Y_i = -1$ for purple, then if $Y_i \cdot f(X_i) > 0$ for all i, $f(X) = 0$ defines a **separating hyperplane**.

Maximal Margin Classifier

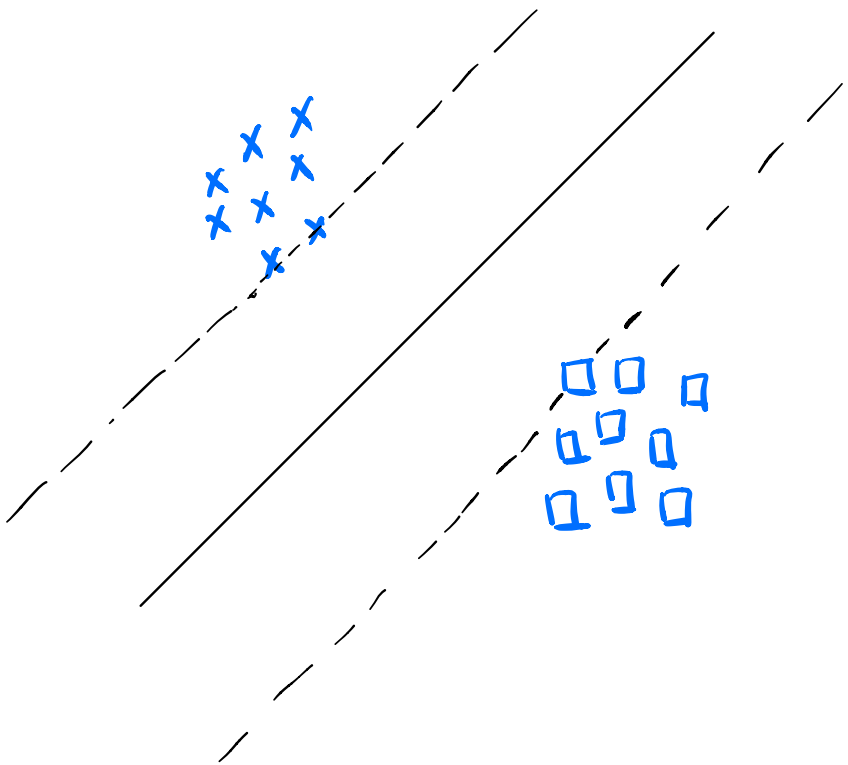
Among all separating hyperplanes, find the one that makes the biggest gap or margin between the two classes.



But why?

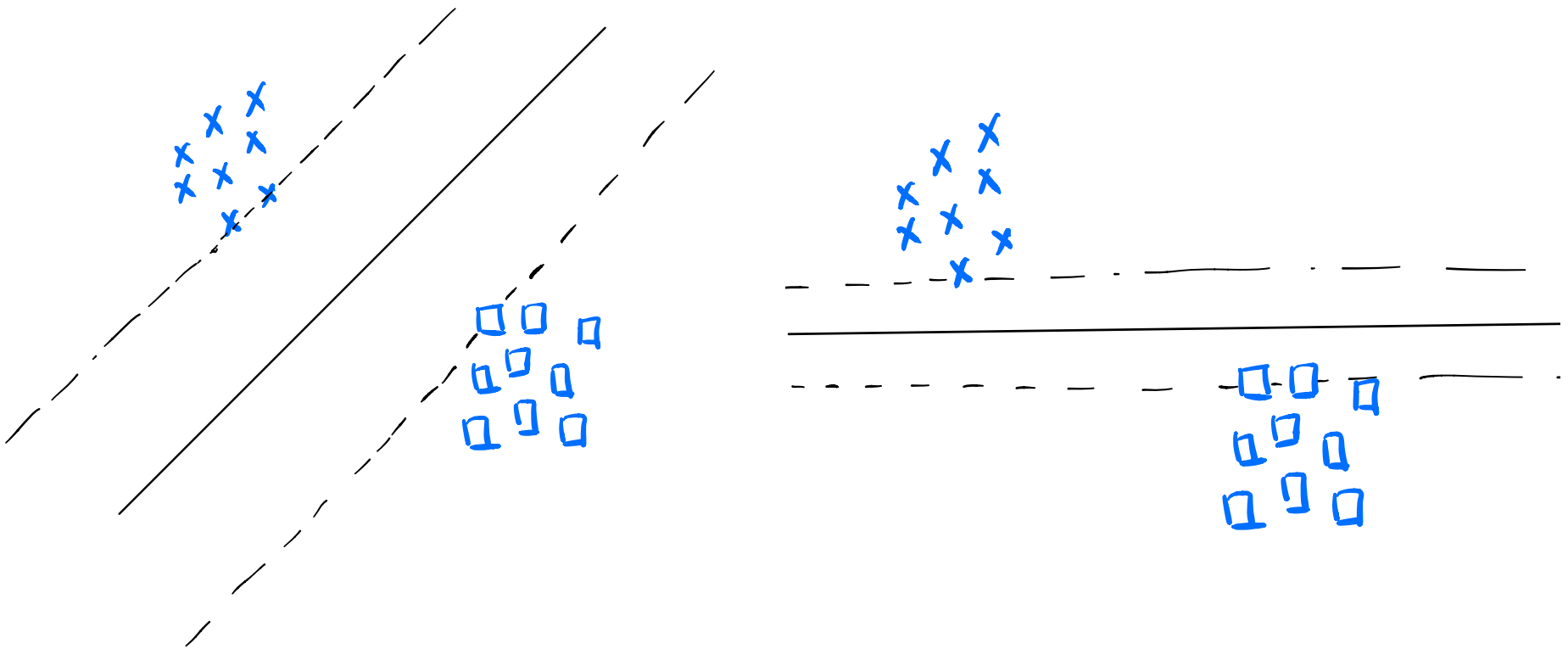
Why biggest margin?

Training



Why biggest margin?

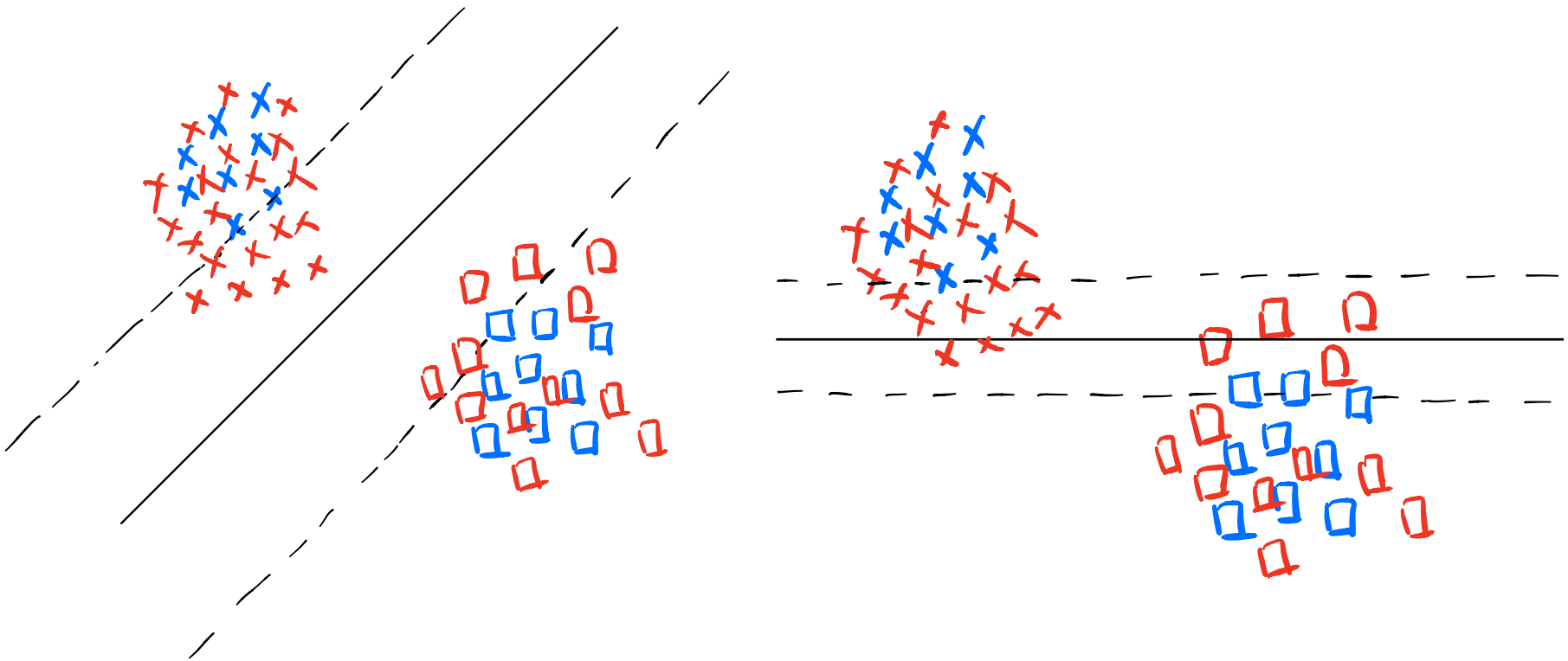
Training



makes no obvious difference.

Why biggest margin?

Testing



Wide margin is clearly better.

Constrained optimization problem

$$\text{maximize}_{\beta_0,\beta_1,\cdots,\beta_p}\ M$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}) \geq M \qquad \text{for all } i = 1, \cdots, N.$$

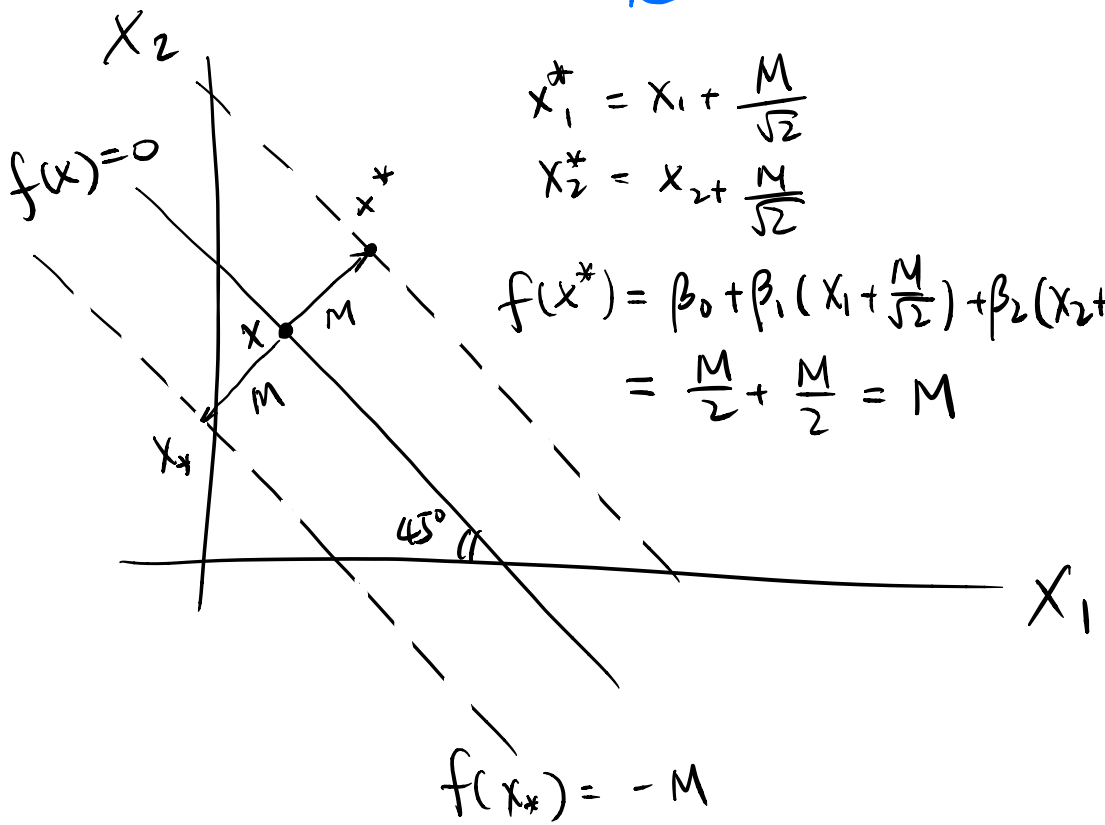
This can be rephrased as a convex quadratic program, and solved efficiently. The function svm() in package e1071 solves this problem efficiently.

Constraint explained

E.g. $f(x) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$

$\beta_0 = -\frac{\sqrt{2}}{2} \quad \beta_1 = \beta_2 = \frac{\sqrt{2}}{2}$

$\beta_1^2 + \beta_2^2 = 1$ ~ unit length



$x_1^* = x_1 + \frac{M}{\sqrt{2}}$

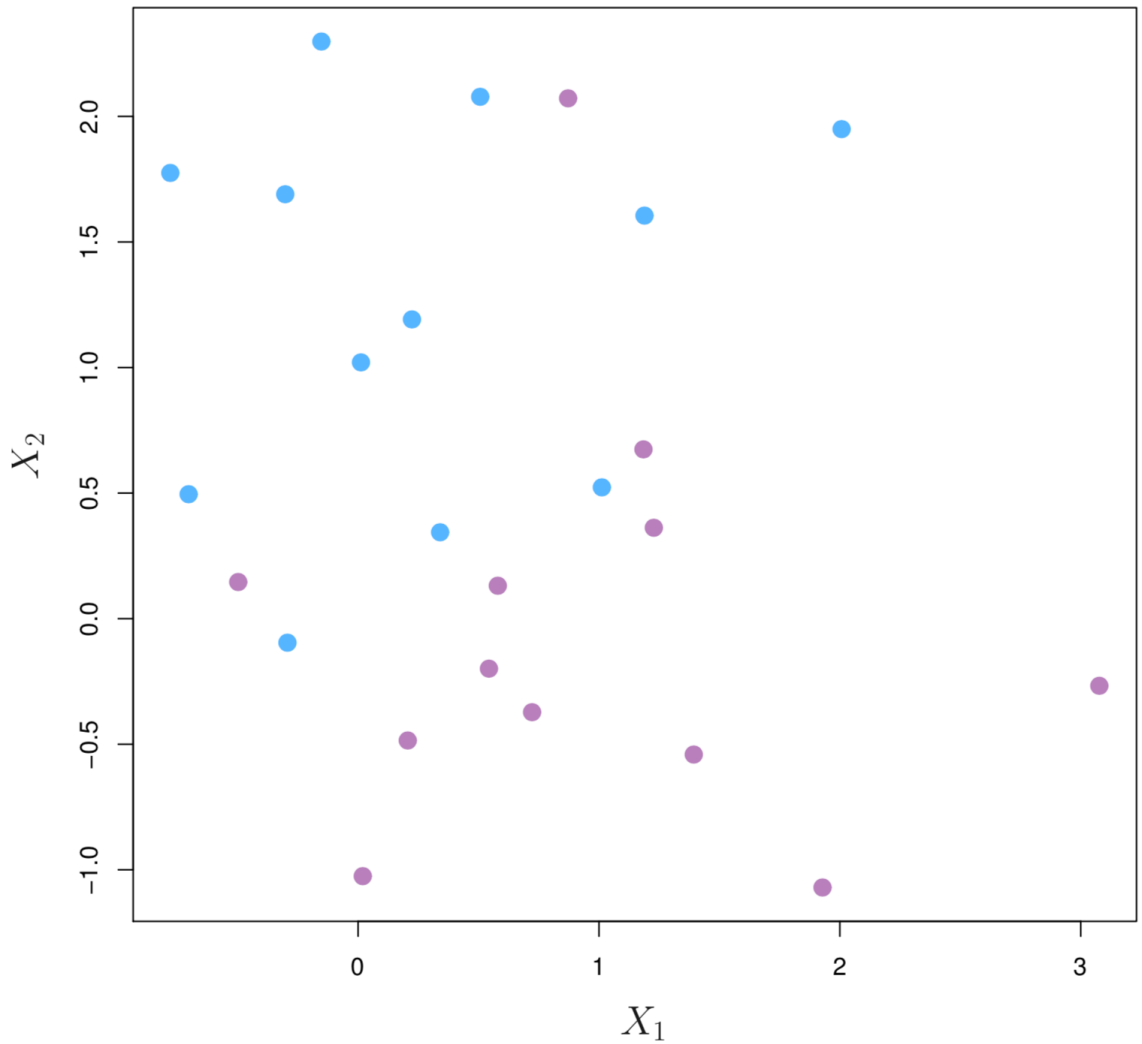
$x_2^* = x_2 + \frac{M}{\sqrt{2}}$

$f(x^*) = \beta_0 + \beta_1(x_1 + \frac{M}{\sqrt{2}}) + \beta_2(x_2 + \frac{M}{\sqrt{2}})$

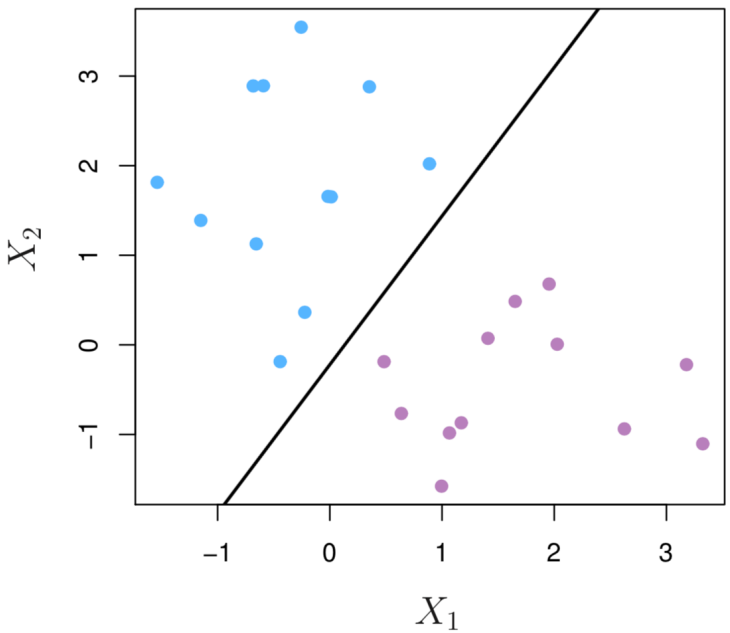
$= \frac{M}{2} + \frac{M}{2} = M$

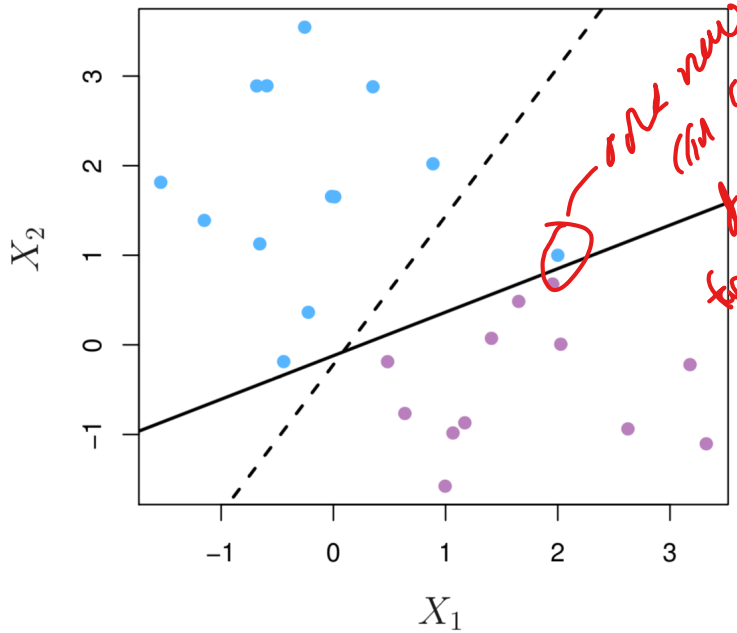
Non-separable Data

The data on the left are not separable by a linear boundary.
This is often the case when $n > p$.



Noisy Data

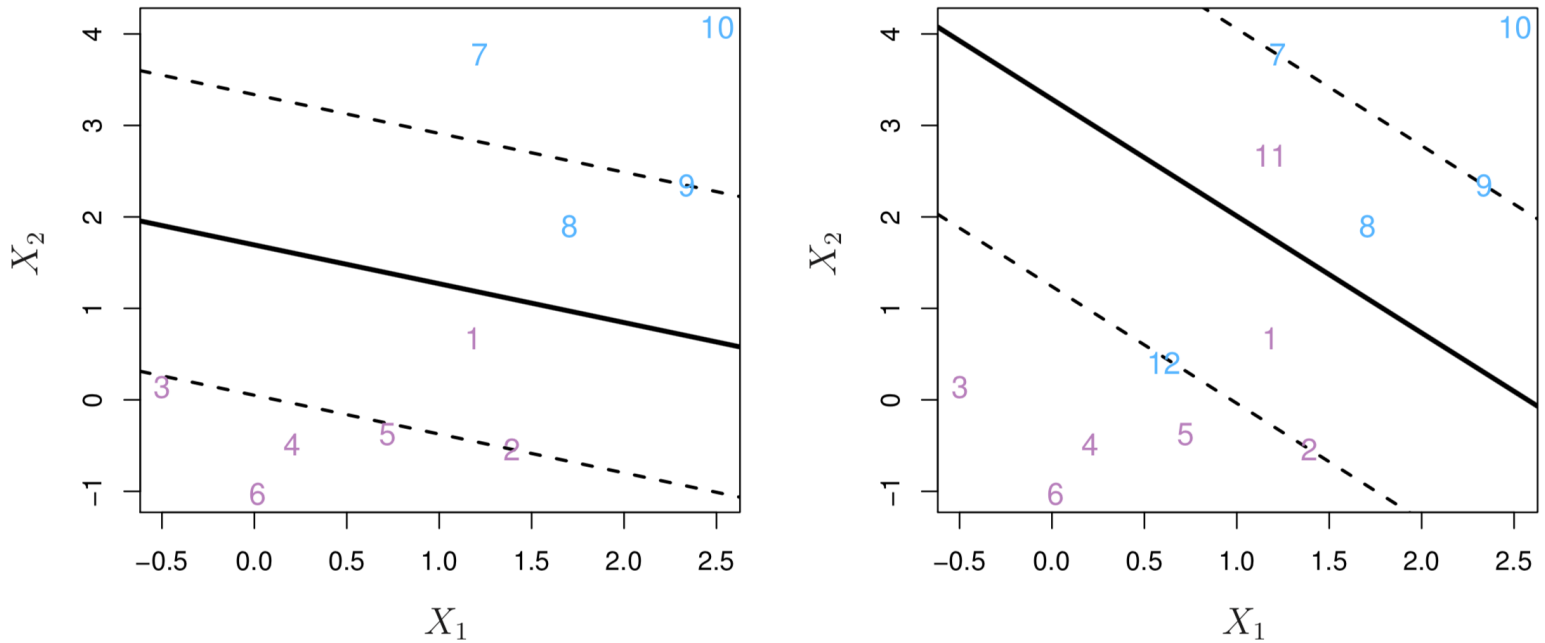




Sometimes the data are separable, but noisy. This can lead to a poor solution for the maximal-margin classifier.

The **support vector classifier** maximizes a **soft** margin.

Support Vector Classifier



For each point on the wrong side of the dashed line, we pay a "price". And we have a total budget of C to spend. The price is proportional to the distance from the point to the dashed line.

For each point on the wrong side of the dashed line, we pay a "price". And we have a total budget of C to spend. The price is proportional to the distance from the point to the dashed line.

$$\text{maximize}_{\beta_0,\beta_1,\cdots,\beta_p,\epsilon_1,\cdots,\epsilon_n} \quad M \quad \text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

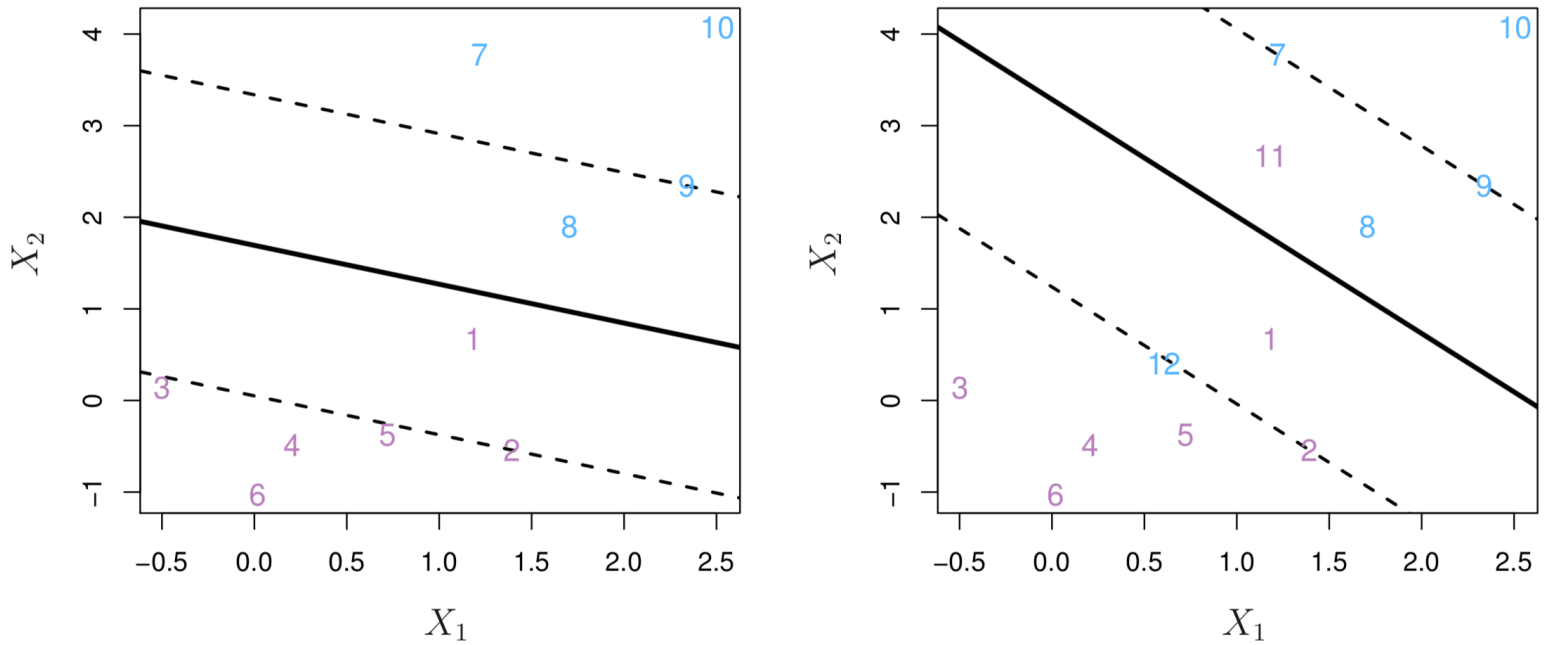
$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}) \geq M(1 - \epsilon_i),$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C.$$

$C = 0 \Rightarrow$ no mistakes.

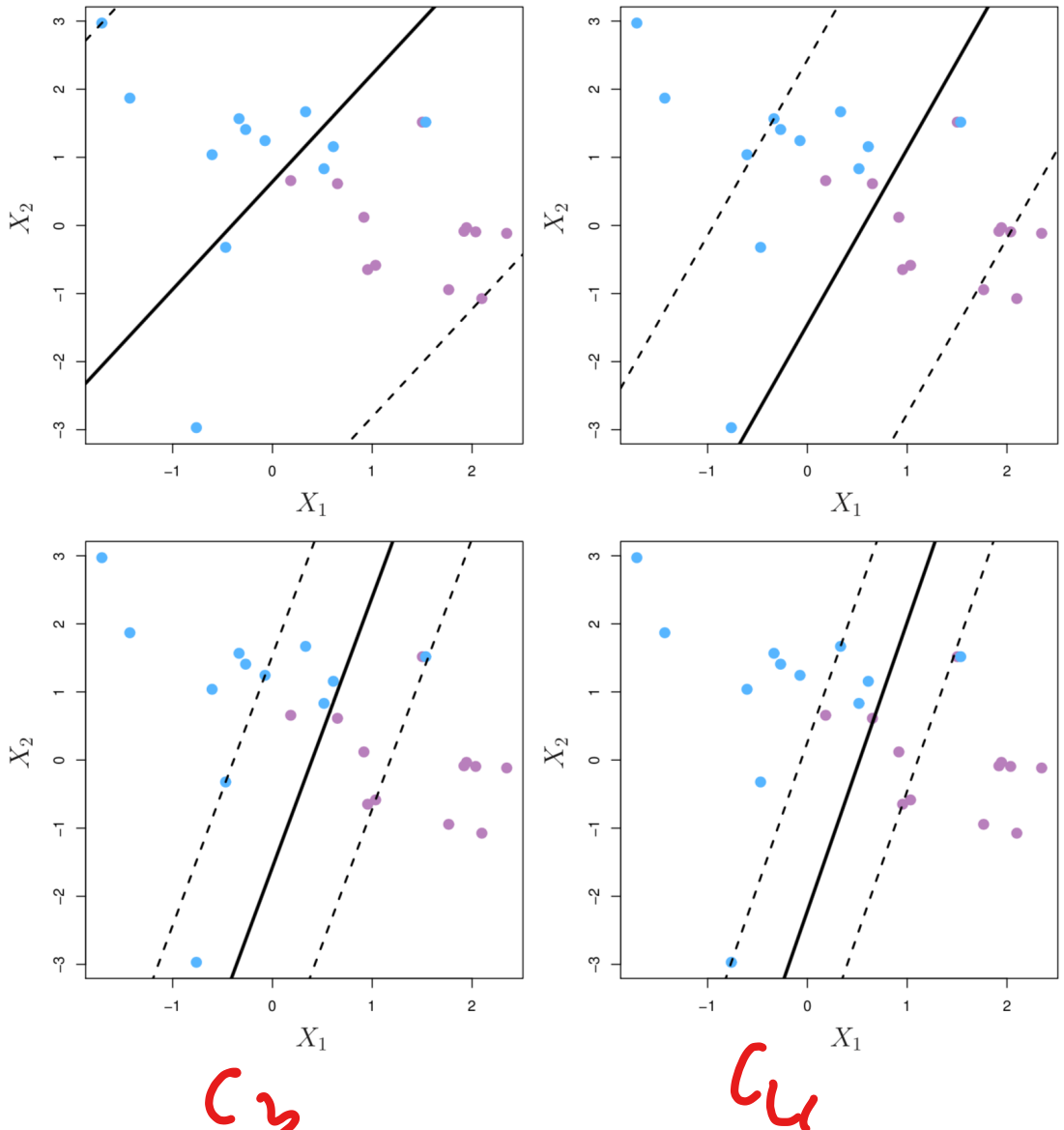
STAT Tuesday 2/8/22 (week 4, lecture 6)

START Mon 21/2/22 (week 4, lecture 7)

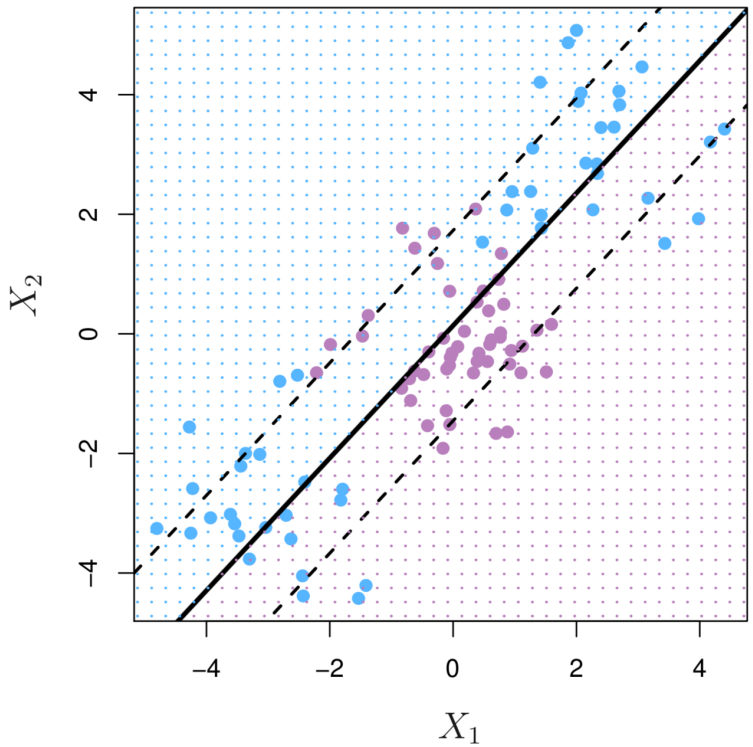


Observations that lie directly on the margin, or on the wrong side of the margin for their class, are known as support vectors.
Left: 1,2,8,9 are support vectors. Right: 1,2,7,8,9,11,12 are support vectors

C is a regularization parameter

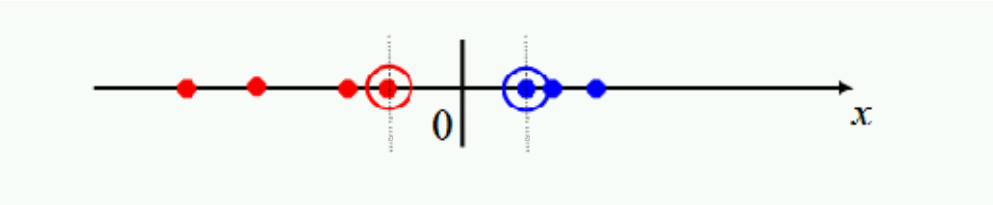


Linear boundary can fail

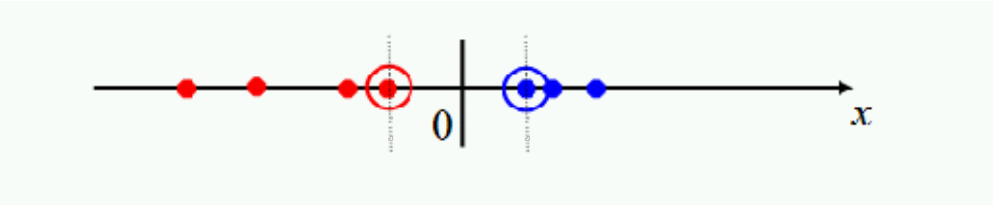


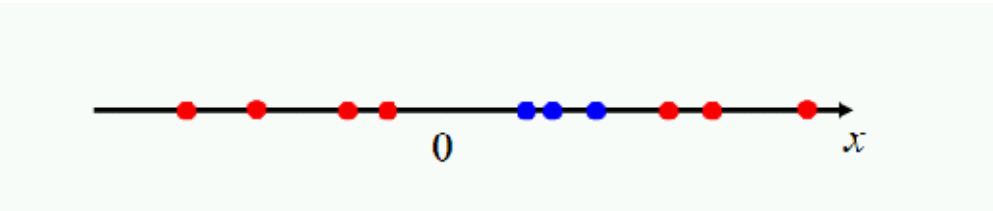
Sometime a linear boundary simply won't work, no matter what value of C. What to do?

One-dimensional Example

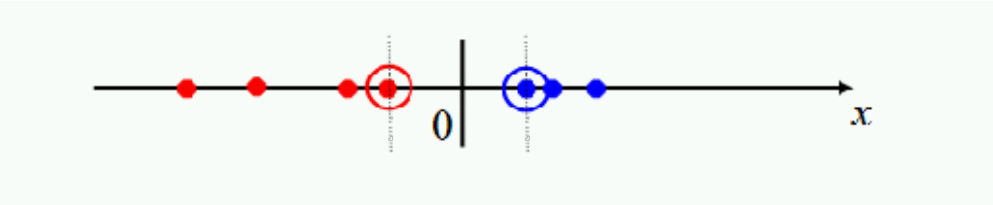


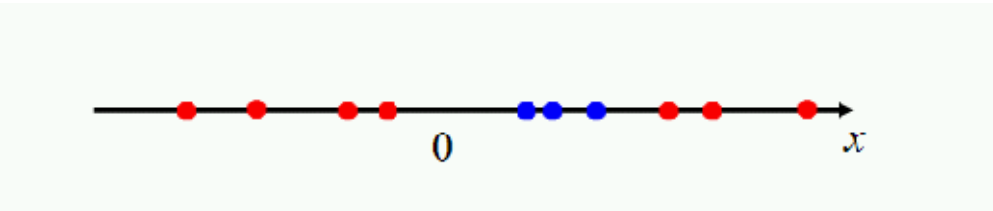
One-dimensional Example

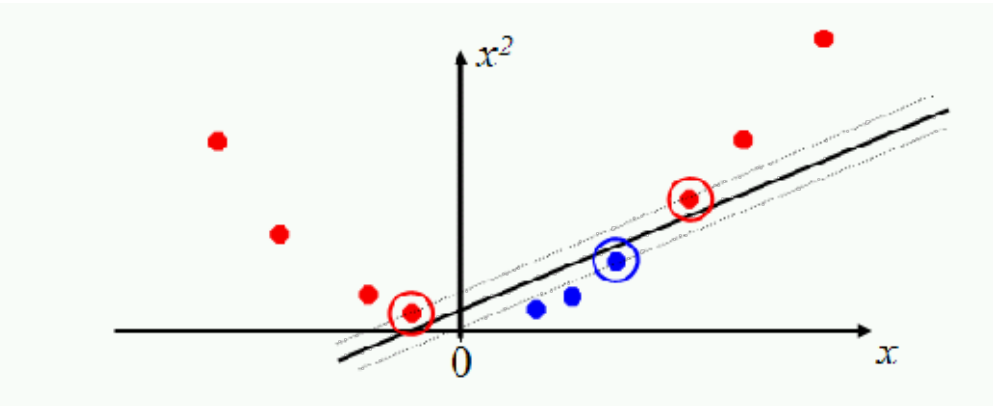




One-dimensional Example







expand dimension.

Feature Expansion

- Enlarge the space of features by including transformations; e.g. $X_1^2, X_1^3, X_1X_2, X_1X_2^2, \cdots$. Hence go from a p-dimensional space to a $q > p$ dimensional space.
- Fit a support-vector classifier in the enlarged space.
- This results in non-linear decision boundaries in the original space.

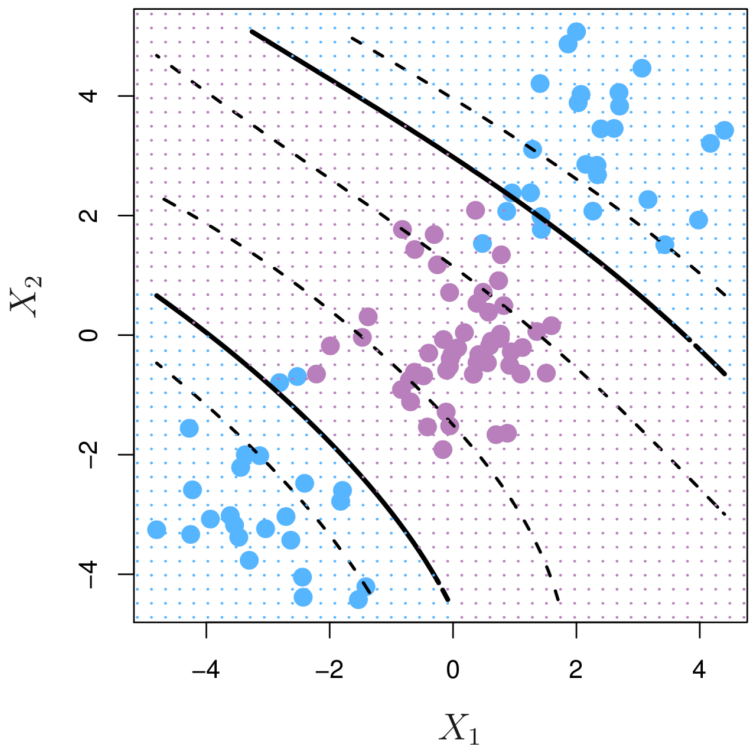
Example: Suppose we use $(X_1, X_2, X_1^2, X_2^2, X_1X_2)$ instead of just (X_1, X_2). Then the decision boundary would be of the form

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1^2 + \beta_4 X_2^2 + \beta_5 X_1 X_2 = 0.$$

This leads to nonlinear decision boundaries in the original space.

Cubic Polynomials

Here we use a basis expansion of cubic polynomials
From 2 variables to 9
The support-vector classifier in the enlarged space solves the problem in the lower-dimensional space.



Nonlinearities and Kernels

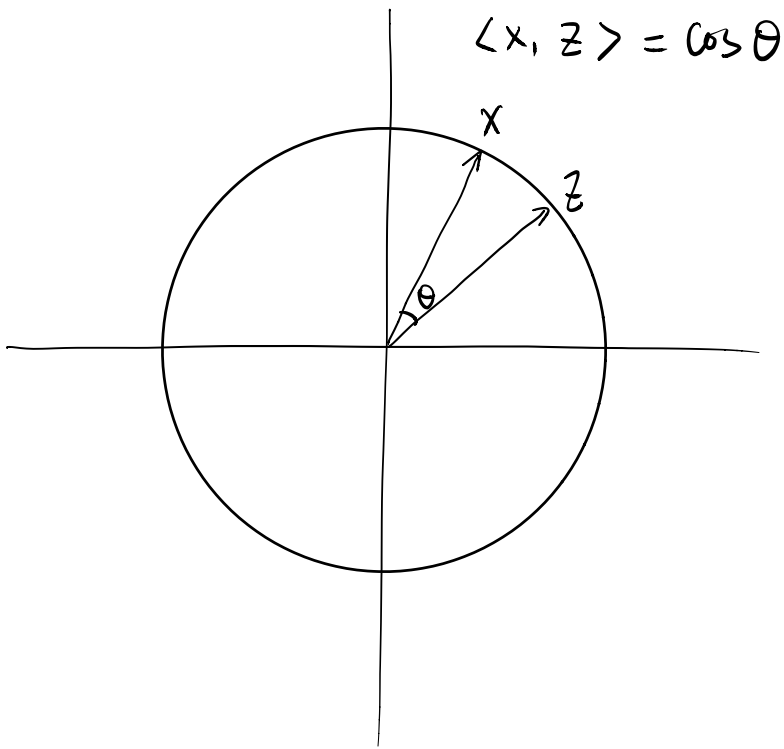
- Polynomials are hard to compute even for moderate degree when p is large.
- There is a more elegant and controlled way to introduce nonlinearities in support-vector classifiers — through the use of **kernels**.
  - Computationally, only need to compute $\binom{n}{2}$ inner products of p-dimensional vectors. Details later.
  - Choosing kernel is not an easy task.
- Before we discuss these, we must understand the role of **inner products** in support-vector classifiers.

Inner products

Inner (dot) product between vectors

$$< x, z >= \sum_{j=1}^{p} x_j z_j$$

Inner product measures the similarity of two vectors.



Inner products and support vectors

- The solution to the linear support vector classifier can be written as

$$\hat{f}(x) = \hat{\beta}_0 + \sum_{i=1}^{n} \hat{\alpha}_i < x, x_i >$$

- To estimate the parameters $\alpha_1, \cdots, \alpha_n$ and β_0, all we need are the $\binom{n}{2}$ inner products $< x_i, x_{i'} >$ for all pairs of training observations.
- It turns out that that α_i is nonzero only for the support vectors.

$$\hat{f}(x) = \hat{\beta}_0 + \sum_{i \in \mathcal{S}} \hat{\alpha}_i < x, x_i >$$

$\mathcal{S}$ is the collection of indices of these support vectors.

- We can replace the inner product by other similarity measures.
- Kernel functions can do this for us. E.g. polynomial kernel

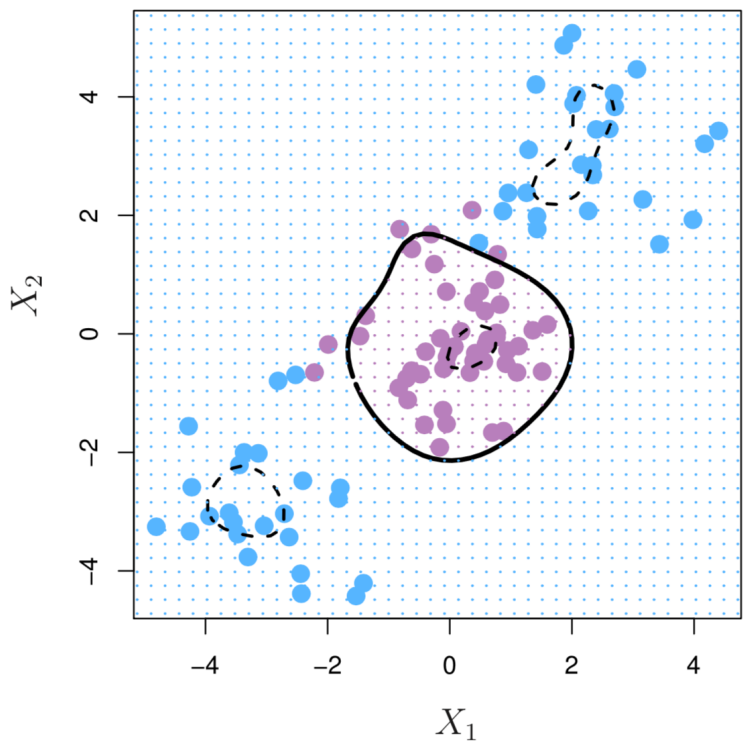
$$K(x_i, x_{i'}) = \left(1 + \sum_{j=1}^{p} x_{ij} x_{i'j}\right)^d$$

- The solution has the form

$$\hat{f}(x) = \hat{\beta}_0 + \sum_{i \in \mathcal{S}} \hat{\alpha}_i K(x, x_i).$$

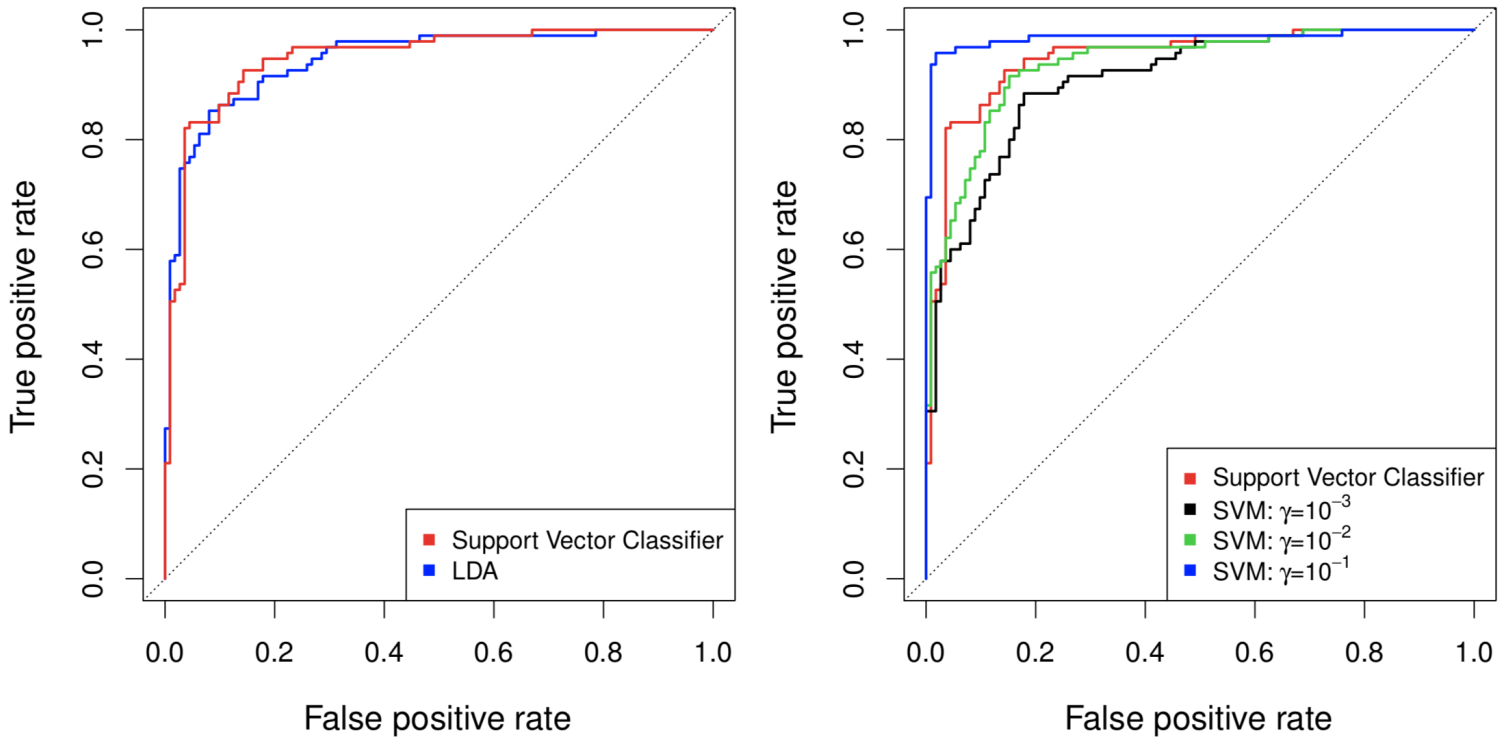
Radial Kernel

$$K(x_i, x_i') = \exp(-\gamma \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2).$$



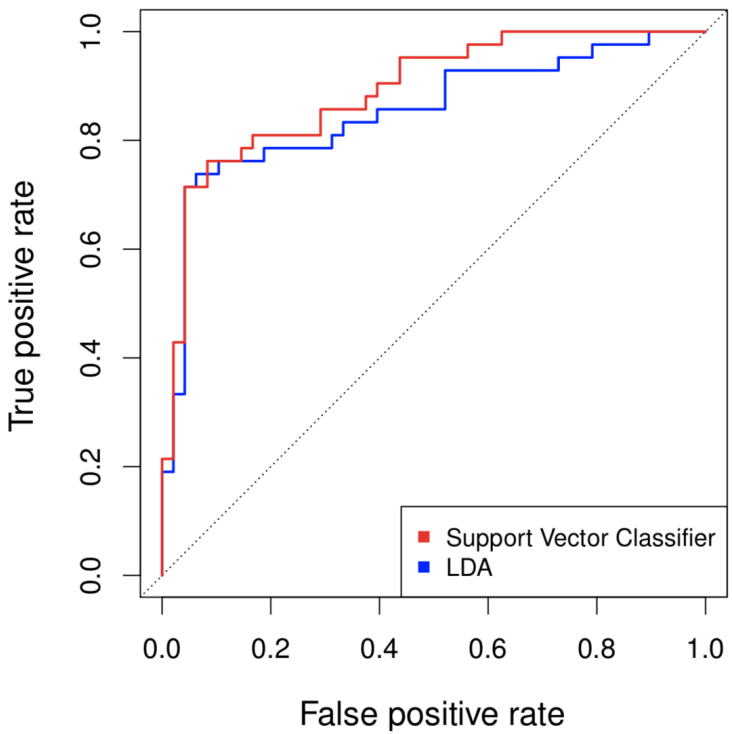
Larger γ leads to more flexible decision boundary.

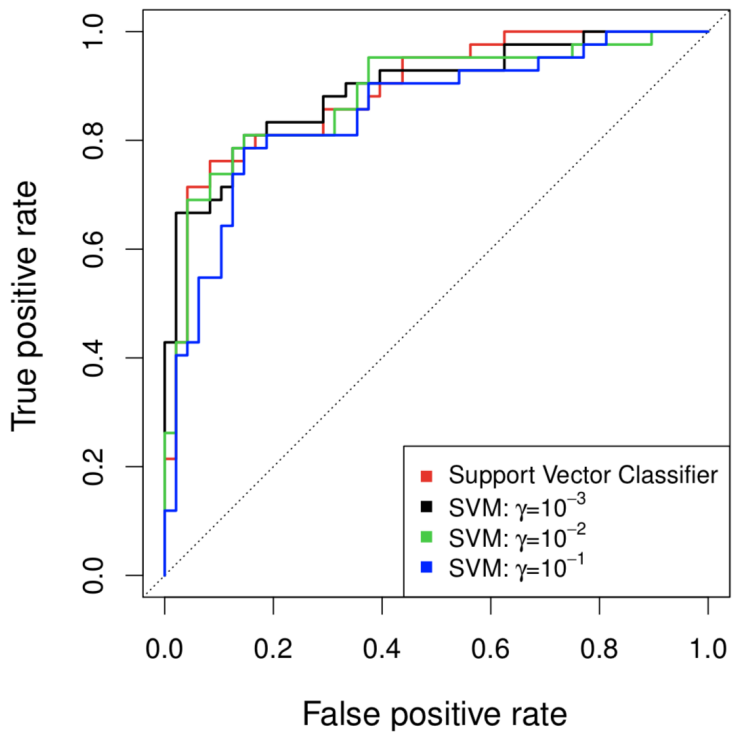
Example: Heart Data



ROC curve is obtained by changing the threshold 0 to threshold t in $\hat{f}(X) > t$, and recording **false positive** and **true positive** rates as t varies. Here we see ROC curves on training data.

Example continued: Heart Test Data





SVMs: more than 2 classes?

The SVM as defined works for $K = 2$ classes. What do we do if we have $K > 2$ classes?

- OVA One versus All. Fit K different 2-class SVM classifiers $\hat{f}_k(x), k = 1, \cdots, K$; each class versus the rest. Classify x^* to the class for which $\hat{f}_k(x^*)$ is largest.
- OVO One versus One. Fit all $\binom{K}{2}$ pairwise classifiers $\hat{f}_k(x)$. Classify x^* to the class that wins the most pairwise competitions.

Which to choose? If K is not too large, use OVO.

Which to use: SVM or Logistic Regression or LDA

- When classes are (nearly) separable, SVM and LDA are better than LR.
- When not, LR, LDA and SVM very similar.
- If you wish to estimate probabilities, use LR or LDA.
- For nonlinear boundaries, kernel SVMs are popular. Can use kernels with LR and LDA as well, but computations are more expensive.
- When sample size is large, kernel SVMs can be very slow.

Finished Thursday 7/10/22 @ 5:7 much work