QBUS3820: Statistical Learning and Data Mining

Lectures 6: Classification I

Semester 1, 2019

Discipline of Business Analytics, The University of Sydney Business School

Lectures 6: Classification I

1. Maximal Margin Classifier
2. Support Vector Classifier
3. Support Vector Machines
4. Naïve Bayes classifier
5. Decision theory for binary classification
6. Model evaluation for binary classification

Support Vector Machines

Support Vector Machines (SVM) are an approach to classification developed by computer scientists in 1990s.

SVMs have grown to be very popular in the machine learning community, as they have been found to perform well in a variety of settings.

Support Vector Machines

- The support vector machine is a generalisation of a simple and intuitive classifier known as the maximal margin classifier, which we introduce in the next section. Though intuitive, the maximal margin classifier only applies to a specific setting (in which we can separate the classes with a linear boundary).

- A support vector classifier (SVC) extends the maximal margin classifier to a broader range of settings.

- Support vector machines further extends the SVC to accommodate nonlinear class boundaries.

Maximal Margin Classifier

Separating hyperplanes

In mathematics, a $p-$dimensional **hyperplane** is defined by the equation

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

A point X with coordinates $X_1, X_2, ..., X_p$ lies on the hyperplane if and only if it satisfies the equation.

E.g. In two dimensions the hyperplane is a line:

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 = 0$$

Separating hyperplanes

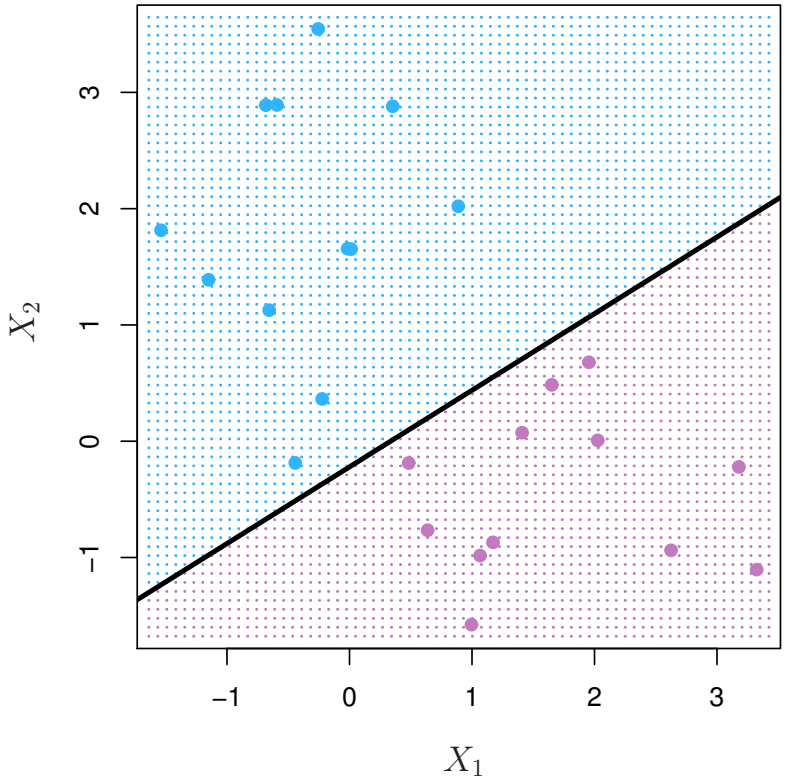
If $\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$ is **non-zero**, then the sign of this value determines which side of the hyperplane the point X lies on.

$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p > 0$ means X falls on one side of the hyperplane, and

$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p < 0$ means X falls on the other side of the hyperplane

Thus, the hyperplane divides the X space into two halves.

Separating hyperplanes (a 2-d example shown)



We classify each observation depending on which side of the hyperplane it lies.

Separating hyperplanes

Consider a binary classification setting:

The training data is $\{y_i, x_{i1}, ..., x_{ip}\}_{i=1}^n$ with $y_i \in \{-1, 1\}$.

A separating hyperplane will have the property that, for all i,

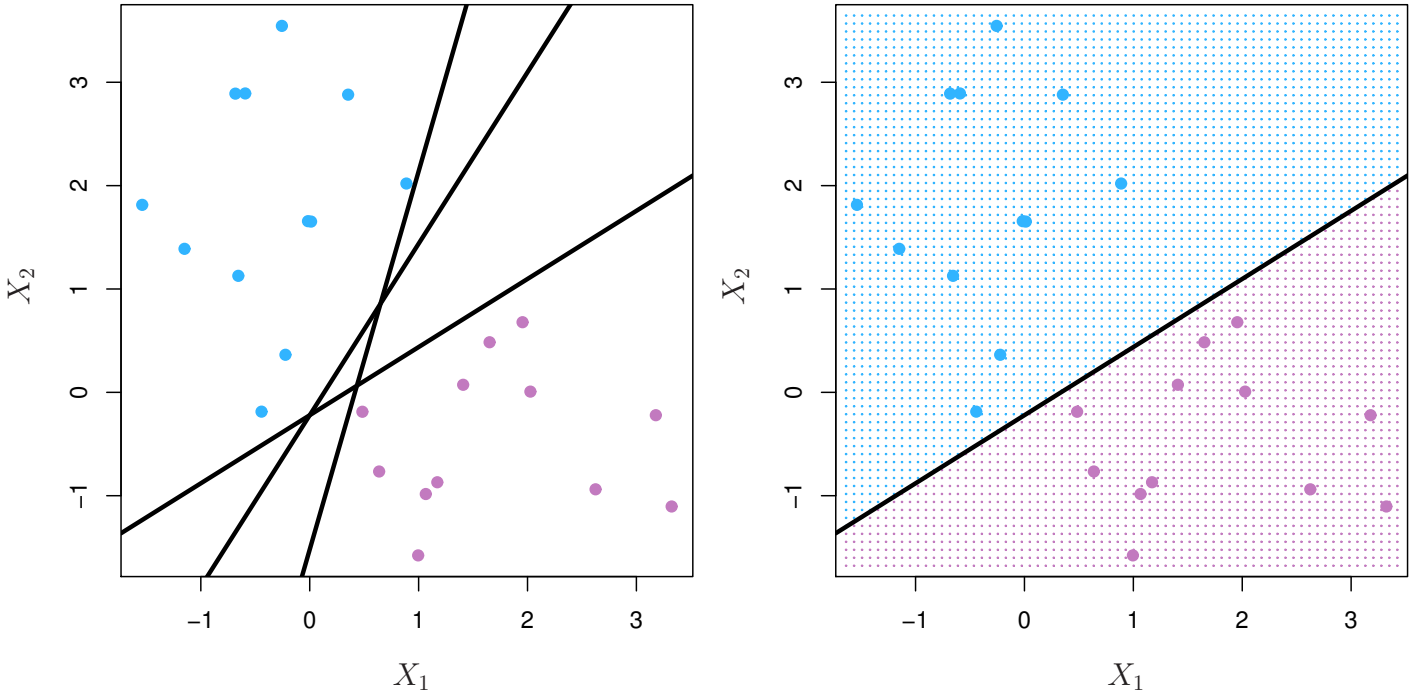
$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} > 0$ if $y_i = 1$ and

$\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip} < 0$ if $y_i = -1$.

In other words, for all i,

$$y_i\Big(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_p x_{ip}\Big) > 0$$

Separating hyperplanes (figure from ISL)

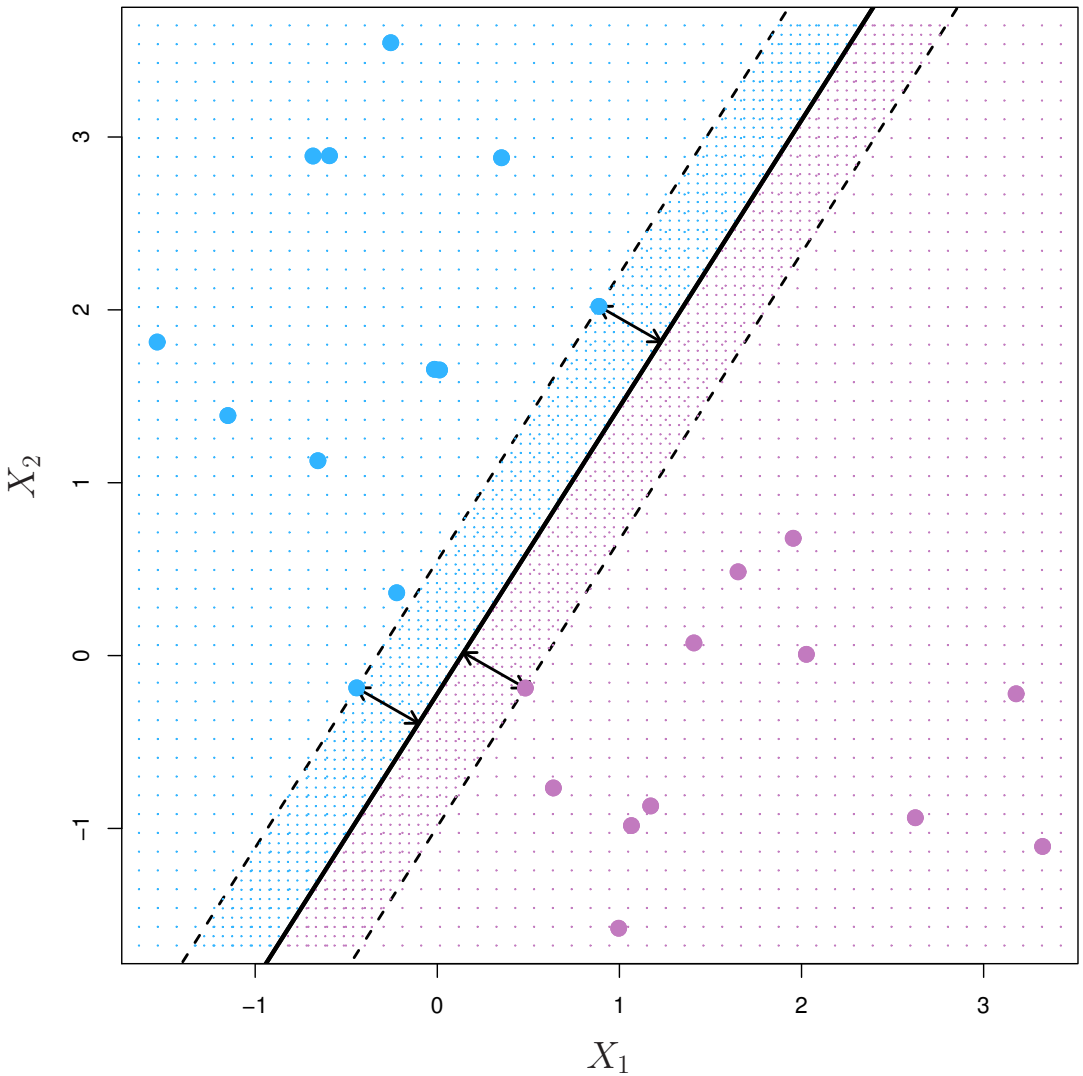


If our data can be perfectly separated using a hyperplane, then there are infinitely many such hyperplanes. In order to construct a classifier, we need a way to decide which hyperplane to use.

Maximal margin hyperplane

- An intuitive choice is the **maximal margin hyperplane** or **optimal separating hyperplane**, which is the separating hyperplane that is farthest from the training data.
- That is, compute the distance from each training observation to a given separating hyperplane. The smallest such distance is the minimal distance between the training observations and the hyperplane, known as the **margin**.
- The maximal margin hyperplane is the separating hyperplane with largest possible margin.

Maximal margin hyperplane



The points on the margin are the **support vectors**.

Maximal margin classifier: mathematical formulation

$$\max_{\boldsymbol{\beta}} \; M$$

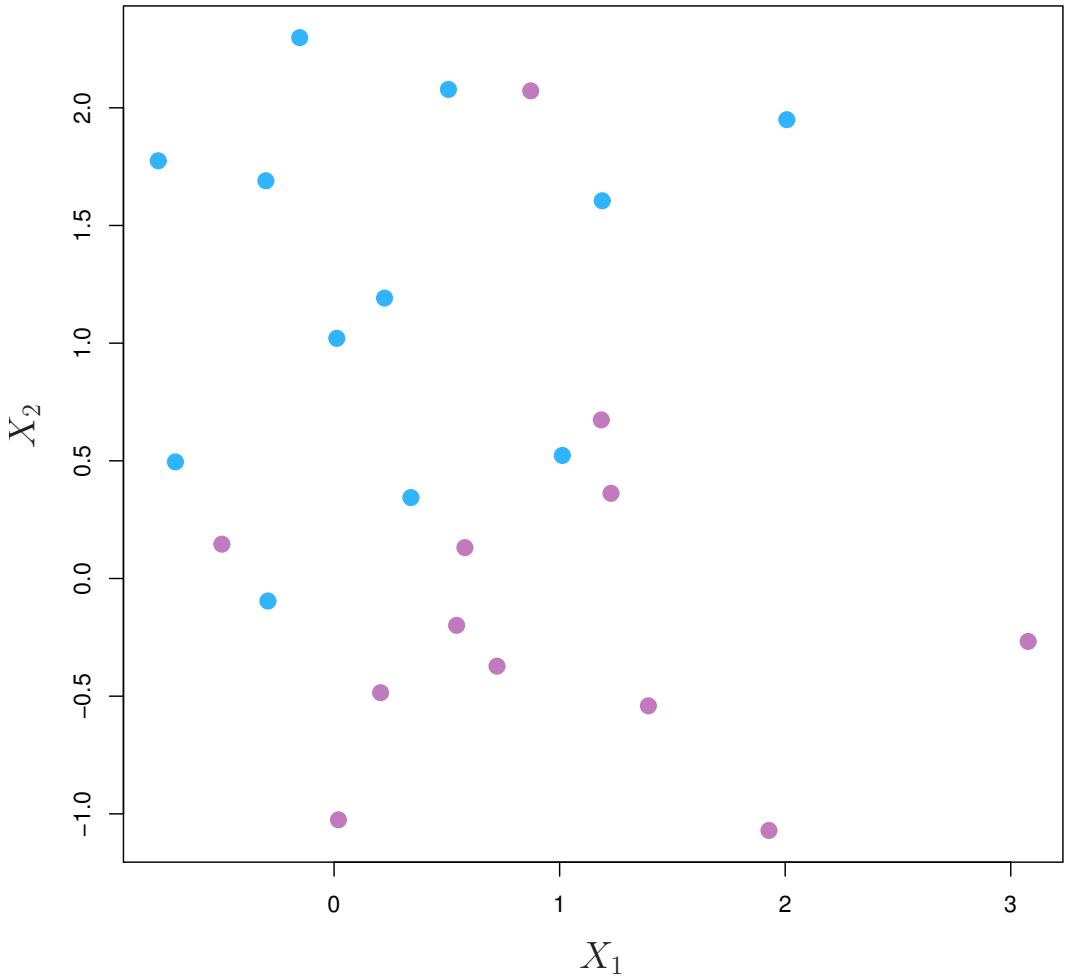
$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) \geq M \;\; i = 1, \ldots, n$$

We maximise the margin under the constraints that ensure:

- Each observation is on the correct side of the hyperplane (each training observation is correctly classified).
- Each observation is at least at a distance M from the hyperplane.

Non-separable data



No separating hyperplane exists in this case.

Support Vector Classifier

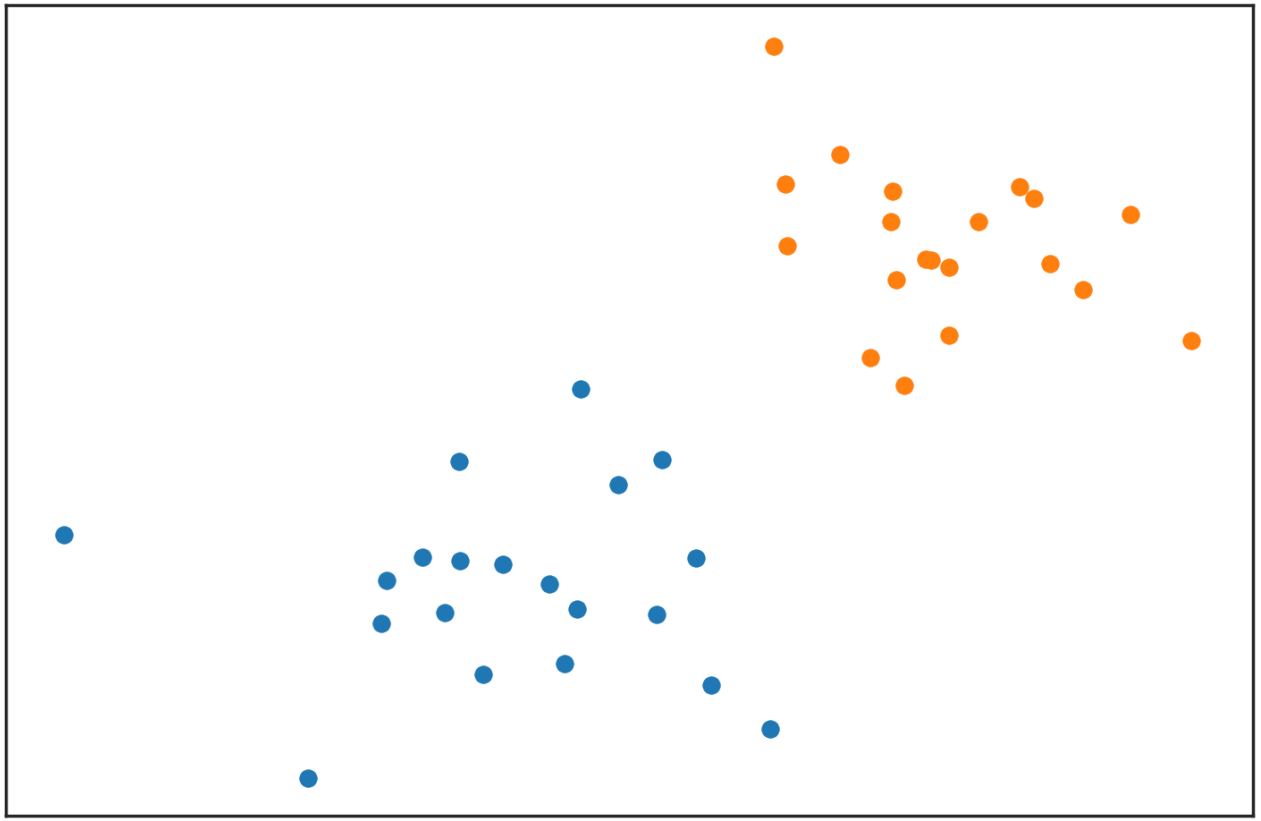
Support Vector Classifier

The **support vector classifier** or **soft margin classifier** extends the previous idea by allowing some observations to be inside the margin, or even misclassified, in the interest of achieving:

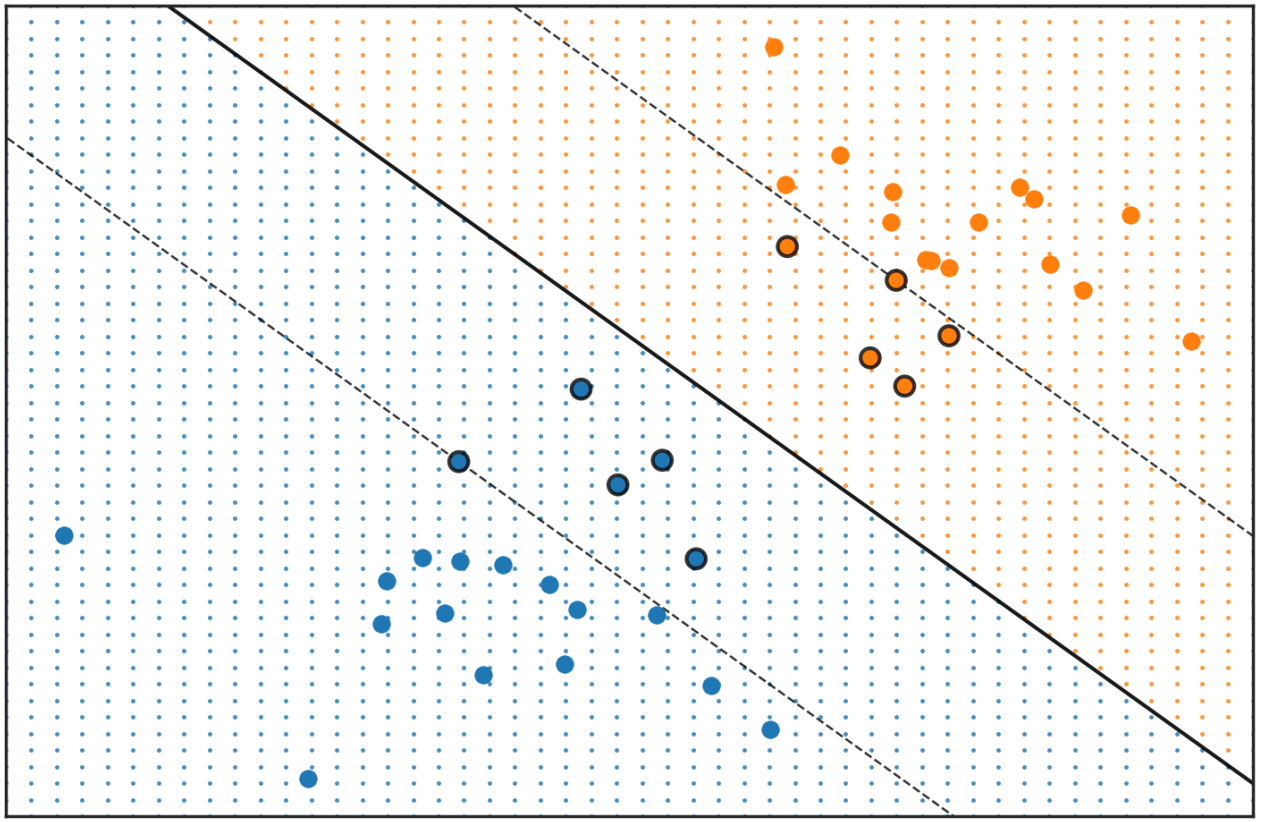
- Greater robustness to individual observations.
- Better classification.

Furthermore, it applies to data that is not linearly separable.

Support Vector Classifier



Support Vector Classifier



Support vector classifier: mathematical formulation

$$\max_{\boldsymbol{\beta},\epsilon_1,\ldots,\epsilon_n} \quad M$$

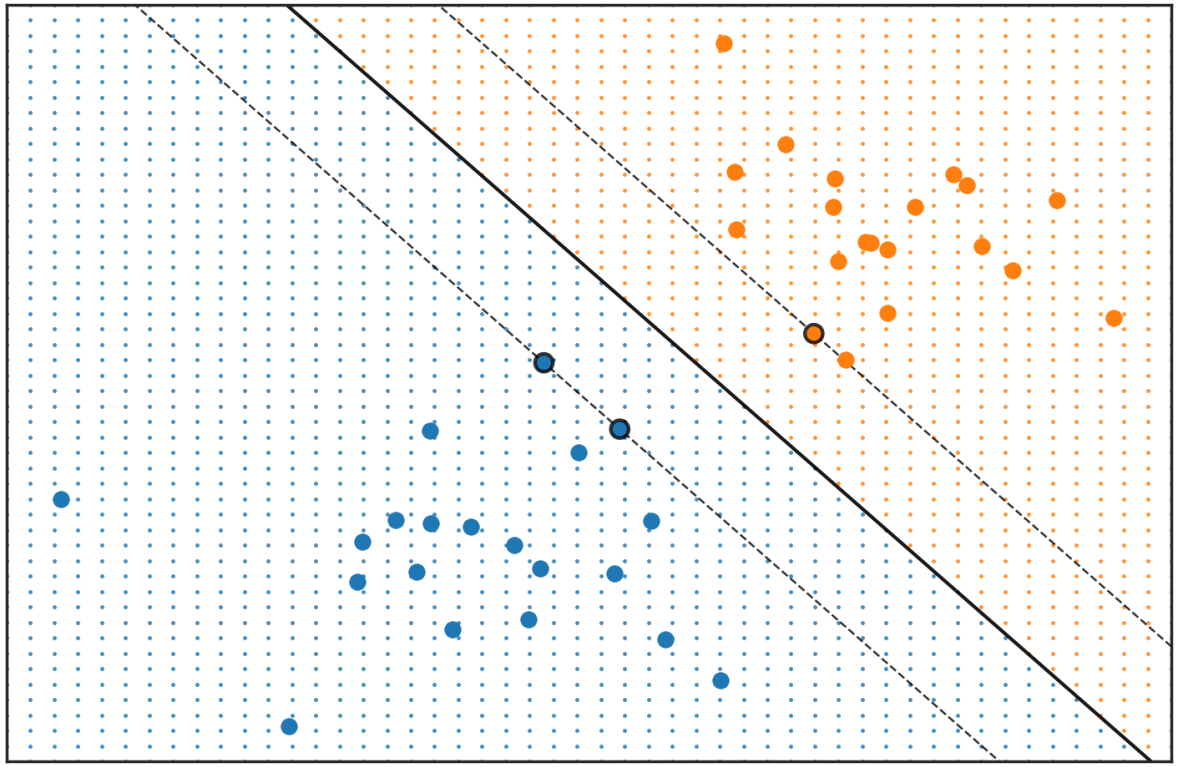
$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i \left(\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right) \geq M(1-\epsilon_i) \quad i = 1, \ldots, n,$$

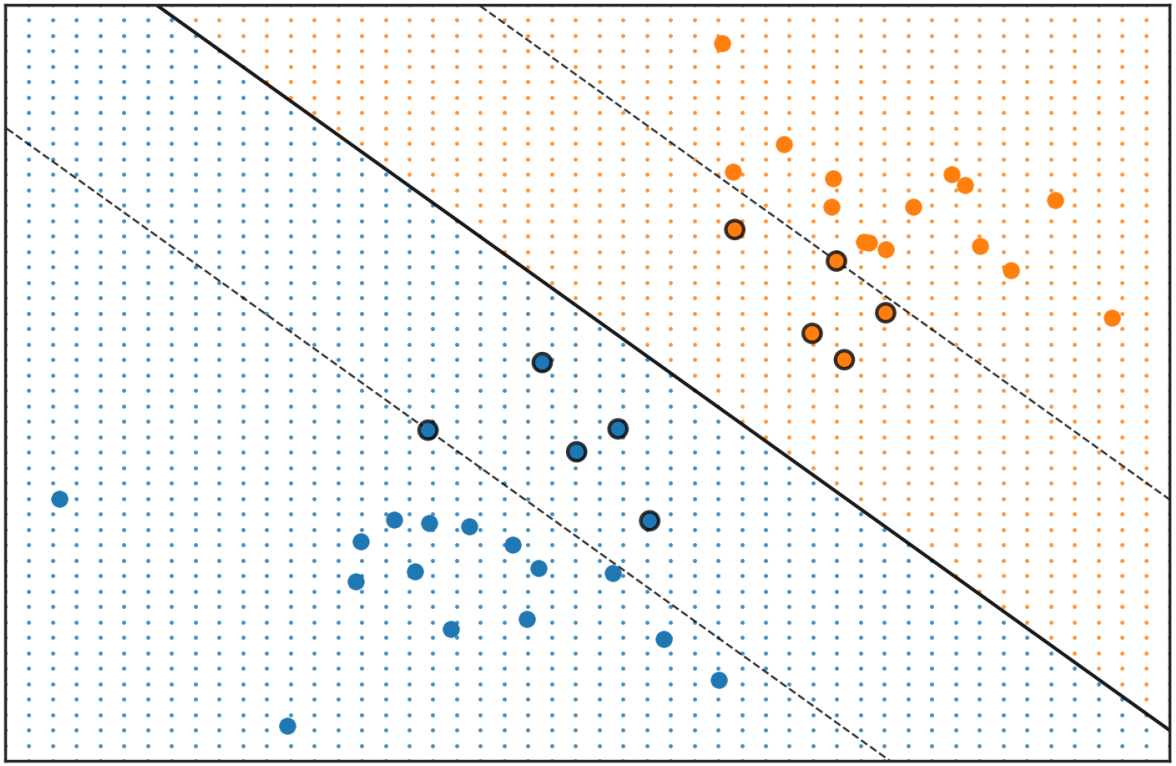
$$\epsilon_i \geq 0, \ \sum_{i=1}^{n} \epsilon_i \leq V,$$

where V is a tuning parameter, which we can interpret as a budget for margin violations.

Maximal margin classifier ($V = 0$)

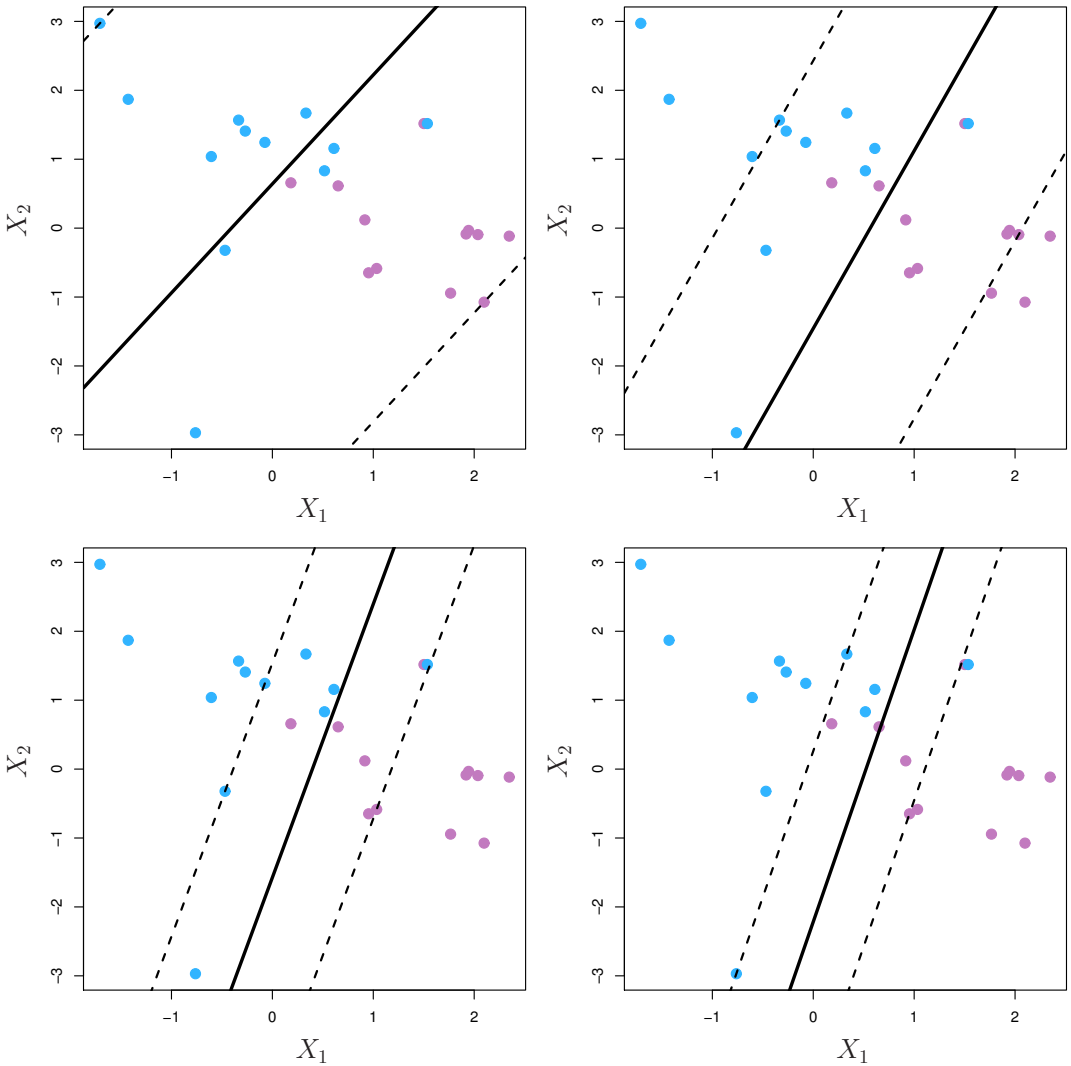


Support Vector Classifier ($V > 0$)



SVC has the fundamental property that an observation that lies strictly on the correct side of the margin does not affect the hyperplane. Only the support vectors (highlighted) do.

SVC: tuning parameter



SVC for different values of V (largest at the top left).

SVC: tuning parameter

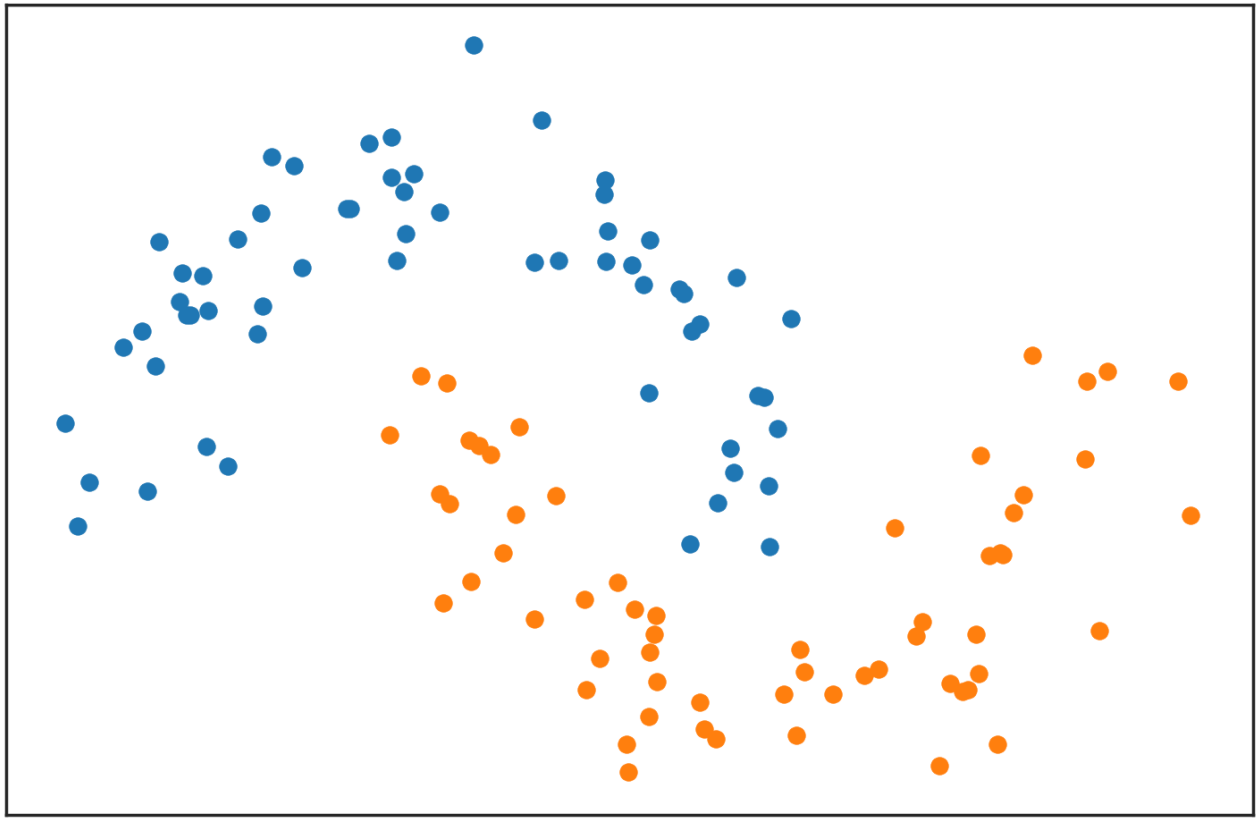
- When V (the budget of margin violations) is large, the margin is wide, many observations violate the margin. This type of classifier has low variance (since many observations are support vectors) but potentially high bias.

- Conversely, if V is small then there will be fewer support vectors, leading to low bias but high variance.

Support Vector Machines

Classification with nonlinear decision boundaries

In applications, we often face nonlinear class boundaries.



Enlarging the feature space

One option is to enlarge the feature space with polynomial functions of the predictors. For example, instead of using features

$$X_1, X_2, \ldots, X_p,$$

we can fit a support vector classifier using

$$X_1, X_1^2, X_2, X_2^2, \ldots, X_p, X_p^2.$$

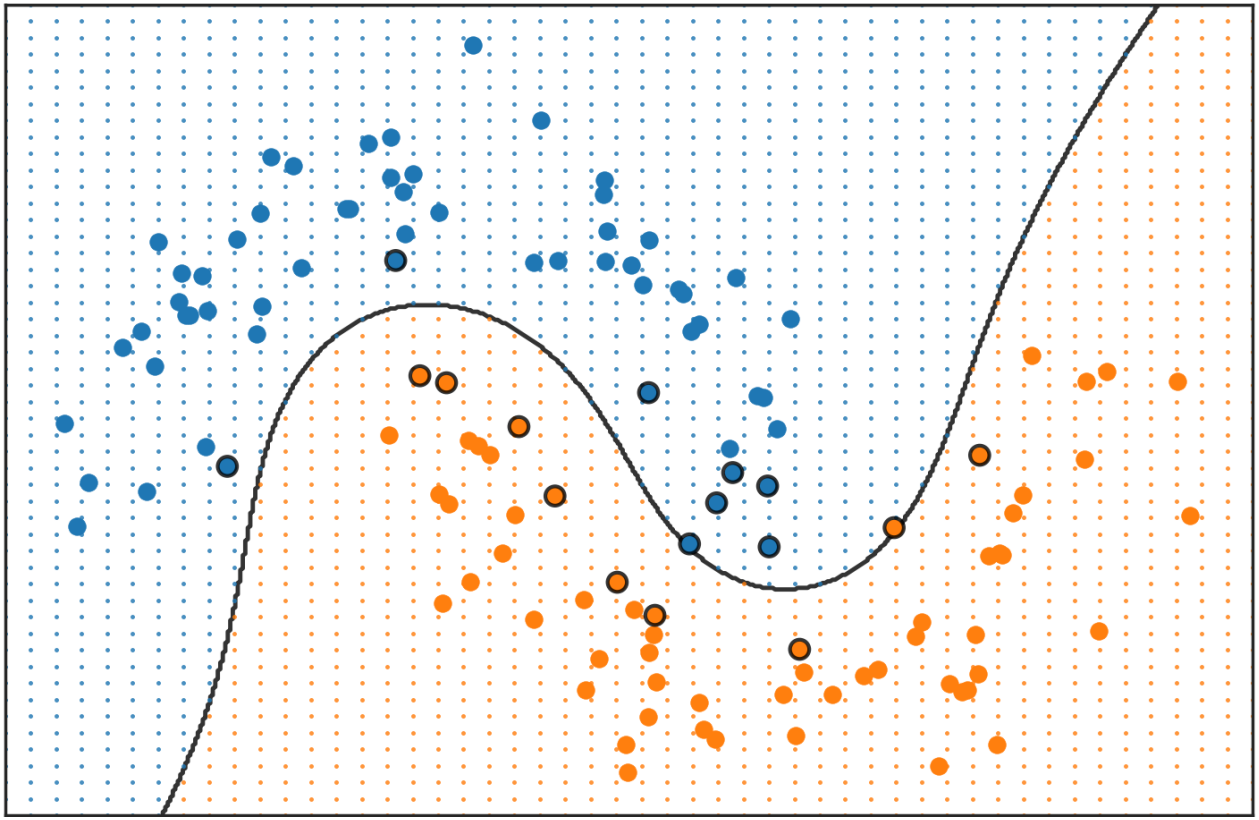
That will lead to a quadratic decision boundary in the original predictor space.

Support Vector Machines (SVM)

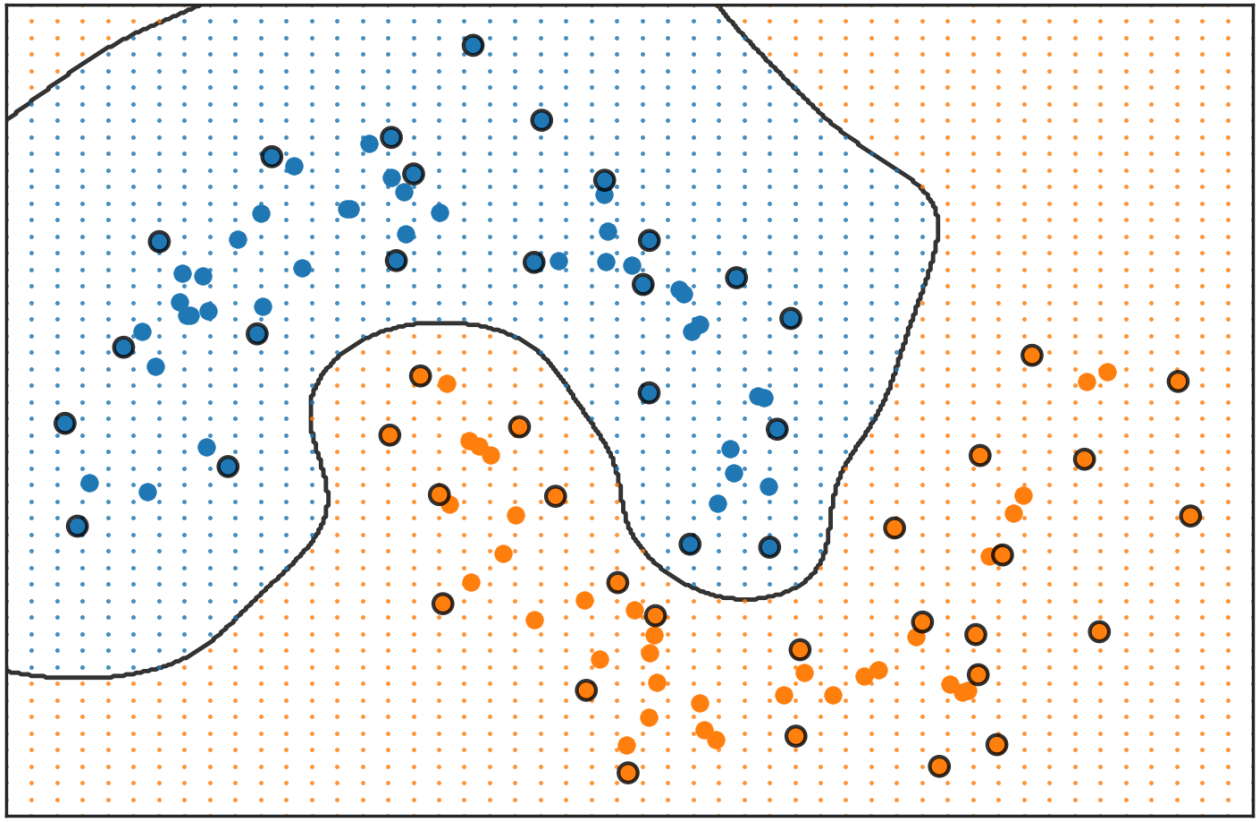
Support vector machines are an extension of the support vector classifier that enlarges the feature space in a specific way using *kernels*, leading to nonlinear decision boundaries.

The details are quite technical and outside the scope of our class.

SVM classifier with a polynomial kernel



SVM classifier with a Gaussian radial kernel



Naïve Bayes classifier

Notation

Suppose that Y takes values in the set $\{1, 2, \ldots, C\}$.

Define $\pi_c = P(Y = c)$.

If X is continuous, write $p(\boldsymbol{x}|Y = c)$ for the density of X conditional on $Y = c$.

If X is discrete, let $p(\boldsymbol{x}|Y = c)$ denote the conditional probability $P(X = x|Y = c)$.

Bayes' rule

Let X be discrete and recall that we defined $\pi_c = P(Y = c)$.

Recall the **Bayes rule** or **Bayes theorem** gives:

$$P(Y = c|X = \boldsymbol{x}) = \frac{P(X=\boldsymbol{x}|Y=c)\pi_c}{P(X=\boldsymbol{x})}$$

$$= \frac{P(X = \boldsymbol{x}|Y = c)\pi_c}{P(X = \boldsymbol{x}|Y = 1)\pi_1 + P(X = \boldsymbol{x}|Y = 2)\pi_2 + \ldots + P(X = \boldsymbol{x}|Y = C)\pi_C}$$

Using our notation, we can write the last expression as:

$$\frac{p(\boldsymbol{x}|Y = c)\pi_c}{p(\boldsymbol{x}|Y = 1)\pi_1 + p(\boldsymbol{x}|Y = 2)\pi_2 + \ldots + p(\boldsymbol{x}|Y = C)\pi_C}$$

Bayes' rule

Now let X be continuous and recall that we write $p(\boldsymbol{x}|Y = c)$ for the density of X conditional on $Y = c$.

Similarly to the discrete case, the Bayes rule gives:

$$P(Y = c|X = \boldsymbol{x})$$

$$= \frac{p(\boldsymbol{x}|Y = c)\pi_c}{p(\boldsymbol{x}|Y = 1)\pi_1 + p(\boldsymbol{x}|Y = 2)\pi_2 + \ldots + p(\boldsymbol{x}|Y = C)\pi_C}$$

Example: medical test

Consider a medical test for cancer. Suppose that the test has a sensitivity of 80%, which means that if a person has cancer, the test will return positive with probability 0.8:

$$P(X = 1|Y = 1) = 0.8.$$

Here X is the outcome of the test and Y is the indicator of the presence of cancer.

In case of a positive test result, what is the probability that a person has cancer, i.e. what is $P(Y = 1|X = 1)$?

Example: medical test

Using Bayes' theorem:

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

$$= \frac{P(X = 1|Y = 1)P(Y = 1)}{P(X = 1|Y = 1)P(Y = 1) + P(X = 1|Y = 0)P(Y = 0)}$$

This equation tells us that in order to calculate the desired probability, we also need to know the prevalence of cancer, i.e. $P(Y = 1)$, and the false positive rate: $P(X = 1|Y = 0)$.

Example: medical test

Suppose that $P(Y = 1) = 0.004$ and $P(X = 1|Y = 0) = 0.1$.

Also, recall that we assumed $P(X = 1|Y = 1) = 0.8$. Then:

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

$$= \frac{P(X = 1|Y = 1)P(Y = 1)}{P(X = 1|Y = 1)P(Y = 1) + P(X = 1|Y = 0)P(Y = 0)}$$

$$= \frac{0.8 \times 0.004}{0.8 \times 0.004 + 0.1 \times 0.996} = 0.031$$

Naïve Bayes classifier

In the classification setting, we refer to $p(\boldsymbol{x}|Y = c)$ as the class conditional densities (or probability mass functions if X is discrete), and we refer to $\pi_c = P(Y = c)$ as class probabilities.

The Naïve Bayes classifier method uses the general approach of modeling the conditional distribution of X given Y, and then using the Bayes' rule to obtain the conditional distribution of Y given X:

$$P(Y = c|X = \boldsymbol{x})$$

$$= \frac{p(\boldsymbol{x}|Y = c)\pi_c}{p(\boldsymbol{x}|Y = 1)\pi_1 + p(\boldsymbol{x}|Y = 2)\pi_2 + \ldots + p(\boldsymbol{x}|Y = C)\pi_C}$$

Naïve Bayes classifier

The **Naïve Bayes classifier** (NBC) is based on the assumption that the predictors are conditionally independent given the class label.

Thus, the class conditional density (or probability mass function if X is discrete) factorises into a product of the individual predictor densities:

$$p(\boldsymbol{x}|Y = c) = \prod_{j=1}^{p} p(x_j|Y = c).$$

Naïve Bayes classifier

- The method is "naive" because we do not think that the features are in fact conditionally independent.
- The simplicity of the NBC method makes it relatively immune to overfitting, which is useful for applications where the number of features is large.
- The assumption of conditional independence makes it easy to mix and match different predictor types.

Naïve Bayes classifier

- Despite being based on an assumption that is not necessarily true, the Naïve Bayes classifier often performs very well in practice compared to more complex alternatives.

- The reason is again the bias-variance trade-off: while the assumption of class-conditional independence may lead to biased probabilities, the simplifications brought by it may lead to substantial savings in variance.

Continuous predictors

For real-valued predictors, a common assumption is that:

$$X_j|Y = c \quad \sim \quad N(\mu_{jc}, \sigma^2_{jc})$$

where μ_{jc} and σ^2_{jc} are the mean and the variance of predictor j conditional on the class c.

Hence:

$$p(x_j|Y = c) = \frac{1}{\sqrt{2\pi\sigma^2_{jc}}} \exp\left(-\frac{(x_j - \mu_{jc})^2}{2\sigma^2_{jc}}\right)$$

Parameters μ_{jc} and σ^2_{jc} need to be estimated from the data.

Continuous predictors

- Typically, we first transform the predictors in order to make the variables approximately normal or symmetric.

- We could also use other distributional assumptions or follow a nonparametric approach to estimate the class conditional densities.

Binary predictors

When the predictors are binary, i.e. X_j only takes values 0 and 1, we use the Bernoulli distribution:

$$X_j|Y = c \quad \sim \quad \text{Bernoulli}(\theta_{jc})$$

where θ_{jc} is the probability that $X_j = 1$ given $Y = c$
(and, thus, $1 - \theta_{jc}$ is the probability that $X_j = 0$ given $Y = c$)

Parameters θ_{jc} need to be estimated from the data.

Application: document classification

Document classification is the problem of classifying text documents into different categories.

A simple approach is to represent each document as a vector of binary variables, where each variable records whether a particular word is present in the document or not. For example, $x_{ij} = 1$ if the word j appears in document i, and $x_{ij} = 0$ otherwise.

This is called a **bag of words** model.

Estimating Naïve Bayes parameters using maximum likelihood

We estimate the parameters in the Naïve Bayes model by maximum likelihood. In particular, we:

1. Estimate the prior class probabilities π_c by computing the sample proportions of each class in the training data.

2. Fit univariate models and estimate the parameters separately for each predictor within each class (the fact that we can do this is a direct consequence of the assumption of conditional independence).

the next two slides give some mathematical details, but you will **not** need to reproduce them on the exam

Estimating Naïve Bayes parameters using maximum likelihood

Let $\boldsymbol{\theta}$ contain all the parameters for the class conditional densities of the predictors, and let $\boldsymbol{\pi}$ contain the class probabilities for Y.

The density (or probability) for observation i is

$$\begin{aligned} p(\boldsymbol{x}_i, y_i; \boldsymbol{\theta}, \boldsymbol{\pi}) &= p(\boldsymbol{x}_i | y_i; \boldsymbol{\theta}) p(y_i; \boldsymbol{\pi}) \\ &= \prod_{j=1}^{p} p(x_{ij} | y_i; \boldsymbol{\theta}_j) p(y_i; \boldsymbol{\pi}) \end{aligned}$$

Thus, the likelihood is:

$$\ell(\boldsymbol{\theta}, \boldsymbol{\pi}) = \prod_{i=1}^{n} \prod_{j=1}^{p} p(x_{ij} | y_i; \boldsymbol{\theta}_j) p(y_i; \boldsymbol{\pi}).$$

Estimating Naïve Bayes parameters: class probabilities

The likelihood is:

$$\ell(\boldsymbol{\theta}, \boldsymbol{\pi}) = \prod_{i=1}^{n} \prod_{j=1}^{p} p(x_{ij}|y_i; \boldsymbol{\theta}_j) p(y_i; \boldsymbol{\pi}).$$

Hence, the log-likelihood is:

$$L(\boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{j=1}^{p} \sum_{i=1}^{n} \log\left(p(x_{ij}|y_i; \boldsymbol{\theta}_j)\right) + \sum_{i=1}^{n} \log\left(p(y_i; \boldsymbol{\pi})\right)$$

Note that the log-likelihood decomposes into a sum of terms, each involving a different parameter: $\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_p$, and $\boldsymbol{\pi}$. We can therefore maximize all these terms separately.

Estimating Naïve Bayes parameters using maximum likelihood

As a consequence:

$$\widehat{\pi}_c = \frac{n_c}{n}$$

where n is the sample size, as always, and $n_c = \sum_{i=1}^{n} I(y_i = c)$ is the number of observations that fall in class c.

Estimating Naïve Bayes parameters: binary predictors

Suppose that the predictors are binary, such that

$$X_j|Y = c \quad \sim \quad \text{Bernoulli}(\theta_{jc})$$

The MLE of each parameter $\theta_{jc} = P(X_j = 1|Y = c)$ is:

$$\widehat{\theta}_{jc} = \frac{n_{jc}}{n_c}$$

where $n_{jc} = \sum_{i=1}^{n} I(x_{ij} = 1)I(y_i = c)$ is the number of observations that fall in class c for which predictor j equals 1.

Estimating Naïve Bayes parameters: Gaussian case

Suppose that the class conditional distribution is Gaussian:

$$X_j | Y = c \quad \sim \quad N(\mu_{jc}, \sigma^2_{jc})$$

The MLEs are:

$$\widehat{\mu}_{jc} \quad = \quad \frac{1}{n_c} \sum_{i:\, y_i = c} x_{ij}$$

i.e. the sample mean of predictor X_j using just the subjects from class c

$$\widehat{\sigma}^2_{jc} \quad = \quad \frac{1}{n_c} \sum_{i:\, y_i = c} (x_{ij} - \widehat{\mu}_{jc})^2$$

i.e. the "sample variance" of X_j using just the subjects from class c

Decision theory for binary classification

Classification outcomes

In most business problems, there are distinct losses associated with each classification outcome. Consider, for example, the case of transaction fraud detection.

		Classification	
		Legitimate	**Fraud**
Actual	**Legitimate**	No loss	Investigation cost
	Fraud	Fraud loss	Fraud loss avoided

The cost of investigating a suspicious transaction is likely to be much lower than the loss in case of fraud.

Classification outcomes

We will use the following terminology.

		Classification	
		$\widehat{y} = 0$	$\widehat{y} = 1$
Actual	$Y = 0$	True negative	False positive
	$Y = 1$	False negative	True positive

Loss matrix

The context of the business problem will often specify a **loss matrix** or **cost-benefit matrix** for classification as follows.

		Classification	
		$\widehat{y} = 0$	$\widehat{y} = 1$
Actual	$Y = 0$	L_{TN}	L_{FP}
	$Y = 1$	L_{FN}	L_{TP}

Example: credit scoring

In credit scoring, we want to classify a loan applicant as creditworthy $(Y = 1)$ or not $(Y = 0)$ based on the probability that the customer will not default.

		Classification	
		$\widehat{y} = 0$	$\widehat{y} = 1$
Actual	$Y = 0$	Default loss avoided	Default loss
	$Y = 1$	Profit opportunity lost	Profit

A false positive is a more costly error than a false negative in this business scenario. Our decision making should therefore take this into account.

Decision rule

The decision to classify a subject as positive or negative is based on the following decision rule:

$$\widehat{y}(\boldsymbol{x}) = \begin{cases} 1 \text{ if } \widehat{P}(Y=1|X=\boldsymbol{x}) > \tau. \\ 0 \text{ if } \widehat{P}(Y=1|X=\boldsymbol{x}) \leq \tau. \end{cases}$$

Here $\widehat{P}$ corresponds to the estimated conditional probability, and τ is a decision threshold parameter. Recall that in the case of binary classification with the zero-one loss we use $\tau = 0.5$.

Optimal decision

It has been shown (but we will not go into the proof) that the optimal value of the threshold (the one minimising expected loss) is:

$$\tau^* = \frac{L_{\mathsf{FP}} - L_{\mathsf{TN}}}{L_{\mathsf{FP}} + L_{\mathsf{FN}} - L_{\mathsf{TP}} - L_{\mathsf{TN}}}$$

Example: zero-one loss

With the zero-one loss, we have that $L_{\mathsf{FP}} = L_{\mathsf{FN}} = 1$ and $L_{\mathsf{TP}} = L_{\mathsf{TN}} = 0$, i.e. the loss matrix is:

		Classification	
		$\widehat{y} = 0$	$\widehat{y} = 1$
Actual	$Y = 0$	0	1
	$Y = 1$	1	0

$$\tau^* = \frac{L_{\mathsf{FP}} - L_{\mathsf{TN}}}{L_{\mathsf{FP}} + L_{\mathsf{FN}} - L_{\mathsf{TP}} - L_{\mathsf{TN}}} = \frac{1}{2} \quad \text{as before}$$

Example: credit scoring

In the credit scoring example, we can set the loss matrix as:

		Classification	
		$\widehat{y} = 0$	$\widehat{y} = 1$
Actual	$Y = 0$	0	L_{FP}
	$Y = 1$	L_{FN}	0

where L_{FN} equals missed profit and L_{FP} equals default loss.

$$\tau^* = \frac{L_{\mathsf{FP}} - L_{\mathsf{TN}}}{L_{\mathsf{FP}} + L_{\mathsf{FN}} - L_{\mathsf{TP}} - L_{\mathsf{TN}}} = \frac{L_{\mathsf{FP}}}{L_{\mathsf{FP}} + L_{\mathsf{FN}}}.$$

Example: credit scoring

We've shown that the optimal threshold for the loan decision is:

$$\tau^* = \frac{L_{\mathsf{FP}}}{L_{\mathsf{FN}} + L_{\mathsf{FP}}}.$$

We expect the loss from default to be much higher than the profit from a loan to a creditworthy customer (L_{FP} much larger than L_{FN}). Note that this leads to a high value of the threshold τ^*.

In other words, it is only worth it to lend to customers that have a high probability of repayment.

Model evaluation for binary classification

Confusion matrix

A **confusion matrix** counts the number of true negatives, false positives, false negatives, and true positives for the test data.

		Classification		
		$\widehat{y} = 0$	$\widehat{y} = 1$	Total
Actual	$Y = 0$	True negatives (TN)	False positives (FP)	N
	$Y = 1$	False negatives (FN)	True positives (TP)	P
	Total	Negative predictions	Positive predictions	

True positive and true negative rates

The **true positive rate** (a.k.a. **sensitivity** or **recall**) is:

$$\frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{\text{True positives}}{\text{Actual positives}} \approx P(\widehat{y} = 1 | Y = 1)$$

The **true negative rate** (a.k.a. **specificity**) is:

$$\frac{\text{TN}}{\text{TN} + \text{FP}} = \frac{\text{True negatives}}{\text{Actual negatives}} \approx P(\widehat{y} = 0 | Y = 0)$$

False positive and false negative rates

The **false positive rate** is

$$\frac{\text{FP}}{\text{TN} + \text{FP}} = \frac{\text{False positives}}{\text{Actual negatives}} = 1 - \text{Specificity} \approx P(\widehat{y} = 1 | Y = 0)$$

The **false negative rate** is

$$\frac{\text{FN}}{\text{TP} + \text{FN}} = \frac{\text{False negatives}}{\text{Actual positives}} = 1 - \text{Sensitivity} \approx P(\widehat{y} = 0 | Y = 1)$$

Decision rule

Recall that the decision to classify a subject as positive or negative is based on the following decision rule:

$$\widehat{y}(\boldsymbol{x}) = \begin{cases} 1 \text{ if } \widehat{P}(Y=1|X=\boldsymbol{x}) > \tau. \\ 0 \text{ if } \widehat{P}(Y=1|X=\boldsymbol{x}) \leq \tau. \end{cases}$$

Trade-off between true positive and true negative rates

- There is a trade-off between the true positive and true negative rates, since a classifier can always obtain the maximum true positive (negative) rate by setting $\tau = 0$ ($\tau = 1$) and automatically returning all positives (negatives).

- Equivalently, there is a trade-off between achieving a higher true positive rate and achieving a lower false positive rate.

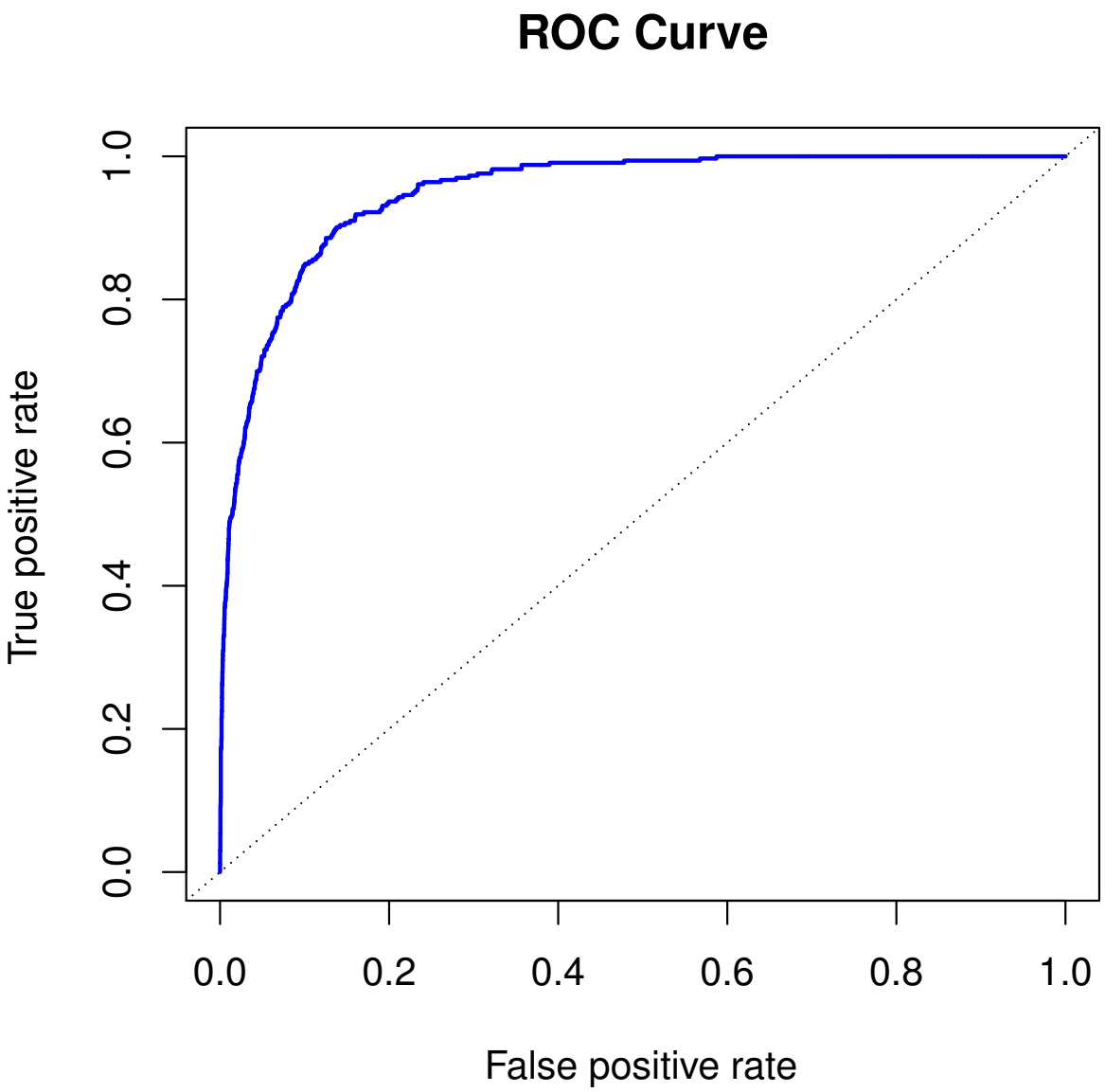
ROC curve

A **receiver operating characteristic** or **ROC** curve plots the true positive rate against the false positive rate for a range of threshold values τ.

ROC plots tell us the false positive rate that we need to accept if we want to obtain a particular level of the true positive rate.

We often summarise the quality of ROC curve as a single number using the **area under the curve** or **AUC**. Higher AUC scores are better, and the highest possible AUC value is one.

ROC curve


ROC Curve
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0

Imbalanced classes

Many classification scenarios (such as fraud detection) concern rare events, leading to a very large proportion of negatives in the data.

In this situation we say that the classes are highly **imbalanced**.

The true negative rate is not very informative for these problems, as it will tend to be high regardless of the quality of the classifier.

Precision

In the imbalanced scenario, we are usually more interested in the proportion of detections that are actually positive. We define the **precision** as

$$\frac{\mathsf{TP}}{\mathsf{TP}+\mathsf{FP}} = \frac{\text{True positives}}{\text{Positive classifications}} \approx P(Y=1|\widehat{y}=1)$$

Review questions

- In what way does the support vector classifier extend the maximal margin classifier?
- In what way does the support vector machine extend the support vector classifier?
- What is the key assumption of the Naive Bayes classifier?
- What is a confusion matrix? Write down what the matrix looks like.
- What are true positive rate, true negative rate, and precision?
- Why is there a trade-off between the true positive rate and the true negative rate?