Machine Learning (CP322)

Winter 2026

Statistical Learning

what is it?, models, regression, classification, prediction, inference, accuracy, bias, variance

Presented by

Sukhjit Singh Sehra

Wilfrid Laurier University
Department of Computer Science

LAURIER
WILFRID LAURIER UNIVERSITY

Overview

- Is statistical learning = machine learning?
- Models
- Prediction & inference
- Accuracy & interpretability
- Bias & Variance
- Regression & classification

This sets the stage for the remainder of the course.

Machine Learning?

> *“Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence.”*
>
> — *Wikipedia page on Machine Learning*

> *“"Statistical learning" redirects here.”*
>
> — *Wikipedia page on Machine Learning*

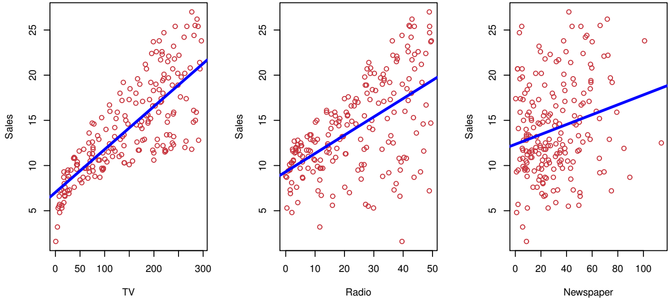
Machine Learning is just a fancy new name. It sure sounds cool!

Example: Advertising

- We are hired to increase the sales of a particular product.
- We are given the `Advertising` data set.
  - `sales` for 200 different markets.
  - advertising budgets for `TV`, `radio` and `newspaper`.
- Our clients can *not directly* influence the sales.
- But they *can* control the advertising budgets.
- We need to determine whether there is an association between advertising and sales.

This is a typical scenario.

Example: Advertising



- The plots display `sales` as a function of the advertising budgets.
- Each plot is a *scatter plot* of `sales` versus a budget.
- We have overlaid a *least squares line* fit on each plot.

We can already spot some promising relationships.

Example: Advertising

- The advertising budgets are *input variables*.
- We denote the input variables with X.
- We use subscripts to distinguish them.
- For example, X_1 = `TV`, X_2 = `radio`, X_3 = `newspaper`.
- The input variables are also called *predictors*, *independent variables*, *features* or simply *variables*.
- `sales` is the *output variable*.
- Output variables are also called *dependent variables* or *responses*.
- We denote the output variables with Y.

Don't get confused by different names.

Models

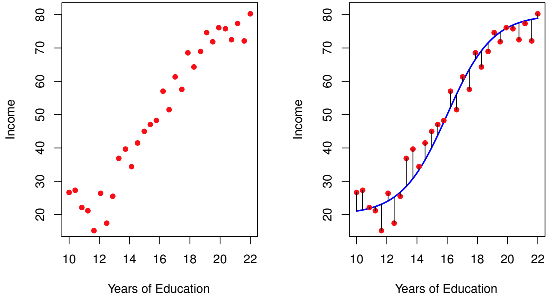
- Suppose we observe a *quantitative* response Y,
  and p different predictors, $X_1, X_2, \ldots X_p$.
- We assume that there is some relationship between Y and X.
- We can write this in the very general form

$$Y = f(X) + \epsilon$$

- Here f is a fixed but unknown function of $X = (X_1, X_2, \ldots, X_p)$.
- And ϵ is a random error term, independent of X with mean zero.
- We say f represents the *systematic* information
  that X provides about Y.

This is the most general form of a *model*.

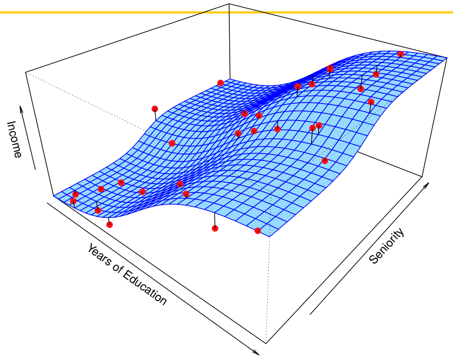
Example: Income



- The figure shows a scatter plot of `income` versus `years of education`.
- The blue curve is the true functional relationship $f(X) = f(X_1)$.
- The vertical lines represent the error terms ϵ.

This is a simulated data set.

Example: Income



- The figure shows a 3D scatter plot of `income` versus `years of education` and `seniority`.
- The blue surface is the true functional relationship $f(X) = f(X_1, X_2)$.
- The vertical lines represent the error terms ϵ.

Visualising higher dimensions is hard.

What is Statistical Learning?

In essence, statistical learning refers to to a set of approaches for estimating the function f.
In this lecture we cover the key theoretical concepts that arise in estimating f, as well as techniques for evaluating the estimates obtained.

This equally applies to machine learning.

Why estimate f?

There are two main reasons why we wish to estimate f.

1. *Prediction*: assuming the inputs X are readily available, we simply want ot predict
$$\hat{Y} = \hat{f}(X)$$
without being interested in the exact form of $\hat{f}$.
2. *Inference*: we want to understand the functional relationship
$$Y = Y(X_1, X_2, \ldots, X_p)$$
between the predictors and the response.

We discuss each in more detail.

Prediction

- Often the inputs X are readily available but the response Y is not.
- Since the error term ϵ averages to zero, we can then predict

$$\hat{Y} = \hat{f}(X)$$

  where $\hat{f}$ represents the estimate of f and $\hat{Y}$ is the resulting prediction of Y.
- In this scenario we are not interested in the exact form of $\hat{f}$.
- We are mostly concerned with the accuracy of $\hat{Y}$.

Here $\hat{f}$ is often treated as a *black box*. But we don't have to!

Prediction Accuracy

- The accuracy of $\hat{Y}$ depends on two quantities:
  1. the *reducible error*
  2. the *irreducible error*
- The reducible error is due to $\hat{f}$ not being a perfect estimate of f.
  (We can potentially come up with better statistical learning techniques to improve $\hat{f}$).
- Even if our estimate of the relationship was perfect ($\hat{f} = f$),
  our estimate $\hat{Y}$ would still have an error!
- This irreducible error is due to our *observations* not being perfect.
  (The error term ϵ).

You have to understand what you can improve and what you can't.

Prediction Accuracy

- Assuming that $\hat{f}$ and X are fixed, we can show that

$$\begin{aligned} E[(Y-\hat{Y})^2] &= E[(f(X)+\epsilon-\hat{f}(X))^2] \\ &= \underbrace{E[(f(X)-\hat{f}(X))^2]}_{\text{Reducible}} + \underbrace{\text{Var}(\epsilon)}_{\text{Irreducible}} \end{aligned}$$

  where $E[(Y-\hat{Y})^2]$ is the average, or *expected value*, of the squared difference between the predicted and the true value of Y, and $\text{Var}(\epsilon)$ is the *variance* of the error term ϵ.
- The average, or expected value, $E[z]$ is often written as $\overline{z}$ or $\langle z \rangle$.
- The variance is then $\text{Var}(z) = \langle (z-\overline{z})^2 \rangle = \langle z^2 \rangle - \langle z \rangle^2 = \overline{z^2} - \overline{z}^2$.

All techniques we learn aim to estimate f while minimising the reducible error.

Inference

- We are often interested in *how* $X_1, X_2, \ldots, X_p$ affect Y.
- We want to estimate f, but not necessarily predict Y.
- The aim is to understand the *functional relationship* between Y and X.

We can *not* treat $\hat{f}$ as a black box in this scenario.

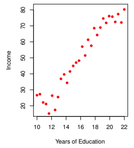
Inference: Questions

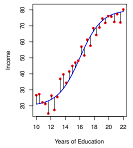
- *Which predictors are associated with the response?*
  We want to identify the *important* predictors.
- *What is the relationship between the response and each predictor?*
  For example, predictors can have positive or negative influence on the response. Any functional relationship can occur, possibly depending on the values of the other predictors.
- *Can the relationship between Y and each predictor be summarised as a linear equation?*
  It greatly simplifies things when the relationship can be captured by a linear model. Historically, models have been mostly chosen to be linear. However, the true relationships are very rarely linear.

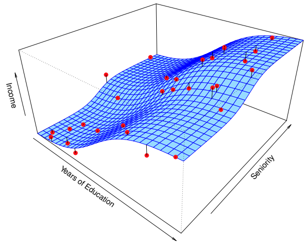
Linear is good, but rarely true.

How do we estimate f?

- We have n *observations* of X and Y.
- In the plots on the right n is 30.
- This is the *training data*.
- We use these observations to *train*, or *teach*, our model how to estimate f.
- The goal is to find a function $\hat{f}$ such that $Y \approx \hat{f}(X)$, for each observation (X, Y).







Statistical learning methods are either *parametric* or *non-parametric*.

Parametric Methods

1. *Make an assumption about the functional form of f.*
   A very simple assumption is a linear model:

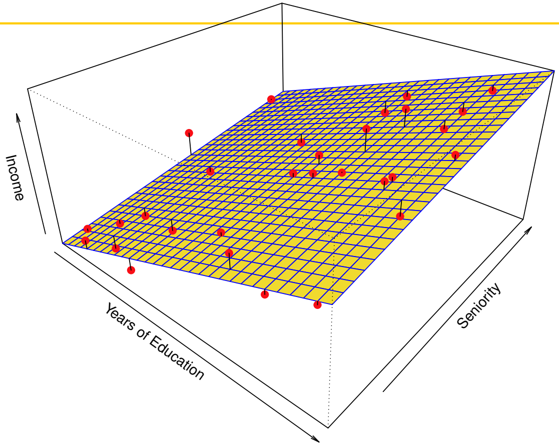
   $$f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

   With a linear model we only need to estimate $p + 1$ parameters. We will cover linear models in great detail in the next lecture.
2. *Use the training data to train, or fit, the model.*
   For a linear model we need to estimate the parameters $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$ such that:

   $$Y \approx \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \cdots + \hat{\beta}_p X_p$$

Non-linear models are of course possible, but harder to fit.

Linear Model Example



$$\texttt{income} \approx \hat{\beta}_0 + \hat{\beta}_1 \times \texttt{years of education} + \hat{\beta}_2 \times \texttt{seniority}$$

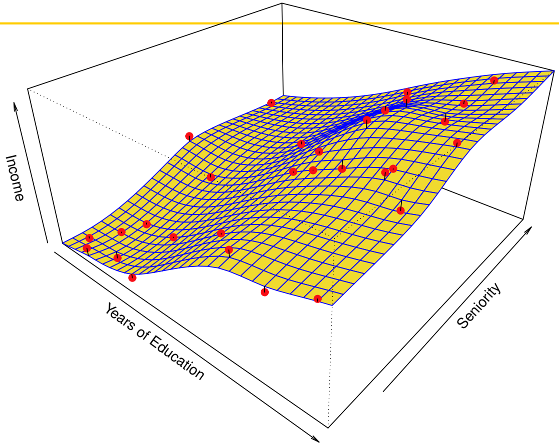
Non-parametric Methods

1. *No assumption about the functional form of f is made.*
2. *Seek an estimate of f that smoothly approaches the data points in the training data.*

+ Avoids choosing potentially very wrong models.
+ Can approximate complex functional relationships.
– Often needs many more observations for the fit.
– Prone to over-fitting.

Oddly, non-parametric methods can have a lot of parameters.

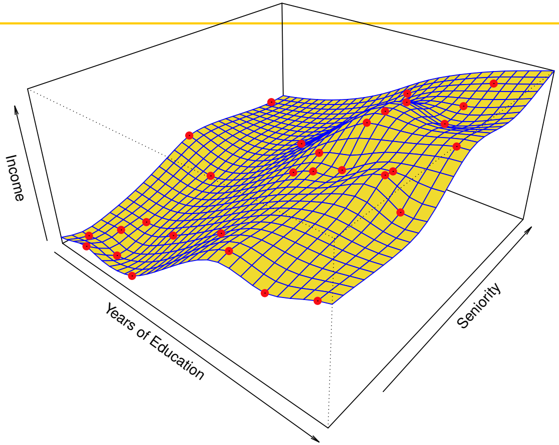
Non-parametric Example



Approximation to f to predict `income` using a *thin-plate spline*.

Better fit to the training data than the linear model.

Non-parametric Example



A less smooth spline fit that fits the training data perfectly.

This is an example of *over-fitting*!

Accuracy versus Interpretability Trade-off

Why would we choose a less flexible approach?

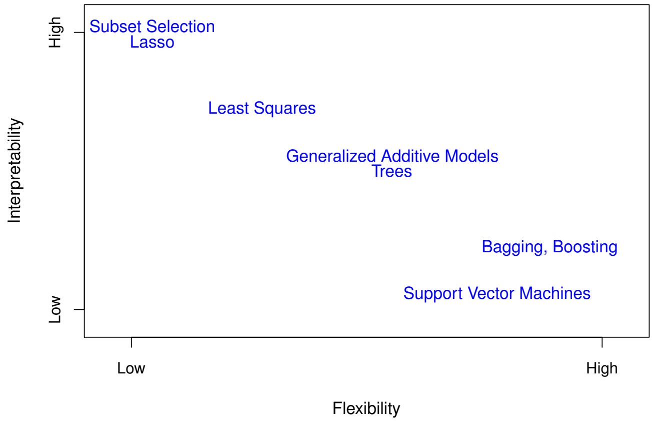
- Interpretability is often desirable (inference).
- For example, in a linear model we can directly infer whether a relationship is positive or negative from the sign of the β coefficient.

Why would we choose a more flexible approach?

- We might be mostly interested in the accuracy of our estimate of f (prediction).
- In the extreme, this is a *black box* approach where we don't care about the interpretation at all.

This is a *trade-off*, we have to decide.

Accuracy versus Interpretability Trade-off



In general, more flexible statistical learning methods have lower interpretability.

We'll learn more about the methods mentioned in the plot later

Supervised versus Unsupervised Learning

Supervised Learning

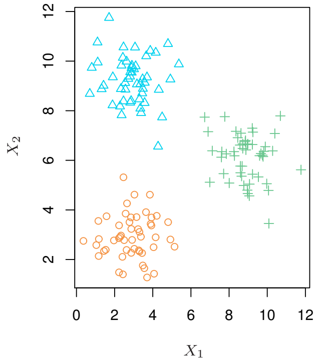
- We have n observations of the p predictors X_j *and* the corresponding response Y.
- In this case we can fit a model to the training data.
- The *supervision* comes in through the presence of the known response Y in the training data.

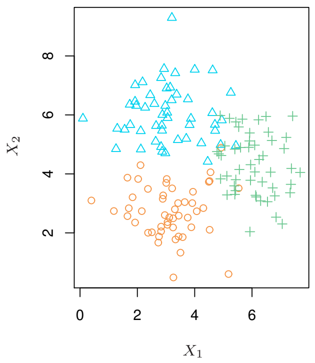
Unsupervised Learning

- We have n observations of the p predictors X_j, but *no access* to the corresponding response Y in the training data.
- All we can do is look for *patterns* in the training data.

This distinction is not always clear-cut.

Unsupervised Example





Clustering data involving three groups. Left: little overlap. Right: larger overlap.

We need to automate this for large data sets.

Regression versus Classification

- Variables, in particular responses, can be *quantitative* or *qualitative*.
  Quantitative Responses – Regression Problems
- Gas mileage, crime rate, sales.
  Qualitative Responses – Classification Problems
- Sex (male, female), default on debt (yes, no), brand (Apple, Samsung, Blackberry).

In terms of statistical learning methods the distinction is not crisp.

Assessing Model Accuracy

- In this course we introduce a wide range of statistical learning methods.
- Why are we doing this?
- Why not just teach the *best* method?
- No single method dominates all others over all possible data sets.
- It is *our* task to decide which method produces the best results on a given data set.
- We therefore need a way to *measure* how well we are doing.

There is no free lunch in statistical learning.

Measuring Fit Quality

- How can we measure the quality of a fit?
- We need to quantify how close our estimate $\hat{Y} = \hat{f}(X)$ is to the true values Y.
- Clearly, the difference between the two is a measure of this, but we don't care about the sign.
- We therefore use the *mean squared error*, MSE:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2 = E[(Y - \hat{f}(X))^2] = \langle (Y - \hat{f}(X))^2 \rangle$$

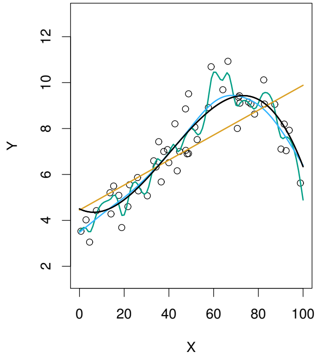
Other choices for the *loss function* are possible. This is the most straight-forward and common choice.

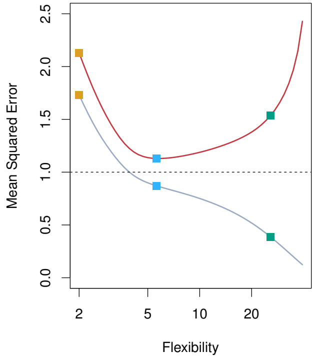
Measuring Fit Quality

- We could simply compute the MSE on the training data.
- Most fit, or learning, methods work by minimising the MSE (or another suitable *loss function*) on the training data set.
- However, we want to make predictions on *future* observations, not available at the time of training.
- A better measure of how well we are doing is obtained by computing the MSE on a *test data set* that is *independent* of the training data set.
- We can always produce an independent test data set by only using a subset of the available data for training and the remainder for testing (*cross validation*).

The MSE from the training data set overestimates the fit quality.

Measuring Fit Quality: Example 1

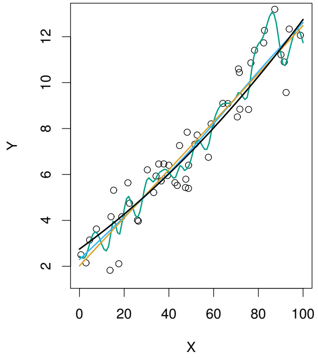


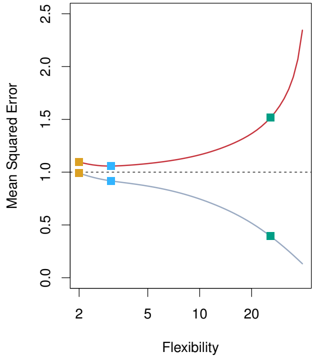


Left: true f (black), linear regression (orange), and spline fits (blue, green).
Right: training (grey) and test (red) MSE.

The most flexible model over-fits the training data.

Measuring Fit Quality: Example 2

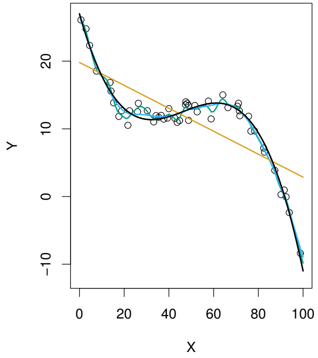


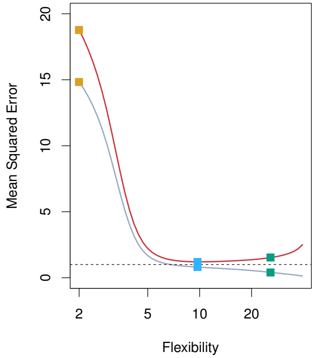


Left: true f (black), linear regression (orange), and spline fits (blue, green).
Right: training (grey) and test (red) MSE.

The truth is more linear, so linear regression does well.

Measuring Fit Quality: Example 3





Left: true f (black), linear regression (orange), and spline fits (blue, green).
Right: training (grey) and test (red) MSE.

The truth is highly non-linear, so linear regression does badly.

The Bias-Variance Trade-off

- We denote the observations from a test data set as (x_0, y_0).
- Then the test data set MSE is:

$$\text{MSE}_{\text{test}} = \langle (y_0 - \hat{f}(x_0))^2 \rangle$$

- We can then show that this can be decomposed like this:

$$\langle (y_0 - \hat{f}(x_0))^2 \rangle = \underbrace{\text{Var}(\hat{f}(x_0)) + (\text{Bias}(\hat{f}(x_0))^2}_{\text{Reducible}} + \underbrace{\text{Var}(\epsilon)}_{\text{Irreducible}}$$

where the *bias* is:

$$\text{Bias}(\hat{f}(x_0)) = \langle \hat{f}(x_0) \rangle - \langle f(x_0) \rangle$$

In practice, we don't know the true f and can't compute the bias properly.

The Bias-Variance Trade-off

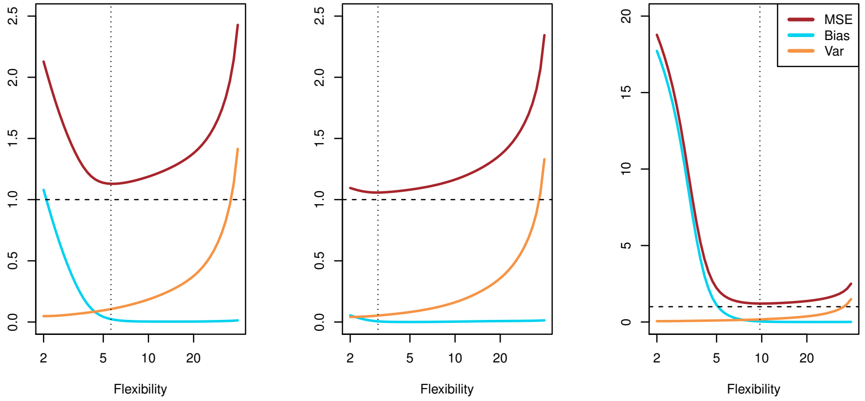
- Given the test MSE

$$\langle (y_0 - \hat{f}(x_0))^2 \rangle = \underbrace{\mathrm{Var}(\hat{f}(x_0)) + (\mathrm{Bias}(\hat{f}(x_0))^2}_{\text{Reducible}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{Irreducible}}$$

  we want to minimize the variance of $\hat{f}(x_0)$ *and* the bias.
- This is often not possible at the same time.
- Highly flexible methods can eliminate the bias.
- That does *not* mean they perform better at prediction.

The bias-variance trade-off translates to a flexibility trade-off.

The Bias-Variance Trade-off



The plots correspond to the three example data sets we have seen before. (Non-linear, almost linear, highly non-linear).

The Classification Setting

- As before, we want to estimate f from the training observations
$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$
and say we train a *classifier*.
- Now the responses y_i are *qualitative*.
- That is, the y_i are *labels* denoting the different *classes* the output variable can belong to.
- For example:
$$y = \texttt{origin}, \texttt{origin} \in \{\texttt{US}, \texttt{EU}, \texttt{JP}\}$$

Classifiers usually predict class *probabilities* from observations.

Classifier Accuracy

- The concepts we have developed for regression also apply to classification.
- We need a way to quantify the accuracy of classifiers.
- We define the *training error rate*:

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i) = \langle I(y_i \neq \hat{y}_i)\rangle, \;\; I(y_i \neq \hat{y}_i) = \begin{cases} 1 & y_i \neq \hat{y}_i \\ 0 & y_i = \hat{y}_i \end{cases}$$

- With the convention that (x_0, y_0) denotes the test observations, the *test error rate is*:

$$\langle I(y_0 \neq \hat{y}_0)\rangle$$

A good classifier minimises the test error rate.

The Bayes Classifier

- The best possible classifier minimises the test error rate on average.
- It assigns each observation to the most likely class given the observed predictors.
- We write the probability of Y belonging to class j, given the predictor observation x_0, as follows.

$$P(Y = j|X = x_0)$$

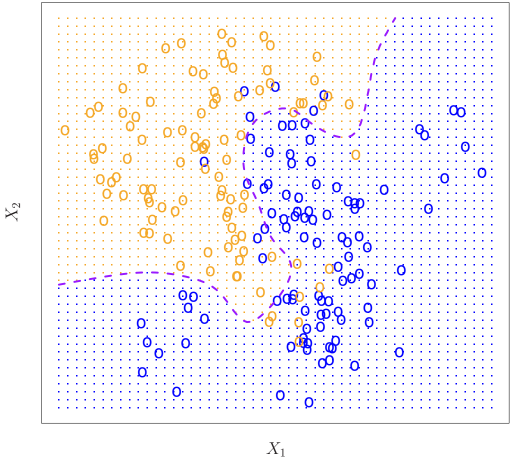
- In a scenario with two classes (say `Up` and `Down`) this classifier predicts $Y =$ `Up` if

$$P(Y = \texttt{Up}|X = x_0) > 0.5$$

  and $Y =$ `Down` otherwise.
- This is called the *Bayes Classifier* and it is ideal.
- On simulated data we can compute the above probability perfectly.

No method can do better than this proper Bayesian posterior probability.

The Bayes Decision Boundary



A simulated data set with 100 observations. The dashed line is the *Bayes decision boundary*.

K-Nearest Neighbours

1. Given an integer K and a test observation x_0, find the K observations x_i in the training data set that are closest to x_0, denoted by $\mathcal{N}_0$.
2. Then estimate the conditional probabilities to observe x_0 from the relative frequencies of the classes in $\mathcal{N}_0$ (*likelihoods*):

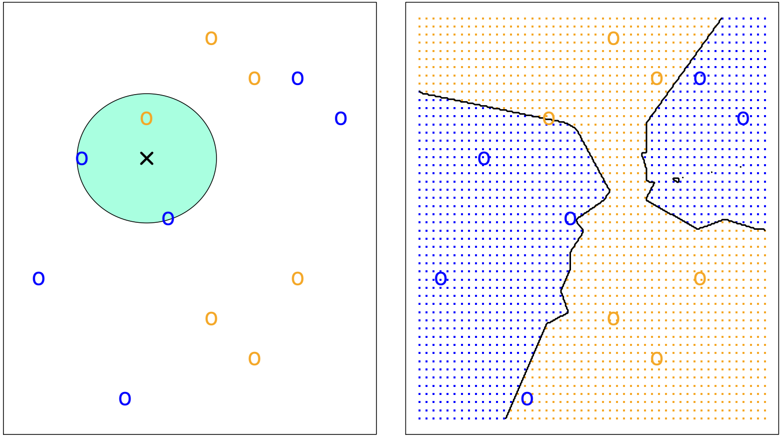
$$P(X = x_0 | Y = j) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j)$$

3. Finally, use Bayes' theorem to compute the conditional *posterior probability*:

$$P(Y = j | X = x_0) = \frac{P(X = x_0 | Y = j) \times P(Y = j)}{P(X = x_0)}$$

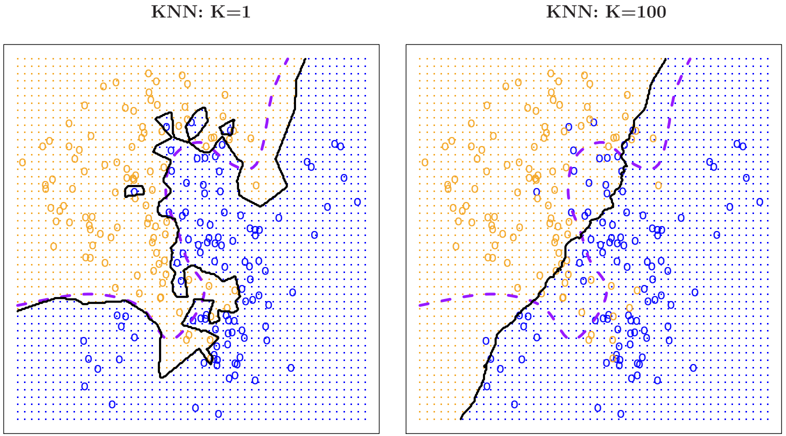
KNN *estimates* $P(Y|X)$. The *flexibility* of KNN scales with $1/K$.

K-Nearest Neighbours



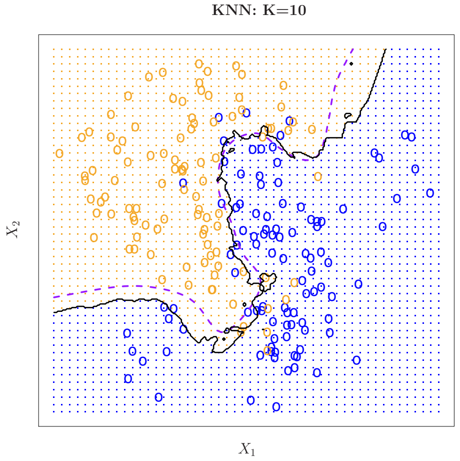
The KNN approach with $K = 3$. The black “x” in the left panel is a test observation.

K-Nearest Neighbours



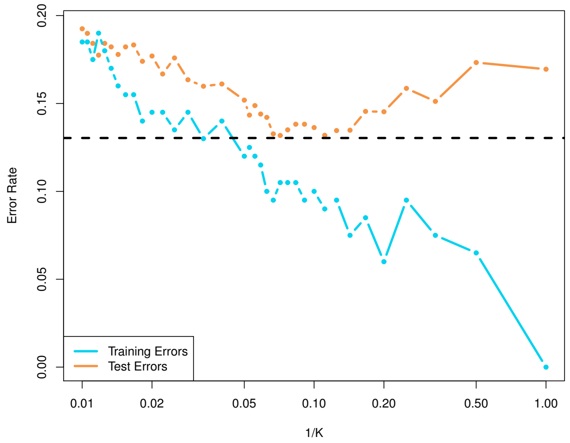
A comparison of decision boundaries with $K = 1$ and $K = 100$.

K-Nearest Neighbours



Comparing to the Bayes decision boundary, $K = 10$ seems a good choice.

K-Nearest Neighbours



Flexibility versus model performance on training and test data sets.

Questions?

Thank you!