# Classification: K-nearest neighbours and Overfitting

Alex Lewin

London School of Hygiene and Tropical Medicine

Spring 2022

# K-Nearest Neighbours

A different way of estimating $P(Y|X) = f(X)$

- Logistic regression $\rightarrow$ parametric:
  specify functional form of $f(X)$, find best fitting parameters
- K-nearest neighbours $\rightarrow$ non-parametric:
  $f(X)$ is estimated as averages over data points

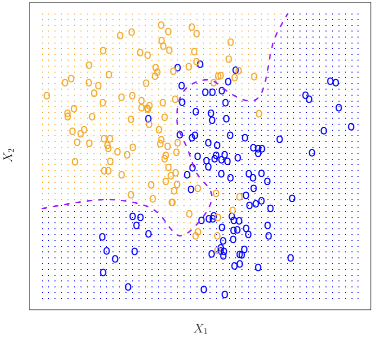
In K-NN, cannot write a simple equation for $f(X)$ in terms of $X$.

No inference here, only prediction.

# K-Nearest Neighbours

2D illustration: here $f(x_1, x_2)$ is going to be our prediction of $Y$ for a new point $(x_1, x_2)$

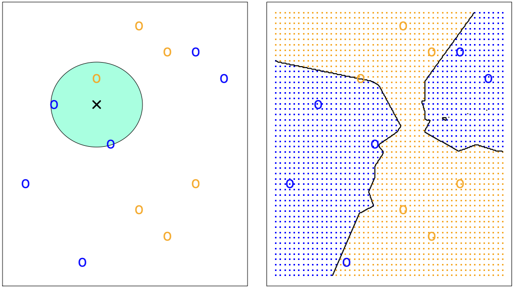
Idea is to use the data points which are similar in $X$-space to our new point.



# K-Nearest Neighbours

1. For the new point $(x_1, x_2)$, call observations "neighbours" if they have similar values of $(X_1, X_2)$.
2. Find the $K$ nearest neighbours of the point $(x_1, x_2)$.
3. This defines a neighbourhood $\mathcal{N}_0$ of $(x_1, x_2)$ in $X$-space.
4. Predict the class of the new point to be the most common class of observations in the neighbourhood.

2D example,
K=3



# K-Nearest Neighbours

Can write the predicted classification probability as

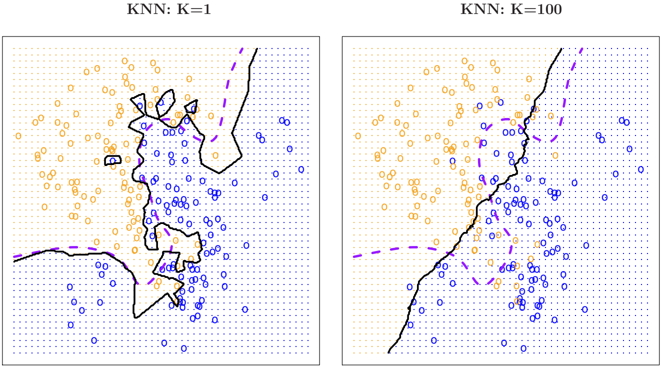
$$P(Y = j|X = x_0) = f(x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} \mathcal{I}[y_i = j]$$

where $\mathcal{N}_0$ is the set of nearest neighbours to $x_0$.

# K-Nearest Neighbours

Note that different choices of $K$ give different results:

- Larger $K$ smooths more (less flexible model, but more general)
- Smaller $K$ estimates are more local (more flexible, but less general)



# Model fit for classification

How do we assess model fit for classification models?

*Recall model fit in linear regression based on sums of squared residuals:*
*Mean square error* $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ *related to Residual mean squares.*

In classification we count the number of errors in prediction:

Simplest error rate:

Error rate $= \frac{1}{n}\sum_{i=1}^{n}\mathcal{I}[y_i \neq \hat{y}_i]$

However, there are 2 ways to make an error in classification ...

# Model fit for classification

There are 2 ways to make an error in classification: false positives, and false negatives.

| | Predicted $\hat{y}_i = 0$ | Predicted $\hat{y}_i = 1$ |
|---|---|---|
| Observed $y_i = 0$ | TN | FP |
| Observed $y_i = 1$ | FN | TP |

Sensitivity: TP/(TP+FN), how many of the real signals do we detect?
Specificity: TN/(TN+FP), how many of the real non-signals do we falsely declare?

Sensitivity $\longleftrightarrow$ false negatives
Specificity $\longleftrightarrow$ false positives

# Model fit for classification

There are 2 ways to make an error in classification: false positives, and false negatives.

| | Predicted $\hat{y}_i = 0$ | Predicted $\hat{y}_i = 1$ |
|---|---|---|
| Observed $y_i = 0$ | TN | FP |
| Observed $y_i = 1$ | FN | TP |

In machine learning:

Recall is the same as sensitivity: TP/(TP+FN)
Precision: TP/(FP+TP), out of the *declared signals*, how many are true?

Recall $\longleftrightarrow$ false negatives
Precision $\longleftrightarrow$ false positives

# Training and Test data

Error rate can be calculated on the data used to fit (train) the model:
$\rightarrow$ Training error.

The training error will be low (relatively) because this data was used to find the best model $\hat{f}(X)$.

Actually we are interested in estimating the error rate when this function $\hat{f}(X)$ is used to make predictions about **new observations**:
$\rightarrow$ this is called Test error.

The test error will in general be higher because this data was not used to find the best model $\hat{f}(X)$. But it gives a better idea of how general the model is (will it fit other data sets well).

# Training and Test data in practice

Ideally have validation data set to test the model on.

But usually split the data set into training and test data.

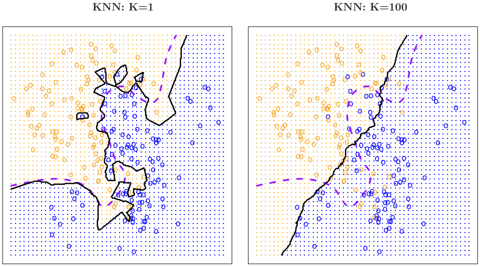
Training data: fit the model.

Test data: test the model predictions.

Back to K-NN:

Fit on training data:

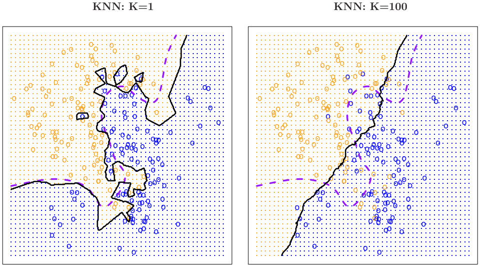
- Larger $K$ smooths more (less flexible model, but more general)
- Smaller $K$ estimates are more local (more flexible, but less general)



# Bias-variance tradeoff

Fit on training data:

- Larger $K$ more general
- Smaller $K$ more flexible




More general (less flexible) model $\longrightarrow$ Lower Variance, Higher Bias.

Less general (more flexible) model $\longrightarrow$ Lower Bias, Higher Variance.

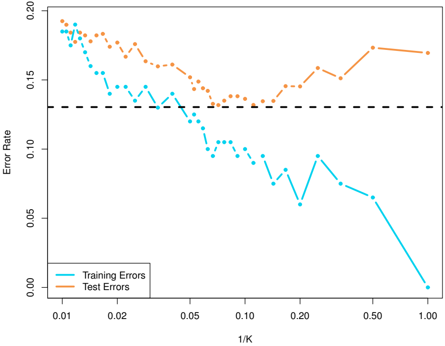
# Training and Test error

Training error keeps on decreasing as $K$ decreases.

Test error decreases at first: this shows model is getting better at fitting the test data.

When $K$ gets too small, test error increases: the training data is not representative of the test data - this is **over-fitting**.

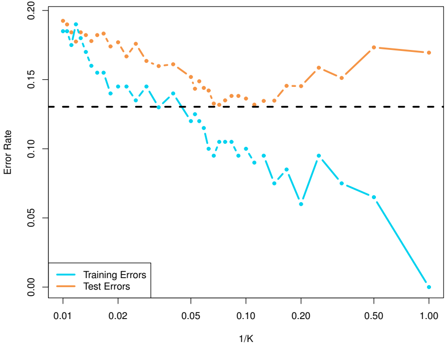
- Blue: training error, Orange: test error
- K decreases to the right



# Training and Test error

Test error has a characteristic U-shape: balance between over-fitting and under-fitting the data.

This helps us choose the optimal $K$ number of neighbours to use in the K-NN model.



# Reading

ISLR book (download a pdf from https://statlearning.com/):

Section 2.2.3 discusses K-NN method and bias-variance tradeoff for classification problems.

Figures in this presentation are taken from ISLR.