Nearest Neighbor Classification

Morteza H. Chehreghani
morteza.chehreghani@chalmers.se

Chalmers University of Technology

April 11, 2019

Reference

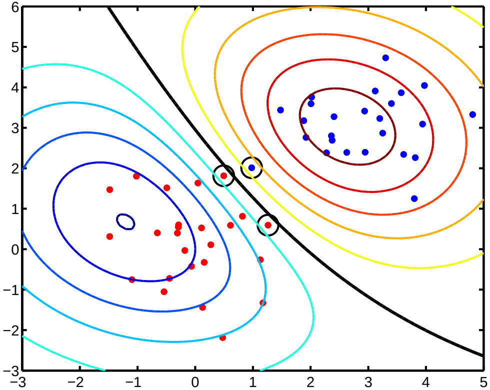
The content and the slides are adapted from

S. Rogers and M. Girolami, A First Course in Machine Learning (FCML), 2nd edition, Chapman & Hall/CRC 2016, ISBN: 9781498738484

Introduction

- Supervised learning
  - Regression
    - Minimised loss (least squares)
    - Maximised likelihood
    - Bayesian approach
  - **Classification**
- Unsupervised learning
  - Clustering
  - Projection

Classification



- A set of N objects with attributes (usually vector) $\mathbf{x}_n$.
- Each object has an associated response (or label) t_n.
- Binary classification: $t_n = \{0, 1\}$ or $t_n = \{-1, 1\}$,
  - (depends on algorithm).
- Multi-class classification: $t_n = \{1, 2, \ldots, K\}$.

Classification syllabus

- 4 classification algorithms.
- Of which:
  - 2 are probabilistic.
    - Bayes classifier
    - Logistic regression.
  - 2 are non-probabilistic.
    - K-nearest neighbours
    - Support Vector Machines.
- There are many others!

Probabilistic vs non-probabilistic classifiers

Classifier is trained on $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and $t_1, \ldots, t_N$ and then used to classify $\mathbf{x}_{\text{new}}$.

- Probabilistic classifiers produce a probability of class membership $P(t_{\text{new}} = k | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$
  - e.g. binary classification: $P(t_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$ and $P(t_{\text{new}} = 0 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$.
- Which to choose depends on application....

Probabilistic vs non-probabilistic classifiers

Classifier is trained on $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and $t_1, \ldots, t_N$ and then used to classify $\mathbf{x}_{\text{new}}$.

- Probabilistic classifiers produce a probability of class membership $P(t_{\text{new}} = k | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$
  - e.g. binary classification: $P(t_{\text{new}} = 1 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$ and $P(t_{\text{new}} = 0 | \mathbf{x}_{\text{new}}, \mathbf{X}, \mathbf{t})$.
- Non-probabilistic classifiers produce a hard assignment
  - e.g. $t_{\text{new}} = 1$ or $t_{\text{new}} = 0$.
- Which to choose depends on application....

Probabilistic vs non-probabilistic classifiers

- Probabilities provide us with more information – $P(t_{\text{new}} = 1) = 0.6$ is more useful than $t_{\text{new}} = 1$.
  - Tells us how **sure** the algorithm is.

Probabilistic vs non-probabilistic classifiers

- Probabilities provide us with more information – $P(t_{\text{new}} = 1) = 0.6$ is more useful than $t_{\text{new}} = 1$.
  - Tells us how **sure** the algorithm is.
- Particularly important where cost of misclassification is high and imbalanced.
  - e.g. Diagnosis: telling a diseased person they are healthy is much worse than telling a healthy person they are diseased.

Probabilistic vs non-probabilistic classifiers

- Probabilities provide us with more information – $P(t_{\text{new}} = 1) = 0.6$ is more useful than $t_{\text{new}} = 1$.
  - Tells us how **sure** the algorithm is.
- Particularly important where cost of misclassification is high and imbalanced.
  - e.g. Diagnosis: telling a diseased person they are healthy is much worse than telling a healthy person they are diseased.
- Extra information (probability) often comes at a cost.
- For large datasets, might have to go with non-probabilistic.

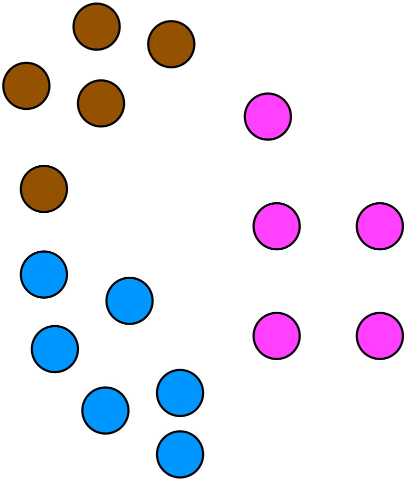
Algorithm 1: K-Nearest Neighbours

- Non-probabilistic.
- Can do binary or multi-class.
- No ‘training’ phase.

Algorithm 1: K-Nearest Neighbours

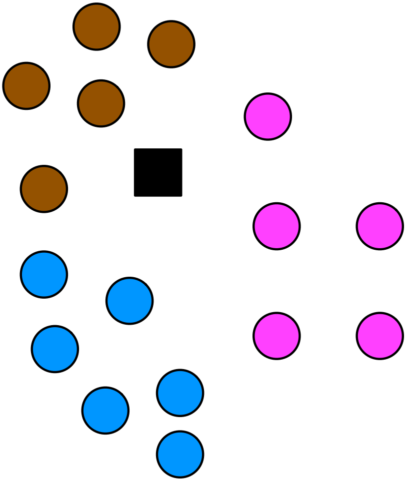
- Non-probabilistic.
- Can do binary or multi-class.
- No ‘training’ phase.
- How it works:
  - Choose K
  - For a test object $\mathbf{x}_{\text{new}}$:
  - Find the K closest points from the training set.
  - Find majority class of these K neighbours.
  - (Assign randomly in case of a tie)

KNN



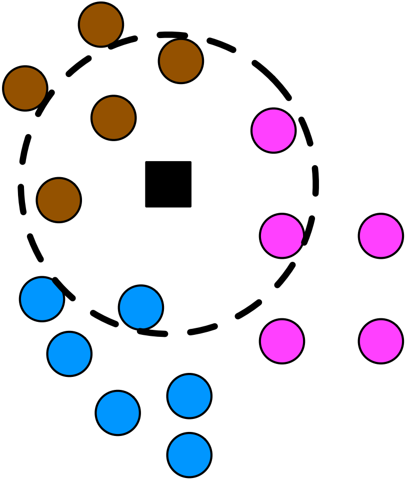
Training data from 3 classes.

KNN



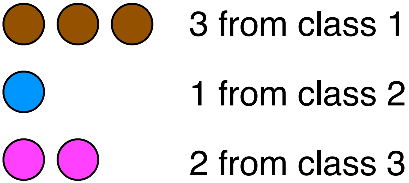
Test point.

KNN



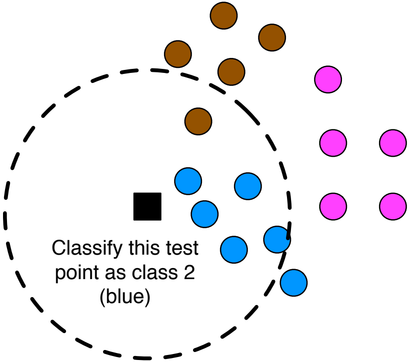
Find $K = 6$ nearest neighbours.

KNN



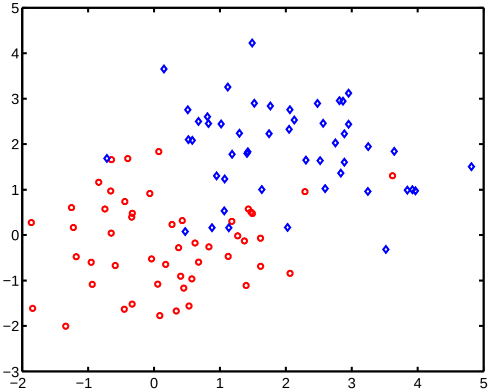
Class one has most votes – classify $\mathbf{x}_{\text{new}}$ as belonging to class 1.

KNN



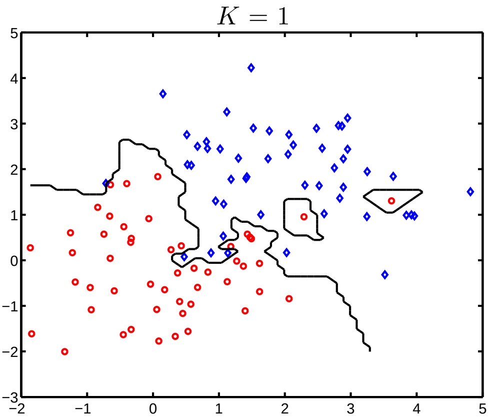
Second example – class 2 has most votes.

KNN – real example



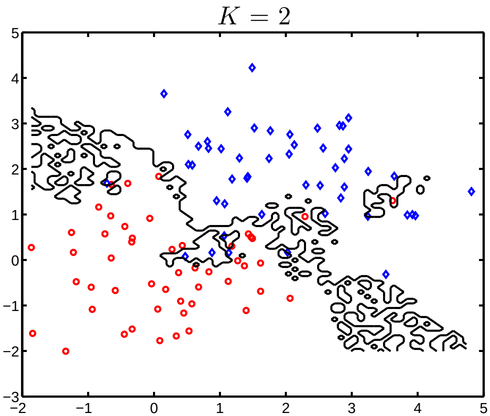
- Binary data.

KNN – real example



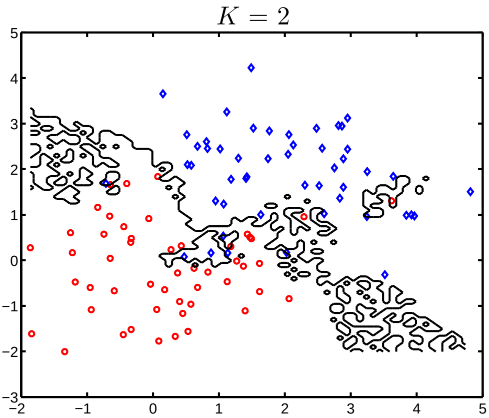
- 1-Nearest Neighbour.
- Line shows decision boundary.
- Too complex – should the islands exist?

KNN – real example



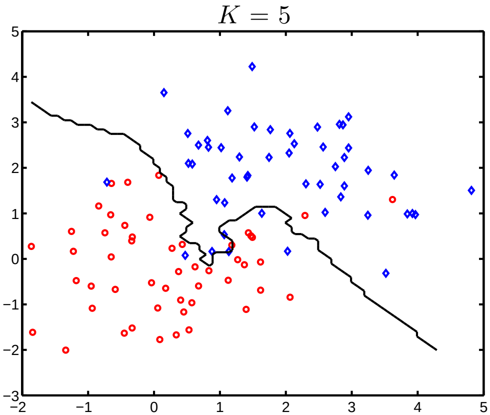
- 2-Nearest Neighbour.
- What's going on?

KNN – real example



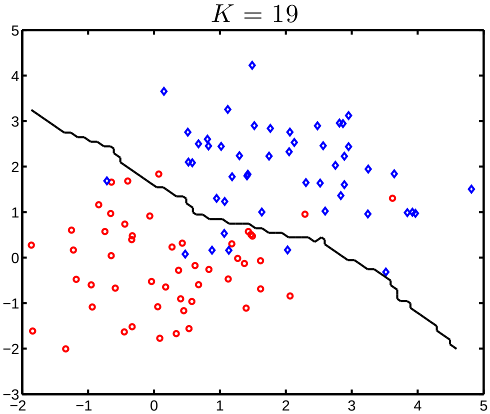
- 2-Nearest Neighbour.
- What's going on?
- Lots of ties – random guessing.

KNN – real example



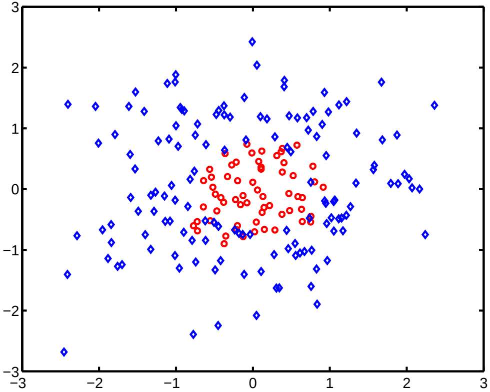
- 5-Nearest Neighbour.
- Much smoother.

KNN – real example



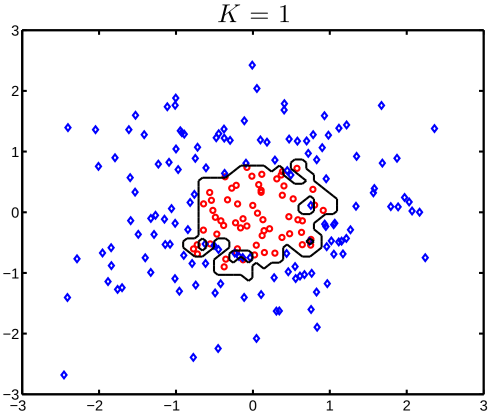
- 19-Nearest Neighbour.
- Very smooth.

KNN – real example 2



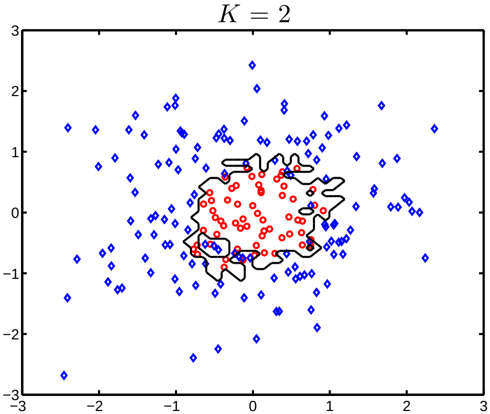
- Binary data.

KNN – real example 2



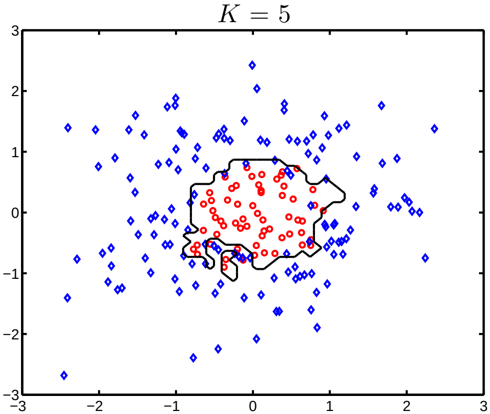
- Non-smooth – too complex again?

KNN – real example 2



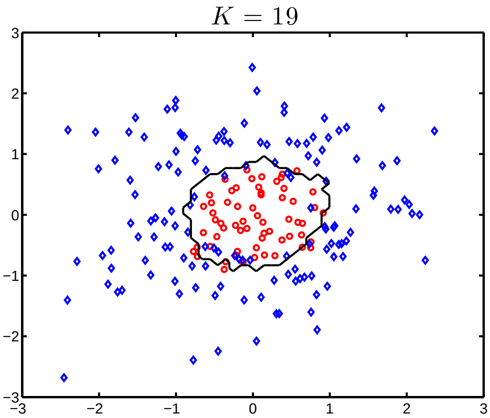
- Random effects again...

KNN – real example 2



- Getting smoother.

KNN – real example 2



- Smoother still.

Problems with KNN

- Class imbalance
  - As K increases, small classes will disappear!
  - Imagine we had only 5 training objects for class 1 and 100 for class 2.
  - For $K \geq 11$, class 2 will **always** win!

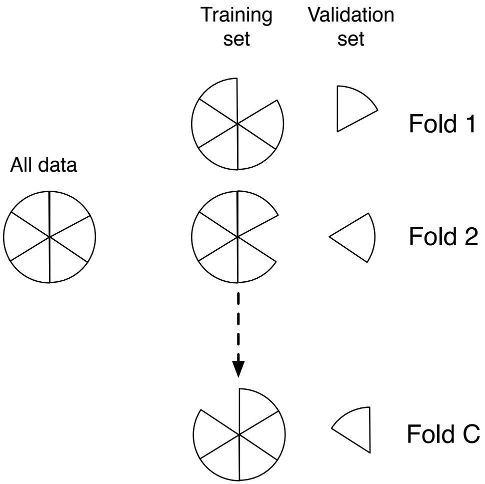
Problems with KNN

- Class imbalance
  - As K increases, small classes will disappear!
  - Imagine we had only 5 training objects for class 1 and 100 for class 2.
  - For $K \geq 11$, class 2 will **always** win!
- How do we choose K?
  - Right value of K will depend on data.
  - Cross-validation!

Cross-validation for classification

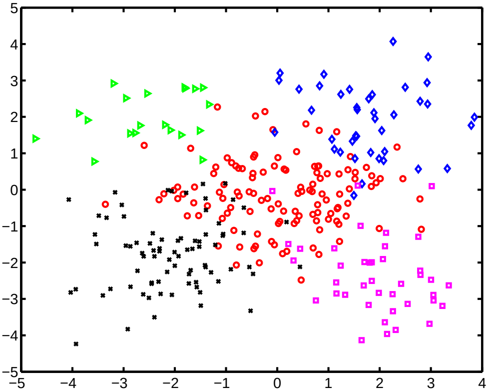
- E.g. to find K in KNN:
- Exactly the same as we have seen before.
- Split the data up – use some to train, some to validation.
- Need a measure of 'goodness'.
- Use number of mis-classifications.....
- ....and use K that minimises it!

Remember...



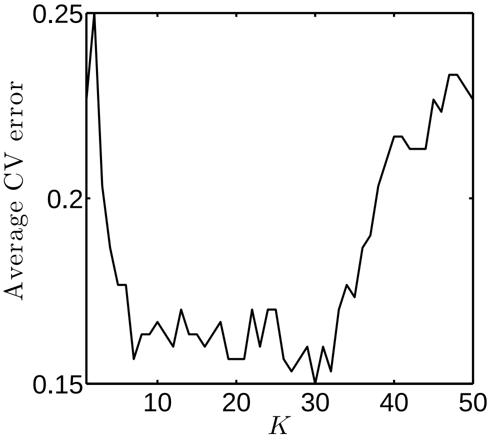
Average number of misclassifications over the C folds.

Example – 5 classes



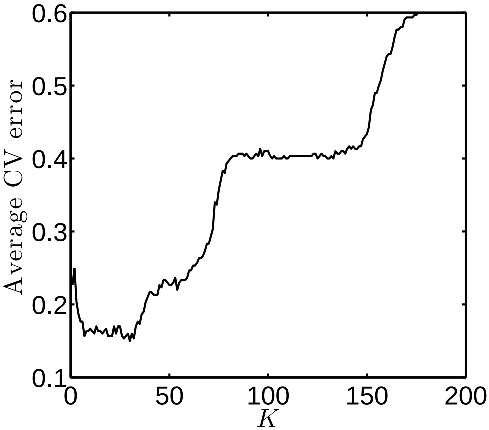
- 5 classes.
- Smallest has 20 instances, biggest 120.

Example – 5 classes



- Curve shows average misclassification error for 10-fold CV.
- Minimum at approximately $K = 30$.

Example – 5 classes



- As K increases, classes 'disappear'
- Causes the 'steps' in error.

KNN – summary

- Non-probabilistic.
- Fast.
- Only one parameter to tune (K).
- Important to tune it well....
- ...can use CV.

KNN – summary

- Non-probabilistic.
- Fast.
- Only one parameter to tune (K).
- Important to tune it well....
- ...can use CV.
- There is a probabilistic version.
  - Not covered in this course.