COM2004/3004

Data Driven Computing

Non-parametric classifiers

Dr. Po Yang

The University of Sheffield
po.yang@sheffield.ac.uk

Outline

Recap

Non-parametric Classifiers

Nearest Neighbour Classifier

k-Nearest Neighbour Classifier

Recap

Recap

- Minimising risk
- Linear Classifier
- Analysis of the Gaussian Bayes Classifier
- Parameterising a linear classifier

Non-parametric Classifiers

Non-parametric classifiers

In this lecture we will,

- Explain the terms parametric classifier and non-parametric classifier.

Non-parametric classifiers

In this lecture we will,

- Explain the terms parametric classifier and non-parametric classifier.
- Compare the strengths and weaknesses of these two approaches.

Parametric Classifiers

- The classifiers we have seen so far have been parametric classifiers.

Overview

Parametric Classifiers

- The classifiers we have seen so far have been parametric classifiers.
- This means that they are governed by a fixed number of learnable parameters, e.g.

Parametric Classifiers

- The classifiers we have seen so far have been parametric classifiers.
- This means that they are governed by a fixed number of learnable parameters, e.g.
    - The mean and variance of a normal distribution

Parametric Classifiers

- The classifiers we have seen so far have been parametric classifiers.
- This means that they are governed by a fixed number of learnable parameters, e.g.
  - The mean and variance of a normal distribution
  - The weights of a linear classifier

Parametric Classifiers

- The classifiers we have seen so far have been parametric classifiers.
- This means that they are governed by a fixed number of learnable parameters, e.g.
  - The mean and variance of a normal distribution
  - The weights of a linear classifier
- So what do we mean by a non-parametric classifier?

Non-Parametric Classifiers

- Non-parametric classifiers use the data directly at classification time

Non-Parametric Classifiers

- Non-parametric classifiers use the data directly at classification time
- No explicit model of the data

Non-Parametric Classifiers

- Non-parametric classifiers use the data directly at classification time
- No explicit model of the data
- i.e., so they're not governed by parameters

Overview

Non-Parametric Classifiers

- Non-parametric classifiers use the data directly at classification time
- No explicit model of the data
- i.e., so they're not governed by parameters
- No explicit learning stage

Parametric versus non-parametric classifiers

- parametric classifiers

Parametric versus non-parametric classifiers

- parametric classifiers
    - Everything we've seen so far

Parametric versus non-parametric classifiers

- parametric classifiers
    - Everything we've seen so far
    - Simple underlying model of the classes is assumed (e.g Gaussian)

Parametric versus non-parametric classifiers

- parametric classifiers
  - Everything we've seen so far
  - Simple underlying model of the classes is assumed (e.g Gaussian)
  - Parameters of the model learnt from the training data

Parametric versus non-parametric classifiers

- parametric classifiers
    - Everything we've seen so far
    - Simple underlying model of the classes is assumed (e.g Gaussian)
    - Parameters of the model learnt from the training data
    - Number of parameters fixed and typically $<<$ amount of training data

Parametric versus non-parametric classifiers

- parametric classifiers
  - Everything we've seen so far
  - Simple underlying model of the classes is assumed (e.g Gaussian)
  - Parameters of the model learnt from the training data
  - Number of parameters fixed and typically $<<$ amount of training data
  - Once parameters are known the training data can be discarded.

Parametric versus non-parametric classifiers

- parametric classifiers
  - Everything we've seen so far
  - Simple underlying model of the classes is assumed (e.g Gaussian)
  - Parameters of the model learnt from the training data
  - Number of parameters fixed and typically $<<$ amount of training data
  - Once parameters are known the training data can be discarded.
- non-parametric classifiers

Parametric versus non-parametric classifiers

- parametric classifiers
  - Everything we've seen so far
  - Simple underlying model of the classes is assumed (e.g Gaussian)
  - Parameters of the model learnt from the training data
  - Number of parameters fixed and typically $<<$ amount of training data
  - Once parameters are known the training data can be discarded.
- non-parametric classifiers
  - No real model of the classes

Parametric versus non-parametric classifiers

- parametric classifiers
    - Everything we've seen so far
    - Simple underlying model of the classes is assumed (e.g Gaussian)
    - Parameters of the model learnt from the training data
    - Number of parameters fixed and typically $<<$ amount of training data
    - Once parameters are known the training data can be discarded.
- non-parametric classifiers
    - No real model of the classes
    - Data not assumed to belong to any particular distribution

Parametric versus non-parametric classifiers

- parametric classifiers
  - Everything we've seen so far
  - Simple underlying model of the classes is assumed (e.g Gaussian)
  - Parameters of the model learnt from the training data
  - Number of parameters fixed and typically $<<$ amount of training data
  - Once parameters are known the training data can be discarded.
- non-parametric classifiers
  - No real model of the classes
  - Data not assumed to belong to any particular distribution
  - More flexible, but often expensive and often requiring a lot of data to learn things that were assumed by parametric approaches

Parametric versus non-parametric classifiers

- parametric classifiers
  - Everything we've seen so far
  - Simple underlying model of the classes is assumed (e.g Gaussian)
  - Parameters of the model learnt from the training data
  - Number of parameters fixed and typically $<<$ amount of training data
  - Once parameters are known the training data can be discarded.
- non-parametric classifiers
  - No real model of the classes
  - Data not assumed to belong to any particular distribution
  - More flexible, but often expensive and often requiring a lot of data to learn things that were assumed by parametric approaches
  - Poor choice when data is known to come from simple distributions!

Examples of non-parametric approaches include,

- Nearest Neighbour
- Decision Trees
    - Classification and Regression Tree (CART) model
- Support Vector Machines

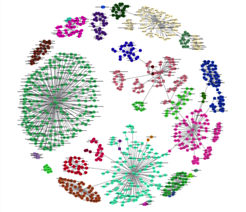

Figure : k-nearest neighbour

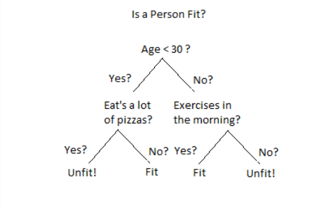

Figure : Decision tree

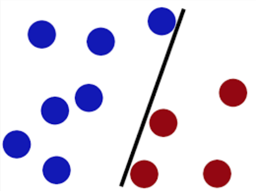

Figure : SV machine

Overview ii

We will be focusing on,

- Nearest Neighbour
- k-Nearest Neighbour (k-NN)

These approaches will operate by comparing a sample against previously seen examples.

Nearest Neighbour Classifier

Nearest neighbour classification

We will be covering,

- What is a nearest neighbour classifier

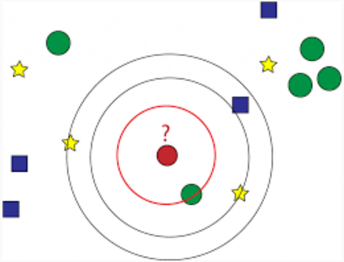

Figure : Nearest neighbour classifier

Nearest neighbour classification

We will be covering,

- What is a nearest neighbour classifier
- Some common applications

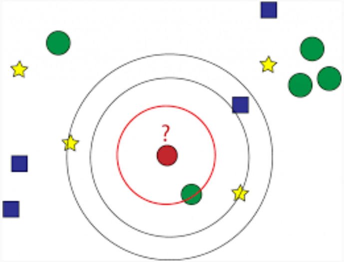

Figure : Nearest neighbour classifier

Nearest neighbour classification

We will be covering,

- What is a nearest neighbour classifier
- Some common applications
- Analysis of the decision boundary

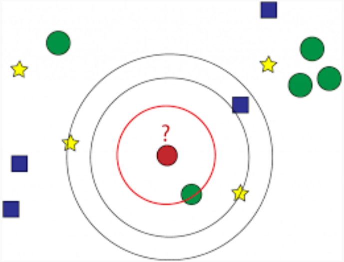

Figure : Nearest neighbour classifier

Nearest Neighbour Classification

Nearest neighbour rule

- label new sample by copying the class of the nearest sample in a labeled database (i.e., training data set)

Nearest Neighbour Classification

Nearest neighbour rule

- label new sample by copying the class of the nearest sample in a labeled database (i.e., training data set)
    - no prior assumptions about the distributions of instances in the database

Nearest Neighbour Classification

Nearest neighbour rule

- label new sample by copying the class of the nearest sample in a labeled database (i.e., training data set)
    - no prior assumptions about the distributions of instances in the database
- advantage: simple and high performance

Nearest Neighbour Classification

Nearest neighbour rule

- label new sample by copying the class of the nearest sample in a labeled database (i.e., training data set)
  - no prior assumptions about the distributions of instances in the database
- advantage: simple and high performance
  - often exceeds accuracy of more complicated classification methods

Nearest Neighbour Classification

Nearest neighbour rule

- label new sample by copying the class of the nearest sample in a labeled database (i.e., training data set)
  - no prior assumptions about the distributions of instances in the database
- advantage: simple and high performance
  - often exceeds accuracy of more complicated classification methods
- problem: computationally intensive

Nearest Neighbour Classification

Nearest neighbour rule

- label new sample by copying the class of the nearest sample in a labeled database (i.e., training data set)
  - no prior assumptions about the distributions of instances in the database
- advantage: simple and high performance
  - often exceeds accuracy of more complicated classification methods
- problem: computationally intensive
  - efficient nearest neighbour classification is a non-trivial problem when the database is very large

Nearest Neighbour Classification

Input

$\mathbf{y}$ – sampled to classify,

$\mathbf{x}_1, ..., \mathbf{x}_N$ – training data,

$\omega_1, ..., \omega_N$ – labels

$dist()$ - a distance function

Algorithm

set minimum distance, d_{min}, to infinity

for $i = 1$ to N

compute $d = dist(\mathbf{y}, \mathbf{x}_i)$
if $d < d_{min}$

$output\_label = \omega_i$
$d_{min} = d$

return $output\_label$

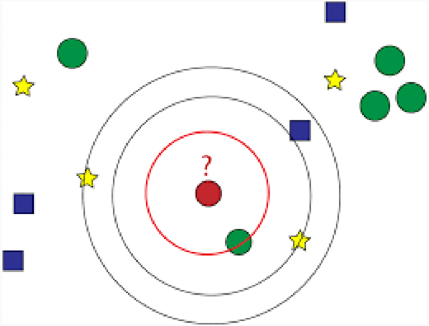

Figure : Nearest neighbour classifier

Nearest Neighbour Classification

(from 'http://cs-people.bu.edu/athitsos/nearest-neighbors/')

Nearest Neighbour Classification

Applications

- recognition problems
  - face recognition, fingerprints verification, speaker identification, optical characters recognition
- data mining
  - plagiarism detection, synonym detection
- recommendation systems
  - music, film, shopping recommendations
- information retrieval
  - spelling correction, concept matching, search for DNA sequences, related webpage search

Nearest Neighbour Classification

Issues

- definition of similarity

Nearest Neighbour Classification

Issues

- definition of similarity
  - similarity between two faces?

Issues

- definition of similarity
  - similarity between two faces?
  - distance between multiple webpages?

Nearest Neighbour Classification

Issues

- definition of similarity
    - similarity between two faces?
    - distance between multiple webpages?
- choice of similarity measure

Nearest Neighbour Classification

Issues

- definition of similarity
  - similarity between two faces?
  - distance between multiple webpages?

- choice of similarity measure
  - Euclidean distance?

Nearest Neighbour Classification

Issues

- definition of similarity
  - similarity between two faces?
  - distance between multiple webpages?

- choice of similarity measure
  - Euclidean distance?
  - Manhattan distance?

Nearest Neighbour Classification

Issues

- definition of similarity
    - similarity between two faces?
    - distance between multiple webpages?

- choice of similarity measure
    - Euclidean distance?
    - Manhattan distance?
    - Cosine distance?

Nearest Neighbour Classification

Issues

- definition of similarity
  - similarity between two faces?
  - distance between multiple webpages?

- choice of similarity measure
  - Euclidean distance?
  - Manhattan distance?
  - Cosine distance?

- efficient classification

Nearest Neighbour Classification

Issues

- definition of similarity
  - similarity between two faces?
  - distance between multiple webpages?
- choice of similarity measure
  - Euclidean distance?
  - Manhattan distance?
  - Cosine distance?
- efficient classification
  - very large database

Nearest Neighbour Classification

Issues

- definition of similarity
  - similarity between two faces?
  - distance between multiple webpages?
- choice of similarity measure
  - Euclidean distance?
  - Manhattan distance?
  - Cosine distance?
- efficient classification
  - very large database
  - similarity calculation may not be simple

Nearest Neighbour Classifier

Decision Boundary

Decision Boundary

What does the decision boundary of a nearest neighbour classifier look like?

Is it linear?

Is it smooth?

Voronoi tessellation and decision boundary

Voronoi tessellation

- conditions:
  - $\mathbf{x}_1, \ldots, \mathbf{x}_N$ are L-dimensional feature vectors
  - nearer neighbour rule is used
  - distance measure $d(\mathbf{x}_i, \mathbf{x}_j)$
- the feature vectors define a partition of the L-dimensional space into N regions R_i :

$$R_i = \{\mathbf{x} : \ d(\mathbf{x}, \mathbf{x}_i) < d(\mathbf{x}, \mathbf{x}_j), \ i \neq j\}$$

  - R_i contains all points in space that are closer to $\mathbf{x}_i$ than any other points of the feature set

Decision Boundary of Voronoi tessellation

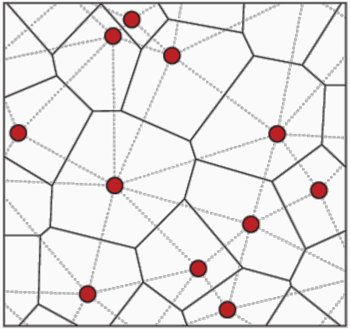

Figure : An example in a 2-D feature space.

Examples

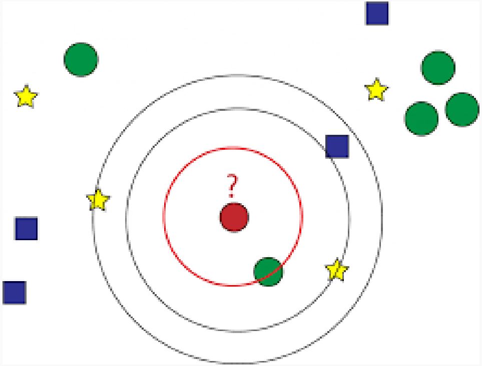

Figure : Nearest neighbour classifier

More complex visualisation: http://vision.stanford.edu/teaching/cs231n-demos/knn/

Nearest Neighbour Classifier

Summary

Nearest Neighbour Classifer

- Nearest neighbour classification is a non-parametric technique

Nearest Neighbour Classifer

- Nearest neighbour classification is a non-parametric technique
- We classify a sample by outputting the label of the closest labeled sample

Nearest Neighbour Classifer

- Nearest neighbour classification is a non-parametric technique
- We classify a sample by outputting the label of the closest labeled sample
- Need to define a distance measure (typically Euclidean distance)

Nearest Neighbour Classifer

- Nearest neighbour classification is a non-parametric technique
- We classify a sample by outputting the label of the closest labeled sample
- Need to define a distance measure (typically Euclidean distance)
- A basic implementation can be slow because need to compute distance to all samples in training set

Nearest Neighbour Classifer

- Nearest neighbour classification is a non-parametric technique
- We classify a sample by outputting the label of the closest labeled sample
- Need to define a distance measure (typically Euclidean distance)
- A basic implementation can be slow because need to compute distance to all samples in training set
- The decision boundary is piece-wise linear (i.e., made up of segments of a line, plane, hyperplane)

k-Nearest Neighbour Classifier

k-Nearest Neighbour

In this lecture we will be

- Presenting the algorithm.

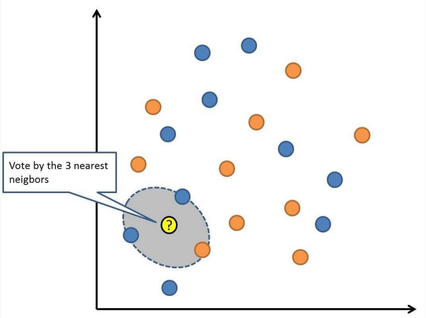

Figure : 3-nearest neighbour classifier

k-Nearest Neighbour

In this lecture we will be

- Presenting the algorithm.
- Asking how does the value of k affect the classifier's behaviour?

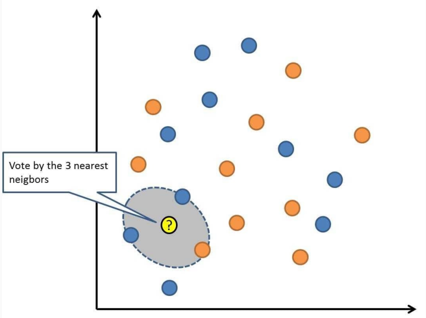

Figure : 3-nearest neighbour classifier

k-Nearest Neighbour

In this lecture we will be

- Presenting the algorithm.
- Asking how does the value of k affect the classifier's behaviour?
- Discussing the computational cost.

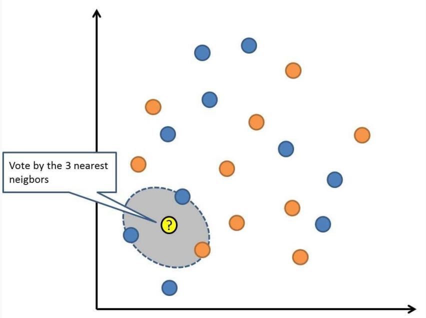

Figure : 3-nearest neighbour classifier

k-Nearest Neighbour

A generalisation of the nearest neighbour classifier that we discussed in the previous segment.

- find the k nearest neighbours
- assign a new sample to the class most common amongst its k nearest neighbours

Note, when $k = 1$ it is equivalent to the standard nearest neighbour classifier.

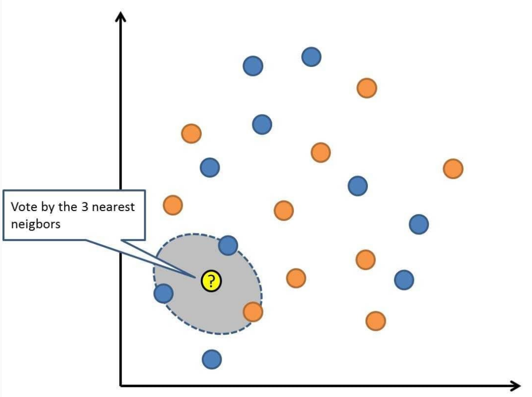

Figure : 3-nearest neighbour classifier

Algorithm of k-Nearest Neighbour

Algorithm

1. Load the training data
2. Select a distance function
3. Choose the value of k
4. Find the distance of test point to all training data points
   - keep track of the k closest points
5. Assign a class to the test point based on the majority of classes present in the chosen points

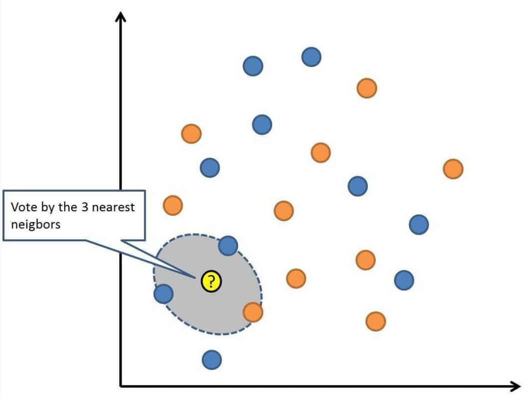

Figure : 3-nearest neighbour classifier

Youtube: https://www.youtube.com/watch?v=Mhv-HxGSgHU

Attention

Notes,

- no explicit training step, but the labeled data is sometimes referred to as the training set

Attention

Notes,

- no explicit training step, but the labeled data is sometimes referred to as the training set
- the algorithm is sensitive to the local structure, e.g., sensitive to outliers or poorly labeled data.

Visualise: `http://vision.stanford.edu/teaching/cs231n-demos/knn/`

Attention

Notes,

- no explicit training step, but the labeled data is sometimes referred to as the training set
- the algorithm is sensitive to the local structure, e.g., sensitive to outliers or poorly labeled data.

Visualise: http://vision.stanford.edu/teaching/cs231n-demos/knn/

- the contributions can be weighted

Attention

Notes,

- no explicit training step, but the labeled data is sometimes referred to as the training set
- the algorithm is sensitive to the local structure, e.g., sensitive to outliers or poorly labeled data.

Visualise: http://vision.stanford.edu/teaching/cs231n-demos/knn/

- the contributions can be weighted
  - the nearer neighbours contribute more than the distant ones

Attention

Notes,

- no explicit training step, but the labeled data is sometimes referred to as the training set
- the algorithm is sensitive to the local structure, e.g., sensitive to outliers or poorly labeled data.

Visualise: http://vision.stanford.edu/teaching/cs231n-demos/knn/

- the contributions can be weighted
  - the nearer neighbours contribute more than the distant ones

    (*e.g.*) give each neighbour a weight of $\frac{1}{d}$, where d is the distance to the neighbour

k-Nearest Neighbour Classifier

Choosing the value of k

Effect of k

Choice of parameter k

- increasing k
  - reduces the effect of noise
  - makes class boundaries smoother
  - too large and can over-smooth the boundaries (under-fitting)
- optimum value is data dependent
  - can be selected by some heuristics (*e.g.*) cross validation

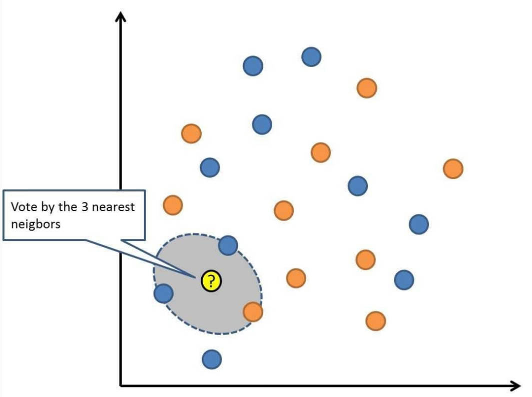

Figure : 3-nearest neighbour classifier

Computational Complexity

Consider classifying a single test sample given N training samples, each with F features.

- N distances will need to be computed regardless of the value of k
- Finding shortest k distances does not require sorting the complete list
- The overhead for tracking k best distances will be small compared to the cost of computing the distances.
- So the cost scales linearly with number of training samples, N.
- The cost of the distance measurement will typically scale with the number of features F
- So overall computational cost should scale as $\mathcal{O}(\mathcal{N} \times \mathcal{F})$

k-Nearest Neighbour Classifier

Summary

Summary

- k-Nearest Neighbour is a generalisation of the nearest neighbour algorithm.
- Labels according to a majority vote of the closest k training samples.
- Larger k will lead to a smoother decision boundary, reduced influence of outliers.
- If too large then can over-smooth and performance will decrease.
- k is often tuned using validation data.
- For the basic algorithmn, computational cost proportional to size of training data set.