Case-Based Reasoning

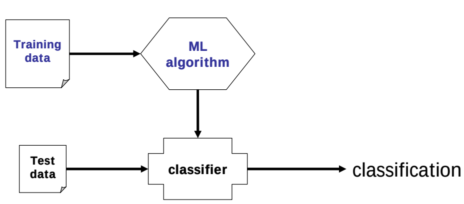
Lazy Learning

Prof. Ian Watson

Eager Learning

- ML algorithms like ID3, C4.5 or Neural Networks are *eager* learners
  - Use a training data set to
  - Generalize rules, induce a tree or a function that can be applied to categorize future inputs
  - Processing time is done up-front before query time
  - After querying they discard any inputs

Eager Learning



Eager learning methodology

- Obtain data set
- Identify target (output) attribute (this is what we want to predict)
- Analyse input features
  - Estimate which are predictive of target
  - Are combinations of input features required (eg a simple ratio of two inputs)
- Analyse data set and remove noisy items
- Divide data set in training and test sets

Eager learning methodology

- Identify possible ML algorithms based on:
  - Data types (discrete, continuous)
  - Classification or regression task
  - Type of output required
    - Function
    - Decision tree
    - Neural network

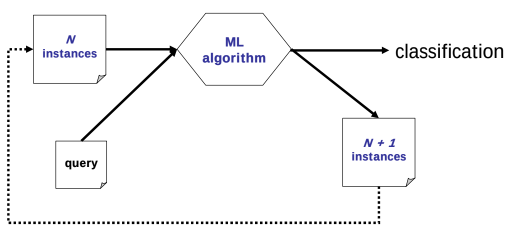
Eager learning methodology

- Run algorithm(s) on training data
- Validate on test data
- Better still do 10 fold cross validation
- Tweak parameters of algorithm
- Repeat validation
- Consider using an ensemble of algorithms

Lazy learners

- Lazy learners have three characteristics:
  - They defer all (most) processing until query/run-time
  - They discard any generated functions/answers
  - They retain the query with the stored data

Lazy Learning



Lazy vs. Eager

- Lazy learners have low computational costs at training (~0)
- But may have high storage costs
- High computational costs at query
- Lazy learners can respond well to dynamic data where it would be necessary to constantly re-train an eager learner

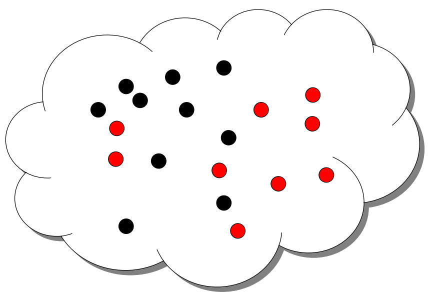
The University of Auckland

Instance-based learners

- store *all* the training data
- when a new query instance is encountered, a set of related instances are retrieved from memory and used to classify the instance
- can construct a different approximation function of the target function for each distinct query instance

The University of Auckland

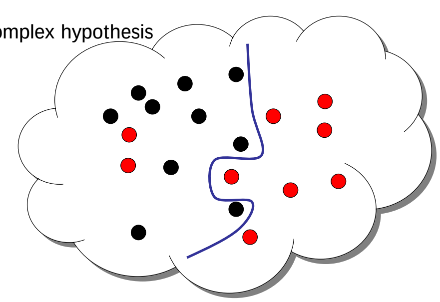
Eager learners



The University of Auckland

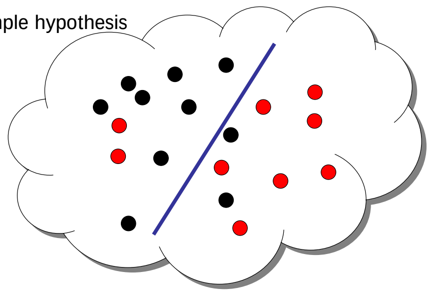
Eager learners

A complex hypothesis



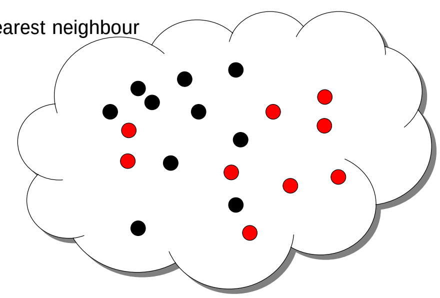
Eager learners

A simple hypothesis



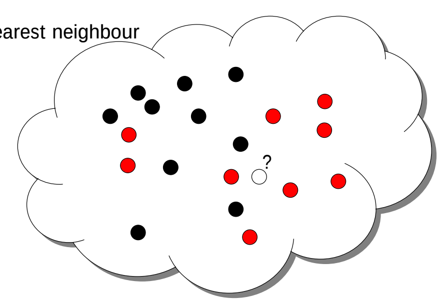
Lazy learners

k-nearest neighbour



Lazy learners

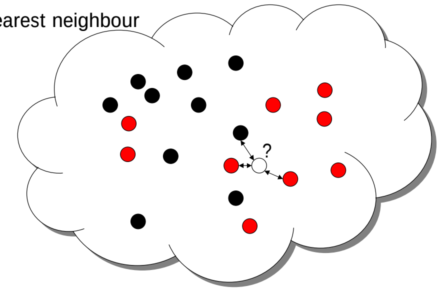
k-nearest neighbour



The University of Auckland

Lazy learners

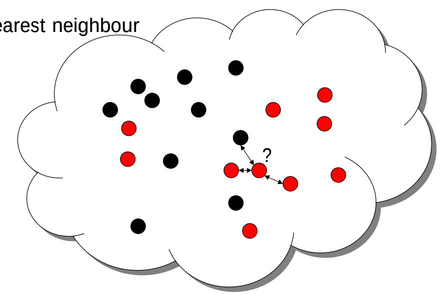
k-nearest neighbour



The University of Auckland

Lazy learners

k-nearest neighbour



The University of Auckland

Instance-based learners

- significant advantage
- when the target function is potentially very complex
- but can be described by a collection of simple local approximations

Instance-based learners

- Disadvantages
  - cost of classifying new instances can be high, so efficiently indexing training instances very important
  - Similarity has to be determined for each attribute or feature

Instance-based learners

- Nearest neighbour (k-NN)
  - most basic method - all instances are points in an *n*-dimensional space
  - distance is defined as standard Euclidean distance
  - K-NN finds the *nearest* neighbours to a query in the *n*-dimensional space
  - values may be discrete or real

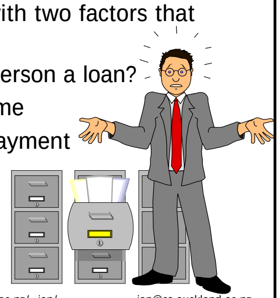
Nearest Neighbour

$$Similarity(T,S) = \sum_{i=1}^{n} f(T_i, S_i) \times w_i$$

where:
T is the target case
S is the source case
n is the number of attributes in each case
i is an individual attribute from 1 to *n*
f is a similarity function for attribute *i* in cases *T* and *S* and
w is the importance weighting of attribute *i*

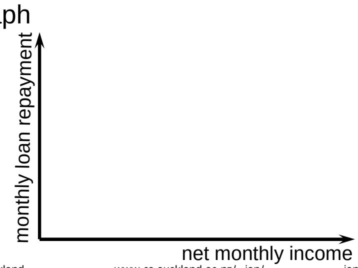
Nearest Neighbour

- imagine a decision with two factors that influence it
- should you grant a person a loan?
  - net monthly income
  - monthly loan repayment



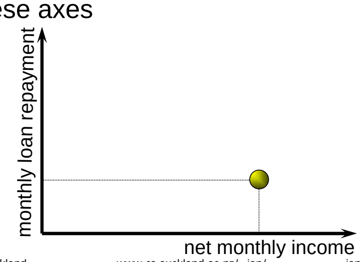
Nearest Neighbour

- these factors can be used as axes for a graph



Nearest Neighbour

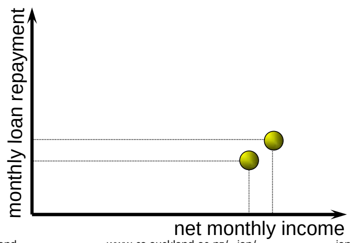
- a previous loan can be plotted against these axes



25

Nearest Neighbour

- and a second loan

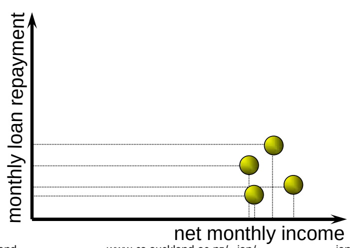


 www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

26

Nearest Neighbour

- and more loans

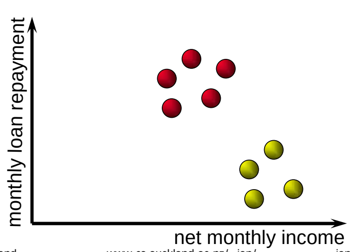


 www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

27

Nearest Neighbour

- and even more loans

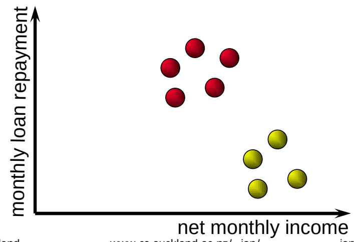


 www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

28

Nearest Neighbour

- past cases (loans) may form clusters

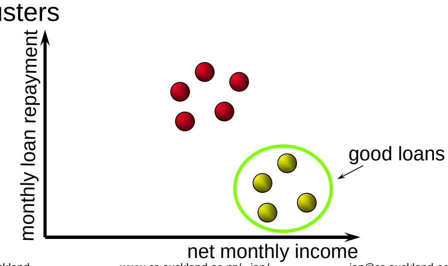


 www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

29

Nearest Neighbour

- past cases (loans) may tend to form clusters

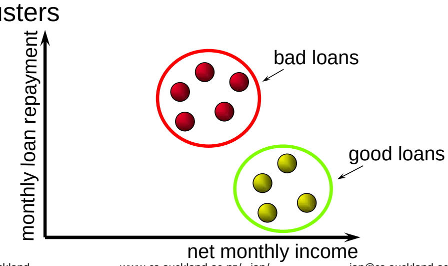


 www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

30

Nearest Neighbour

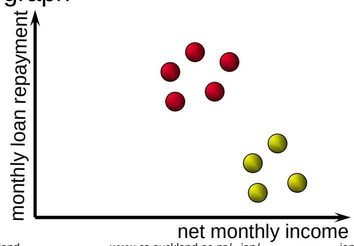
- past cases (loans) may tend to form clusters



 www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

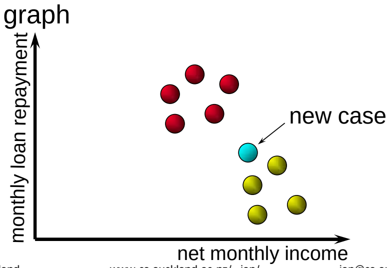
Nearest Neighbour

- a new loan prospect can be plotted on the graph



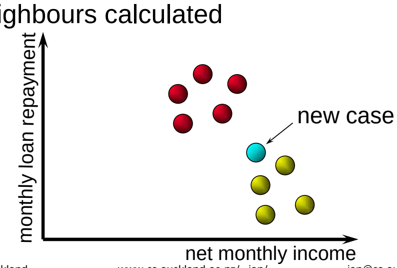
Nearest Neighbour

- a new loan prospect can be plotted on the graph



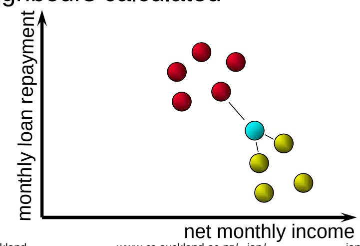
Nearest Neighbour

- and the distance to its nearest neighbours calculated



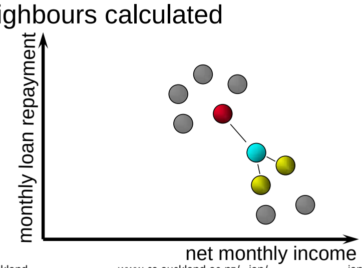
Nearest Neighbour

- and the distance to its nearest neighbours calculated



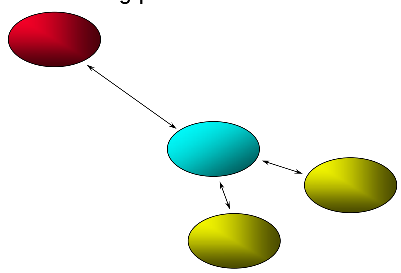
Nearest Neighbour

- and the distance to its nearest neighbours calculated



Nearest Neighbour

- the best matching past case is the closest

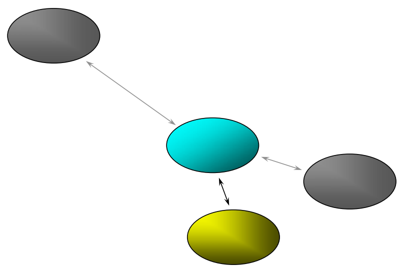


37

The University of Auckland

Nearest Neighbour

- the best matching past case is the closest



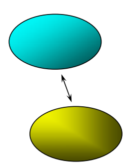
 www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

38

The University of Auckland

Nearest Neighbour

- this suggests a precedent



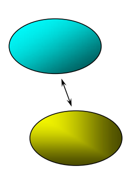
 www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

39

The University of Auckland

Nearest Neighbour

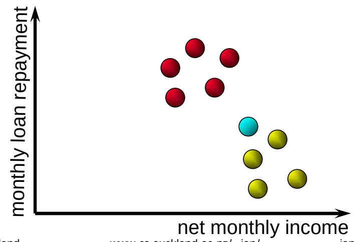
- this suggests a precedent
- the loan will be successful



 www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

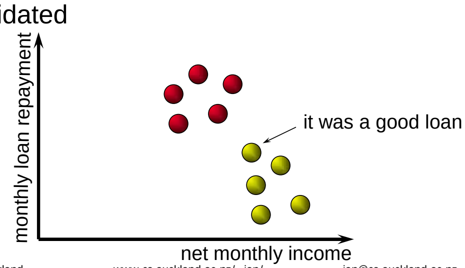
Nearest Neighbour

- over time the prediction can be validated



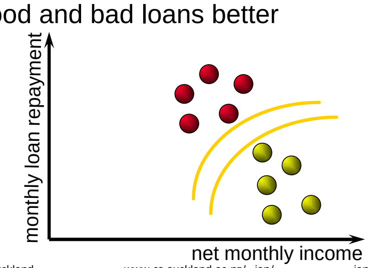
Nearest Neighbour

- over time the prediction can be validated



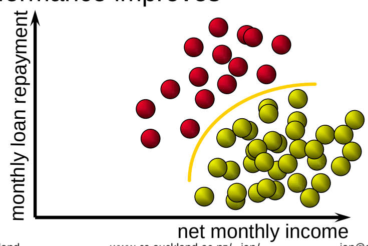
Nearest Neighbour

- the system is learning to differentiate good and bad loans better

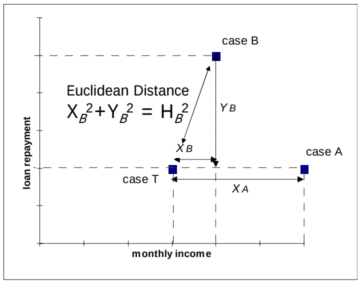


Nearest Neighbour

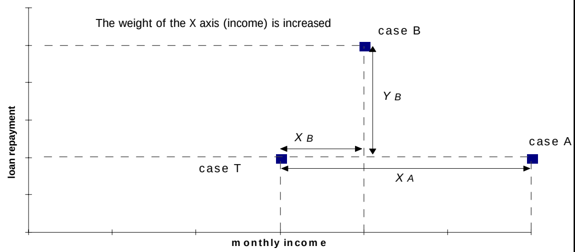
- as more cases are acquired its performance improves



Nearest Neighbour



Nearest Neighbour



46

Nearest Neighbour

- Require a unique similarity function for each attribute or feature (not always a trivial problem) – *local similarity* $f(T_i,S_i)$
- Local similarities are combined to give a *global similarity* – sim(T,S)
- k-NN Requires every feature of the query to be compared to every feature of every instance/case at run-time
- Not very efficient ☹

 www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

47

Nearest Neighbour

- distance weighted k-Nearest neighbour is a highly effective algorithm for many practical problems robust to noisy data if the training set is large enough
- bias is that the classification of an instance is most similar to other instances that are nearby in Euclidean distance
- But then again that's the point

 www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

48

Nearest Neighbour

- because distance is calculated on all attributes - irrelevant attributes are a problem - curse of dimensionality
- some approaches weight attributes to overcome this - stretching the Euclidean space – determined automatically using cross-validation
- alternatively eliminate the least relevant attributes - they used leave-one out cross-validation – ideal for IBL

 www.cs.auckland.ac.nz/~ian/ ian@cs.auckland.ac.nz

Nearest Neighbour

- could locally stretch an axis...but more degrees of freedom...so more chance of overfitting...useful if problem space is not uniform...problem of over fitting
- much less common, but it is used in CBR
- efficient indexing of instances can be done with kd-trees (we'll discuss later)
- possible to pre-compute a position of each instance in the Euclidean space then simply position query in the space

Summary

- IBLs (k-NN is an IBL) delay processing until prediction time they form a different local approximation for each query instance
- can model complex functions by a combination of less-complex local approximations
- information present in the training data is never lost
- can be computationally expense to label new instances
- finding appropriate distance metric can be difficult
- irrelevant attributes can have a negative impact