COMP219: Artificial Intelligence

Lecture 26: Linear Models and Non-parametirc Models

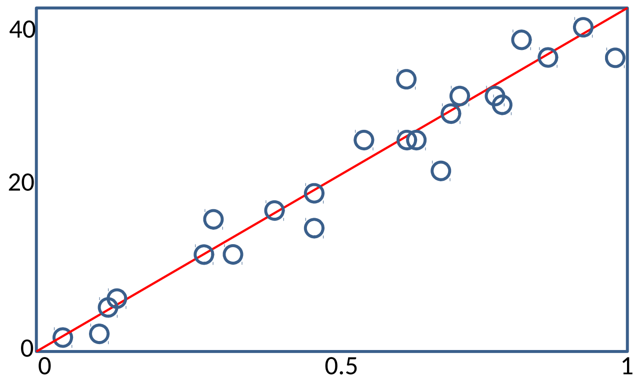
Overview

- Last time
  - Types of learning; supervised learning; decision trees
- Today
  - More supervised learning methods
    - Regression and classification with linear models
    - Non-parametric models
      - *K*-nearest neighbours
  - A brief look at unsupervised learning
- Learning outcomes covered today:

  Identify or describe the major approaches to learning in AI and apply these to simple examples.

Linear Models

- Linear functions of continuous valued inputs have been used for hundreds of years
- Fitting a line to a function



Univariate Linear Regression

- The task of "fitting a straight line"
- A univariate linear function with input x and output y has the form

  $y = w_1 x + w_0$

  where w_0 and w_1 are real valued coefficients (weights) to be learned

Linear Regression

- We define **w** to be the vector $[w_0,w_1]$ and define
  $h_{\mathbf{w}}(x)=w_1x+w_0$
- The task of finding the $h_{\mathbf{w}}$ that best fits the data is called linear regression

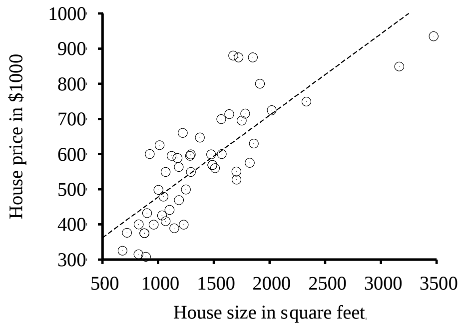
Fitting a Straight Line

- To fit a line to the data we have to find the values of the weights $[w_0, w_1]$ that *minimise* the empirical loss
- Traditionally we calculate loss using the squared loss function summed over all the training examples

$$Loss(h_w) = \sum_{j=1}^{N} (y_j - (w_1x_j + w_0))^2$$

which finds a value for the distance of each training example from the line drawn using $[w_0, w_1]$

Example: Houses for Sale



- Data points plot price vs floor space of houses for sale in Berkeley in July 2009
- Linear function hypothesis that minimises squared error loss:
  $y = 0.232x + 246$

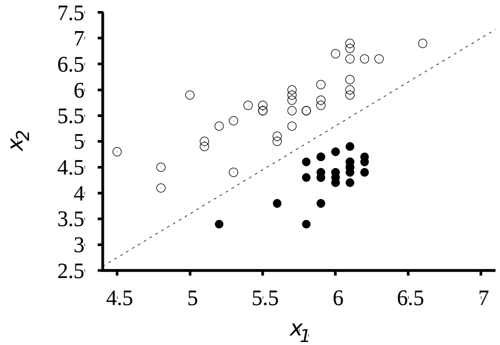
Linear Classification

- Linear functions can be used for classification as well as regression
- A decision boundary is a line that separates two classes
- A *linear* decision boundary is called a linear separator and data that admit such a separator are called linearly separable

Example: Earthquake or Explosion?

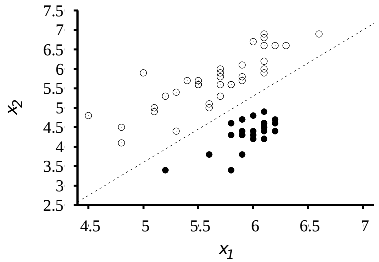
- We have a training data set with information about 2 classes:
  - earthquakes (interesting to seismologists)
  - underground explosions (interesting for arms control)
- Each data point has 2 inputs (x_1, x_2) which describe body (x_1) and surface (x_2) wave magnitudes computed from a seismic signal
- The task of classification is to learn a hypothesis h that will take new data points (x_1, x_2) and return either 0 for earthquakes or 1 for explosions

Seismic Example continued

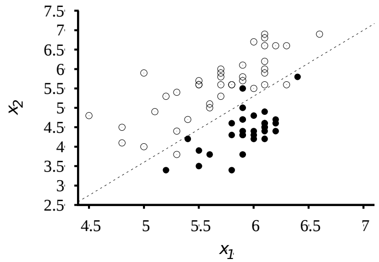


- Plot of two seismic data parameters for earthquakes (white circles) and explosions (black circles) and a decision boundary (linear separator)
- Explosions (class 1) are to the right of the line (higher x_1 and lower x_2) – the line can be thought of as a threshold function

Seismic Example continued



(a)



(b)

- Including more/different training data can affect the decision boundary
- In (b) above, including more data points into the same domain has meant that the earthquakes and explosions are no longer linearly separable

Parametric Models

- Linear regression uses the training data to estimate a fixed set of parameters $\mathbf{w}$ that defines our hypothesis $h_{\mathbf{w}}(x)$ and at that point we no longer need the training data

- A learning model that summarises data with a set of parameters of fixed size is called a **parametric model**

- No matter how much data you give a parametric model, it always needs the same number of parameters
- However, if there are a large number of examples available and the correct function is wiggly not linear, the model shouldn't be restricted to linear functions

Non-parametric Models

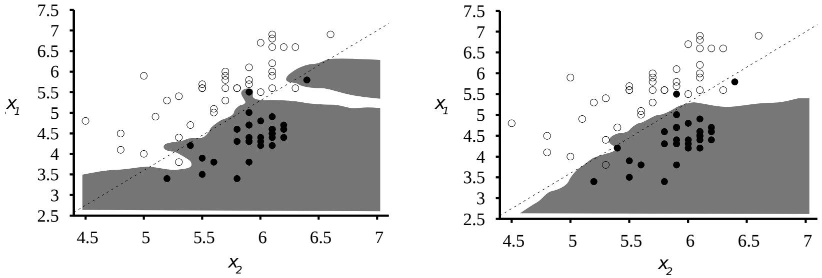
- **Non-parametric models** cannot be characterised by a bounded set of parameters
- e.g. Suppose our hypothesis retains all of the training examples and uses them to predict the next example; this is non-parametric as the number of parameters is unbounded
  - This approach is called instance-based learning
  - Simplest method is table lookup – but this method does not generalise well (if x not in table, return a default value)

Nearest Neighbour Models

Improve on table lookup with a small variation:

- k-nearest neighbours lookup: given x_q find the k examples that are nearest to x_q
- To classify, first find $\mathrm{NN}(k, x_q)$ then take the plurality vote of the neighbours
  - In binary classification, majority vote
  - To avoid ties, k is always odd
- For regression, take the mean or median of k neighbours

Seismic Example Revisited



(a) (b)

- (a) *k*-nearest-neighbour model showing the explosion class decision boundary with *k*=1 (note the overfitting, i.e. when a model describes noise instead of the underlying relationship)
- (b) with *k*=5 the overfitting problem is removed for this dataset

Exercise

Measuring Distance

- To find the nearest neighbours, we need to measure the distance between examples
- For Boolean attribute values, we measure the Hamming distance, i.e. number of attributes on which the two points differ
- NB: if we use the raw numbers for each attribute then the total distance is affected by a difference in scale in any dimension
  – e.g. if we change measurements from cm to miles in dimension i but keep all others the same, we will get different nearest neighbours
- Therefore we need to apply normalisation to the measurements in each dimension, i.e. rescale them
  – e.g. to numbers between 0 and 1

Example: Normalisation

- To normalise height data of a group one method is to rescale the data to values between 0 and 1 using

$$x' = \frac{x - min}{max - min}$$

where x is the original value and x' is the normalised value

- Example:
  [155, 158, 160, 162, 164, 166, 169, 171, 172, 175]

  min = 155; max=175 so $x' = \frac{x-155}{20}$

  Normalised data:
  [0, 0.15, 0.25, 0.35, 0.45, 0.55, 0.7, 0.8, 0.85, 1]

Exercise

Supervised Learning - Summary

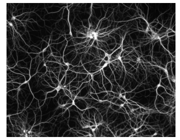
- We have looked at:
  – Decision trees
  – Linear regression
  – Linear classification
  – Non-parametric models
    - *k*-nearest neighbours

Unsupervised Learning

- It is not always possible to acquire example data to use to train a learning algorithm
- In unsupervised learning the agent learns patterns in the input even though no explicit feedback is supplied
- There are two complementary approaches:
  - *Self-organisation*, which tries to understand the principles of organisation of natural systems and use them to create efficient algorithms (e.g. Kohonen self-organising maps)
  - *Statistical approach*, which tries to extract the most relevant information from the distribution of unlabelled data (**clustering**)

Self-Organisation

- Self-organisation is observed in a wide range of natural processes
  - *Physics*: formation of crystals, star formation, chemical reactions,...
  - *Biology*: folding of proteins, social insects, flocking behaviour, brain functioning,...
  - *Social science*: critical mass, group thinking, herd behaviour,...
  - *Computer science*: cellular automata, multi-agent systems, random graphs,...

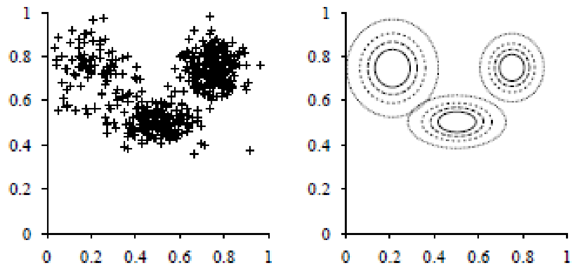


NASA

Clustering and its applications

- Clustering is the problem of detecting potentially useful and *distinct* clusters/categories in a collection of unlabelled objects
  - e.g. suppose we record the spectra of 100,000 stars. Astronomers have to perform unsupervised clustering to identify categories of stars (e.g. "red giant", "white dwarf")
- Example applications:
  - *Marketing*: group customers on properties and buying records
  - *Finance*: fraud detection
  - *Counter-terrorism*: identifying groups from Internet usage
  - *Insurance*: identifying high cost and fraudulent policy holders
  - *City planning*: group houses according to their type, value and location
  - *Earthquake studies:* identify dangerous zones
  - *WWW*: document classification, weblog data of access patterns

Clustering



- Shows 500 data points each with 2 continuous attributes (e.g. stars with spectral intensities at 2 different frequencies)
- Model shows 3 clusters

Summary

- More on supervised learning
  - Regression and classification with linear models
  - Non-parametric models
    - *k*-nearest neighbours
- Unsupervised learning
  - Clustering

- Next time
  - Reinforcement models