School of Electronic Engineering and Computer Science
Queen Mary University of London

CBU5201 Principles of Machine Learning
Supervised learning: Classification I

Dr Chao Liu

Credit to Dr Jesús Requena Carrión

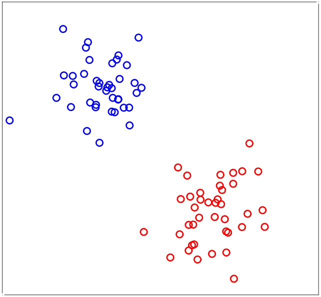
Oct 2023

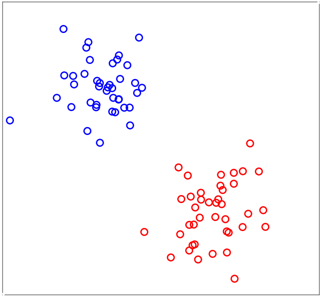
Queen Mary University of London

Agenda

Best, but risky, linear solutions

Draw two linear boundaries that achieve an accuracy $A = 1$. Which one would you choose? Why?

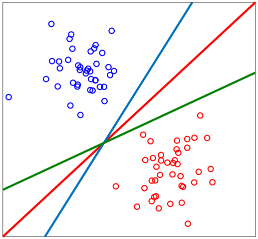




If you prefer one over the other, you might be inadvertently assessing their **generalisation ability** *and modelling the* **distribution of samples***. Your unconscious ML mind is working faster than your conscious mind!*

Keep that boundary away from me!

As we get closer to the decision boundary, life gets harder for a classifier: it is **noise territory** and we should beware of **jumpy samples**.

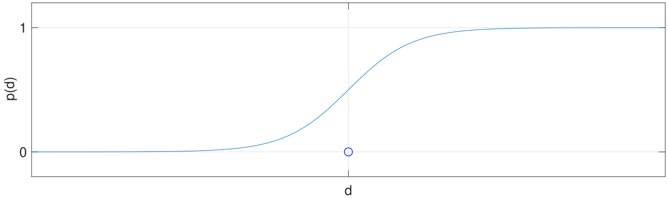


The **further** we are from the boundary, the **higher our certainty** that we are classifying samples correctly.

The logistic model

The logistic function $p(d)$ is defined as

$$p(d) = \frac{e^d}{1 + e^d} = \frac{1}{1 + e^{-d}}$$



Note that

- $p(0) = 0.5$.
- As $d \to \infty$, $p(d) \to 1$.
- As $d \to -\infty$, $p(d) \to 0$.

The logistic model

Given a linear boundary $\boldsymbol{w}$ and a predictor vector $\boldsymbol{x}_i$, the quantity $\boldsymbol{w}^T\boldsymbol{x}_i$ can be interpreted as the **distance** from the sample to the boundary.

If we set $d = \boldsymbol{w}^T\boldsymbol{x}_i$ in the logistic function, we get:

$$p(\boldsymbol{w}^T\boldsymbol{x}_i) = \frac{e^{\boldsymbol{w}^T\boldsymbol{x}_i}}{1 + e^{\boldsymbol{w}^T\boldsymbol{x}_i}}$$

For a fixed $\boldsymbol{w}$, we will simply denote it as $p(\boldsymbol{x}_i)$ to simplify the notation:

- *When* $\boldsymbol{w}^T\boldsymbol{x} \to \infty$, *the logistic function* $p(\boldsymbol{x}_i) \to 1$
- *When* $\boldsymbol{w}^T\boldsymbol{x} \to -\infty$, *the logistic function* $p(\boldsymbol{x}_i) \to 0$

We will use the logistic function to quantify the notion of **certainty** in classifiers. This certainty is a quantity between 0 and 1.

The logistic model

Consider a linear classifier $\boldsymbol{w}$ that labels samples such that $\boldsymbol{w}^T\boldsymbol{x}_i > 0$ as o and samples such that $\boldsymbol{w}^T\boldsymbol{x}_i < 0$ as o.

Notice that:

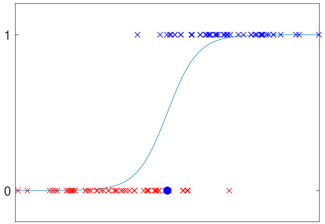
- If $\boldsymbol{w}^T\boldsymbol{x}_i = 0$ ($\boldsymbol{x}_i$ is on the boundary), $p(\boldsymbol{x}_i) = 0.5$.
- If $\boldsymbol{w}^T\boldsymbol{x}_i > 0$ ($\boldsymbol{x}_i$ is in the o region), $p(\boldsymbol{x}_i) \to 1$ as we move away from the boundary .
- If $\boldsymbol{w}^T\boldsymbol{x}_i < 0$ ($\boldsymbol{x}_i$ is in the o region), $p(\boldsymbol{x}_i) \to 0$ as we move away from the boundary.

Here is the crucial point, so use **all your neurons**:

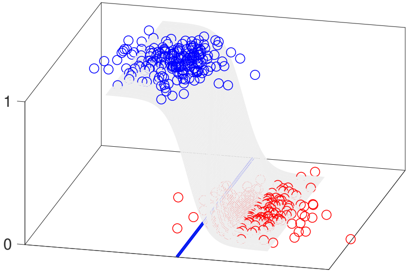
- $p(\boldsymbol{x}_i)$ is the **classifier's certainty** that $y_i =$ o true.
- $1 - p(\boldsymbol{x}_i)$ is the **classifier's certainty** that $y_i =$ o true.

Visualising logistic regression

1D predictor space



2D predictor space



The logistic classifier

We can obtain the classifier's certainty that $\boldsymbol{x}_i$ belongs to either o or o.
Can we calculate the certainty for a **labelled dataset** $\{(\boldsymbol{x}_i, y_i)\}$?

The answer is yes, by **multiplying** the individual certainties:

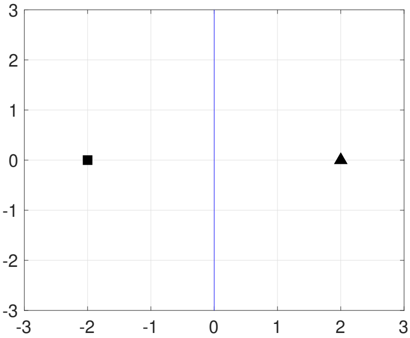
$$L = \prod_{y_i = \text{o}} (1 - p(\boldsymbol{x}_i)) \prod_{y_i = \text{o}} p(\boldsymbol{x}_i)$$

L is known as the **likelihood function** and defines a **quality metric**.
Taking logarithms, we obtain the **log-likelihood**:

$$l = \sum_{y_i = \text{o}} \log\left[1 - p(\boldsymbol{x}_i)\right] + \sum_{y_i = \text{o}} \log\left[p(\boldsymbol{x}_i)\right]$$

The linear classifier that maximises L or l is known as the **Logistic Regression** classifier. It can be found using **gradient descent**.

Example I



- Let's define $d_i = \boldsymbol{w}^T \boldsymbol{x}_i$
- We can rewrite the logistic function as

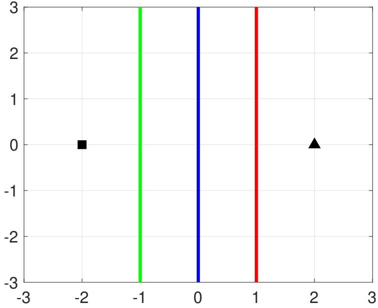
$$p(d_i) = \frac{e^{d_i}}{1 + e^{d_i}}$$

- For instance $p(0) = 0.5$, $p(1) \approx 0.73$, $p(2) \approx 0.88$, $p(-1) \approx 0.27$ and $p(-2) \approx 0.12$

Assume this linear classifier labels samples on the right half-plane as $\triangle$ and samples on the left half-plane as $\square$.

Then $p(\triangle) \approx 0.88$, $1 - p(\square) \approx 0.88$ and $L = p(\triangle)\,(1 - p(\square)) \approx 0.77$.

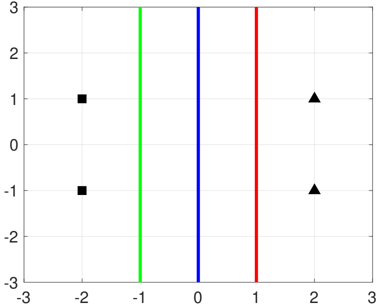
Example II



The global certainty of each classifier (i.e. boundary) is:

- $L = p(\triangle)\,(1 - p(\square)) \approx 0.70$
- $L = p(\triangle)\,(1 - p(\square)) \approx 0.77$
- $L = p(\triangle)\,(1 - p(\square)) \approx 0.70$

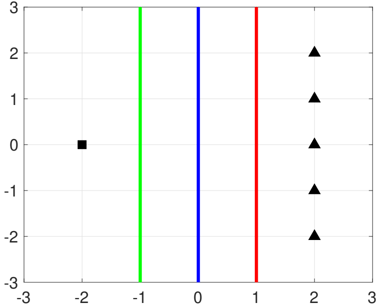
Example III



The global certainty of each classifier (i.e. boundary) is:

- $L \approx 0.49$
- $L \approx 0.60$
- $L \approx 0.49$

Example IV



The global certainty of each classifier (i.e. boundary) is:

- $L \approx 0.20$
- $L \approx 0.47$
- $L \approx 0.57$

Agenda

Formulating classification problems

Linear classifiers

Logistic model

Nearest neighbours

Summary

Parametric and non-parametric approaches

Linear classifiers belong to the family of **parametric** approaches: a shape is assumed (in this case linear) and our dataset is used to find the best boundary amongst all the boundaries with the preselected shape.

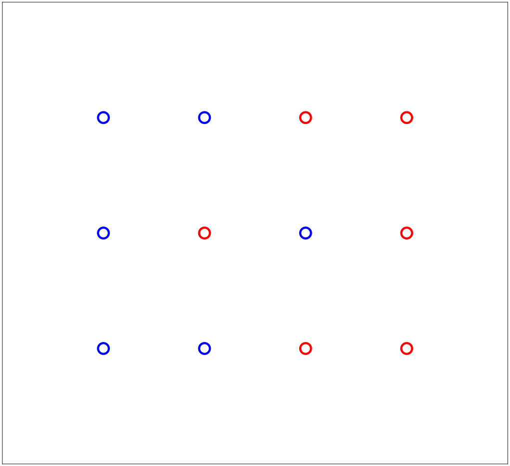
Non-parametric approaches offer a more flexible alternative, as they do not assume any type of boundary. In this section, we will study a popular non-parametric approach, namely **k Nearest Neighbours** (kNN).

Nearest Neighbours

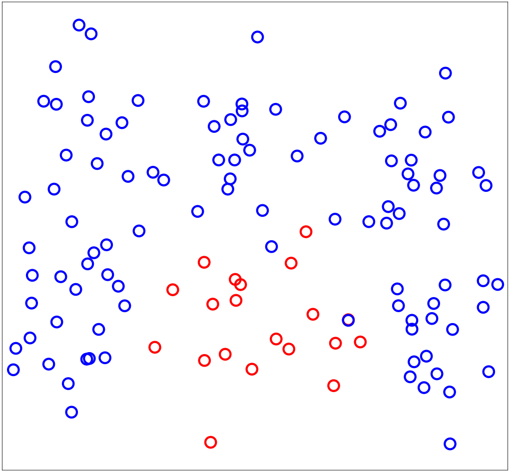
In nearest neighbours (NN), new samples are assigned the **label of the closest (*most similar*) training sample**. Therefore:

- Boundaries are not defined explicitly (although they exist and can be obtained).
- The whole training dataset needs to be **memorised**. That's why sometimes we say NN is an **instance-based method**.

Boundaries in Nearest Neighbours classifiers



Boundaries in Nearest Neighbours classifiers



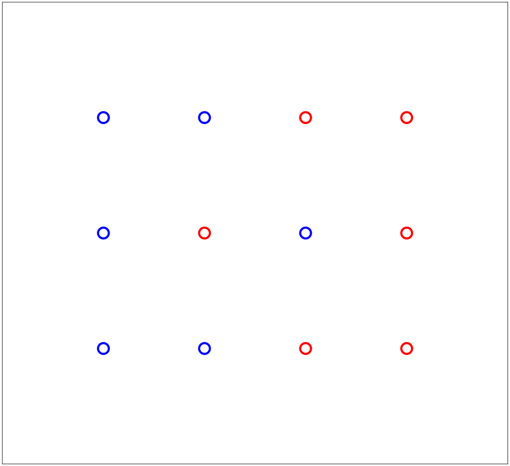
k Nearest Neighbours

Boundaries in nearest neighbours classifiers can be too complex and hard to interpret. Can we smooth them?

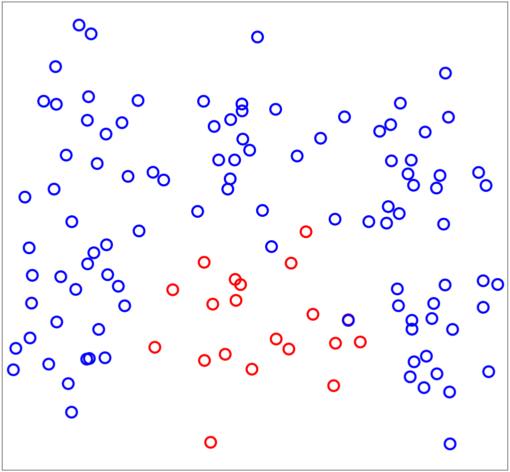
K nearest neighbours (kNN) is a simple extension of nearest neighbours that proceeds as follows. Given a new sample $\boldsymbol{x}$:

- We calculate the distance to all the training samples $\boldsymbol{x}_i$.
- Extract the K closest samples (neighbours).
- Obtain the number of neighbours that belong to each class.
- Assign the label of the most popular class among the neighbours.

Boundaries in kNN classifiers



Boundaries in kNN classifiers



k Nearest Neighbours

Note that:

- There is always an implicit boundary, although it is not used to classify new samples.
- As K increases, the boundary becomes less complex. We move away from **overfitting** (small K) to **underfitting** (large K) classifiers.
- In binary problems, the value of K is usually an odd number. The idea is to prevent situations where half of the nearest neighbours of a sample belong to each class.
- kNN can be easily implemented in multi-class scenarios.

Agenda

Formulating classification problems

Linear classifiers

Logistic model

Nearest neighbours

Summary

Machine learning classifiers

- Classifiers are partitions of the predictor space into **decision regions** separated by **boundaries**.
- Each decision region is associated with one label.
- In machine learning, classifiers are built using a **dataset** (otherwise it's not machine learning!).

Flexibility and complexity in classifiers

- The notions of flexibility, complexity, interpretability, overfitting and underfitting also apply to classifiers.
- Linear boundaries are simple and rigid; kNN produces boundaries whose complexity depends on the value of K.
- Logistic regression is a strategy to train linear classifiers. It's called *regression* because indirectly we solve a regression problem or the classifier's certainty.
- Weirdly, kNN does not involve training as it uses all the samples each time a new sample is to be classified.

Hey, hold on a second, what's going on?

- In machine learning we use a quality metric to define what we mean by the *best* model.
- We have presented two quality metrics: **accuracy** and **error rate**.
- However, **neither** the logistic regression nor kNN classifiers **use the notion of accuracy**.
- What's going on?