UNIVERSITY OF BIRMINGHAM

Week 3. k-Nearest Neighbours (kNN)

Dr. Shuo Wang

Overview

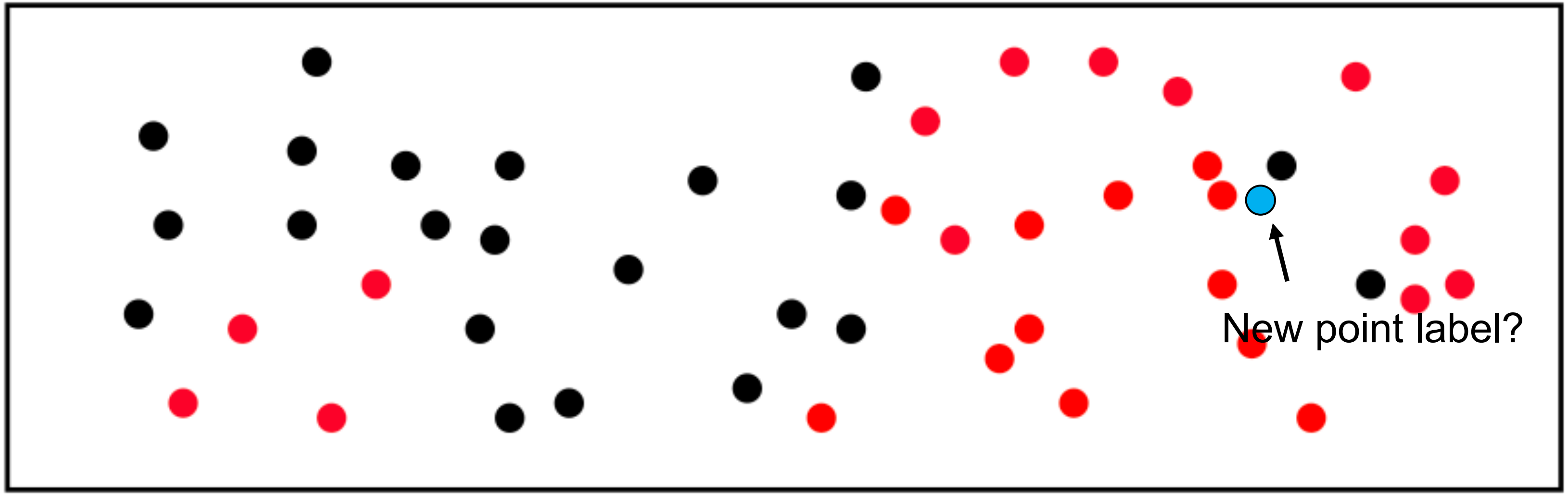
- Intuitive understanding
- The kNN algorithm
- Pros/cons

kNN Basics

- Full name: k-Nearest Neighbours (kNN, or k-NN).
- It is nonparametric.
- It is instance-based.
- It is a lazy algorithm.

Intuitive Understanding

Instead of approximating a model function $f(x)$ globally, kNN approximates the label of a new point based on its nearest neighbours in training data.



Q1: How to choose k? e.g. let k = 3 to avoid issues.

Q2: how to we measure the distance between examples?

Distance metrics (or similarity metrics)

Given two points $\boldsymbol{x}^{(1)} = \left(x_1^{(1)}, x_2^{(1)}, \dots, x_d^{(1)}\right), \boldsymbol{x}^{(2)} = \left(x_1^{(2)}, x_2^{(2)}, \dots, x_d^{(2)}\right)$ in a d-dimensional space:

- Minkowski distance (or L^p norm)

$$D\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\right) = \sqrt[p]{\sum_{i=1}^{d} \left|x_i^{(1)} - x_i^{(2)}\right|^p}$$

- When p=1, it becomes Manhattan distance

$$D\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\right) = \sum_{i=1}^{d} \left|x_i^{(1)} - x_i^{(2)}\right|$$

- When p=2, it becomes Euclidean distance

$$D\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\right) = \sqrt{\sum_{i=1}^{d} \left|x_i^{(1)} - x_i^{(2)}\right|^2}$$

Distance metrics in kNN (common choice)

- Euclidean distance for real values (also called L^2 distance).

$$D\left(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}\right) = \sqrt{\sum_{i=1}^{d} \left| x_i^{(1)} - x_i^{(2)} \right|^2}$$

- Hamming distance for discrete/categorical values, e.g. $x \in \{rainy, sunny\}$.

$$D\left(x^{(1)}, x^{(2)}\right) = \begin{cases} 0, if\ x^{(1)} = x^{(2)} \\ 1, \quad otherwise \end{cases}$$

kNN algorithm

Input: neighbour size $k > 0$, training set $\{(\boldsymbol{x}^{(n)}, y^{(n)}): n = 1,2 \dots N\}$, a new unlabelled data $\boldsymbol{x}^{(j)}$

for n = 1, 2… N // each example in the training set

Calculate $D(\boldsymbol{x}^{(j)}, \boldsymbol{x}^{(n)})$ // distance between $x^{(j)}$ *and* $x^{(n)}$

Select k training examples closest to $\boldsymbol{x}^{(j)}$

Return $y^{(j)}$ = the plurality vote of labels from the k examples. (classification) or

$y^{(j)}$ = average/median of the y values of the k examples. (regression)

Another visual example

kNN for a 3-class problem with k=5

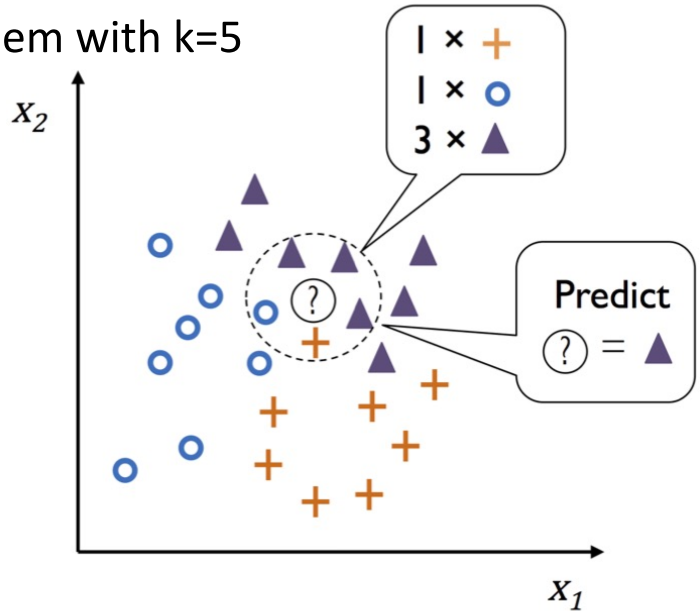


Image from “Python Machine Learning”, by Sybastian Raschka

How to choose k?

- Recall: Overfitting and Underfitting

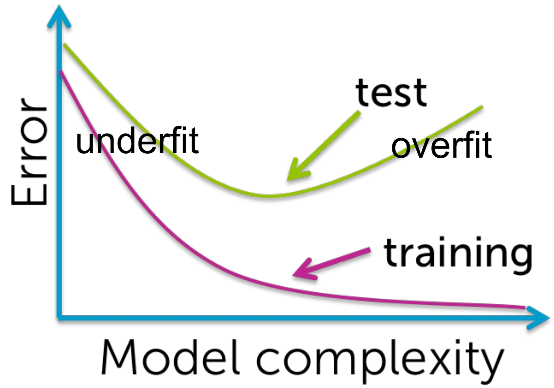


Image from https://neptune.ai/blog/knn-algorithm-explanation-opportunities-limitations

The issue in numeric attribute ranges

- Attributes $\boldsymbol{x} = (x_1, x_2, \dots, x_d)$ may have different ranges.
- What is the problem?
- Solutions?

Normalisation and Standardization

- Method 1 Normalisation: Linearly scale the range of each attribute to be, e.g. in [0,1].

$$x_{j\_new}^{(n)} = \frac{x_j^{(n)} - \min x_j}{\max x_j - \min x_j}$$

- Method 2 Standardization: Linearly scale each dimension to have 0 mean and variance 1 (by computing mean μ and variance σ^2).

$$x_{j\_new}^{(n)} = \frac{x_j^{(n)} - \mu_j}{\sigma_j}, \text{ where } \mu_j = \frac{1}{N}\sum_{n=1}^{N} x_j^{(n)}, \sigma_j = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x_j^{(n)} - \mu_j)^2}$$

kNN algorithm with normalisation

Input: neighbour size $k > 0$, training set $\{(\boldsymbol{x}^{(n)}, y^{(n)}): n = 1,2 \ldots N\}$, a new unlabelled data $\boldsymbol{x}^{(j)}$

Normalise/standardize $\boldsymbol{x}^{(j)} \rightarrow \boldsymbol{x}_{new}^{(j)}$

for n = 1, 2… N // each example in the training set

Normalise/standardize $\boldsymbol{x}^{(n)} \rightarrow \boldsymbol{x}_{new}^{(n)}$

Calculate $D(\boldsymbol{x}_{new}^{(j)}, \boldsymbol{x}_{new}^{(n)})$ // normalized/standardized distance

Select k training examples closest to $x^{(j)}$

Return $y^{(j)}$ = the plurality vote of labels from the k examples. (classification) or

$y^{(j)}$ = average/median of the y values of the k examples. (regression)

Fun project using kNN: where on earth is this photo from?

- Problem: where was this picture taken (country or GPS)?
- http://graphics.cs.cmu.edu/projects/im2gps/

- Get images from Flickr with gps info.
- Represent each image with meaningful features
- Apply kNN.

Q/A

Teams Channel: www.birmingham.ac.uk/
Office Hour: [faculty or individual email]@bham.ac.uk