Machine Learning, Lecture 2: k-nearest neighbours

S. Nõmm

[1] Department of Computer Science, Tallinn University of Technology

12.02.2015

Distance and/or Similarity

Let x and y are two elements (objects). Define measure of distance/similarity between x and y

Metric (some times referred as distance function)

Definition

A function $d : X \times X \to \mathbb{R}$ is called metric if for any elements x, y and z of X the following conditions are satisfied.

1. Non-negativity or separation axiom
$$d(x, y) \geq 0$$
2. Identity of indiscernibles, or coincidence axiom
$$d(x, y) = 0 \Leftrightarrow x = y$$
3. Symmetry
$$d(x, y) = d(y, x)$$
4. Subadditivity or triangle inequality)
$$d(x, z) \leq d(x, y) + d(y, z)$$

Examples: distances in the Euclidean space 1

Do you remember what Euclidean space is?

- Euclidean distance

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- Manhattan distance also referred as city block distance or taxicab distance

$$d(x, y) = \sum_{i=1}^{n} \mid x_i - y_i \mid$$

- Chebyshev distance

$$d(x, y) = \lim_{k \to \infty} \Big(\sum_{i=1}^{n} \mid x_i - y_i \mid^k \Big)^{\frac{1}{k}} = \max_i \big(\mid x_i - y_i \mid \big)$$

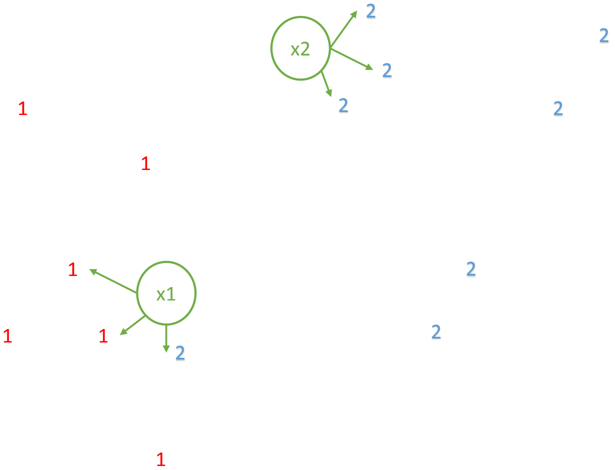
k-nearest neighbour (k-NN) classification

- Let N be a labeled set of points belonging to c different classes such that

$$\sum_{i=1}^{c} N_i = N$$

- Classification of a given point x
  - Find k - nearest points to the point x.
  - Assign x the majority label of neighbouring (k-nearest) points

Example



(k-NN) classification

- k-NN is a supervised learning method
- it is nonparametric learning method (number of the parameters grows with the amount of data)
- k-NN is a memory (or instance) -based learning, (algorithm memorizes the training data).
- k is the hyperparameter.

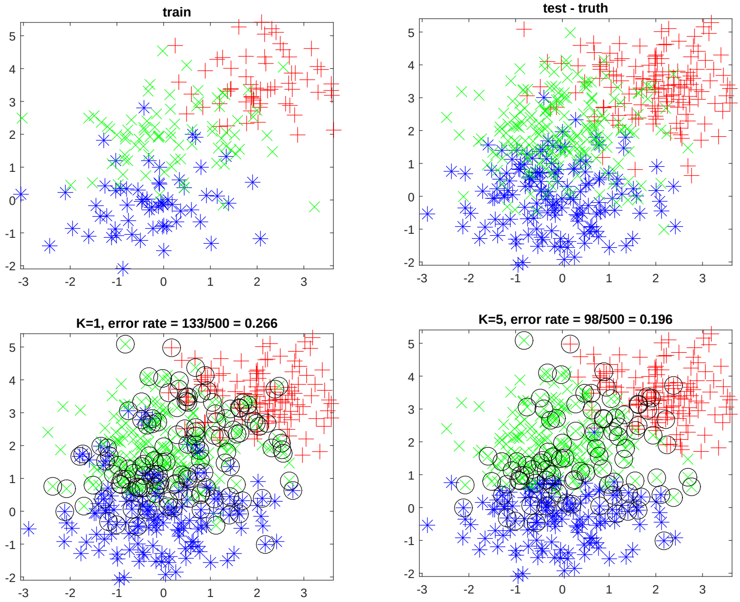
(k-NN) classification

- For an arbitrary point x the probability to belong to the class c is given by

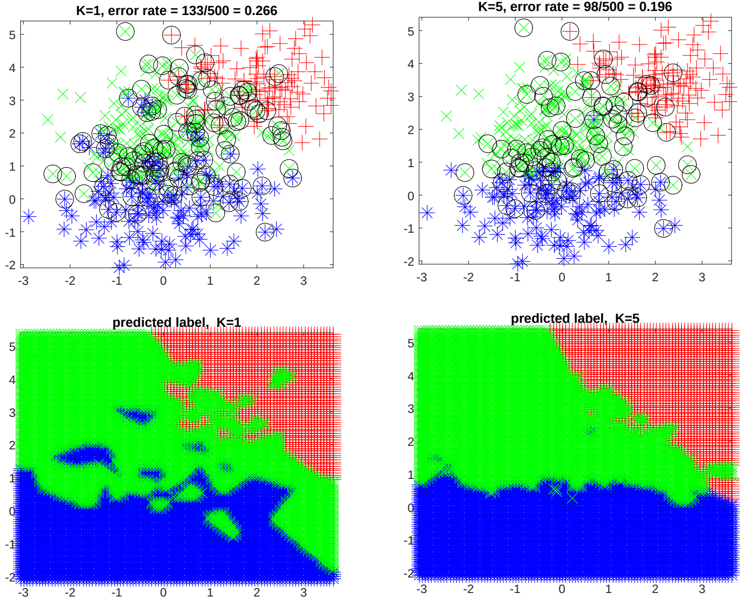
$$p(y = c \mid x, \mathcal{D}, k) = \frac{1}{k} \sum_{i \in N_{k(x,\mathcal{D})}} \mathbb{I}(y_i = c)$$

  here $N_{k(x,\mathcal{D})}$ denotes the indexes of the k nearest points to x in $\mathcal{D}$

Example


train
test - truth
K=1, error rate = 133/500 = 0.266
K=5, error rate = 98/500 = 0.196

Example



K=1, error rate = 133/500 = 0.266
K=5, error rate = 98/500 = 0.196
predicted label, K=1
predicted label, K=5

Decision boundary

- Decision boundary or decision surface (the lines between different colors on the previous slide) is a "hypersurface" that partition the vector space in accordance to two classes it separates.
- Not necessarly surface in the strict sense of this word.
- Decision boundaries characterize the complexity of the model
  - Decision boundary is too "complex" - overfitting.
  - Decision boundary is too "smooth" - underfitting.
- the value k is used to control the complexity of the decision boundary
- Cross-validation may be used to select value k

Examples: distances in the Euclidean space 2

Do you remember what Euclidean space is?

- Mahalanobis distance

$$d(x, y) = \sqrt{(x-y)^T S^{-1} (x-y)}$$

where S is the covariance matrix.

- Cosine distance Cosine similarity is the measure of the angle between two vectors

$$d_c(x, y) = \frac{x \cdot y}{\|x\| \|y\|}$$

Usually used in high dimensional positive spaces, ranges from -1 to 1. Cosine distance is defined as follows

$$d_C(x, y) = 1 - d_c(x, y)$$

Examples 3: Distances between strings. Similarity?

- Levenshtein or SED distance. SED - minimal number of single -charter edits required to change one string into another. Edit operations are as follows:
  - insertions
  - deletions
  - substitutions
- SED(delta, delata)$=1$ delete "a" or SED(kitten,sitting)$=3$: substitute "k" with "s",substitute "e" with "i", insert "g".
- Hamming distance Similar to Levenshtein but with substitution operation only. Frequently used with categorical and binary data.

Data normalization

Normalization - is the process of adjusting values measured on different scales to a common scale. There are different ways to normalize the data:

- Standard score Works well for normally distributed data. For each dimension j compute

$$x'_{i,j} = \frac{x_{i,j-\bar{\mu}_j}}{\sigma_j}.$$

- Feature scaling used to bring all values into the range $[0, 1]$.

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

  may be generalized to bring the values in to and closed interval $[a, b]$

$$x' = a + \frac{\big(x - min(x)\big)(b - a)}{max(x) - min(x)}$$

Note x' denotes normalization, not to be confused with derivative.

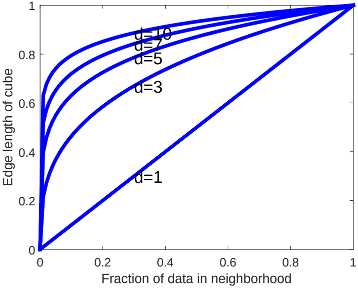
Curse of dimensionality

- k-NN-s are best applied to the cases with "good" distance metric and enough labeled data
- k-NN-s do not perform well in the case of high dimensional problems due to the phenomenon refereed as *curse of dimensionality*.
  - Consider the case when data is distributed uniformly in d-dimensional unit cube.
  - Choose a point x and form a cube around, such that it will include a fraction f of all available points
  - Expected edge length of this cube is

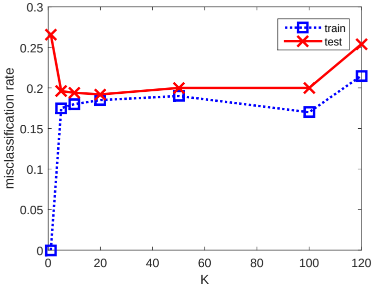
$$E_d\big[s(f)\big] = f^{\frac{1}{d}}$$

Curse of dimensionality

Let $f = 0.01$ Compute yourself the edge length for the values $d = 1, \ldots, 10$. Neighbours that are "far" away may not be good predictors.



Misclassification rate



Mixed Quantitative and Categorical Data

$$d(x,y) = \lambda d_q(x_q, y_q) + (1-\lambda)(d_c(x_c, y_c))$$

here index q denotes quantitative and c categorical data.

- how to choose λ?
- data normalization ?