Machine Learning, Lecture 2: k-nearest neighbours

S. Nõmm

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

12.02.2015

Organisation clarified

No Plagiarism in any of your tests and final project!!!. You should cite all the references, including libraries you use to complete your computational assignments. The student should be able to explain the meaning of all the computations performed, interpret and present the results. Grading: Detailed information about the grading may be found in the first lecture.

- Five (5) home assignments, each gives you max 10 % of the final grade. If you missed the deadline your grade will be reduced by 10% for each day you missed.
- Final project gives you max 50% of the final grade. Precise guidelines will be available from the course page one month before the examination date. During examination each student will be given 3 min. to explain the problem, describe chosen methodology, and present the results. This part will be followed by more detailed examination of the implementation, student may be asked to change implementation on the fly and/or run it on the new data set.
- Note, your own implementation graded higher than usage of third party libraries.

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

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

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

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

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

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

Examples: distances in the Euclidean space 3

- Standardized Euclidian distance
- Correlation distance
- Spearman distance
- Hamming distance
- Levenshtein
- Jaccard distance

Please verify, if all the distances mentioned above (starting with Euclidian ;-)) are metrics?

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.

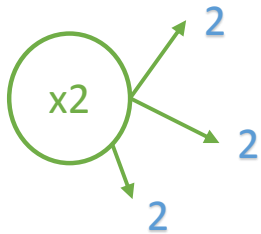
k-nearest neighbour (k-NN) classification

- 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

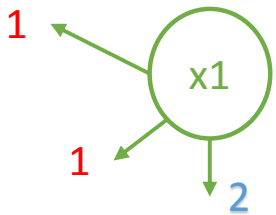


2

1

2

1



2

2

1

(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.

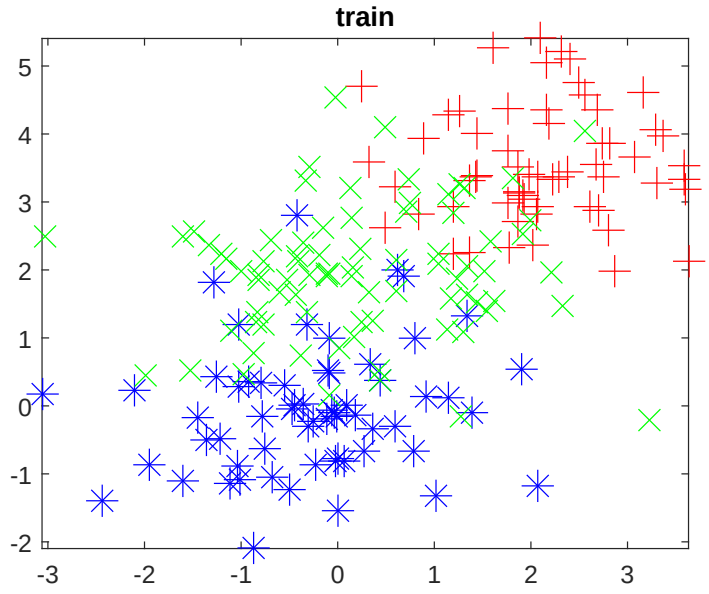
(k-NN) classification

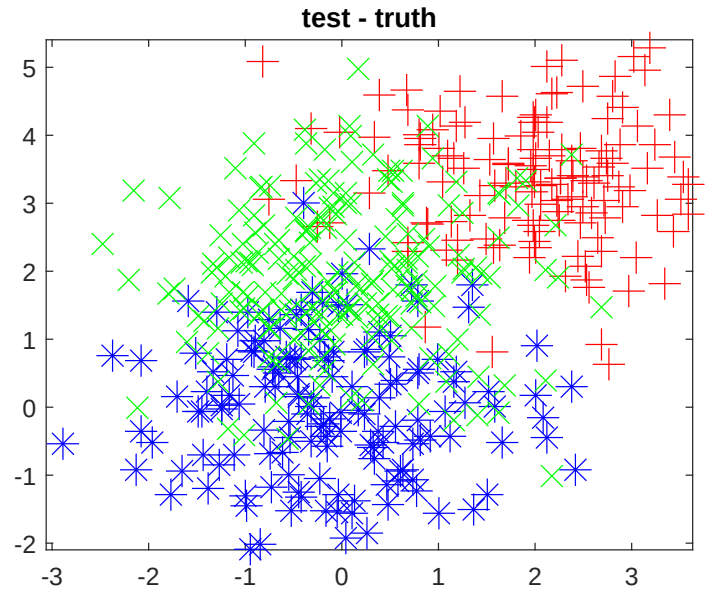
- For an arbitrary point x the probability to belong to the class c is given by

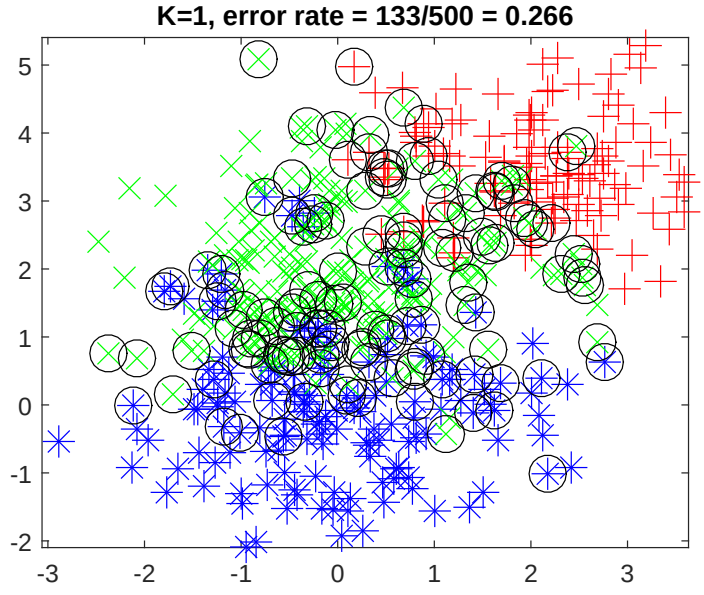
$$p(y = c \mid x, \mathcal{D}, k) = \frac{1}{k} \sum_{i \in N_{k(x,\mathcal{D})}} \mathbb{I}(y_i = c)$$

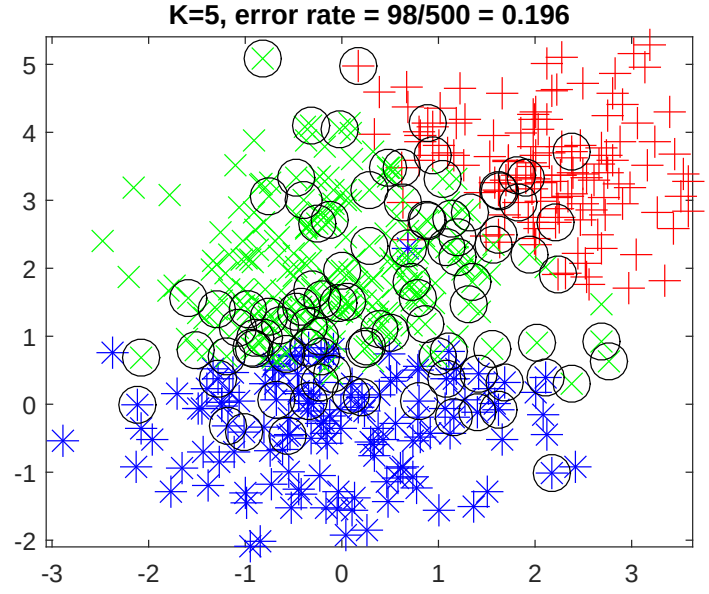
here $N_{k(x,\mathcal{D})}$ denotes the indexes of the k nearest points to x in $\mathcal{D}$

Example

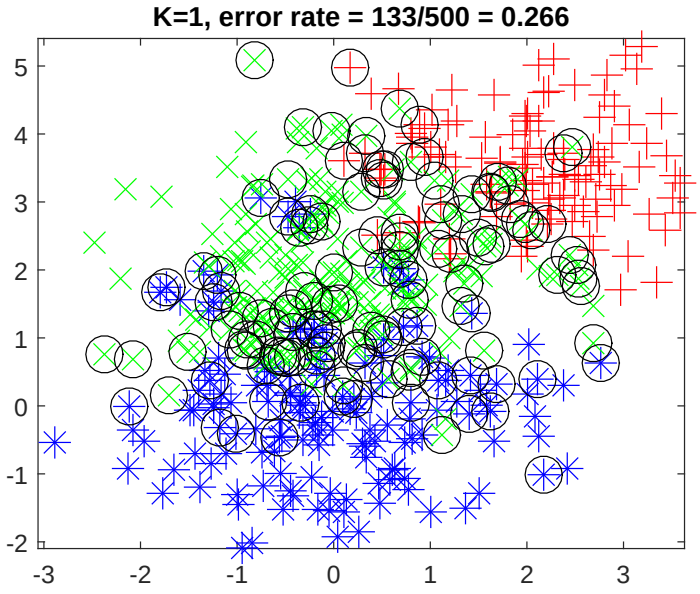

train
5
4
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3

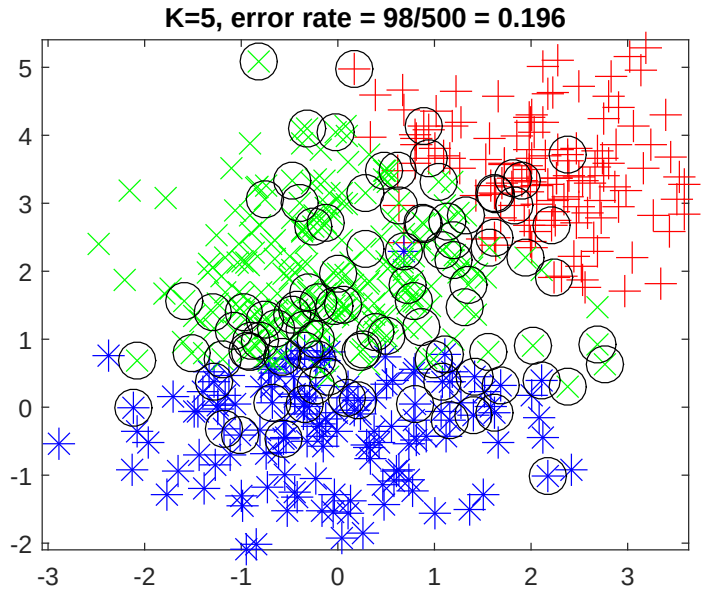

test - truth
5
4
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3

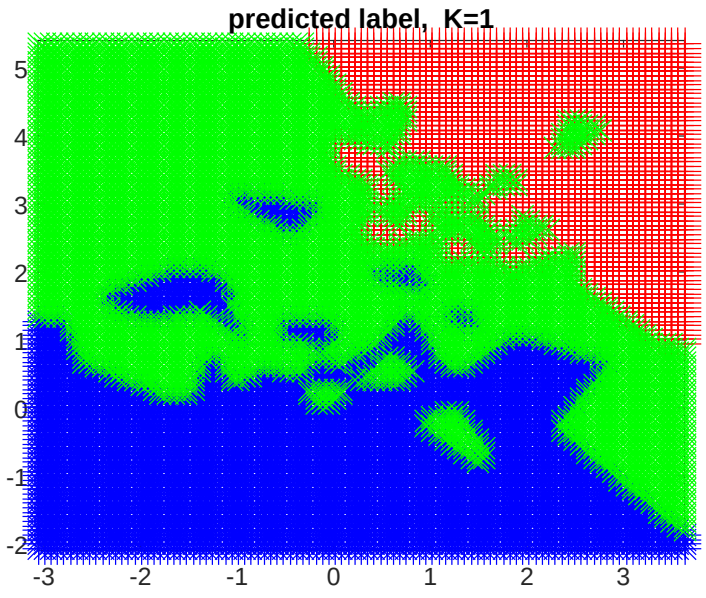

K=1, error rate = 133/500 = 0.266
5
4
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3

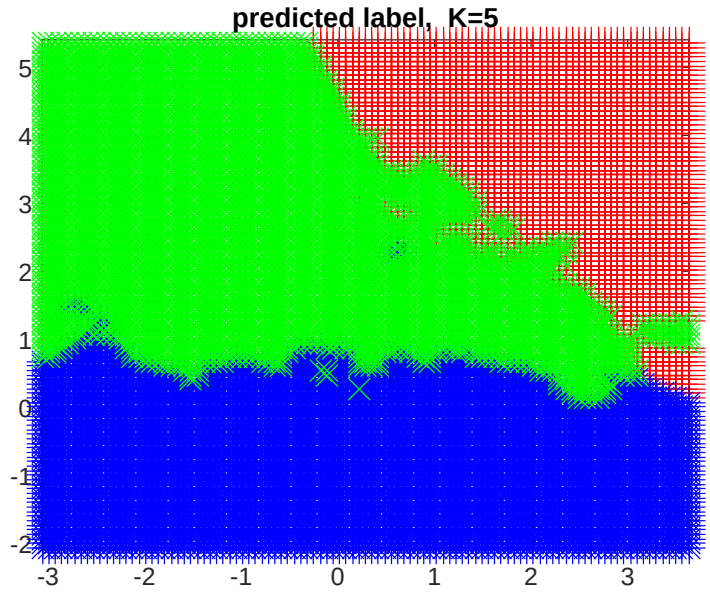

K=5, error rate = 98/500 = 0.196
5
4
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3

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

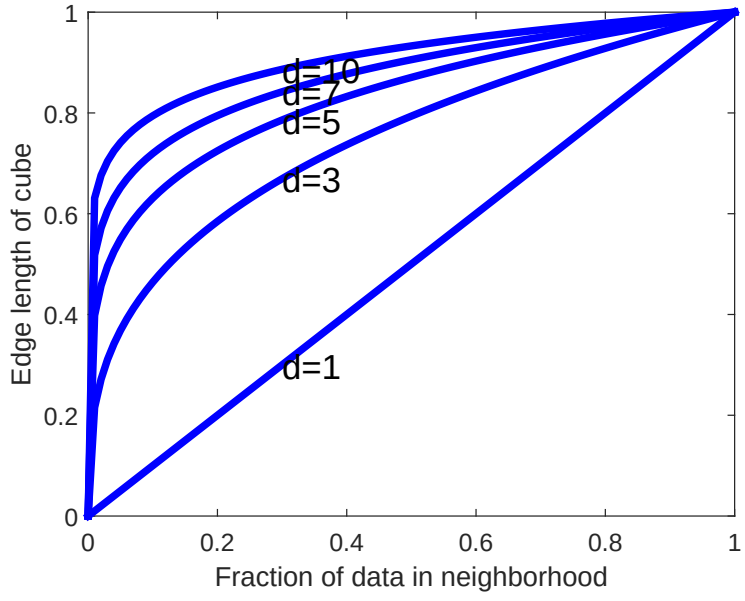
Curse of dimensionality

- 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

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

Curse of dimensionality

Let $f = 0.01$ (



for the values $d = 1, \ldots, 10$. may not be good predictors.

Misclassification rate