Machine Learning
Introduction & distance function

S. Nõmm

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

31.01.2023

Course organization I

- Lectures on Tuesdays 15:30 - 17:00 ICT-A1; Practices on Thursdays 16:00 - 17:13 ICT-401.
- Lectures and Practces are recorded and availble in MS Teams.
- If joining online, please keep camera on and microphone muted. **Only teacher or lecturer may initiate meeting and start recording.**
- Any changes in lecture and practice occurrence will be communicated to the students through the MS teams environment and TalTech moodle.
- Moodle course environment requires the code HAL900 to enroll. All the assignments and grading will be conducted via course Moodle environment whereas, `https://gitlab.cs.ttu.ee/` environment should be used by students to submit their code. NB NO E-MAIL submissions are accepted!!!
- Open web page of the course `https://courses.cs.ttu.ee/pages/ITI8565`

Course organization II: Grading

- Three **mandatory** home assignments! Each home assignment gives 10% of the final grade. Assignment consists of computational part to be performed in Python, report (max 2 pages, template is provided) and short defence with the lecturer or teaching assistants. Precise instructions will be provided with each assignment in TalTech Moodle.
- Two mandatory closed book tests each gives 10% of the final grade.
- Final exam, referred as *final project* (Written report on assign topic (programming is required) + short presentation) gives 50% of the final grade. Exam and final project are synonyms within the frameworks of the course.
- Please note defense of the final project will take place at examination time.
- Note, your own implementation graded higher than usage of third party libraries.

Course organization IV

- For correspondence please use sven.nomm@taltech.ee Please do not call me by phone!!!
- Consultations by appointment only. Do not hesitate to ask for the consultation.
- Some of the course topics require the student to learn certain chapters independently. It is assumed that student will be given general guidance(reference to a particular books) but will learn the theory independently.

References

Present lectures are based on:

- Machine Learning: A Probabilistic Perspective Textbook by Kevin R. Murphy. MIT Press, Aug 24, 2012 - Computers - 1067 pages (available in TUT Library).
- The Elements of Statistical Learning Book by Jerome H. Friedman, Robert Tibshirani, and Trevor Hastie, Springer (available through the TUT library as e-book, UNIID required).
- Pattern Recognition and Machine Learning Book by Christopher Bishop. Springer. (available in TUT Library)
- Data Mining The Text Book by C. Aggarwal. 2015 Springer.

In course of the lectures, whenever necessary additional references will be shared.

Prerequisites

- Knowledge of Calculus, Foundations: function, derivative etc.
- Knowledge of Linear algebra, Foundations: matrixes, matrix operations, polynomials etc.
- Medium level of programming skills.
- Foundations of probability theory and statistics.
- You are STRONGLY advised to write your own notes!!!
- Lecture slides could not substitute attendance, there are quite many things are discussed during lectures which are not reflected in the slides.
- It is assumed that students are: attending the lectures, read books and do home works regularly!!!

Course plan (tentative)

- 31.01 Introduction and distance function.
- 07.02 Cluster analysis I
- 14.02 Cluster analysis II and Outlier analysis.
- 21.02 Supervised learning I Classification.
- 28.02 Supervised learning II Regression.
- 05.03 23:59 **Deadline to submit Home Assignment I.**
- Defense of Home assignment I 09.03.2023
- 07.03 Supervised learning III Gradient descent.
- 14.03 Supervised learning IV Support Vector Machines.
- 21.03 Supervised Learning V Improving model quality.
- 28.03 Test I.

Course plan (tentative)

- 30.03 Markov models.
- 02.04 23:59 **Deadline to submit Home Assignment II.**
- 04.04 Neural networks I.
- Defense of Home assignment II 06.04.2023
- 11.04 Neural networks II.
- 18.04 Deep learning I: RNN.
- 25.04 Deep learning II: Convolutional Neural Networks.
- 02.05 Deep Learning III: Transformers.
- 09.05 Foundations of eXplainable AI.
- 14.05 23:59 **Deadline to submit Home Assignment III.**
- 16.05 Test II
- Defense of Home assignment III 18.05.2023

What is Learning?

According to:
http://www.merriam-webster.com/dictionary/learning

learning noun

: the activity or process of gaining knowledge or skill by studying, practicing, being taught, or experiencing something : the activity of someone who learns

- the act or experience of one that learns
- knowledge or skill acquired by instruction or study
- modification of a behavioral tendency by experience (as exposure to conditioning)

What is *Machine Learning*?

Arthur Samuel, 1959 has defined Machine learning as: Field of study that gives computers the ability to learn without being explicitly programmed.

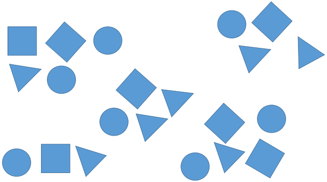
Tom Mitchell, 1997 has defined Machine learning as:
A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Main steps

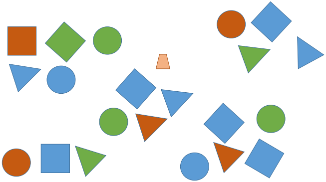
- Data acquisition.
- Data preparation.
- Feature selection.
- Model training.
- Tuning (sometimes).
- Model validation.

From this point it is expected that when solving an exercise student chooses and executes all necessary steps and able to explain their decision.

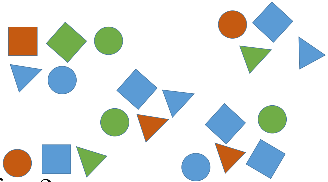
Similarity and distance



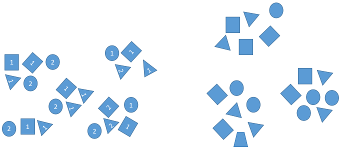
Set 1



Set 3



Set 2



set 4

Distance function

Distance function is one of most fundamental notions in Machine learning and Data mining. Formally defined in pure mathematics as *metric* function. It provides measure of similarity or distance between two elements.

Definition

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

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

Distance function: Examples 1 (Most common distance functions)

- Euclidean distance

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

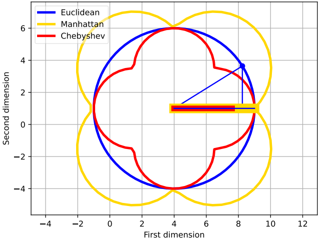
- Manhattan distance also referred as city block distance or taxicab distance

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

- Chebyshev distance

$$S(x,y) = \max_i \big(\mid x_i - y_i \mid \big)$$

Distance function: Examples 2



Euclidean distance

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

Manhattan distance

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

Chebyshev distance

$$S(x, y) = \max_i \left(\mid x_i - y_i \mid \right)$$

Distance function:Examples 3 Minkowsky distance

$$S(x, y) = \Big(\sum_{i=1}^{d} \mid x_i - y_i \mid^p\Big)^{1/p}$$

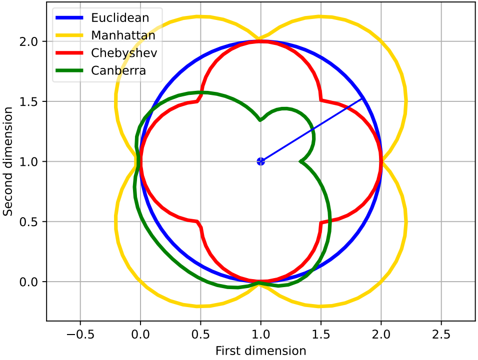
- $p < 1$ triangle inequality is violated, therefore for the values of p smaller than one, equation above is not a distance function.
- $p = 1$ case of Manhattan distance.
- $p = 2$ case of Euclidian distance.
- $p \to \infty$ case of Chebyshev distance.

Distance function:Examples

- Canberra distance

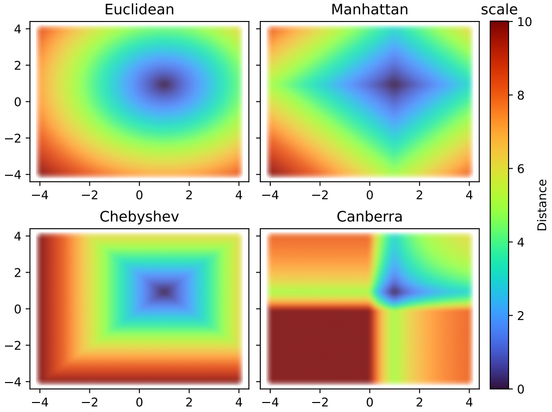
$$S(x,y)=\sum_{i=1}^{n}\frac{|x_i-y_i|}{|x_i|+|y_i|}$$

weighted version of Manhattan distance.



Distance function: Examples 4

Heat map representation of the Minkovsky distance function and Canberra distance function. Colors correspond to the values of the distance function.

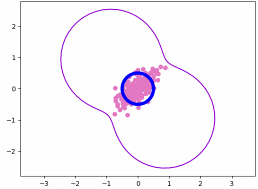


Distance function: Examples 5

Mahalanobis distance

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

where C is the covariance matrix. Takes into account impact of the data set.



Distance function: Examples 6

- Canberra distance
$$S(x,y)=\sum_{i=1}^{n}\frac{|x_i-y_i|}{|x_i|+|y_i|}$$
weighted version of Manhattan distance.
- 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
$$S_C(x,y)=1-S_c(x,y)$$

Distance function: Examples 7

- 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.
- Specialized similarity measures Distance and similarity functions applicable to the graphs, temporal data etc. These topics are left outside of the framework of the present course.

Impact of High Dimensionality (Curse of Dimensionality)

Curse of dimensionality - term introduced by Richard Bellman. Referred to the phenomenon of efficiency loss by distance based data-mining methods. Let us consider the following example.

- Consider the unit cube in d - dimensional space, with one corner at the origin.
- What is the Manhattan distance from the arbitrary chosen point inside the cube to the origin?

$$S(\bar{0}, \bar{Y}) = \sum_{i=1}^{d} (Y_i - 0)$$

  Note that Y_i is random variable in $[0, 1]$
- The result is random variable with a mean $\mu = d/2$ and standard deviation $\sigma = \sqrt{d/12}$
- The ratio of the variation in the distances to the mean value is referred as *contrast*

$$G(d) = \frac{S_{max} - S_{min}}{\mu} = \sqrt{\frac{12}{d}}$$

Impact of High Dimensionality (Curse of Dimensionality)

Data Preparation

Note: some steps listed below are frequently considered in pair of corresponding problem.

- **Feature Extraction.** Feature Extraction is the process of selecting the attributes and features relevant to the goal of analysis.
- **Data Cleaning.** Handling missing entries, handling incorrect and inconsistent entries, scaling and normalization.
- **Data Reduction and Transformation.** Sampling, feature subset selection dimensionality reduction. Conversion between different data types e.g. Numeric to Categorical data or Categorical to Numeric data.

Conversion

- **Discretization:** Numeric to Categorical Data: Divide range of numeric attribute into finite number of intervals. To each data point assign categorical value of the interval containing its numerical attribute.
  - Equi-width ranges: the ranges have the same length.
  - Equi-log ranges: $\log(b) - \log(a)$ have the same length for all the intervals. Here a is a beginning and b is the ending of the intervals.
  - Equi-depth ranges: Each range contains the same number of intervals.
- **Biniarization:** Categorical to Numeric Data
- Text to Numeric Data
- Time Series to Discrete Sequence Data
- Time series to Numeric Data
- Discrete Sequence to Numeric Data
- Spatial to Numeric Data
- Graphs to Numeric Data

Data Cleaning

- Missing Entries
- Incorrect and Inconsistent Entries
- Scaling and Normalization: Different features represent different scales and not always comparable.
  - Normalization Let j^{th} attribute has mean μ_j and standard deviation σ_j then j^{th} attribute value x_i^j of the record $\bar{X}_i$ may be normalized as follows

$$z_i^j = \frac{x_i^j - \mu_j}{\sigma_j} \tag{1}$$

  - Min - max scaling:

$$y_i^j = \frac{x_i^j - \min(x^j)}{\max(x^j) - \min(x^j)} \tag{2}$$

Data Reduction and Transformation

- Sampling
  - Sampling for Static data
    - Biased sampling
    - Stratified sampling
  - Reservoir Sampling for Data Streams
- Feature Subset Selection
- Dimensionality Reduction
  - Principal Component Analysis
  - Singular Value Decomposition
  - Latent Semantic Analysis