Introduction to Data Science

Course 094201

Lab 4:

Classification and Data Manipulation

Spring 2019

מטרות המעבדה

במעבדה זו תדרשו לעבוד עם מסווג KNN שנלמד בהרצאה.

מטרות:

- הבנת אופן עבודת ה KNN.
- הבנת תהליך cross-validation.
- שיטות נרמול של משתנים.

The dataset and the code

- The code and the data can be found at the moodle.
- Copy lab4_students to your local folder and unzip the code
- The dataset is described in the next slide.
- You are asked to add the **normalization** functionality and check how the KNN classifier’s effectiveness is affected.

The data - Pima Indians Diabetes Database

The data contains information regarding patients. All patients here are females at least 21 years old of Pima Indian heritage. The class we want to learn is the **presence of diabetes.**

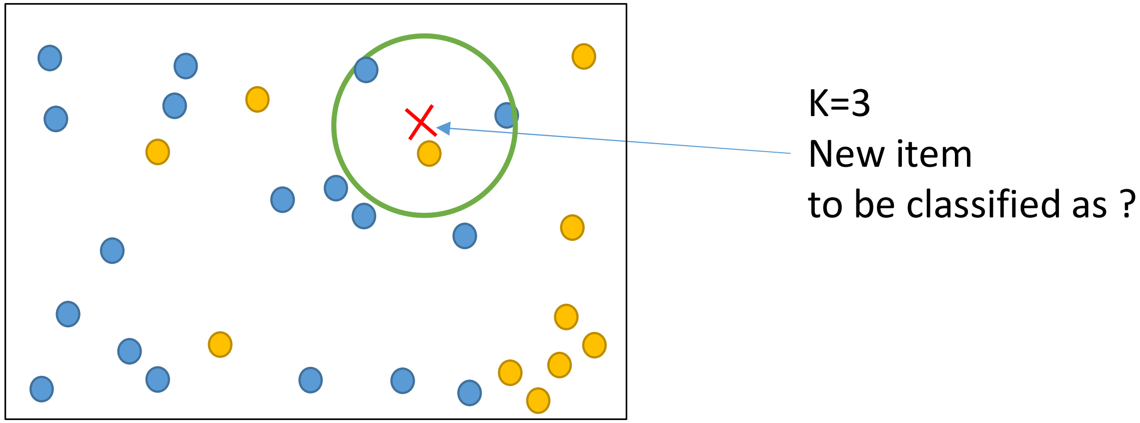
Attribute Information:

1. Number of times pregnant
2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
3. Diastolic blood pressure (mm Hg)
4. Triceps skin fold thickness (mm)
5. 2-Hour serum insulin (mu U/ml)
6. Body mass index (weight in kg/(height in m)^2)
7. Diabetes pedigree function
8. Age (years)
9. Class variable (0 or 1)

Reference: Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., & Johannes, R.S. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. *In Proceedings of the Symposium on Computer Applications and Medical Care* (pp. 261--265). IEEE Computer Society Press.

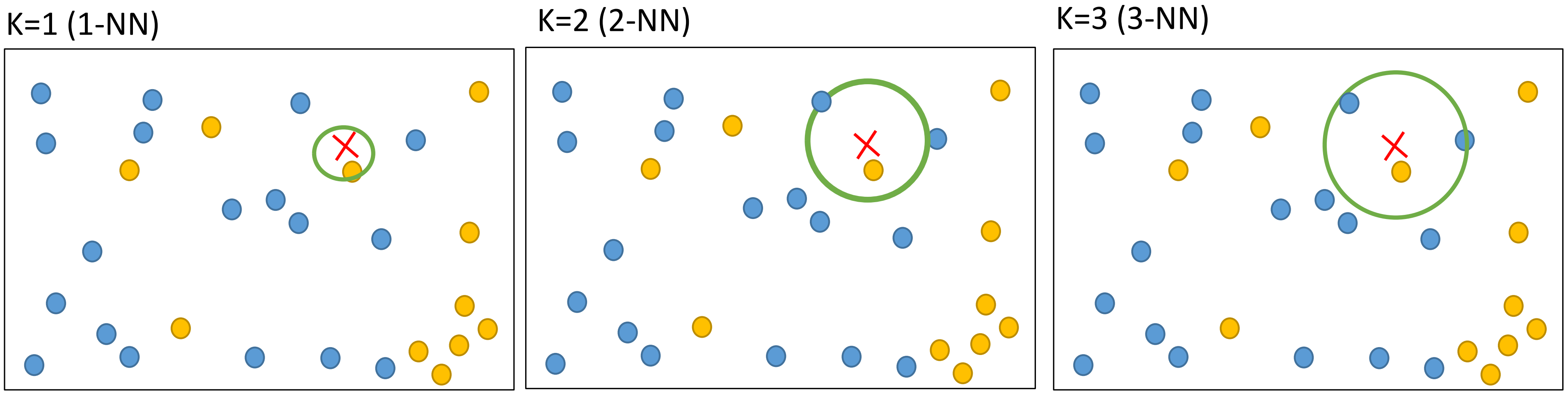
K-Nearest Neighbors (KNN) classifier

- Distance based classifier
- Lazy classifier: does not generalize a model from the data, but rather keeps all the data in memory
  - Large space complexity
  - No training phase (all the data points are stored during training)
- The new item is classified as the majority of its nearest K neighbors



K=3
New item
to be classified as ?

KNN



- KNN makes no assumptions about the data or about the number of classes
- Distance metric is required (at the above example Euclidean distance is used)
- In binary classification problems, the number of neighbors is usually set to be odd

Data Normalization - Motivation

- Often the different variables (features) are measured in different units and thus have large differences in scale
  - For example, today we will work with a dataset where each item is a patient represented by:
    - Diastolic blood pressure (mm Hg)
    - Age in years
    - 2-Hour serum insulin (mu U/ml)
    - And more…
- Most distance metrics (e.g.: Euclidean distance, L1-norm) will be affected: larger scaled features will dominate the others
  - Unless there is some explicit feature weighting, all features should have the same importance **before** a model is learned

Data Normalization - Example

- Distance based classifiers will be biased towards features with larger scale
- Classifiers which use different numerical algorithms might exhibit problems
  - For example, some more advanced algorithms (you'll learn in other courses) might have convergence time issues as a result of different scales of features
- **Solution: we want to normalize all the variables (features) to the same scale**

Data Normalization – to unit L$_p$ norm

Given a dataset where:
X is a quantitative variable (e.g., age), x_i is the value for item i (e.g., x_i=26)
n is the number of items in the dataset

We divide each value by the norm of the vector (x_1, x_2,…, x_n)

L1-norm - sum normalization	L2-norm
$x_i^{L_1} = \frac{x_i}{\sum_{1 \le i \le n} \lvert x_i \rvert}$	$x_i^{L_2} = \frac{x_i}{\sqrt{\sum_{1 \le i \le n} x_i^2}}$

Data Normalization – Z Norm

- We **standardize** all the points by using the mean and standard deviation of each feature.

$$x_i^{z-score} = \frac{x_i - \mu}{\sigma}$$

- μ is the population mean -> can be estimated using $\bar{x}$
- σ is the population standard deviation -> can be estimated using s
- The resulting values have the mean of 0 and quantify the distance between x_i and μ in units of σ

Data Normalization – Min-Max Norm

- Widely used is min-max normalization, also called range normalization
- For each variable we use its minimum and maximum values:
  - We first **shift x_i to the origin**
  - We **scale the shifted value by its range**. The result is in [0,1]

$$x_i^{min-max} = \frac{x_i - x_i^{min}}{x_i^{max} - x_i^{min}}$$

The code – what do we have?

1. בקוד הניתן עם התרגיל ישנן המחלקות הבאות:

מחלקה\קובץ	תיאור
KNN	המסווג KNN.
CrossValidation	מריצה את המסווג, מודדת יעילות
Metrics	אוסף של פונקציות שמודדות איכות של סיווג
Normalization	אוסף של מחלקות המשנות את הנתונים.
Point	מחלקה המייצגת אובייקט יחיד לסיווג.

Run the code

- הקוד הינו במצב **מוכן להרצה** וכדי להתנסות בו עליכם לקרוא לפונקציות המתאימות.
- **שימו לב**: התוכנה מקבלת פרמטר יחיד והוא שם קובץ הקלט

1. כתבו פונקציה בקובץ main היוצרת מסווג KNN עם k=5 שכנים, הריצו אימון עם כל הנתונים וסווגו את האיבר הראשון בקובץ הנתונים. בידקו מה הפלט של המסווג ומהו ה class הנכון של האובייקט.
2. כתבו פונקציה בקובץ main המריצה 10 fold cross-validation והדפיסו את הפלט. הבינו את משמעותו.

Assignment

- Normalizer interface contains 2 member functions:
  - fit: a function that receives normalization type and a vector of Points objects and set up the normalizer parameters that will be required for normalizing a new point
    - For example: for sum-normalization (L_1) we need to hold the sum of each variable
  - transform: a function that receives a point and returns a normalized version of that point

Assignment (contd.)

- Implement classes with normalizer interface, using the following normalization methods:
  1. Sum-normalization (L_1)
  2. Min-max normalization
  3. Z-Norm
     - For sum-normalization (L_1) we need the sum of each variable
     - For min-max normalization we need the minimal and the maximal values of each variable
     - For Z-Norm we need to save the mean and the standard deviation of each variable