CSE 5243 - Introduction to Data Mining

Instructor: Jason Van Hulse

Homework 1

Submitted by:

Siva Meenakshi Renganathan
meenakshirenganathan.1@osu.edu

Objective:

The objective of this homework is to pre-process the datasets and calculate the closest five neighbors to each tuple in the given dataset using two different distance calculation methods.

i.e. For a given dataset with n tuples the output will be n(2k+1) dataframe as follows (here k=4)

Transaction ID	1st	1-dist	2nd	2-dist	3rd	3-dist	4th	4-dist
1	18	0.1	5	0.141	40	0.141	28	0.141

There are two different datasets:

1) Iris dataset
2) Income dataset.

Implementation and Design:

R is chosen as the programming language mainly because of its extensive statistic capabilities. R is a very popular open source platform exclusively used for Statistical calculations and Data mining purposes. R also has great graphical features to create different types of plots on the dataset. RStudio is a software built on R which gives a GUI to the R's command line interface. R was purposely designed to make data analysis and statistics easier and hence is not as fast language as other programming languages.

Resources Used:

Since R is a language evolved for Data mining, it has a lot of built-in functions that can read, write and operate on data-sets. The algorithm used to measure distances uses only default R library.

PRE-PROCESSING:

Iris and Income datasets have different types of attributes, so each needs some specific pre-processing which is discussed later.

IRIS DATASET:

Iris dataset has five attributes: Petal_length, Petal_width, Sepal_length, Sepal_width and a class attribute. For computation of Euclidean and Manhattan distances this class attribute is omitted.

Read.csv() function in R reads the dataset and loads it into a dataframe. The difference between a dataframe and matrix in R is that dataframe can hold different attribute types while matrices cannot. So a dataframe is always preferred over matrices in R.

Since Iris is a complete dataset without any missing values or outliers it does not need any pre-processing.

INCOME DATASET:

Income dataset is a relatively larger dataset with different attribute types. It also has a lot of missing values and outliers. These have to be handled before calculating the distances.

Handling Missing values aka NA:

- If the type of the missing value is numeric it is replaced with the mean of that column.
  Mean = Sum of all values/No of values.
- If the type of the missing value is non-numeric it is replaced with the mode of that column.
  Mode = Most frequently occurring variable

Normalization:

Normalization is the process of converting the range of an attribute to 0 and 1.

Normalized value = Value – Min value/(Max value –Min value)

The dataset contains different attributes like age, capital_gain, capital_loss, working hours per week. The range of these values are not the same and hence each of these columns will have a different weight on the distance calculation without normalization. In some cases the attribute fnlwgt is too high that without normalization it eclipses every other attributes and takes the lion share in distance computation. To avoid it the data is normalized.

Age, Fnlwgt, Education_category, Capital_gain, Capital_loss, Hours_per_week are the numeric columns here and they are normalized using the formula above.

Categorical Attributes:

Income dataset has categorical attributes like Workclass, Occupation, Relationship, Marital_status, Race, Gender, Nativer_country. Distance calculation for these attribues is simple. If two tuples have the same value for these attributes, the distance is zero and if they have different values, the distance is 1.

Education column is omitted because it is represented by an Ordinal attribute called Education_cat.

DISTANCE CALCULATION:

Each row in the dataset is reinforced with other rows and the Euclidean and Manhattan distances of that row with other rows are calculated and it is stored in a temporary matrix. This temporary matrix has two attributes (Transaction id, Distance). The temporary matrix is then sorted based on the distance in ascending order and the first four entries, which are the closest 4 neighbors are added to the result dataframe. With this temporary matrix, k can be updated without any major changes to the code.

Euclidean distance:

$$dist = \sqrt{\sum_{k=1}^{n}(p_k - q_k)^2}$$

Where p and q are the two rows and it has n columns.

Euclidean distance calculates the root mean square value of the difference between two rows.

Manhattan distance:

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

Where x and y are the two rows and it has n columns.

Manhattan distance is the simple sum of the difference between two rows.

ANALYSIS:

Two different distance calculation metrics are used to calculate distance namely Euclidean and Manhattan. It is evident from the results that these two metrics calculate different distances because one tries to minimize the error by taking a mean square while the other computes simple sum.

IRIS DATSET:

Euclidean neighbors calculated for row - 1

Transaction.ID	X1st	X1st.dist	X2nd	X2nd.dist	X3rd	X3rd.dist	X4th	X4th.dist	X5th	X5th.dist
1	18	0.1	5	0.1414214	40	0.1414214	28	0.1414214	29	0.1414214

Manhattan neighbors calculated for row - 1

Transaction.ID	X1st	X1st.dist	X2nd	X2nd.dist	X3rd	X3rd.dist	X4th	X4th.dist	X5th	X5th.dist
1	18	0.1	5	0.2	40	0.2	28	0.2	29	0.2

In some cases the neighbors calculated by both the methods are the same although the distances vary.

Euclidean neighbors calculated for row - 3

Transaction.ID	X1st	X1st.dist	X2nd	X2nd.dist	X3rd	X3rd.dist	X4th	X4th.dist	X5th	X5th.dist
3	48	0.1414214	4	0.244949	7	0.2645751	13	0.2645751	46	0.2645751

Manhattan neighbors calculated for row - 3

Transaction.ID	X1st	X1st.dist	X2nd	X2nd.dist	X3rd	X3rd.dist	X4th	X4th.dist	X5th	X5th.dist
3	48	0.2	43	0.3	30	0.3	36	0.4	4	0.4

But in most cases the neighbors are different because the distance calculated by these methods vary which is the key factor in choosing the neighbors

INCOME DATASET:

Manhattan neighbors calculated for row - 1

Transaction.ID	X1st	X1st.dist	X2nd	X2nd.dist	X3rd	X3rd.dist	X4th	X4th.dist	X5th	X5th.dist
9364	26972	0.2264645	16934	0.3305458	12867	0.4410609	16996	0.4477863	30082	0.4916049

Euclidean neighbors calculated for row - 1

Transaction.ID X1st X1st.dist X2nd X2nd.dist X3rd X3rd.dist X4th X4th.dist X5th X5th.dist

9364 26972 0.05128618 16934 0.07959452 16996 0.1443108 7139 0.1596661 31329 0.1625165

Similar results are observed in the Income dataset too.

RESULT: In most cases the predicted 1st neighbor by both these methods happens to be the same.