Kennesaw State University
CS8267 – Machine Learning
Fall 2021

Assignment 1
Unsupervised Learning: K-Means Clustering

Purpose

Clustering is an unsupervised method for grouping patterns based on similarity such that the patterns in the same group are similar to one another, and those in different groups are less similar to one another. Clustering has many uses, including outlier detection and unsupervised image segmentation. The K-Means clustering algorithm is one of the most widely used clustering algorithms. In this assignment, you are going to implement basic k-means and analyze the effect of normalization.

Goals

- Learn how normalization is applied
- Learn how distance based clustering algorithms work
- Understand how to apply clustering
- Understand which data sets k-means clustering is appropriate for

Description

Implement basic K-means algorithm to cluster the sample data set. Two sample data sets will be provided. Make sure your program works with both data sets.

Input

The sample data sets to be used for this project are given below. The kmtest.arff (or kmtest.csv) file is very simple data set with two attributes that you can use to debug your code. For evaluating your program, you will use "Individual house hold electric power consumption Data Set" available at https://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption. You should only use relevant attributes for clustering. You may convert both datasets to csv file format and run your experiments. The ARFF files are formatted as text files with one sample / pattern vector per line of input. Each line will contain a series of attribute values, separated by commas or spaces. The files have an ARFF header consisting of a set of tags beginning with the '@' character. These tags provide

descriptions of the attributes (name and type). The data begins after the @data tag, which denotes the end of the header. Do not use *class* attributes in these files for clustering.

- kmtest.arff (kmtest.csv)
- household_power_consumption.txt

Outputs

For each input data set, your program should produce two outputs for each K.

A. Without Normalization

1. *Basic K-Means Cluster Centers:* A text file ("YourNetIdClusterCenterBasicKInputFile") containing each of the cluster centers (means), one cluster per line. Add one more attribute to your ARFF file as *@attribute cluster real* as the last attribute to represent cluster id if you are using ARFF format. Add a column "cluster" if you are using csv format. Each row of your output file will have values of a cluster center plus cluster id. Cluster ids must start from 1 and increase by 1. If K=5, there will be 5 cluster centers listed in the file. If there are 3 clusters, cluster ids may be 1, 2 or 3. *raygun*ClusterCenterBasic3*kmtest.arff* is a sample output file name.

2. *Basic K-Means Clusters:* A text file ("YourNetIdClusteringBasicKInputFile") containing cluster ids for each pattern vector in the source file (i.e., identifies the cluster that the pattern vector belongs to). Add one more attribute to your ARFF file as *@attribute cluster real* as the last attribute. The cluster attribute corresponds to the cluster ID of the cluster that the data point belongs to.

Note about filename notation: Underlined parts will depend on the user, input file, and parameters (e.g., K). Your file name may look like raygunClusteringBasic4kmtest.csv

B. With Normalization

If the data is to be normalized (if –normalize option provided), you should create clustering centers and clustering input for normalized data. Use min-max normalization as the normalization method where the range is set to [0..1]. First normalize the data and apply clustering on the normalized data. Your program should generate the following files:

a. YourNetIdClusterCenterNormalizedBasicKInputFile : (normalized) cluster centers using basic K-means
b. YourNetIdClusteringNormalizedBasicKInputFile : (normalized) clustering results for basic K-means

Then apply inverse normalization to cluster centers and generate the following files.

c. YourNetIdClusterCenterUnnormalizedBasicKInputFile: cluster centers in the original domain using basic K-means

Remap data values to the original values without normalization. You may use ID of data values to determine which data they were before normalization. Then generate the following files.

d. YourNetIdClusteringUnnormalizedBasicKInputFile: clustering results in the original domain for basic K-means

For normalization option 4 files are generated. Without normalization, 2 files are generated.

Analysis

You will compare how well k-means clustering performs compared to manual clustering for the power consumption dataset. Apply manual clustering as follows where the range of hours for each cluster is given in parentheses:
K=2 -> [0-6), [6-24)
K=3 -> [0-8), [8-16), [16-24)
K=4 - > [0-6), [6-12), [12-28), [18,24)
K=6 -> [0-4), [4-8), [8-12), [12,16), [16,20), [20,24)
K=12 -> [0-2), [2-4), … , [20,22), [22,24)
K=24 -> [0-1), [1-2), … , [22,23), [23,24)

For each K value, compare k-means and manual clustering based on the cluster evaluation measure.

Deliverables

You should submit the following to D2L:

1. Report:
    1. Kmtest: Provide visualizations of your clusters, cluster centers, and the assignment of clusters for unnormalized as well as normalized versions.
    2. Power consumption: Provide the analysis for the number of cluster mentioned above. Discuss your results and explain whether normalization has improved the performance or not.
    3. You should provide snapshots of your runs and outputs. Your outputs should include complete results for kmtest data. For power consumption data, provide cluster centers and partial results of clustering.
    4. Appendix:

    a. Appendix A: This will include your source code.
    b. Appendix B: A simple report that will include "README" on how to execute your code and sample runs of your program.
5. All source code required to build your program. Source code should be submitted in raw format. Do not convert your source code
6. The output files generated by running your program on the sample input files.
7. Submission by Email is not acceptable.
8. 50% penalty for one day late submission (except for medical and other emergency reasons). Submissions after one day late will not be accepted.
9. All submissions are due class time unless stated otherwise.

Notes

The following additional comments apply:

1. Good programming style must be observed. This includes using meaningful variable names, descriptive comments, and readable formatting
2. All your files (submitted and output files) must start with your login name + Cluster (i.e., YourNetIdCluster). The main files for each basic Kmeans should start with YourNetIdClusterBasic.
3. If you submit multiple files, you should either prepare a make file or a script file.
    - If you prepare a make file, you should name it as YourNetIdClusterMakefile.mk
    - If you prepare a script file, you should name it as YourNetIdClusterRun.sh. Make sure that your file only processes your files. For example, "gcc *.cpp" is not acceptable, since it is compiling all files.
    - Make sure that output executable file name is generated correctly (i.e., YourNetIdClusterBasic)
4. The program source code file must contain a header comment at the beginning of the file. This comment must include the following items:
    - Source code file name, e.g., "YourNetIdClusterBasic.cpp."
    - Student Name
    - Date
5. I will run your programs with gcc/c++/javac compiler under linux. So make sure that you do not use any additional libraries and your program is runnable under linux.
6. Your program will take inputs from command-line. "linux$ YourNetIdClusterBasic –i InputFile –K N –c classAttribute –normalize". (N is the number of clusters).For example,
    - raygunClusterBasic –i kmtest.arff –K 5 –c class –normalize
    - –c classAttribute is optional. If it is provided, the classAttribute should not be considered for clustering.
    - –normalize is optional in the command-line. If it is provided, the normalization is applied to all attributes except classAttribute (if

provided). If –normalize is not stated in the command-line, the normalization will not be applied.

- Running programs is based on the programing language you choose. For example, Java programs will be run with 'java' such as "java YourNetIdClusterBasic –i InputFile –K N –c classAttribute –normalize"

7. Students should work independently. Each student is responsible for handing in an original program.

Sample Outputs

---------- For cluster centers ----------

```
@relation ClusterCenter_kmtest

@attribute x real
@attribute y real
@attribute cluster real

@data
10 10 1
14.5 5 2
3.2 4.1 3
10.4 4.2 4
```

---------- For Clustering ----------

```
@relation Clustering_kmtest

@attribute x real
@attribute y real
@attribute cluster real

@data
2 4 3
3 3 3
3 4 3
3 5 3
4 3 3
4 5 3
9 4 4
9 5 4
9 9 1
9 10 1
10 4 4
10 5 4
10 9 1
10 10 1
11 10 1
```

```
15 4 2
15 5 2
15 6 2
16 4 2
16 5 2
16 6 2
```