Internship project: k-means algorithm for clustering

Hakim CHEKIROU

Dr S. BABA-ALI

LRIA, IADM Team

University of science and technology houari boumediene

October 29th 2017

Contents:

Chapter 1
Introduction

1.1 Why data mining?

We live in the data age, in fact petabytes of data are generated every day and stored in various computer systems from every aspect of human activity. The growth in the quantity of data generated is due to the computerization of society and the development of storage tools. This data is produced from various sources such as business, scientific and engineering practices, telecommunication networks, the medical and health industry, social media and the list goes on.

This exponential growth makes truly our time the data age. This explains the need for powerful tools to extract powerful knowledge from this huge amount of information, this situation has led to the birth of data mining.

Data mining can be viewed as the evolution of the information technology. Several important functionalities had been developed and led the way for advanced data analysis. Such functionalities include data collection and database creation, data management and advanced data analysis mechanism. The abundance of collected data in data repositories represents an incredible source of information and has caused the development of data mining tools to extract that knowledge.

1.2 What is data mining?

Data mining is the process of discovering patterns in large datasets involving methods at the intersection of different fields such as machine learning, statistics and data base systems. It's an interdisciplinary subfield of computer science. The overall goal of the data mining process is to extract information from a dataset and turn it into and understandable structure for later use. Another term for data mining is **knowledge discovery from data** or **KDD**, however some view data mining as part of the knowledge discovery process. The knowledge discovery process is an iterative sequence of the following steps (which are later discussed in the third chapter):

1 **Data cleaning**
2 **Data integration**
3 **Data selection**
4 **Data transformation**
5 **Data mining**
6 **Pattern evaluation**
7 **Knowledge presentation**

1.3 What kind of data can be mined?

Data mining can be applied to different forms of data as long as they are meaningful for a target application. The most basic data for the mining process are:

- **Database data**:

  A database system or database management system (DBMS), is a collection of interrelated data, known as a data base, and a set of software programs to manage the data.

  A **relational database** is a collection of tables which consists of a set of attributes and usually stores a large set of tuples.

- **Data warehouses**:
  A data warehouse is a repository of information collected from multiple sources, stored under a unified schema, and usually residing in a single site. They are constructed under a process of data cleaning, data integration, data transformation, data loading and periodic data refreshing.
- **Transactional data:**
  A transactional database stores transactions such as a purchase or a flight booking. Each transaction is made of a unique id number and a set of items making up the transaction.
- **Other kinds of data:**
  Many other types of data can be mined such as:
  - Time related or sequence data
  - Data streams
  - Spatial data
  - Engineering design data
  - Hypertext and multimedia data
  - Graph and networked data
  - Web data

  This kinds of data represent a challenge in the data mining process as to how to handle new structures and specific semantic.

1.4 What kinds of patterns can be mined?

Data mining functionalities are used to specify the kinds of patterns to be found in data mining tasks. In general such tasks can be classified as **descriptive** or **predictive**. Descriptive mining characterise the properties of the data in a target data set. Predictive mining perform induction on training data to make predictions. There are a number of data mining functionalities:

- **Class/Concept Description: Characterisation and Discrimination :**

Class/Concept Description is the process in which individual classes and concepts are described in summarized, concise and yet precise terms. These description can be derived using:

- Data characterisation is the summarisation of the general characteristics or feature of a target class.
- Data discrimination is the comparison of the general features of the target class data objects against the general features of objects from one or multiple contrasting classes.

- **Mining frequent patterns, Associations, and correlation:**
  Frequent patterns are patterns that occurs frequently on data including frequent items, frequent sub-sequences and frequent substructures.
- **Classification and regression for predictive Analysis**
  Classification is the process of finding a model that distinguishes data classes and is used to predict the class label of objects for which the class label is unknown. The model is derived based on that analysis of a set of training data. The derived model can be represented in various forms such as: *classification rules, decision trees, neural networks …*
  Regression is used to predict missing or unavailable numerical data values rather than class labels.
- **Cluster analysis:**
  Clustering analyses data objects without class labels, and constructs clusters of objects so that objects within a cluster have high similarity but are dissimilar to objects in other classes.

- **Outlier analysis:**
  Outliers are objects that don't comply with the general behaviour of the data. Outliers can be viewed as noise in many applications but can be very interesting in applications like fraud detection.

1.5 Which Technologies Are Used?

Data mining is an interdisciplinary domain and that characteristics has contributed to its success and its applications.

- **Statistics:**
  Statistics studies the collection, analysis, interpretation and presentation of the data.
- **Machine learning:**
  Machine learning investigates how computers can learn or improve their performance based on data. Here are some of the biggest problems in machine learning that are related to data mining: *supervised learning, unsupervised learning, semi-supervised learning and active learning.*
- **Data base systems and data warehouses**
- **Information retrieval:**
  Information retrieval is the science of searching information in documents that can be text or multimedia and may reside on the web.

1.6 Which kinds of applications are targeted?

It is impossible to enumerate every application were data mining plays a critical role but here are a few popular applications of data mining

- **Business intelligence:**
  Data mining is used in businesses to acquire a better understanding of their customers, the market, supply and resources and competitors so that they can make more effective decisions.

- **Web search engines:**
  Various data mining techniques are used in all aspects of search engines, ranging from crawling, indexing and searching.

1.7 Major Issues in Data Mining

Data mining is a vast expending field and face major issues that continue to stimulate further investigation and improvements. Some of the major issues are listed above.

- Mining methodology
- User interaction
- Efficiency and scalability
- Diversity of database types
- Data mining and society

Chapter 2
Getting to Know Your Data

Before jumping directly to the phase of extracting data we have to take time to fully understand the data in its different forms. This also means having a look at statistical description of data, how to visualise it and how to measure data similarity and dissimilarity.

2.1 Data Objects and Attributes Types

Data sets are made up of data objects which represents an entity that are usually described by attributes.

Attributes:

An attribute is a data field representing a characteristic or feature of a data object. The type of an attribute is determined by the set of possible values. In the following sub section, we introduce each type.

Nominal attributes:

The values of a nominal attribute are symbols or names which represent a category, code or state. Although the values are names they can be represented with numbers. However mathematical operations on values of nominal attributes are not meaningful.

Binary attributes

A binary attributes is an attributes with only two states 1 and 0, where 1 means that the attribute is present or true and 0 means it's absent or false. A binary attribute is called **symmetric** if both of its states are equally valuable and is called **asymmetric** otherwise.

Ordinal attributes

An ordinal attribute is an attribute with possible values that have a meaningful order but the magnitude between successive values is not known. An example of ordinal attributes are drink_size in a restaurant: small, medium and large.

Numeric attributes

A numeric attribute is quantitative represented in integer or real values, they are split into two categories:

- **Interval scaled attributes**
- **Ratio-scaled attribute**

Discrete versus continuous attributes

A **discrete** attribute has finite or countably infinite set of values. An attribute is countably infinite if the set of possible values is infinite but can be put in a one-on-one correspondence with natural numbers.

Continuous attributes are represented as floating-points variables.

2.2 Basic Statistical Descriptions of Data

For data pre-processing to be efficient, it is crucial to have an overall picture of your data. Many statistical description can be used for that purpose.

Measuring the central tendency

The most effective numeric measure of the “center” are listed above:

- The mean
- The median
- The mode
- The midrange

Measuring the dispersion of data

- Range, Quartiles and interquartile range:
- Five number summary, boxplots and outliers:
  The five number summary of a distribution consists of the median Q_2, the quartiles Q_1 and Q_3 and the smallest and biggest observations. Boxplots incorporate the five number summary to visualise the distribution.

- Variance and standard deviation:
  Then variance of N observations, $x_1, x_2, \dots, x_n$, for a numeric attribute X is

$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x - \bar{x})^2$$

  The standard deviation is the root of the variance

Graphic display of basic statistical description of data

- Quantile plot
- Quantile quantile plot
- Histograms
- Scatter plots

2.3 Data Visualisation

Data visualisation aims to communicate data clearly and effectively through graphical representation to discover data relationships that are otherwise not easily observable. There are many representation approaches for different kinds of data, including:

- Pixel oriented visualisation techniques
- Geometric projection visualisation techniques
- Icon-based visualisation techniques: Chernoff faces and stick figure.
- Hierarchical visualisation techniques: worlds-within-worlds and tree maps.

- Visualizing complex data and relations: tag clouds and graphs.

2.4 measuring data similarity and dissimilarity

In data mining application, such as clustering, outlier analysis and nearest-neighbour classification, we need to access how alike or unalike objects are in comparison to one another. Here we look at similarity and dissimilarity measures.

- The ration of mismatches for nominal attributes.
- The Jaccard coefficient for asymmetric binary attributes.
- The Euclidean, Manhattan, Minkowski and supremum distances for numeric attributes.

Dissimilarity for attributes of mixed types

Suppose that the data set contains *p* attributes of mixed types. The dissimilarity between objects i and j is defined as

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} \, d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}$$

Where $\delta_{ij}^{(f)} = 0$ if either (1) one of the values for the attribute is missing, or (2) both equal 0 and attribute f is asymmetric binary.

For sparse numeric data vectors, such as term-frequency vectors two measures of similarity are used:

a) Cosine similarity:
Let ***x*** and ***y*** be two for comparison. Using the cosine similarity measure, we have

$$sim(x,y) = \frac{x.y}{||x|| \, ||y||}$$

b) Tanimoto coefficient:
When attributes are binary-valued, using the tanimoto coefficient we have

$$sim(x,y) = \frac{x.y}{x.x + y.y - x.y}$$

Chapter 3
Data Preprocessing

Real world databases are noisy, full of missing and inconsistent data due to their huge size and their heterogeneous sources. To prepare the extracting phase our data needs to be preprocessed which means needing to apply different procedures like cleaning, integrating and reducing to obtain high quality data.

3.1 Data Preprocessing: An overview

Data is considered being high quality data if it satisfies the following requirements:

- Accuracy
- Completeness
- Consistency
- Timeliness
- Believability
- Interpretability

3.2 Data Cleaning

Real world data tend to be incomplete, noisy and inconsistent. Data cleaning routines attempt to fill in missing values, smooth out noise while identifying outliers and correct inconsistencies in the data.

a) Missing values

Data isn't always available, we look at the following methods for filling the missing values.

- Ignore the tuple: if its importance is negligible.

- Fill in the missing value manually: this method is not effective for a huge amount of data.
- Use a global constant to fill in the value: like “unknown”, although this method is simple it’s not foolproof.
- Use a measure of central tendency: the mean for symmetric data distributions and the median for the others.
- Use the attribute mean or median for all samples belonging to same class as the given tuple.
- Use the most probable value to fill in the missing value: determined with regression, inference based tools or decision tree induction.

b) Noisy data

Noise is a random error or variance in a measured variable such as outliers. Here we look at the following data smoothing techniques.

- Binning:
  Binning methods smooth a sorted data distribution, which is distributed in bins and within each bin every value is replaced by the mean or median of the bin or by the closest boundary value.
- Regression:
  Regression consists of finding the “best” line to fit two attributes so that one attribute can be used to predict the other.
- Outlier analysis: using clustering to detect outliers.

3.3 Data integration

Data integration is the merging of data from various data sources, if done carefully, it can reduce redundancies and inconsistencies. There are a number of issues to consider during this process such as schema integration and object matching and special attention must be paid to the structure of the data.

Redundancy is another important issue that can be detected using **correlation analysis** to measure how strongly an attribute implies another. For nominal data, we use the χ^2 test. For numeric attributes we can use the correlation coefficient and the covariance.

We also have to detect tuple duplication caused by inaccurate data entry or updating some but not all data occurrences. It may occur that for the same real world entity, attribute value from different sources may differ because of differences in representation, scaling and encoding.

3.4 Data reduction

Most of the time, the data mining tasks are run on huge data sets. Data reduction techniques can be applied to obtain a smaller data set while maintaining the integrity of the original data. Here, we look at data reduction strategies.

a) Dimensionality reduction:
   Dimensionality reduction is the process of reducing the number of random variables or attributes under consideration. Dimensionality reduction methods include:
   - Wavelet transform which is a linear signal processing technique
   - Principal Components Analysis (PCA)
   - Attribute subset selection
     Attribute subset selection reduces the data set size by removing irrelevant or redundant attributes.

b) Numerosity reduction:
   Numerosity reduction techniques replace the original data volume by alternative, smaller forms of data representation. We look at the following methods of numerosity reduction
   - Regression and log linear models: parametric data reduction
   - Histograms

- Clustering
- Sampling
- Data cube aggregation

c) Data compression:
In this technique, transformation are applied on the original data set to obtain a reduced data set. If the original data set can be reconstructed without any information loss, data reduction is called lossless, otherwise we call it lossy.

3.5 Data transformation and data discretisation

This section presents methods of discretisation. In this preprocessing step the data are transformed or consolidated so that the resulting mining process may be more efficient, and the patterns found may be easier to understand. Data discretisation, a form of data transformation, is also discussed.

Strategies for data transformation include the following:

1. **Smoothing:** where noise are removed using binning, regression and clustering.
2. **Attribute construction:** where new attributes are constructed to help the mining process.
3. **Aggregation:** where summary or aggregation operations are applied to the data.
4. **Normalisation:** where the attribute data are scaled so as to fall within a smaller range.
5. **Discretisation:** where the raw values of a numeric attribute are replaced by interval labels or conceptual labels.
6. **Concept hierarchy generation for nominal data:** where attributes such as street can be generalised to higher-level concepts, like city or country.

Chapter 4
Clustering

Clustering or cluster analysis is the process of partitioning a set of data objects into subsets. Each object is a cluster such as each object in the same cluster are similar to one another and dissimilar to objects in another cluster. Different clustering methods may generate different results. Clustering is useful for discovering previously unknown groups within the data.

4.1 k-means algorithm

Suppose a data set D, contains n objects in Euclidean space. The k-means algorithm distributes the objects in D in k clusters, $C_1, \dots, C_k$. Objects within a cluster are similar to one another but dissimilar to objects in other clusters.

Algorithm: k-means. The k-means algorithm for partitioning, where each cluster's center is represented by the means value of the objects in the cluster.

Input:

- K: the number of clusters,
- D: the data set containing n objects.

Output: A set of k clusters.

Method:

(1) Arbitrarily choose k-objects from D as the initial cluster centers;
(2) **Repeat**
(3) (re) assign each object to the nearest cluster;
(4) Update the cluster means, update the cluster's centroid;
(5) **Until** no change

Chapter 5
Conception

5.1 introduction

To conceive this project we will focus on 3 essential parts:

- Pre-processing
- K-means algorithm
- Improvement to the k-means algorithm

5.2 Pre-Processing

To prepare our data for the clustering, we applied these three pre-processing

- **Replacement of the missing values** by the mean for the numeric attributes and the mode for the nominal ones.
- **Discretisation:** where the raw values of a numeric attribute are replaced by interval labels or conceptual labels.
- **Normalisation:** where the attribute data are scaled so as to fall within a smaller range.

$$normalised\_data = \frac{data - \min(data)}{\max(data) - \min(data)}$$

5.3 Evaluation function

For this project we used two quality measures, the silhouette coefficient the inner cluster variance.

The silhouette coefficient:

It is defined as for the i^{th} object as follows:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

Where a_i is the average distance between i^{th} object and the other objects in its cluster, and b_i is the minimum average distance to the objects in the other cluster.

The value of the silhouette coefficient varies from -1 to 1. A negative value is undesirable. We want the silhouette coefficient to be positive and close to 1.

The sum of squared error E:

$$E = \sum_{i=1}^{k} \sum_{p \in C\_i} dist(p_i, C_i)^2$$

Where p is a given object; and c_i the centroid of the cluster C_i;

5.4 attempt at improving k-means

Knowing that the majority of the data sets are of a compact nature, one of the issues with k-means is how to deal with objects that are border line, that is, objects that have a positive silhouette coefficient but are close to 0, which means they are on the border between two clusters.

To deal with this problem, we proposed a method to rearrange this objects.

In this method, we will take the objects that are border line and those who are incorrectly clustered apart from their original clusters, and recalculate the centroids of the original clusters. In the next step we will apply the k-means algorithm on those who are border line with k/2 as the number of clusters. As for the objects that are incorrectly clustered, since they represent a small part of the data set, they will be put in the nearest cluster and the centroids of the clusters will be recalculated.

Our goal is: starting with the resulting clusters of a first run of k-means on a data set, we shall have a better clustering by reconsidering the object that do not have a good silhouette coefficient.

The algorithm:

Algorithm: attempt at improving k-means.

Input:

- K: the number of clusters,
- D: the data set containing n objects.

Output: A set of 3k/2 clusters.

Method:

(1) apply k-means on the data set D;
(2) **for** i =1 to n
(3) calculate the silhouette coefficient S_i for the i^{th} object;
(4) **If** -0.5<S_i <0.1 add the object I to the data set P and remove it from D;
(5) **If** S_i <= -0.5 add the object I to the data set N and remove it from D;
(6) Apply k-means on the data set P with k/2 clusters;
(7) Put each object in N in the nearest resulting clusters;

5.5 Conclusion

In this chapter, we managed to present the method used in our pre-processing, the measures of the clustering quality and our attempt at improving k-means.

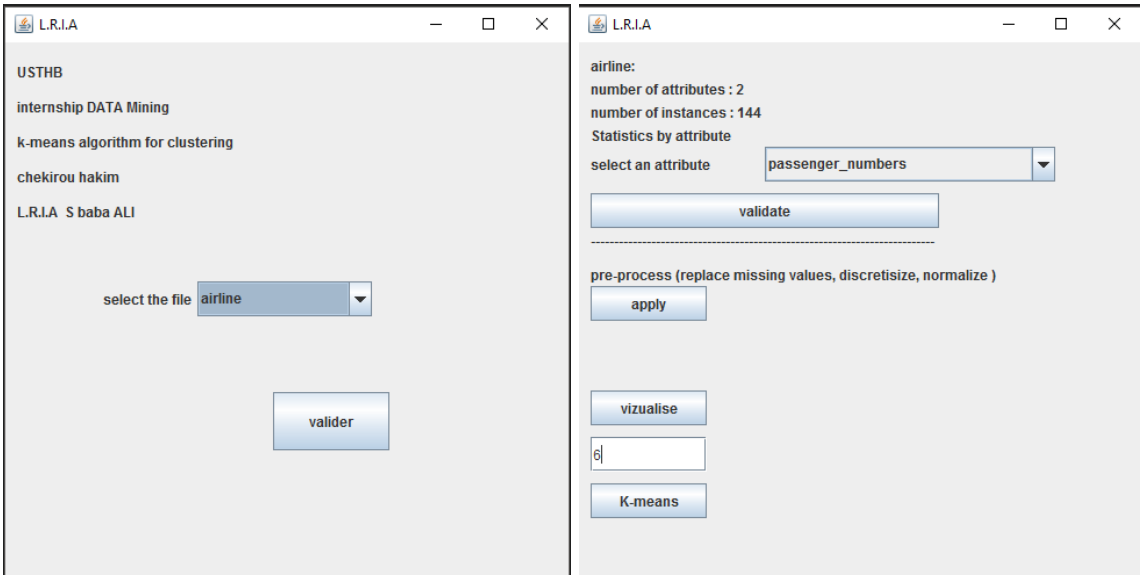
Chapter 6
Results

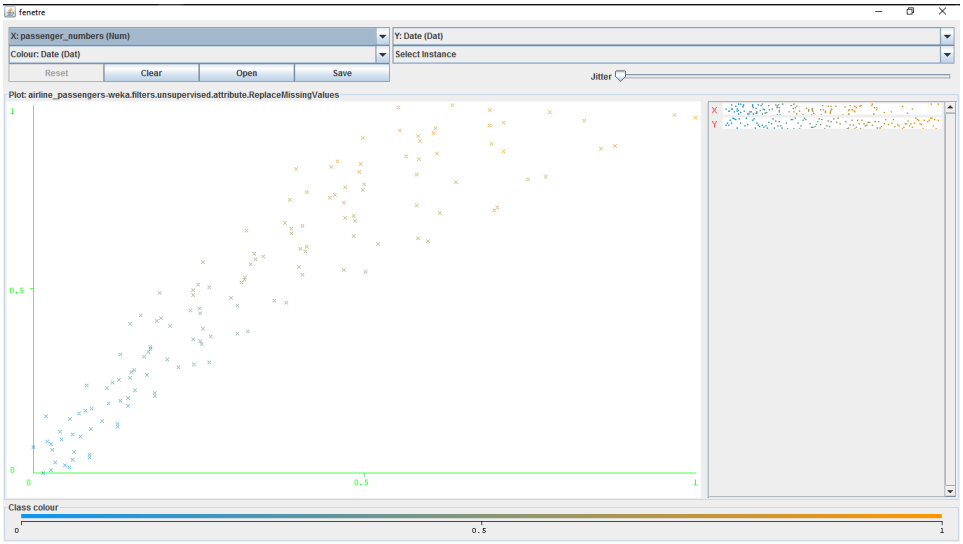
6.1 Introduction

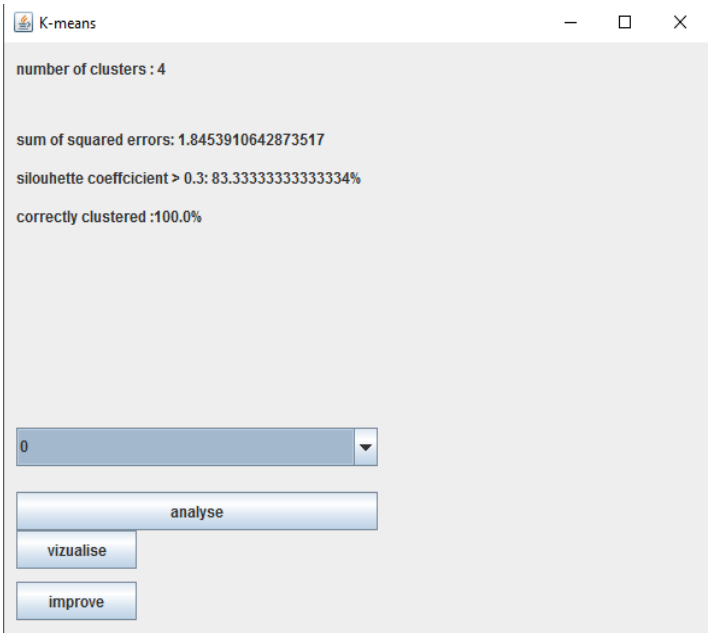
To develop and implement this project, we used <<java-eclipse>>. Eclipse is a free, polyvalent and open source IDE.

6.2 tests and results

Before presenting our tests and results here is a preview of our graphical interface.







Tests:

In this section we present the results of the application of the k-means algorithm and our own attempt at improving it.

We ran the algorithm on different data sets, the k was chosen to be optimal on each data set by running k-means many times and selecting the best one.

Results:

Here is a table that sums up our results.

					k-means			notre methode			
data set	variance	nb d'insta	nb attributes	k	correctly clustered %	silhouette >0,3	E	new k	correctly clustered	silhouette >0,3	E
autos	1,45	205	26	8	97%	40,48%	121,9	12	90,24%	44,30%	117,1
breast-cancer	1,2	286	10	10	100%	42%	129,5	15	96%	40,55%	123,7
cpu	0,16	209	7	7	92,80%	74,10%	8,79	10	93,30%	74,64%	8,57
diabetes	0,43	768	9	2	100,00%	99,00%	149,5	3	99,86%	97,26%	147,9
glass	0,39	214	10	7	97,10%	55,60%	23,23	10	85,50%	61,60%	20,12
kdd_train	2,46	11419	42	8	99,60%	94,30%	2991,2	12	98,10%	93,80%	2499
labor	1,6	57	17	7	100,00%	31,50%	41,2	10	92,98%	47,36%	35,94
lymph	2,29	148	19	10	95,27%	16,20%	170,8	15	84,40%	22,97%	163,3
soybean	3,9	683	36	6	97,00%	11,56%	1629	9	93,11%	16,98%	1568
unbalanced	1,2	856	33	10	97,30%	47,07%	423,1	15	78,70%	51,05%	441,08
vote	3,9	435	17	3	93,70%	52,60%	923	4	78,60%	53,33%	936
vowel	1,03	990	14	10	96,10%	55,75%	252,8	15	85,05%	37,20%	248,5
weather	0,84	14	5	4	100,00%	64,20%	6,4	6	100%	85,70%	3,1
zoo	3,12	101	18	7	100%	63,36%	102	10	94%	69,30%	93,3

6.3 Discussion

From our results, we can see that:

- The vast majority of the objects are correctly clustered during the first run of k-means.
- The percentage of objects that are not border line is highly dependent on the data set, as we can see we have data sets like kdd_train that have 99.6% correctly clustered objects and 94.3% who are not border line objects and others like diabetes 99% and 65%.
- We can observe a decrease of the percentage of correctly clustered objects, but is restrained to less than 15%
- We can also observe a general decrease in the sum of squared errors E which implies that the data object within the clusters are more similar.
- A general increase of the percentage of objects who are bigger than 0.3% which means that the objects are better clustered.

6.3 Conclusion

In this chapter we presented the results of the various tests. The results were mixed, some were positive, others can be subject to improvement.

Chapter 7
General Conclusion

Our attempt at improving the k-means algorithm was proven positive in ways that the data was more similar within the clusters and the number of objects clustered in a better way is bigger, but there was still a decrease in the number of correctly clustered objects.

To improve this algorithm, one may optimize the parameters of the method, specially the k for the clustering of the border line objects.