# DATA CLUSTERING: APPLICATIONS IN ENGINEERING

**Zdravko Krpić**
Faculty of Electrical Engineering,
University of Osijek
Kneza Trpimira 2B, HR-31000 Osijek
E-mail: zdravko.krpic@etfos.hr

**Goran Martinović**
Faculty of Electrical Engineering,
University of Osijek
Kneza Trpimira 2B, HR-31000 Osijek
Phone: ++385 31 224766 ; E-mail: goran.martinovic@etfos.hr

**Ivan Vazler**
Department of Mathematics, University of Osijek
Gajev trg 6, HR-31000 Osijek
Phone: ++385 95 8204838; E-mail: ivazler@mathos.hr

**Abstract**
Dividing a set $S = \{x_i = (x_1^{(i)}, \ldots, x_n^{(i)})^T \in \mathbf{R}^n : i = 1, \ldots, m\}$ (a set of vectors from a vector space $\mathbf{R}^n$ ) into disjunct subsets $\pi_1, \ldots, \pi_k, 1 \le k \le m$ , such that

$$\bigcup_{i=1}^{k} \pi_i = S,$$
$$\pi_i \bigcap \pi_j = 0,\ i \ne j,$$
$$|\pi_j| \ge 1,\ j = 1, \ldots, k,$$

determines a partition of the set $S$ . The elements of such partition $\pi_1, \ldots, \pi_k$ are called *clusters*.

For practical clustering applications the number of all clusters is too big and the problem of determining the optimal partition in the least-squares sense is an NP-hard problem.

In this paper we will consider some well-known algorithms for searching for an optimal LS-partition, list some of the numerous applications of cluster analysis in engineering and give some practical applications.

**Key words:** *data clustering, engineering, least squares*

## 1. INTRODUCTION

In short, clustering problems are problems of identifying groups of individuals or objects that are similar to each other but different from those in other groups. Many web portals and internet businesses track consumer habits and take advantage of these similarities to target specific offers to subgroups that are most likely to be receptive to them. Many search engines cluster their databases so they can offer similar results (like bookstores suggesting other books by the same author, or books with similar topics, or books from the same publisher, and so on).

Dividing a set $S = \{x_i = (x_1^{(i)}, \ldots, x_n^{(i)})^T \in \mathbf{R}^n : i = 1, \ldots, m\}$ (a set of vectors from a vector space $\mathbf{R}^n$) into disjunct subsets $\pi_1, \ldots, \pi_k, 1 \le k \le m$, such that

$$\begin{aligned} &\bigcup_{i=1}^{k} \pi_i = S, \\ &\pi_i \bigcap \pi_j = 0, \; i \ne j, \\ &\left|\pi_j\right| \ge 1, \; j = 1, \ldots, k, \end{aligned}$$

determines a partition of the set $S$, which will be denoted by $\Pi(S) = \{\pi_1, \ldots, \pi_k\}$. The elements of such partition $\pi_1, \ldots, \pi_k$ are called *clusters*. The set of all partitions of the set $S$ containing $k$ clusters which satisfy the properties above will be denoted by $\Psi(S,k)$.

The number of all $k$-partitions is

$$|\Psi(S,k)| = \frac{1}{k!} \sum_{j=1}^{k} (-1)^{k-j} \binom{k}{j} j^m$$

and the goal of clustering is to find the optimal partition in some sense.

For practical clustering applications that number is too big. The problem of clustering can be divided in several subproblems. Objects being clustered need to be represented in a way that the clustering algorithms can easily measure their similarity or dissimilarity (or distance). Determining the goal function and an algorithm for clustering (which in most cases finds only the approximation of the optimal partition) is another problem. Depending on the algorithm, the problem of determining the number of clusters can also arise.

Clustering algorithms can be classified in several categories:

- Hierarchical clustering - Based on a tree model of data, it can be either agglomerative or divisive. Agglomerative clustering begins with each object in its own cluster and in each step clusters are joined based on similarity. Its bad side is that once two objects are in the same cluster, they stay in it till the algorithm ends. Divisive clustering works similarly in the opposite direction.

- Partitional clustering - These are methods that iteratively improve the partitioning by moving elements from one cluster to another, usually starting from a random partition. K-means and k-medoids are the most known such algorithms.
- Neural network-based clustering
- High dimensional and large-scale data clustering - These are methods based on reducing the dimensionality of the problem. They include random sampling methods, density-based methods and grid-based methods.

If we define a criteria function on the set $\Psi(S,k)$ of all partitions of the set $S$ containing $k$ clusters by

$$f : P(S) \to [0,+\infty), \quad f(\pi_j) = \sum_{x_i \in \pi_j} \|x_i - c_j\|^2$$

$$F : \Psi(S,k) \to [0,+\infty), \; F(\Pi) = \sum_{j=1}^{k} f(\pi_j), \; \Pi = \{\pi_1, \ldots, \pi_k\},$$

then we can define a partition $\Pi^{(o)}$ which is optimal in the least-squares sense, i.e.

$$F(\Pi^{(o)}) = \min_{\Pi \in \Psi(S,k)} F(\Pi).$$

The problem of determining the optimal partition in the least-squares sense is an NP-hard problem.

The k-means algorithm can be used for searching the optimal partition in the LS sense.

**Algorithm 1: K-Means**

**Input:** Arbitrary $k$-partition $\Pi^{(0)}$ of the set $S = \{x_i = (x_1^{(i)}, \ldots, x_n^{(i)})^T \in \mathbf{R}^n : i = 1, \ldots, m\}$

**Output:** Locally LS-optimal $k$-partition $\Pi^{(l)}$

```
$it \leftarrow 0$
calculate cluster centres $c^{(it)} = (c_1^{(it)}, \ldots, c_k^{(it)}),\ c_i^{(it)} = \frac{1}{|\pi_i^{(it)}|} \sum_{x_j \in \pi_i^{(it)}} x_j$
repeat
    $it \leftarrow it + 1$
    $\Pi^{(it)} = \{\pi_1^{(it)}, \ldots, \pi_k^{(it)}\}$ such that $\pi_i^{(it)} = \left\{ x_j \in S : \arg\min_p (\|x_j - c_p^{(it)}\|) = i \right\}$
    calculate new cluster centres $c^{(it)} = (c_1^{(it)}, \ldots, c_k^{(it)}),\ c_i^{(it)} = \frac{1}{|\pi_i^{(it)}|} \sum_{x_j \in \pi_i^{(it)}} x_j$
until $c^{(it)} = c^{(it-1)}$
return $\Pi^{(it)}$
```

It is easily shown that the k-means algorithm monotonously reduces the criteria function $F$. The algorithm often stops before reaching the optimal partition in the LS sense. The partition on which the algorithm stops depends on the choice of the initial partition, and since the algorithm is usually very quick, it is very common to run it multiple times with different starting partitions to increase the chance of obtaining a better resulting partition.

## 2. APPLICATIONS

### 2.1. Text clustering

There are many uses of clustering in text analysis. Clustering can be used to derive keywords, group articles with similar topics (initially unknown), find possible synonyms in monolingual dictionaries and in many other areas. To cluster textual data one must first transform the textual data to a format that can be used in clustering algorithms. One way to represent articles would be to use their references and represent their connections with a graph. This representation is suitable for clustering with hierarchical clustering algorithms. Another way to pre-process textual data would be to represent it in the form of vectors (which can be done in many ways depending of our goal). The method of representation used here is the vector-based model described in Berry (2004) and Srivastava and Sahami (2009).

The most common way of creating a vector space model can be divided in two stages. The first stage is the extraction of content bearing terms (words or short phrases) and setting their weight proportional to the count of the corresponding term in the document. The second stage is to modify the weights so that the important terms get more emphasis. The set of $m$ documents would be represented by a set of vectors $S = \{x_i = (x_1^{(i)},\ldots,x_n^{(i)})^T \in \mathbf{R}^n : i = 1,\ldots,m\}$ where dimension $n$ of the vectors is equal to the number of terms in the whole document collection.

The first task in stage one is to determine all the terms. Some terms in a document don't describe any important content (e.g. pronouns,...). Other terms may appear in all (or most) documents or only several documents. These words are usually filtered from the documents and do not appear in the vector representation. In many languages some terms can be condensed to one due to conjugation or declension. In the first stage we create vectors $f_i, i = 1,\ldots,m$ of frequencies of terms. The value of $f_j^{(i)}$ is set to the frequency of the $j^{th}$ term in the $i^{th}$ document. Note that the vectors are very sparse, since many terms appear only in several documents.

In the second stage, the term frequencies are multiplied by the inverse document frequency of a term in the document collection. If we denote by $W = \mathrm{diag}(\ln(m / w_1),\ldots,\ln(m / w_n))$ a diagonal matrix where $w_j$ is the total number of documents containing the $j^{th}$ term, then the vector representation of document $i$ is

$$x'_i = W f_i, \quad i = 1,\ldots,m.$$

This is done so that terms occurring in almost all documents don't influence the clustering results as much.

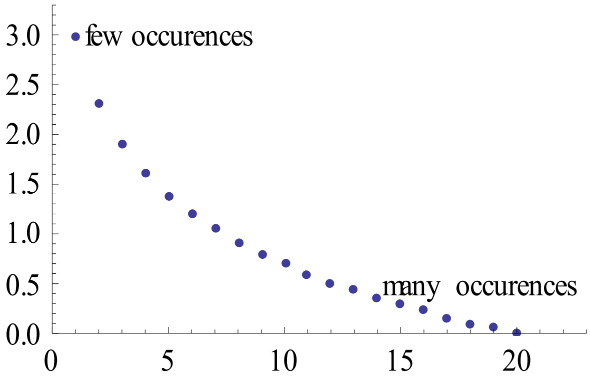

*Figure 1: Example of inverse document frequency weights*

The last thing left to do is to normalize the vectors so that we observe the relative frequency of terms. Normalization can be done in different norms $(1,2,\infty)$ resulting in data on a corresponding $n$-dimensional sphere. After the preprocessing step we have

$$x_i = \frac{x'_i}{\|x'_i\|}, \quad i = 1,\ldots,m.$$

Although the k-means algorithm is not very good for clustering high dimensional data the clustering of normalized vectors using the k-means should be done with an extra step so that the centres of clusters also lie on a unit sphere. If the Euclidian norm is used for normalization, a simple normalization of the centroids yields the centres of clusters constrained on a unit sphere. This modified algorithm is often called *spherical k-means* algorithm and is obtained by adding the normalization step

$$c^{(it)} \leftarrow c^{(it)} / \|c^{(it)}\|$$

after steps 2 and 6 in **Algorithm 1**.

**Example 1.**

We tried out this method of grouping of 101 short news articles in Croatian found on the internet. There were 1290 different terms. We did not combine terms based on conjugation and declension and therefore the results are not as good as they could be. Also, many of the articles are too short to have enough grouping terms. Despite those shortcomings, the results are satisfactory. These are the most frequent terms by clusters:

1) utakmica, ugovor, Zagreb, Maksimir, postigao, gol, prvak, navijači,...
2) kazna, sjedala, prodaja, korisnik, pad, automobil, vozila,...
3) pokrenuta, odvjetnik, bivši, uzeo, zbrisao, banke, mito,...
4) kapital, rebalans, povećati, cigareta, banke, pdv, porez, proračun,...

...

Terms in those clusters are characteristic terms for football, automobiles, banking and politics.

## 2.2. Image analysis

Cameras and other imaging equipment are cheap, available and used in many areas. With that, the need for an automatic image analysis has arisen. Often the goal is to distinguish similar or dissimilar areas of an image. In medical sciences clustering can be applied to various body scans for tumor diagnostics. On satellite images, clustering can be used to discern urban areas, fields, forests... Clustering can also be used to find differently textured areas of an image. To do any of these things, images must often be pre-processed to show the distinguishing features.

The most common image attribute used for image clustering is *colour*. Other simple image characteristics applicable for clustering parameters are *hue* and *saturation*. More complex image properties include pixel distances and patterns. Some pattern recognition applications also require large image databases against which candidate images are compared.

**Example 2.**

In this example k-means clustering method is used on a 256 colour greyscale image, as proposed in Saha and Bandyopadhyay (2008). The image is 256 pixels in width and height. Different areas of interest are extracted from it based on the pixel colour value. This application is common in satellite image analysis, but it has some other uses, such as those described in Tatiraju and Mehta (2008). The number of clusters represent granularity of segments needed for the extraction from the image.

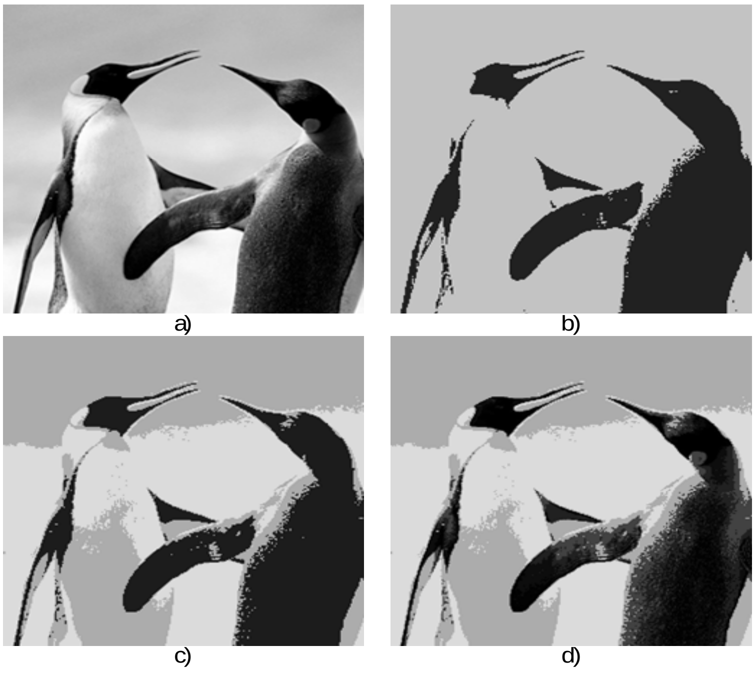

*Figure 2: Clustering of a greyscale image based on pixel colour value: a) Original image, b) k=2, c) k=3, d) k=8.*

Figures 2 and 3 show two images segmented into different numbers of clusters. As the number of cluster increases, more detailed image analysis is done, but increasing the number of clusters beyond a certain threshold can make clustering meaningless.

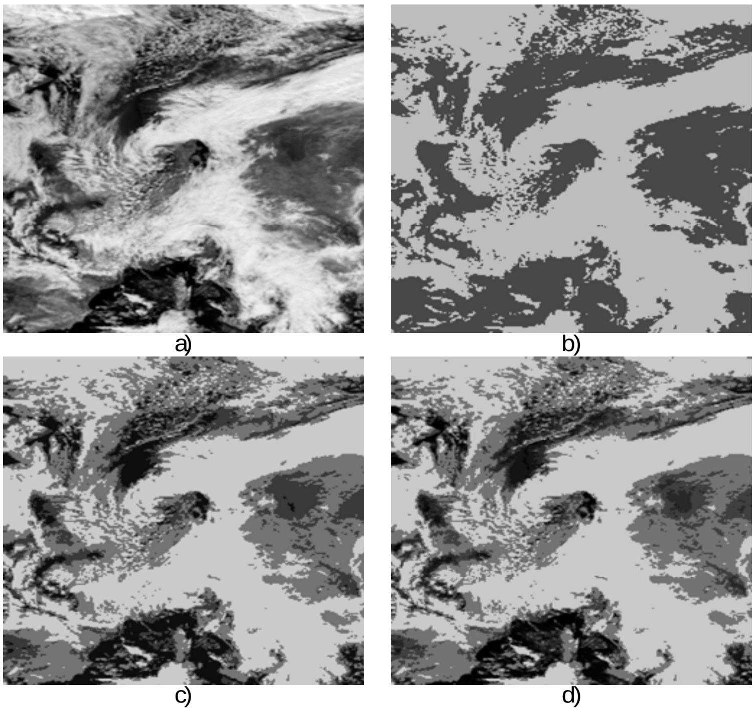

*Figure 3: k-means clustered satellite image: a). Original image, b) Separation of heavy clouds with k=2, c) Separation of light and heavy clouds with k=4, d) With 8 clusters, there is no gain in enhancing cloud separation.*

In order to discover important areas of a greyscale image we need to know how many clusters to look for. There are many criteria for determining that number, and many of them require clustering of the data for each $k$ . One of the easier ways is to use histogram analysis of the image. The procedure is as follows:

- Let $G$ denote the set of all grey levels. For every shade of grey $i \in G$ we find its frequency $f_i$ , the number of pixels with that colour.
- We detect the set of local maximums in the image histogram

$$S_1 = \{(i, f_i) | (f_{i-1} < f_i) \& (f_i > f_{i+1})\}$$

- We remove the local maximums with frequencies below some empirical threshold (for example $f_{thr} = \max(f_i)/100$ ).

$$S_2 = \{(i, f_i) \in S_1 | f_i > f_{thr}\}$$

- From $S_2$ we remove the elements having close peaks (their difference in grey levels is below some empirically determined threshold $t$).

$$S_3 = S_2 - \{(i, f_i) \in S_2 \mid \exists (j, f_j) \in S_2 \text{ such that } (i > j) \,\&\, (i - j \le t) \,\&\, (f_i \le f_j)\}$$

- The number $k = |S_3|$ is the number of clusters to look for, and the elements remaining in $S_3$ are good initial centres for clustering.

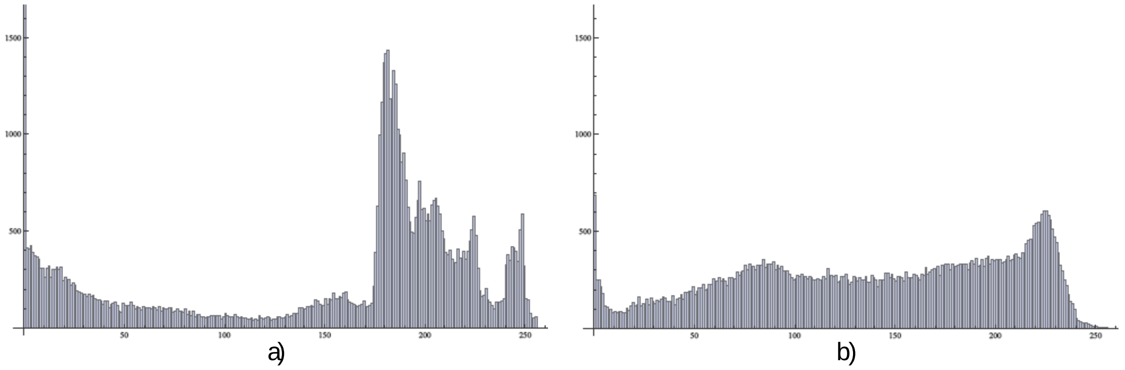

*Figure 4: Grey level histograms of images from a) Figure 2. b) Figure 3.*

In Figure 4 we can see that the first image has fluctuating grey level frequencies so we would use the frequency threshold $f_{thr}$ to remove low level peaks. The grey level frequencies of the second image are more uniform so we would use the threshold $t$ to reduce the number of peaks.

## 2.3. Application in computer science

A possible application of clustering in computer science can be in application mapping in heterogeneous environments. Advances of this research can be found in Siegel (2009). The heterogeneous computing environment comprises of different computers under different loads. The goal is to find optimal groups of computers (computer clusters) which are capable of performing various application tasks. These systems can be found in various computer cluster installations, computer grids and cloud computing systems. Finding the optimal computer(s) for solving an application-given problem can often be NP hard, as there are many parameters which describe each computer. Another difficulty arises due to the different nature of these parameters (processor speed in MHz, RAM capacity in megabytes, network throughput in megabits per second, etc.). This implies that normalization of parameters has to be done first. After that, since the mapping system uses different preferences on different parameters for every application task, analysis has to be performed which evaluates parameter impact on candidate suitability. This is done by using weights, and multiplying normalized parameter values with them. The metric used for measuring the distance and for calculating the similarity matrix depends on a number of parameters for each mapping candidate. In Table 1, ten parameters which describe mapping candidates are presented. The range of values (Minimum value and Maximum value) used during calculation is also given for each candidate.

After normalization, static and dynamic data are combined together to form current computer mapping candidate (MC) state.

*Table 1: Candidate parameters*

| | Parameter | Measuring unit | Minimum value | Maximum value |
|---|---|---|---|---|
| Static | *Processor speed* | MHz | 1000 | 4800 |
| | *memory capacity* | MB | 128 | 8192 |
| | *hard disk capacity* | GB | 10 | 1000 |
| | *network throughput* | Mbit/s | 1 | 1000 |
| | *Operating system* | 1- 3 | 1 | 3 |
| Dynamic | *Processor load* | % | 0 | 100 |
| | *Memory load* | % | 0 | 100 |
| | *Disk space usage* | % | 0 | 100 |
| | *Network traffic* | % | 0 | 100 |

## Example 3.

For the purpose of visualization simplicity, greatest weights were given to the first two parameters (available CPU speed and available RAM memory), meaning that these are mostly required by the application task. Running the mapping in this environment, with k-means clustering ( $k = 4$ ) gives selections shown on Figure 5.

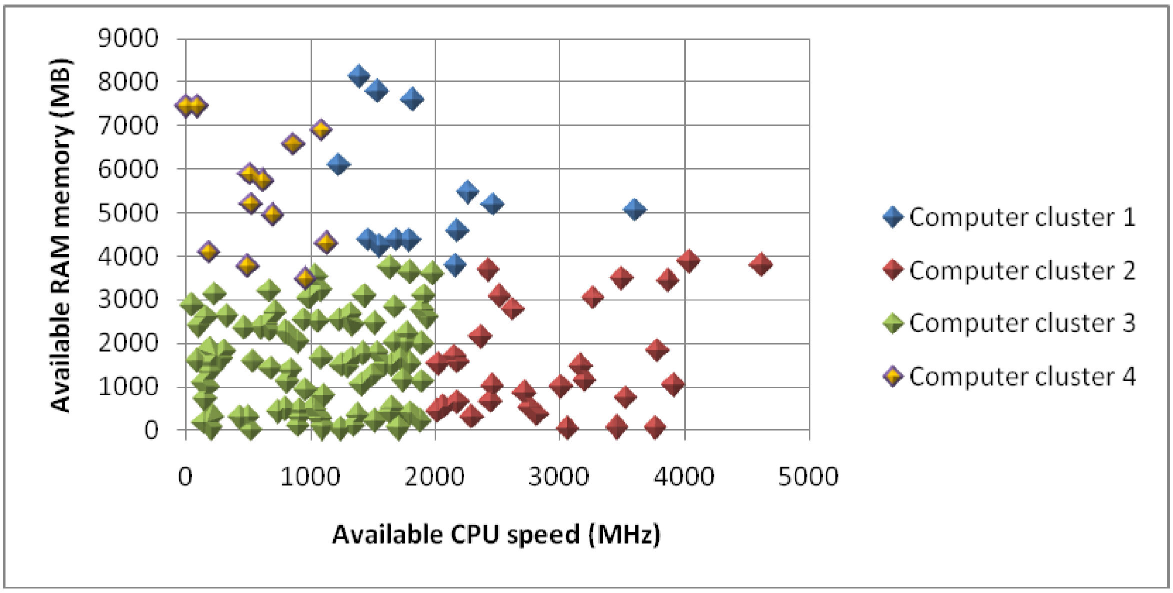

*Figure 5: Results of a mapping system, using k-means clustering method*

It is obvious that only computer cluster 1 has the benefits from both parameters, forming the most powerful computer cluster. Computer cluster 2 comprises computers with great processing power, but they lack RAM memory. However, this cluster has its uses. Many applications depend almost exclusively on processor speed. The third computer cluster holds most of the computers, which have lower performance. This cluster contains unwanted candidates, which are either heavily loaded already, or their hardware is insufficient. Last computer cluster holds computers with large amount of RAM memory. These are the appropriate candidates for application tasks which are hungry memory-wise.

In conclusion, there are many different applications of clustering. They differ in data representation, goal functions or method of clustering and that is the reason behind the increasing number of articles in this field.

## REFERENCES

Bandyopadhyay, S. and Saha, S. (2008), "Unsupervised pixel classification in satellite imagery using a new multiobjective symmetry based clustering approach", *TENCON*, India, pp. 1-6.

Berry, M. W. (2004), *Survey of text mining: Clustering, classification, and retrieval*, Springer, Berlin

Dubes, R.C. and Jain, A. K. (1988), *Algorithms for clustering data*, Prentice Hall, New Jersey

Everitt, B. S., Landau, S. and Leese, M. (2001), *Cluster analysis*, Wiley, London

Frank, E. and Witten, I. H. (2005), *Data mining: Practical machine learning tools and techniques*, Morgan Kaufmann

Gan, G., Ma, C. and Wu, J. (2007), *Data clustering: theory, algorithms, and applications*, SIAM, Philadelphia

Han, J. and Kamber, M. (2006), *Data mining: concepts and techniques*, Morgan Kaufmann

Hartigan, J. A. (1975) *Clustering algorithms*, Wiley

Jajuga, K., Sokolowski, A. and Bock, H. H. (2002), *Classification, clustering and data analysis*, Springer, Berlin

Jing, T., Oscar, C. A., Ruobing, Z., Weiyu, Y. and Zhiding, Y. (2008), *An adaptive unsupervised approach toward pixel clustering and color image segmentation*, Elsevier

Kaufman, L. and Rousseeuw, P. J. (2005), *Finding groups in data: an introduction to cluster analysis*, Jonh Wiley & Sons, Hoboken

Kogan, J. (2007), *Introduction to clustering large and high-dimensional data*, Cambridge University Press

Mehta, A. and Tatiraju, S. (2008), *Image segmentation using k-means clustering, EM and Normalized Cuts,* Department of EECS report, University Of California

Siegel H. J. (2009), "Stochastically robust resource management in heterogeneous parallel computing systems", *ISPAN*, USA, pp. 1-2.

Srivastava, A. and Sahami, M. (2009), *Text mining: Classification, clustering, and applications*, Chapman & Hall