University of Alberta

The algorithmic enhancements of user-supervision to the Fuzzy C-Means (FCM) and the successful scenarios of their incorporation

By

Pawel Brzemiński ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science

Department of Electrical and Computer Engineering

Edmonton, Alberta

Spring 2005

0-494-08032-9

Your file Votre référence
ISBN:
Our file Notre reterence
ISBN:

University of Alberta

Library Release Form

Name of Author: Paweł Brzemiński

Title of Thesis: The algorithmic enhancements of user-supervision to Fuzzy C-Means (FCM) and the successful scenarios of their incorporation.

Degree: Master of Science

Year this Degree Granted: 2005

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

Signature

#205 8715-104 Street
Edmonton, AB, T6E 4G7
Canada

Date: Dec. 21, 2004

Dedication

I would like to dedicate this thesis to my parents Janina and Krzysztof and my brother Michał. Thank you for the patience you gave to me while I was growing up.

Abstract

This study presents and evaluates the algorithmic enhancements to Fuzzy C-Means (FCM): the class assignment information in the PS-FCM (Partially Supervised FCM) and the proximity hints in the P-FCM (Proximity-based FCM). It was proposed and examined the PSP-FCM (Partially Supervised Proximity-based FCM), which is a hybrid method of two previous approaches that collaboratively combines both sources of user supervision.

The detailed experimental part embraces the clustering and classification with user supervision on the synthetic data, Machine Learning datasets and a comprehensive case study of WWW documents' (Web pages) collection. It compares and quantifies the amount of user supervision in two modes: random and the misclassification mode, and then presents the most successful scenarios of their application. The results presented may serve as a practical guide for any other applications that incorporates user knowledge, its amount, and quality in a specific experimental design.

Acknowledgement

I would like to sincerely acknowledge, as well as render special thanks to my supervisor Dr Witold Pedrycz for his insightful comments and guidance. His many suggestions have been invaluable.

I am deeply indebted to my parents and brother for their constant and unconditional support.

Finally, I would like to express my gratitude to my friends, both in Canada and Poland, whose presence in my life at the time of my endeavors towards attaining a Master of Science diploma significantly increased my motivation, and helped me considerably in successfully overcoming difficulties: Adrianna and Milan Galandak, Anton Nguyen, Clinton Martin, Kinga and Stefan Slowikowscy, Łukasz Krzymień, Magda Żuber, Sandra Mandić, Tomek Marciniak and Victor Prosolin.

Table of contents

List of Tables

List of Figures

1. Introduction

The evolutionary process created the most advanced classifier known: the human neural system. Observe a person driving a car and be astonished at the speed with which human beings can process complex information, recognize perceived objects, their special location and act accordingly. It is easy to see the great benefit that would be derived from having computers performing human-related recognition activities. The other aspect coming from usage of computers is associated with a huge amount of data to process, which is easily beyond the capabilities of a human being. The usage of computers for the recognition purposes marks the origin for Pattern Recognition field.

The term Pattern Recognition includes the procedures and methodologies that are used to make judgements based on a datum's category [3]. Classically, Pattern Recognition can be divided into three phases: (a) data acquisition (observation space), (b) data pre-processing and feature extraction (feature space), and (c) classification (decision space) [1]. Interestingly, one of the disciplines derived from Pattern Recognition is the Knowledge Discovery in Databases (KDD) process.

KDD itself is an interdisciplinary field, which merges together database management, statistics, machine learning and other related areas aiming at extracting knowledge from data. The whole KDD process consists of the iterative sequence of steps [4] that are directly related to the Pattern Recognition phases. KDD steps form a methodological approach of extracting useful knowledge from large data collections; databases. They are more specialized but develop along Pattern Recognition phases and are as follows:

1) Data cleaning
2) Data integration
3) Data selection
4) Data transformation
5) Data Mining
6) Pattern evaluation
7) Knowledge representation

Data cleaning removes significantly different observations from the data set (outliers). The subsequent steps integrate the data, and then a data subset is selected for further processing. The next challenge of imitating human-like classification processes appears in finding the proper way of abstracting complex objects. Then the objects of complex structure are described in terms of some subset of their features the best characterizing the object. The features are carefully selected and quantitatively measured to allow comparison with features from other objects. The core of extracting information from data is the fifth step of Data Mining. It spans the methods and algorithms for discovering interesting knowledge in the data. Pattern evaluation deals with the identification of truly interesting patterns and the evaluation of Data Mining techniques. The focus of this work will adhere to the methods of Data Mining and Pattern evaluation.

1.1 Fuzzy sets in Pattern Recognition

This work will examine clustering and classification as Data Mining tools. Clustering is a method of unsupervised learning, where all patterns from a dataset are assigned to *k* groups of similar patterns (homogeneity within a cluster), where *k* is smaller than the number of patterns in a dataset. Dissimilar objects will be located in different clusters (heterogeneity between clusters) [2]. There is no training and no a priori knowledge used to influence the process. The ultimate goal of clustering is to generate a partitioning of given dataset into a number of clusters.

Unlike clustering, classification uses a certain training procedure, usually performed on a certain number of labelled patterns called training set. From a training set it is possible to derive a classifier(s). The rest of the patterns is denoted as a test set and is used to validate a classifier's performance. Since the number of classes is fixed, the classifier is used to decide which class a pattern is assigned to given the properties (features) of the pattern. Based on the results, suitable modifications of the classifier's parameters can be carried out. The overall goal of classification is to find the underlying structure in the provided training set, and to learn by modifying classifier's structure and/or parameters that allows classifying of new patterns into one of the existing classes.

Real world data rarely form a concise and transparent structure. Deeper analysis reveals that the clusters' boundaries are more likely to be obscured than crisp. Some clusters will overlap, the smaller groups could be nested inside larger ones. In such scenarios fuzzy sets are useful. Fuzzy sets have proven advantages in Knowledge Discovery and Data Mining systems. Why is the fuzzy sets approach so useful for Data Mining? There are several reasons for it:

(i) The fuzzy sets theory establishes the interface between higher level concepts represented as features and the computer computations driven by the quantitative measurements

(ii) The concept of fuzzy sets is well suited for the real world data, which often do not have crisp boundaries and do have overlapping clusters. This can be expressed as a multiple class membership that is a pattern may belong to certain degree to more than one cluster.

(iii) The membership functions of the fuzzy sets model unsure patterns and are able to identify unclear patterns in the dataset

The advantages of fuzzy sets are beneficial for this work as well. Specifically, the work was confined to clustering methods based on fuzzy sets, which are able to deal very well with unclear patterns. The Fuzzy C-Means algorithm will be of prime interest when clustering and will also be applied as a foundation for the construction of fuzzy classifiers.

The purpose of this study is motivated by the challenges that exist in Pattern Recognition and Data Mining, as presented in the following chapters.

1.2 Bibliography

1. Bow, S., "Pattern recognition: applications to large data-set problems", Marcel Dekker, New York, 1984.
2. Cios, K.J, Pedrycz W., Swiniarski, R. W., "Data Mining: Methods for Knowledge Discovery", Kluwer Academic Publishers, Boston, 1998.
3. Duda, R.O., Hart, P.E, Stork, D.G, "Pattern Classification", John Wiley & Sons, 2001.
4. Han, J., Kamber M., "Data Mining: Concepts and Techniques", Morgan Kaufmann, San Francisco, 2000.

2. Main Objectives

2.1 Motivation

The fuzzy clustering algorithms exhibit obvious advantages for Data Mining's purposes. The clustering process is an unsupervised learning technique, thus there is an absence of any guidance or training. In some cases there is not enough training information in order to train a classifier, but it seems appropriate to take advantage of a small but important piece of information to improve the clustering. Often this kind of knowledge is acquired as experience and is only available for data experts/designers whose knowledge of their domains is very good. In this situation the human operator is able to conclude if the processes appearing in the system are correct, and act in the scenarios when the behaviour should be corrected. This partial, segmented or stepwise involvement of the user in the process of pattern recognition provides field knowledge to improve the unsupervised learning method performance. Henceforth it will be referred to as partial user supervision or simply user supervision. Interest lies in the different types of user supervision, what they embrace, and how they could be realized and integrated with the Fuzzy C-Means (FCM) algorithm [1]. The role of partial user supervision in clustering will be examined and the observations quantified. This project's objectives are as follows:

(i) Study in detail algorithmic enhancements of Fuzzy C-Means algorithm: partial supervised FCM [7] and proximity based FCM [2,3] algorithms

(ii) Focus on the efficient manner of combining these aforementioned types of user supervision into one method – partially supervised proximity based FCM (PSP-FCM) merging two previous techniques

(iii) Provide a well-justified and fully quantified protocol of practical relevance advising how the mechanisms of supervision could be implemented in a given experimental setting

It is also the intention of this project to compare different modes of incorporating user input. For instance, the first mode can constitute random usage of the available knowledge for improvement of clustering/classification accuracy. The idea is to proceed with randomly selected patterns and then require their proper class assignment and/or induce the grade of similarity between them. However, the project will also consider the gradual application of user input only for misclassified patterns in order to correct undesired errors. We will examine if both modes are equivalent, study their practical applications and advantages, measure the required effort of their usage, and the possible improvement of accuracy they provide. Several other related issues will be studied: the applicability of finding or recovering structure in data by Fuzzy C-Means in a noisy, incomplete or distorted data environment, robust statistic based algorithms and relational methods.

Partial supervision can be especially beneficial in three situations, where:

(i) The complete feature space is not available due to lack of data, incomplete measurements or physical unavailability of information.

(ii) The exploration of the structure is strongly influenced by the outliers contained in the dataset.

(iii) It is difficult or cumbersome to construct the feature space of patterns because of the complexity of these features. World Wide Web (WWW) documents are a good example because not only is the textual information important, but also the multimedia content, and the structure of links or layout. The numerical representation of these elements together is highly non-trivial.

Although fuzzy clustering methods are proved to produce high quality results in favourable and less optimal conditions [1], the accuracy of the clustering or classification decreases at difficult tasks. It is worth testing their performance in the presence of the real world data, which often form a noisy, incomplete or distorted data environment. This kind of environment directly causes deterioration of performance.

This work endeavours to find an intermediate state between supervised, trainable methods and completely unsupervised learning. In the supervised scenario, the quality of the training data is important to provide a good approximation of the testing set. In some cases, the amount of information that is possessed is not sufficient for training. Conversely, in unsupervised techniques the user does not have any influence on the process (besides choosing input parameters), which is guided by an internally defined similarity measure, inter and intra cluster distance, or an optimization criterion (alternating optimization algorithms) [2].

Although the user input is important, relying entirely on user knowledge does not seem to be a proper approach too. The essence of the right approach is to combine collaboratively two dimensions of the following: (i) the physical nature of the dataset and (ii) the designer's knowledge potential. The combination creates a fertile field of applications that have fuzzy sets. The ability of fuzzy sets and the algorithms based on them model the data with non-crisp boundaries, and it is well established in the techniques of Pattern Recognition and Data Mining [1,3].

The second reason for applications of fuzzy methods lies in their suitability of emulating of human cognitive processes, and this is especially important in the area of Pattern Recognition [5]. The application and rationale for supervision mechanisms can be twofold: (i) it will provide a new line of user–oriented support for the clustering. The data experts would be able to model the dataset by their influence ranging from minimal to significant depending on the amount of information available, (ii) such support from the users side can play a key role in the situation when the dataset is difficult to explore and standard methods provide insufficient results. This may happen in presence of outliers or distorted structure of dataset.

Finally, it is important in this study to provide justification for the user supervision component experimentally by examining its influence on the clustering and classification processes, and illustrating the performance boost of clustering and classification on the datasets in the areas of our interest. The performance assessment will be based on three categories of datasets. First the algorithms will be illustrated on the synthetic datasets. The second category constitutes Machine Learning datasets. In the last data collection, WWW documents from different thematic groups will be the subject of clustering.

2.2 Bibliography

1. Bezdek, J.C., "Pattern recognition with fuzzy objective function algorithms", Plenum Press, New York, 1981.
2. Duda, R.O., Hart, P.E, Stork, D.G, "Pattern Classification", John Wiley & Sons, 2001.
3. Han, J., Kamber M., "Data Mining: Concepts and Techniques", Morgan Kaufmann, San Francisco, 2000.
4. Loia, V., Pedrycz, W., Senatore, S., P-FCM: a proximity-based fuzzy clustering for user-centered web applications, *International Journal of Approximate Reasoning*, Vol. 34, Issues 2-3, 2003, pp. 121-144.
5. Pedrycz, W., Fuzzy sets in pattern recognition: methodology and methods, *Pattern Recognition*, Vol. 23, No. ½, 1990, pp. 121-146.
6. Pedrycz, W., Loia, V., Senatore, S., P-FCM: proximity—based fuzzy clustering, *Fuzzy Sets and Systems*, Vol. 148, Issue 1, 16, 2004, pp. 21-41.
7. Pedrycz, W., Waletzky, J., Fuzzy clustering with partial supervision, *IEEE Trans. on Systems, Man, and Cybernetics* 5, 1997, pp. 787-795.

3. Literature Review

This section of our study presents the algorithms, methods and concepts from two fields: fuzzy clustering with user supervision, and Web Mining methods and concepts. The literature of fuzzy clustering, and the literature regarding user supervision guiding the process of data discovery will be reviewed. Then, we will carefully examine the existing trends and challenges in Web Mining, which can be a fertile field for applications of user supervision components.

3.1 Fuzzy clustering and user supervision support literature

Numerous approaches of unsupervised fuzzy clustering methods in the literature [3,4,5,9,13,20,22] seem to confirm its valuable assets in revealing unknown information about data, and organizing it into groups of similar patterns. The most known and widely used method here is the Fuzzy c-Means (FCM) algorithm [3]. There are situations when the extensive dimensionality growth may increase unnecessary complexity of computations. The solution to this particular problem forms a family of relational clustering algorithms [4,5,12, 13] with a Relational Fuzzy C-Means as a representative. The patterns in this case instead are being represented as multidimensional feature vectors are implicitly referred to as pair wise relative similarities (dissimilarities) to each other. RFCM and other relational methods store these similarities in a similarity (dissimilarity) matrix where rows and columns correspond to patterns. Essentially, RFCM is an equivalent method to FCM. The evident advantages of RFCM can be applied to the pure relational data or the reduction of the patterns' dimensionality (similarity can be computed only once before running of the algorithm). The drawback of using relational methods is an increased computational complexity.

Although fuzzy clustering methods match well with the nature of real world data, and perform well in many environments, problems can appear in some cases because we are rarely provided with clean, complete data in real world scenarios. In this case the most obvious source of authorized information are users equipped with their field knowledge, experience and/or intuition. The idea of incorporating user input is not new, and several approaches of supervised and semi-supervised learning were created as a result [1, 2,21,23].

The incorporation of a designer's knowledge in fuzzy clustering appears in various extensions of the situation described above: (a) an internal optimization mechanism ultimately guided by the user [21,23]; (b) weight vectors magnifying the impact of selected patterns (or features) in the clustering process. [1,2]; (c) the objective function is extended by an additive, supervision component [24].

Proximity-based FCM (P-FCM) [21,23] exhibits a concept of an arbitrary modeling of a dataset by entering user 'hints'. One is able to enter subjective proximity information for any pair of patterns existing in the dataset ranging from very dissimilar to nearly identical. The provision of user guidance allows the person to more easily grasp a difficult, more abstract concept of similarity in a very convenient way.

A form of exploiting an expert knowledge acquired a priori to the experiments was introduced by Bensaid *et al.* [1] and was developed along the lines of the weight vectors idea. This approach was applied to Magnetic Resonance Imaging (MRI) clustering. A part of the dataset is labelled in the initialization phase with known labels by a human expert; the rest of the patterns remain unclassified.

In contrast to the existing approaches of (a) and (b), another way of allowing the user to influence the clustering is using Partially Supervised FCM [24]. The concept of Partially Supervised Fuzzy C–Means lies in the foundation of providing the user with a flexible way of incorporating his domain knowledge, and imposing the assignment of patterns to particular clusters. The guidance to the algorithm is provided via user hints in the form of an extension of the general objective function by an additional component.

It is important to mention briefly robust clustering algorithms, which are especially designed to deal with outliers in datasets. Kersten [16,17,18] proposed a Fuzzy C-Median (FCMED) clustering algorithm derived on a basis of a fuzzy median. The fuzzy median is a robust statistic, which is able to accept almost 50% of outliers before it loses its ability to generate meaningful results. Two other approaches include relational algorithms: Fuzzy C–Medoids (FCMdd) by Krishnapuram [20] and a robust non-Euclidean fuzzy relational data clustering method (robust-NE-FRC) [9]. Dave and Sen [9] presented a broad spectrum of existing relational clustering techniques using objective functional. They introduced robust-NE-FRC, a relational algorithm dealing efficiently with noise and outliers. A low-complexity fuzzy relational algorithm (FCMdd) and its robust version were successfully applied to WWW documents and snippet clustering. Good results were recorded, although the algorithm itself does not guarantee finding of a global minimum. It is advisable to test several random initializations to determine which produce the best results.

3.2 Web Mining: concepts and techniques

Dealing with WWW data imposes additional challenges to the concepts and methods applied to this field. WWW data by their nature and large size causes all kinds of problems such as incompleteness, non-uniformity and inconsistency. The problems described can only be partially overcome by data cleaning, integration or transformation. This study will focus on applying user guidance in the context of Web Data Mining where standard approaches perform poorly. There were several attempts to prevail over emerging problems.

In articles about the WWW, it is appropriate to include PageRank used in Google and HITS algorithms [7, 19], respectively. Both approaches introduce the concept of authoritative pages, which are pointed many other WWW pages (back-links, and give the estimate of page importance – PageRank) and hubs, which contain many forward-links (outgoing links from a page) and suggestions about other authoritative pages (HITS). Both algorithms are not clustering algorithms and can be used to compute the ranks of web pages within a given set of pages, but are unable to form complete thematic clusters of a given collection of Web documents. Moreover, to obtain meaningful results they have to be supported by auxiliary text information from the WWW pages.

Roussinov and Zhao [25] constructed Context Sensitive Similarity Discovery (CSSD) - a method for computing similarity relationships among concepts surfacing during an electronic brainstorming session. In contrast to the vector space model, which their technique outperforms, CSSD is capable of grasping a similarity between documents that do not contain any keywords in common but their topic is relevant.

An evident shortcoming of CSSD is the construction of a similarity network, which in practice is difficult to create a priori from unknown, unlabelled data. Even though the authors believe that this method would work with other clustering algorithms, the limiting factor can be a choice of parameters resulting in creating a similarity matrix very different from natural relationships between the concepts. It seems that this method would perform well using documents of a smaller size, and with a certain amount of user supervision.

Broder *et al.* [8] demonstrated a scalable method for syntactic clustering on the WWW that focused mainly at finding duplicate and contained documents, that is locating highly similar alternatives to a given URL. Their idea uses representation of documents with contained in it contiguous sequences of words (shingles). Limitations of this method derive from its internal architecture that it is not efficient for basic clustering and complexity of data processing, which is storage and computationally expensive, especially in the context of forming clusters in the real time.

Another method, which uses the vector space model is the modified Adaptive Resonance Theory (ART) algorithm proposed by Vlajic and Card [28]. The authors proposed solving the problem of synonymy and reduction of dimension by applying a thesaurus. In practice, the method of creating a thesaurus requires human assistance, which together with the fact that the algorithm is applied only to a single domain (neural networks) and the authors provided insufficient measures of analyzing the performance of the proposed method, brought the applicability of ART in a broader situation into question.

Guillaume [11] proposed another approach to the clustering of XML documents. He found a relationship between documents as expressed by the links between them, thus the entire problem was reduced to a graph-partitioning problem. The proposed method, although successful for XML documents, exploits XML links, and this exploitation is the main reason that clustering is limited to a narrow domain.

Boley *et al.* [6] demonstrated further two techniques of WWW document clustering. The first method (ARHP) exploits the concept of association rule discovery. In this technique items are documents, and features (words) are transactions so the affinities among documents are captured by frequent item sets. The second method (PDDP) is a principal direction algorithm, hierarchically creating a binary tree hierarchy of clusters in which the root is an entire document set. Both algorithms reveal great assets, are fast, scalable and efficient. However, they may not represent a natural organization of data. ARHP creates compact, highly cohesive clusters but leaves some documents unassigned to any of the clusters. Dendroram outputted from PDDP is based on a binary split of every cluster that will obviously favour clusters of the same size.

Self-organizing maps (SOM) proved to be a useful tool for clustering pages with users' navigational patterns [27], clustering document collections [14], finding WWW communities [10] and browsing the results of clustering. The first approach [27] is limited to a narrow domain with difficult parameters estimation, and they might not scale well. The third approach [10] cannot reveal real clusters, for a WWW community is not a cluster but a collection of web pages that have more hyperlinks between each other than other pages from outside of the community.

Although the WEBSOM map [14] was successfully applied to the clustering of documents in large collection, it may not scale for larger WWW collections, as the word and document maps will significantly increase.

Aforementioned algorithms, except self-organizing maps, and most of the standard approaches (especially partitioning methods, k-means or k-medoids) produce crisp clusters. This approach is more appropriate for clusters, which can be well separated from each other rather then for overlapping, very close thematically or nested clusters.

The variability of WWW data requires a more careful approach, possibly introduced by fuzzy sets. The natural extensions to Web approaches are fuzzy sets allowing arbitrary modeling the assignment of an object (e.g. Web page) to a cluster by fuzzy membership values. A lot of work in the realm of fuzzy clustering was done by Bezdek and Hathaway [3, 4, 12, 26], who introduced numerous fuzzy algorithms i.e. FCM, RFCM, NERFCM. According to Krishnampuram [20], FCM, RFCM and NERFCM [3, 4, 12] algorithms seem to be more stable in practice and have proved its efficiency and accuracy. The RFCM algorithm is a relational version of the FCM, with a restriction that the similarity data has to be derived from Euclidean distance. The NERFCM relaxes this restriction imposed on a dissimilarity matrix by introducing β-spread transformations. In fact, the RFCM algorithm is closely related to another technique presented by Kaufman and Rousseeuw – the FANNY algorithm [15].

Fuzzy algorithms have strong mathematical foundations, are fairly efficient, accurate, and suitable for enhancements. In our context it is important that they are able to reveal 'unsure' patterns; those that are not associated closely with any particular cluster. A possible limitation can be a choice of parameters (this is a subject of the optimization as we show in later sections), computational complexity, which can be also overcome (Krishnampuram *et al.* [20]), and sensitivity to outliers. Dave and Sen [9] introduced robust-NE-FRC and proposed transformations generalizing approaches of converting non-robust algorithms into robust ones.

Some relational fuzzy clustering algorithms were applied to Web data mining. Runkler and Bezdek [26] used Relational Alternating Cluster Estimation (RACE) algorithm for WWW logs' and newsgroups' articles clustering. Krishnampuram *et al.* [20] presented a low-complexity fuzzy relational algorithm (FCMdd) similar in concept to RFCM. Although Runkler and Bezdek observed acceptable results, using other weighting scheme for keywords besides that of Levenshtein distance might have produced better results.

3.3 Bibliography

1. Benkhalifa, M., Bensaid, A. and Mouradi, A., Text categorization using the semi-supervised fuzzy c-means algorithm, *18th International Conference of the North American Fuzzy Information Processing Society - NAFIPS*, 1999, pp. 561-565.
2. Bensaid, A.M., Hall, L.O., Bezdek, J.C., Clarke, L.P., Partially supervised clustering for image segmentation, *Pattern Recognition*, Vol. 29, No. 5, 1996, pp. 859-871.
3. Bezdek, J.C., "Pattern recognition with fuzzy objective function algorithms", Plenum Press, New York, 1981.
4. Bezdek, J.C., Hathaway, R.J., A note on two clustering algorithms for relational network data, *SPIE Vol. 1293 Applications of Artificial Intelligence VIII*, 1990, pp. 268-277.
5. Bezdek, J.C., Hathaway, R.J., Windham, M.P., Numerical comparison of the RFCM and AP algorithms for clustering relational data, *Pattern Recognition*, Vol. 24, No. 8, 1991, pp. 783-791.
6. Boley, D., Gini, M., Gross, R., Han, E.H.S., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., Moore, J., Partitioning-based clustering for Web document categorization, *Decision Support Systems*, 27, 1999, pp. 329–341.
7. Brin, S., Motwani, R., Page, L., Winograd, T., What can you do with a Web in your pocket?, *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 1998.
8. Broder, A.Z., Glassman, S.C., Manasse, M.S., Zweig, G., Syntactic clustering of Web, *Computer Networks and ISDN Systems*, 29, 1997, pp. 1157–1166.
9. Dave, R.N., Sen, S., Robust fuzzy clustering of relational data, *IEEE Transactions on Fuzzy Systems*, Vol. 10, No. 6, 2002.
10. Flake, G.W., Lawrence, S., *et al.*, Self-Organization and Identification of Web Communities, *Computer*, 35(3), 2002, pp. 66–72.

11. Guillaume, D., Murtagh, F., Clustering of XML documents, *Computer Physics Communications,* 127, 2002, pp. 215–227.

12. Hathaway, R.J., Bezdek, J.C., NERF c-means: Non-Euclidian relational fuzzy clustering, *Pattern Recognition,* 27(3), 1994, pp. 429–437.

13. Hathaway, R.J., Bezdek, J.C., Davenport, W.J, On relational data versions of c-means algorithms, *Pattern Recognition Letters,* Vol. 17, 1996, pp. 607-612.

14. Kaski, S., Honkela, S. T., *et al.*, WEBSOM - Self-organizing maps of document collections, *Neurocomputing*, 21, 1998, pp. 101–117.

15. Kaufman, L., Rousseeuw, P.J., “Finding Groups in Data: An Introduction to Cluster Analysis”, *New York: Wiley*, 1990.

16. Kersten, P. R., Fuzzy Order Statistics and Their Application to Fuzzy Clustering, *IEEE Transactions on Fuzzy Systems*, Vol. 7, No. 6, 1999.

17. Kersten, P. R., Implementation issues in the fuzzy c-medians clustering algorithm, *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, Vol. 2, 1997.

18. Kersten, P. R., The Fuzzy Median and the Fuzzy MAD, *Proceedings of ISUMA-NAFIPS*, 1995.

19. Kleinberg, J., Authoritative sources in a hyperlinked environment, *Journal of the ACM,* 46(5), 1999, pp. 604–632.

20. Krishnapuram, R., Joshi, A., Nasraoui, O., Yi, L., Low-complexity fuzzy relational clustering algorithms for Web Mining, *IEEE Transactions on Fuzzy Systems,* Vol. 9, No. 4, 2001.

21. Loia, V., Pedrycz, W., Senatore, S., P-FCM: a proximity-based fuzzy clustering for user-centered web applications, *International Journal of Approximate Reasoning*, Vol. 34, Issues 2-3, 2003, pp. 121-144.

22. Pedrycz, W., Fuzzy sets in pattern recognition: methodology and methods, *Pattern Recognition*, Vol. 23, No. ½, pp. 121-146, 1990.

23. Pedrycz, W., Loia, V., Senatore, S., P-FCM: proximity—based fuzzy clustering, *Fuzzy Sets and Systems,* Vol. 148, Issue 1, 16, 2004, pp. 21-41.

24. Pedrycz, W., Waletzky, J., Fuzzy clustering with partial supervision, *IEEE Trans. on Systems, Man, and Cybernetics* 5, 1997, pp. 787-795.

25. Roussinov, D., Zhao, J. L., Automatic discovery of similarity relationships through Web mining, *Decision Support Systems,* 987, 2002.

26. Runkler, T.A., Bezdek, J.C., Web mining with relational clustering, *International Journal of Approximate Reasoning,* 32, 2003, pp. 217–236.

27. Smith, K.A., Ng, A., Web page clustering using self-organizing map of user navigation patterns, *Decision Support Systems,* 995, 2002.

28. Vlajic N., Card, H.C., Categorizing Web pages on the subject of neural networks, *Journal of Network and Computer Applications,* 1998, pp. 91–105.

4. Datasets Description

The synthetic data sets used in this study were chosen to illustrate the idea of operating of the algorithms and assessing their performance. In addition to the artificial data, real-world datasets were also used. Machine Learning and Web Mining datasets constitute more challenging data collections, allowing more meaningful evaluation of the assumed methodologies and techniques, as well as emphasizing their practical applicability.

4.1 Synthetic datasets

The synthetic multivariate datasets generated in this study have normal distribution with the mean vector and covariance matrix $N(\mu,\Sigma)$. This regular and predictable structure was subjected to interference from a group of noise points. Outliers can distort the original structure and make it substantially more difficult to explore. In this case the outliers are from a group of points generated from a heavy-tailed Cauchy distribution, and will be used to distort the structure. In the experiments with the synthetic datasets, the primary interest lies in simulating real-word scenarios. The challenge is reconstruction with the available methods the original (known) structure from the dataset. This allows us to test the performance of the algorithms by extensive experimentation using different parameters, and then gathering general observations of their functioning. These actions will possibly lead to their tuning and improvements. The synthetic datasets can be arbitrarily modified as to the number of points and outliers while the parameters of the distributions provide more exhaustive information to the researcher during the experiment.

4.2 Machine Learning datasets

The UCI Repository of machine learning databases [1] provided several datasets for this study. They are used by the machine learning community for the empirical analysis of algorithms. The datasets may cover a range of topics from a chemical wine analysis to picture segmentation, or biological data manipulation, therefore a variety of clustering and classification tasks can be performed on them.

4.2.1 Wine recognition database

These data were collected from the results of a chemical analysis of wines grown in the same region in Italy, but derived from 3 different cultivars. The analysis determined the quantities of the 13 constituents found in each of the 3 types of wines. The class distribution is as follows: class A – 59 instances, class B – 71, class C – 48. All 13 attributes are continuous. Table 4-1 and Table 4-2 show the range of the features' values, the mean and standard deviation of the features in each class.

Table 4-1.Constituents of the wine chemical analysis: range of values.

#	Chemical		Range of values	
	constituent	Class 1	Class 2	Class 3

1	Alcohol	[12.85;14.83]	[11.03;13.86]	[12.2;14.34]
2	Malic acid	[1.35;4.04]	[0.74;5.8]	[1.24;5.65]
3	Ash	[2.04;3.22]	[1.36;3.23]	[2.1;2.86]
4	Alkalinity of ash	[11.2;25.0]	[10.6;30.0]	[17.5;27.0]
5	Magnesium	[89.0;132.0]	[70.0;162.0]	[80.0;123.0]
6	Total phenols	[2.2;3.88]	[1.1;3.52]	[0.98;2.8]
7	Flavanoids	[2.19;3.93]	[0.57;5.08]	[0.34;1.57]
8	Nonflavanoid phenols	[0.17;0.5]	[0.13;0.66]	[0.17;0.63]
9	Proanthocyanins	[1.25;2.96]	[0.41;3.58]	[0.55;2.7]
10	Color intensity	[3.52;8.9]	[1.28;6.0]	[3.85;13.0]
11	Hue	[0.82;1.28]	[0.69;1.71]	[0.48;0.96]
12	OD280/OD315 of diluted wines	[2.51;4.0]	[1.59;3.69]	[1.27;2.47]
13	Proline	[680.0;1680.0]	[278.0;985.0]	[415.0;880.0]

Table 4-2. Constituents of the wine chemical analysis: means and standard deviations.

#	Chemical constituent	Means ± Standard Deviations		
		Class 1	Class 2	Class 3
1	Alcohol	13.74 ± 0.46	12.27 ± 0.53	13.15 ± 0.53
2	Malic acid	2.01 ± 0.68	1.93 ± 1.01	3.33 ± 1.08
3	Ash	2.45 ± 0.22	2.24 ± 0.31	2.43 ± 0.18
4	Alkalinity of ash	17.03 ± 2.54	20.23 ± 3.34	21.41 ± 2.25
5	Magnesium	106.33 ± 10.49	94.54 ± 16.75	99.31 ± 10.89
6	Total phenols	2.84 ± 0.33	2.25 ± 0.54	1.67 ± 0.35
7	Flavanoids	2.98 ± 0.39	2.08 ± 0.70	0.78 ± 0.29
8	Nonflavanoid phenols	0.29 ± 0.07	0.36 ± 0.12	0.44 ± 0.12
9	Proanthocyanins	1.89 ± 0.41	1.63 ± 0.60	1.15 ± 0.40
10	Color intensity	5.52 ± 1.23	3.08 ± 0.92	7.39 ± 2.31
11	Hue	1.06 ± 0.11	1.05 ± 0.20	0.68 ± 0.11
12	OD280/OD315 of diluted wines	3.15 ± 0.35	2.78 ± 0.49	1.68 ± 0.27
13	Proline	1115.71 ± 221.52	519.50 ± 157.21	629.89 ± 115.09

4.2.2 Image segmentation data

The instances of this dataset were drawn randomly from a database of 7 outdoor images. The images were hand-segmented to create a classification for every pixel. Each instance is a 3x3 pixel region. The whole dataset contains 2,100 instances with a distribution of 300 instances per class. Each instance is described by a 19-dimensional vector of various statistical descriptors (Table 4-3). The seven classes are as follows: brick-face, sky, foliage, cement, window, path and grass. Table 4-4, 4-5 and Table 4-6, 4-7 show the range of values and means with standard deviations.

Table 4-3. Statistical descriptors as features for image segments.

#	Statistical descriptor	Description
1	region-centroid-col	The column of the center pixel of the region.
2	region-centroid-row	The row of the center pixel of the region
3	region-pixel-count	The number of pixels in a region = 9

4	short-line-density-5	The results of a line extraction algorithm that counts how many lines of length 5 (any orientation) with low contrast, less than or equal to 5, go through the region
5	short-line-density-2	Same as short-line-density-5 but counts lines of high contrast, greater than 5
6	vedge-mean	Measure the contrast of horizontally adjacent pixels in the region. There are 6, the mean and standard deviation are given. This attribute is used as a vertical edge detector
7	vegde-sd	(see 6)
8	hedge-mean	Measures the contrast of vertically adjacent pixels. Used for horizontal line detection
9	hedge-sd	(see 8)
10	intensity-mean	The average over the region of (R + G + B)/3
11	rawred-mean	The average over the region of the R value
12	rawblue-mean	The average over the region of the B value
13	Rawgreen-mean	The average over the region of the G value
14	exred-mean	Measure the excess red: (2R - (G + B))
15	exblue-mean	Measure the excess blue: (2B - (G + R))
16	exgreen-mean	Measure the excess green: (2G - (R + B))
17	value-mean	3-d nonlinear transformation of RGB
18	saturation-mean	(see 17)
19	hue-mean	(see 17)

Table 4-4. Statistical descriptors as features for image segments: range of values. Classes 1-3.

#	Statistical descriptor	Range of values		
		Class 1	Class 2	Class 3
1	region-centroid-col	[2;245]	[2;253]	[119;253]
2	region-centroid-row	[164;251]	[171;201]	[12;154]
3	region-pixel-count	[9;9]	[9;9]	[9;9]
4	short-line-density-5	[0;0.22]	[0;0.11]	[0;0.11]
5	short-line-density-2	[0;0.11]	[0;0.22]	[0;0.11]
6	vedge-mean	[0.5;3.38]	[0.83;4.66]	[0;5.50]
7	vegde-sd	[0.32;3.25]	[0.51;5.49]	[0;5.73]
8	hedge-mean	[0.77;8]	[1.00;9.77]	[0;9.27]
9	hedge-sd	[0.49;2.94]	[0.91;7.36]	[0;3.35]
10	intensity-mean	[8.92;20.66]	[39.85;52.55]	[0;15.37]
11	rawred-mean	[6.33;21.44]	[35.77;47.33]	[0;11.33]
12	rawblue-mean	[6.44;17.77]	[47.33;65]	[0;22]
13	Rawgreen-mean	[13.77;23.33]	[35;45.77]	[0;13.88]
14	exred-mean	[-10.11;2.33]	[-18.55;0]	[13;0.11]
15	exblue-mean	[-12.44;0]	[22.44;37.33]	[0;22.66]
16	exgreen-mean	[6.33;18.88]	[-22;0]	[10.22;0]
17	value-mean	[13.77;23.33]	[47.33;65]	[0;22]
18	saturation-mean	[0.23;0.59]	[0.24;0.32]	[0;0.88]
19	hue-mean	[1.28;2.33]	[-2.17;0]	[-2.37;0]

Table 4-5. Statistical descriptors as features for image segments: range of values. Classes 4-7.

#	Range of values			
	Class 4	Class 5	Class 6	Class 7
1	[2;252]	[3;124]	[1;254]	[1;254]
2	[13;159]	[64;147]	[12;249]	[11;249]
3	[9;9]	[9;9]	[9;9]	[9;9]

4	[0;0.22]	[0;0.11]	[0;0.22]	[0;0.33]
5	[0;0.22]	[0;0.22]	[0;0.22]	[0;0.22]
6	[0.22;17.5]	[0;25.5]	[0;22.16]	[0;29.22]
7	[0.17;16.44]	[0;14.96]	[0;375.09]	[0;991.71]
8	[0.27;10.33]	[0;27.27]	[0;26.61]	[0;44.72]
9	[0.25;5.72]	[0;21.61]	[-1.58;184.52]	[0;1386.32]
10	[20.51;72.88]	[0;49.81]	[0;129.25]	[0;143.44]
11	[19.88;65]	[0;41.77]	[0;118.33]	[0;137.11]
12	[22.33;90.11]	[0;61.11]	[0;144]	[0;150.88]
13	[18.66;64.55]	[0;46.55]	[0;125.44]	[0;142.55]
14	[-31.11;0]	[-34.44;0]	[-46.22;0.66]	[-49.66;9.88]
15	[5.44;59.22]	[0;52.22]	[-12.11;70.33]	[-11;82]
16	[-28.11;0]	[-17.77;3]	[-26.66;24.66]	[-33.88;20.33]
17	[22.33;90.11]	[0;61.11]	[0;144]	[0;150.88]
18	[0.15;0.36]	[0;1]	[0;1]	[0;1]
19	[-2.18;0]	[-3.04;0]	[-2.45;2.86]	[-3.01;2.91]

Table 4-6. Statistical descriptors as features for image segments: means and standard deviations. Classes 1-3.

#	Statistical descriptor	Means ± Standard Deviations		
		Class 1	Class 2	Class 3
1	region-centroid-col	110± 14 72.99	168.55±67.28	191.48±33.42
2	region-centroid-row	213.04±23.27	187.22±7.46	101.01±47.71
3	region-pixel-count	9 0±0	8.99±1.96	9±0
4	short-line-density-5	0.03±0.05	0.01±0.04	0.00±0.01
5	short-line-density-2	0.00±0.01	0.00±0.03	0.00±0.01
6	vedge-mean	1.56±0.68	2.17±0.86	0.82±1.23
7	vegde-sd	1.15±0.71	1.60±0.84	0.70±1.14
8	hedge-mean	2.04±1.18	3.31±1.71	0.64±1.31
9	hedge-sd	1.34±0.59	2.25±1.39	0.49±0.75
10	intensity-mean	13.19±2.36	47.37±2.95	3.83±4.55
11	rawred-mean	11.12±2.69	42.72±2.59	2.74±3.30
12	rawblue-mean	10.23±2.49	58.15±3.99	6.08±7.01
13	Rawgreen-mean	18.24±2.21	41.25±2.38	2.67±3.43
14	exred-mean	-6.23±1.93	-13.96±2.08	-3.28±3.91
15	exblue-mean	-8.88±2.14	32.33±3.39	6.75±7.60
16	exgreen-mean	15.12±2.51	-18.37±2.07	-3.47±3.80
17	value-mean	18.24±2.21	58.15±3.99	6.08±7.01
18	saturation-mean	0.46±0.08	0.29±0.01	0.33±0.29
19	hue-mean	1.98±0.16	-2.00±0.05	-1.37±0.96

Table 4-7. Statistical descriptors as features for image segments: means and standard deviations. Classes 4-7.

#	Means ± Standard Deviations			
	Class 4	Class 5	Class 6	Class 7
1	112.75±66.66	62.43±37.06	133.77±69.89	118.67±73.94
2	85.44±39.56	118.29±19.63	130.99±63.29	118.13±53.90
3	9.00±0	8.99±0	9.00±0	9. ± 0
4	0.02±0.04	0.01±0.03	0.01±0.04	0.01±0.04
5	0.00±0.02	0.01±0.03	0.00±0.02	0.00±0.02
6	2.79±2.78	2.49±3.71	2.02±2.50	1.70±2.77
7	1.80±2.14	2.11±2.94	3.49±22.99	8.93±61.75
8	1.90±1.77	2.88±4.73	2.49±2.89	2.49±4.04

9	1.27 ± 1.04	2.40 ± 3.97	3.14 ± 12.54	13.63 ± 75.38
10	49.70 ± 13.01	9.64 ± 9.88	51.35 ± 38.36	33.02 ± 41.69
11	43.57 ± 11.28	6.11 ± 7.62	45.42 ± 34.93	29.60 ± 38.56
12	62.08 ± 16.29	14.77 ± 13.71	60.46 ± 44.28	39.08 ± 46.74
13	43.44 ± 11.59	8.04 ± 8.76	48.17 ± 36.28	30.39 ± 40.12
14	-18.39 ± 6.11	-10.58 ± 8.66	-17.79 ± 11.42	-10.27 ± 12.23
15	37.16 ± 10.56	15.38 ± 13.04	27.34 ± 20.83	18.17 ± 18.95
16	-18.77 ± 4.62	-4.79 ± 4.89	-9.54 ± 13.47	-7.89 ± 10.49
17	62.08 ± 16.29	14.82 ± 13.74	61.73 ± 43.11	39.94 ± 46.30
18	0.30 ± 0.03	0.65 ± 0.26	0.35 ± 0.17	0.48 ± 0.23
19	-2.07 ± 0.06	-2.16 ± 0.58	-1.28 ± 1.73	-1.32 ± 1.53

4.2.3 Wisconsin Diagnostic Breast Cancer (WDBC) and Wisconsin Prognostic Breast Cancer (WPBC) data

Experiments were performed using two breast cancer databases: the Wisconsin Diagnostic Breast Cancer (WDBC) and the Wisconsin Prognostic Breast Cancer (WPBC). Each pattern from these databases represents follow-up data for one breast cancer patient. Only those cases exhibiting invasive breast cancer and no evidence of distant metastases at the time of diagnosis are included.

The first dataset, from WDBC, contains 569 patterns (class distribution: 357 benign, 212 malignant). The 10 features examined are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image (Table 4-8). For each of these features the mean, standard error and "worst", or the largest mean of the three largest values, were computed for each image, resulting in 30 real-valued features.

In the second set there are 194 patterns (class distribution: 148 non-recurring before 24 months, 46 recurring). 4 incomplete patterns were eliminated from the initial 198 records. This dataset has, in addition to the 30 features formed in the way mentioned above, two additional (Table 4-9). For both datasets the means and standard deviations for each class were computed (Table 4-10, Table 4-11, Table 4-12 and Table 4-13).

Table 4-8. Features for the WDBC and the WPBC datasets.

#	Feature	Description
1	Radius	mean of distances from center to points on the perimeter
2	Texture	standard deviation of gray-scale values
3	Perimeter	-
4	Area	-
5	Smoothness	local variation in radius lengths
6	Compactness	perimeter^2 / area - 1.0
7	Concavity	severity of concave portions of the contour
8	concave points	number of concave portions of the contour
9	symmetry	-
10	fractal dimension	"coastline approximation" – 1

Table 4-9. Additional features for the WPBC dataset.

#	Feature	Description
1	Tumor size	diameter of the excised tumor in centimeters

2	Lymph node status	number of positive auxiliary lymph nodes observed at time of surgery

Table 4-10. Features of breast cancer cases for the WDBC dataset: range of values.

#	Range of values	
	Class 1	Class 2
1	[6.98;17.85]	[10.95;28.11]
2	[9.71;33.81]	[10.38;39.28]
3	[43.79;114.6]	[71.9;188.5]
4	[143.5;992.1]	[361.6;2501]
5	[0.05;0.16]	[0.07;0.14]
6	[0.01;0.22]	[0.04;0.34]
7	[0;0.41]	[0.02;0.42]
8	[0;0.08]	[0.02;0.20]
9	[0.10;0.27]	[0.13;0.30]
10	[0.05;0.09]	[0.04;0.09]
11	[0.11;0.88]	[0.19;2.87]
12	[0.36;4.88]	[0.36;3.56]
13	[0.75;5.11]	[1.33;21.98]
14	[6.80;77.11]	[13.99;542.2]
15	[0.00;0.02]	[0.00;0.03]
16	[0.00;0.10]	[0.00;0.13]
17	[0;0.39]	[0.01;0.14]
18	[0;0.05]	[0.00;0.04]
19	[0.00;0.06]	[0.00;0.07]
20	[8.94;0.02]	[0.00;0.01]
21	[7.93;19.82]	[12.84;36.04]
22	[12.02;41.78]	[16.67;49.54]
23	[50.41;127.1]	[85.1;251.2]
24	[185.2;1210]	[508.1;4254]
25	[0.07;0.20]	[0.08;0.22]
26	[0.02;0.58]	[0.05;1.05]
27	[0;1.25]	[0.02;1.17]
28	[0;0.17]	[0.02;0.29]
29	[0.15;0.42]	[0.15;0.66]
30	[0.05;0.14]	[0.05;0.20]

Table 4-11. Features of breast cancer cases for the WDBC dataset: standard deviations.

Statistical descriptor #	Means ± Standard Deviations		
	Dataset	Class 1	Class 2
1	14.12 ± 3.52	12.14 ± 1.78	17.46 ± 3.20
2	19.28 ± 4.30	17.91 ± 3.99	21.60 ± 3.77
3	91.96 ± 24.29	78.07 ± 11.80	115.36 ± 21.85
4	654.88 ± 351.91	462.79 ± 134.28	978.37 ± 367.93
5	0.09 ± 0.01	0.09 ± 0.01	0.10 ± 0.01
6	0.10 ± 0.05	0.08 ± 0.03	0.14 ± 0.05
7	0.08 ± 0.07	0.04 ± 0.04	0.16 ± 0.07
8	0.04 ± 0.03	0.02 ± 0.01	0.08 ± 0.03
9	0.18 ± 0.02	0.17 ± 0.02	0.19 ± 0.02
10	0.06 0.00	0.06 ± 0.00	0.06 ± 0.00
11	0.40 ± 0.27	0.28 ± 0.11	0.60 ± 0.34
12	1.21 ± 0.55	1.22 ± 0.58	1.21 ± 0.48
13	2.86 ± 2.02	2.00 ± 0.77	4.32 ± 2.56
14	40.33 ± 45.49	21.13 ± 8.84	72.67 ± 61.35

15	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
16	0.02 ± 0.01	0.02 ± 0.01	0.03 ± 0.01
17	0.03 ± 0.03	0.02 ± 0.03	0.04 ± 0.02
18	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
19	0.02 ± 0.00	0.02 ± 0.00	0.02 ± 0.01
20	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
21	16.26 ± 4.83	13.37 ± 1.98	21.13 ± 4.28
22	25.67 ± 6.14	23.51 ± 5.49	29.31 ± 5.43
23	107.26 ± 33.60	87.00 ± 13.52	141.37 ± 29.45
24	880.58 ± 569.35	558.89 ± 163.60	1422.28 ± 597.96
25	0.13 ± 0.02	0.12 ± 0.02	0.14 ± 0.02
26	0.25 ± 0.15	0.18 ± 0.09	0.37 ± 0.17
27	0.27 ± 0.20	0.16 ± 0.14	0.45 ± 0.18
28	0.11 ± 0.06	0.07 ± 0.03	0.18 ± 0.04
29	0.29 ± 0.06	0.27 ± 0.04	0.32 ± 0.07
30	0.08 ± 0.01	0.07 ± 0.01	0.09 ± 0.02

Table 4-12. Features of breast cancer cases for the WPBC dataset: range of values.

#	Range of values	
	Class 1	Class 2
1	[1;125]	[1;78]
2	[10.95;24.63]	[12.34;27.22]
3	[10.38;39.28]	[14.34;30.99]
4	[71.9;166.2]	[81.15;182.1]
5	[361.6;1841]	[477.4;2250]
6	[0.07;0.14]	[0.08;0.12]
7	[0.04;0.31]	[0.06;0.23]
8	[0.02;0.42]	[0.05;0.33]
9	[0.02;0.20]	[0.03;0.19]
10	[0.13;0.30]	[0.14;0.23]
11	[0.05;0.09]	[0.05;0.07]
12	[0.19;1.81]	[0.22;1.73]
13	[0.44;3.50]	[0.36;2.91]
14	[1.15;13.28]	[1.60;11.56]
15	[13.99;253.8]	[18.85;316]
16	[0.00;0.03]	[0.00;0.01]
17	[0.00;0.13]	[0.00;0.10]
18	[0.01;0.14]	[0.01;0.09]
19	[0.00;0.03]	[0.00;0.02]
20	[0.00;0.06]	[0.00;0.05]
21	[0.00;0.01]	[0.00;0.01]
22	[12.84;32.49]	[15.51;35.13]
23	[17.04;49.54]	[16.67;40.14]
24	[85.1;214]	[101.7;232.2]
25	[508.1;3432]	[733.2;3903]
26	[0.08;0.22]	[0.10;0.18]
27	[0.05;1.05]	[0.12;0.74]
28	[0.02;1.17]	[0.22;0.73]
29	[0.02;0.29]	[0.11;0.27]
30	[0.15;0.66]	[0.22;0.48]
31	[0.05;0.20]	[0.06;0.13]
32	[0.4;10]	[0.4;10]
33	[0;27]	[0;27]

Table 4-13. Features of breast cancer cases for the WPBC dataset: standard deviations.

Statistical	Means ± Standard Deviations

descriptor #	Dataset	Class 1	Class 2
1	46.93 ± 34.52	53.58 ± 34.91	25.56 ± 22.72
2	17.40 ± 3.17	17.11 ± 3.06	18.33 ± 3.36
3	22.30 ± 4.33	22.46 ± 4.51	21.75 ± 3.69
4	114.78 ± 21.43	112.81 ± 20.63	121.09 ± 22.91
5	969.09 ± 353.15	934.00 ± 331.98	1081.98 ± 397.26
6	0.10 ± 0.01	0.10 ± 0.01	0.10 ± 0.01
7	0.14 ± 0.05	0.14 ± 0.05	0.14 ± 0.04
8	0.15 ± 0.07	0.15 ± 0.07	0.16 ± 0.06
9	0.08 ± 0.03	0.08 ± 0.03	0.09 ± 0.03
10	0.19 ± 0.02	0.19 ± 0.02	0.18 ± 0.02
11	0.06 ± 0.00	0.06 ± 0.00	0.06 ± 0.00
12	0.60 ± 0.30	0.58 ± 0.31	0.66 ± 0.30
13	1.27 ± 0.52	1.29 ± 0.55	1.20 ± 0.42
14	4.25 ± 2.18	4.11 ± 2.16	4.73 ± 2.21
15	70.29 ± 48.01	66.66 ± 45.82	81.96 ± 53.35
16	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
17	0.03 ± 0.01	0.03 ± 0.01	0.03 ± 0.01
18	0.04 ± 0.02	0.04 ± 0.02	0.03 ± 0.01
19	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
20	0.02 ± 0.00	0.02 ± 0.00	0.01 ± 0.00
21	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
22	20.99 ± 4.24	20.46 ± 3.96	22.67 ± 4.70
23	30.18 ± 6.06	30.35 ± 6.22	29.62 ± 5.55
24	140.13 ± 28.82	136.65 ± 26.79	151.33 ± 32.41
25	1401.75 ± 587.04	1329.02 ± 527.86	1635.76 ± 703.14
26	0.14 ± 0.02	0.14 ± 0.02	0.14 ± 0.01
27	0.36 ± 0.16	0.36 ± 0.17	0.35 ± 0.13
28	0.43 ± 0.17	0.43 ± 0.18	0.44 ± 0.14
29	0.17 ± 0.04	0.17 ± 0.04	0.18 ± 0.03
30	0.32 ± 0.07	0.32 ± 0.07	0.31 ± 0.06
31	0.09 ± 0.02	0.09 ± 0.02	0.08 ± 0.01
32	2.86 ± 1.95	2.67 ± 1.89	3.47 ± 2.02
33	3.21 ± 5.47	2.69 ± 5.21	4.86 ± 6.01

4.2.4 Dermatology database

This database contains the data that pertains to patients with differential diagnosis of erythemato-squamous diseases. The diseases in this group are psoriasis, seboreic dermatitis, lichen planus, pityriasis rosea, cronic dermatitis, and pityriasis rubra pilaris. The differential diagnosis of erythemato-squamous diseases is a real problem in dermatology because they share many clinical and histopathological features.

This database contains 34 attributes, 33 of which are continuously valued and one that is nominal. Patients were first evaluated clinically for 12 features. Afterwards, skin samples were taken for the evaluation of 22 histopathological features. The values of the histopathological features are determined by an analysis of the samples under a microscope. The family history feature has a value of 1 if any of these diseases has been observed in the family, and a value of 0 otherwise. Every other feature (clinical and histopathological) was given a value within a range of 0 to 3. Here the value of 0 indicates that the feature was not present, 3 indicates the largest amount possible, and values of either 1 or 2 indicate relative intermediate values. Table 4-14, Table 4-15, Table 4-16, Table 4-17, Table 4-18, Table 4-19, Table 4-20, Table 4-21 and Table 4-22 show the class distribution, the range, and the mean and standard deviation of each of the features in each class.

Table 4-14. Class distribution.

	Class Distribution	
Class code	Class name	Number of instances
1	Psoriasis	112
2	seboreic dermatitis	61
3	lichen planus	72
4	pityriasis rosea	49
5	cronic dermatitis	52
6	pityriasis rubra pilaris	20

Table 4-15. Clinical attributes of the dermatology database: range of values. Classes 1-3.

	Clinical		Range of values	
#	Attribute	Class 1	Class 2	Class 3
1	Erythema	[1;3]	[0;3]	[0;3]
2	Scaling	[1;3]	[1;3]	[0;3]
3	definite borders	[0;2]	[1;3]	[0;3]
4	Itching	[0;3]	[0;3]	[0;3]
5	Koebner phenomenon	[0;2]	[0;3]	[0;3]
6	polygonal papules	[0;0]	[0;0]	[0;3]
7	follicular papules	[0;1]	[0;2]	[0;0]
8	oral mucosal involvement	[0;0]	[0;0]	[0;3]
9	knee and elbow involvement	[0;1]	[0;3]	[0;2]
10	scalp involvement	[0;2]	[0;3]	[0;1]
11	family history	[0;1]	[0;1]	[0;1]
34	Age	[0;2]	[0;2]	[2;3]

Table 4-16. Clinical Attributes of the dermatology database: range of values. Classes 4-6.

#	Clinical		Range of values	
	Attribute	Class 4	Class 5	Class 6
1	erythema	[0;3]	[1;3]	[1;3]
2	scaling	[0;3]	[1;2]	[1;2]
3	definite borders	[0;3]	[0;2]	[0;2]
4	itching	[0;3]	[0;3]	[0;2]
5	koebner phenomenon	[0;0]	[0;3]	[0;0]
6	polygonal papules	[0;0]	[0;0]	[0;0]
7	follicular papules	[0;2]	[0;0]	[1;3]
8	oral mucosal involvement	[0;0]	[0;0]	[0;0]
9	knee and elbow	[0;1]	[0;0]	[0;3]

	involvement			
10	scalp involvement	[0;0]	[0;0]	[0;2]
11	family history	[0;0]	[0;0]	[0;1]
34	Age	[0;1]	[0;0]	[0;1]

Table 4-17. Histopathological attributes of the dermatology database: range of values. Classes 1-3.

#	Histopathological		Range of values	
	Attribute	Class 1	Class 2	Class 3
12	melanin incontinence	[0;0]	[0;0]	[0;3]
13	eosinophils in the infiltrate	[0;2]	[0;2]	[0;2]
14	PNL infiltrate	[0;3]	[0;3]	[0;0]
15	fibrosis of the papillary dermis	[0;0]	[0;0]	[0;2]
16	exocytosis	[0;3]	[0;2]	[0;3]
17	acanthosis	[0;3]	[0;3]	[0;2]
18	hyperkeratosis	[0;3]	[0;3]	[0;3]
19	parakeratosis	[0;3]	[0;3]	[0;3]
20	clubbing of the rete ridges	[0;0]	[0;3]	[0;0]
21	elongation of the rete ridges	[0;2]	[1;3]	[0;0]
22	thinning of the suprapapillary epidermis	[0;1]	[0;3]	[0;0]
23	spongiform pustule	[0;2]	[0;3]	[0;0]
24	munro microabcess	[0;0]	[0;3]	[0;3]
25	focal hypergranulosis	[0;0]	[0;0]	[0;3]
26	disappearance of the granular layer	[0;0]	[0;3]	[0;2]
27	vacuolisation and damage of basal layer	[0;0]	[0;1]	[0;3]
28	spongiosis	[0;3]	[0;0]	[0;3]
29	saw-tooth appearance of retes	[0;0]	[0;0]	[0;3]
30	follicular horn plug	[0;1]	[0;0]	[0;1]
31	perifollicular parakeratosis	[0;1]	[0;0]	[0;0]
32	inflammatory monoluclear inflitrate	[0;3]	[0;3]	[0;3]
33	band-like infiltrate	[0;3]	[0;3]	[0;3]

Table 4-18. Histopathological attributes of the dermatology database: range of values. Classes 4-6.

#	Histopathological		Range of values	
	Attribute	Class 4	Class 5	Class 6
12	melanin incontinence	[0;0]	[0;0]	[0;0]
13	eosinophils in the	[0;1]	[0;1]	[0;0]

	infiltrate			
14	PNL infiltrate	[0;0]	[0;1]	[0;1]
15	fibrosis of the papillary dermis	[1;3]	[0;0]	[0;0]
16	exocytosis	[0;2]	[0;3]	[0;3]
17	acanthosis	[1;3]	[0;2]	[1;3]
18	hyperkeratosis	[0;2]	[0;2]	[0;2]
19	parakeratosis	[0;2]	[0;2]	[0;2]
20	clubbing of the rete ridges	[0;2]	[0;0]	[0;1]
21	elongation of the rete ridges	[0;3]	[0;0]	[0;1]
22	thinning of the suprapapillary epidermis	[0;1]	[0;0]	[0;0]
23	spongiform pustule	[0;1]	[0;0]	[0;1]
24	munro microabcess	[0;0]	[0;1]	[0;0]
25	focal hypergranulosis	[0;0]	[0;0]	[0;1]
26	disappearance of the granular layer	[0;0]	[0;2]	[0;0]
27	vacuolisation and damage of basal layer	[0;0]	[0;0]	[0;0]
28	spongiosis	[0;3]	[0;3]	[0;3]
29	saw-tooth appearance of retes	[0;0]	[0;1]	[0;0]
30	follicular hom plug	[0;1]	[0;0]	[0;3]
31	perifollicular parakeratosis	[0;0]	[0;0]	[1;3]
32	inflammatory monoluclear inflitrate	[0;3]	[0;3]	[0;3]
33	band-like infiltrate	[0;3]	[0;3]	[0;3]

Table 4-19. Clinical attributes of the dermatology database: means and standard deviations. Classes 1-3.

#	Clinical Attribute	Means ± Standard Deviations		
		Class 1	Class 2	Class 3
1	erythema	2.27 ± 0.60	2.28 ± 0.62	2.08 ± 0.59
2	scaling	2.06 ± 0.54	2.19 ± 0.62	1.62 ± 0.65
3	definite borders	0.95 ± 0.80	2.09 ± 0.58	2.09 ± 0.67
4	itching	1.62 ± 0.95	0.94 ± 1.08	2.27 ± 0.79
5	koebner phenomenon	0.03 ± 0.25	0.66 ± 0.87	1.34 ± 1.05
6	polygonal papules	0 0	0 0	2.27 ± 0.69
7	follicular papules	0.01 ± 0.12	0.03 ± 0.22	0 0
8	oral mucosal involvement	0 0	0 0	1.91 ± 0.76
9	knee and elbow involvement	0.06 ± 0.24	1.63 ± 1.00	0.02 ± 0.23
10	scalp	0.11 ± 0.41	1.52 ± 0.97	0.02 ± 0.16

	involvement			
11	family history	0.04 ± 0.21	0.28 ± 0.45	0.01 ± 0.11
34	Age	0.03 ± 0.25	0.02 ± 0.21	2.72 ± 0.45

Table 4-20. Clinical attributes of the dermatology database: means and standard deviations. Classes 4-6.

#	Clinical Attribute	Means ± Standard Deviations		
		Class 4	Class 5	Class 6
1	Erythema	1.50 ± 0.67	1.89 ± 0.58	2.05 ± 0.51
2	Scaling	1.13 ± 0.62	1.51 ± 0.50	1.75 ± 0.44
3	definite borders	0.84 ± 0.89	1.18 ± 0.72	1.05 ± 0.75
4	Itching	1.88 ± 1.04	0.46 ± 0.76	0.49 ± 0.60
5	Koebner phenomenon	0	1.18 ± 0.80	0
6	polygonal papules	0	0	0
7	Follicular papules	0.23 ± 0.54	0	2.20 ± 0.61
8	oral mucosal involvement	0	0	0
9	knee and elbow involvement	0.03 ± 0.19	0	1.70 ± 0.80
10	scalp involvement	0	0	0.5 ± 0.82
11	family history	0	0	0.49 ± 0.51
34	Age	0.01 ± 0.13	0	0.05 ± 0.22

Table 4-21. Histopathological attributes of the dermatology database: means and standard deviations. Classes 1-3.

#	Histopathological Attribute	Means ± Standard Deviations		
		Class 1	Class 2	Class 3
12	melanin incontinence	0	0	2.05 ± 0.66
13	eosinophils in the infiltrate	0.45 ± 0.67	0.03 ± 0.22	0.16 ± 0.44
14	PNL infiltrate	1.08 ± 0.82	1.11 ± 0.90	0
15	fibrosis of the papillary dermis	0	0	0.05 ± 0.33
16	exocytosis	2.19 ± 0.70	0.26 ± 0.62	2.26 ± 0.67
17	acanthosis	1.77 ± 0.76	2.09 ± 0.62	2.11 ± 0.66
18	hyperkeratosis	0.21 ± 0.52	0.82 ± 0.90	0.29 ± 0.54
19	parakeratosis	0.98 ± 0.93	1.99 ± 0.64	1.20 ± 0.85
20	clubbing of the rete ridges	0	2.11 ± 0.73	0
21	elongation of the rete ridges	0.16 ± 0.48	2.25 ± 0.62	0
22	thinning of the suprapapillary epidermis	0.01 ± 0.12	2.05 ± 0.75	0
23	spongiform pustule	0.16 ± 0.45	0.85 ± 0.93	0
24	munro microabcess	0	1.15 ± 0.95	0.04 ± 0.35
25	focal hypergranulosis	0	0	1.98 ± 0.70

26	disappearance of the granular layer	0	1.19 ± 1.11	0.23 ± 0.63
27	vacuolisation and damage of basal layer	0	0.00 ± 0.09	2.30 ± 0.59
28	Spongiosis	2.16 ± 0.77	0	1.09 ± 1.17
29	saw-tooth appearance of retes	0	0	2.29 ± 0.63
30	follicular horn plug	0.01 ± 0.12	0	0.01 ± 0.11
31	perifollicular parakeratosis	0.01 ± 0.12	0	0
32	inflammatory monoluclear infiltrate	1.60 ± 0.78	1.86 ± 0.71	2.24 ± 0.59
33	band-like infiltrate	0	0	2.05 ± 0.66

Table 4-22. Histopathological attributes of the dermatology database: means and standard deviations. Classes 4-6.

#	Histopathological	Means ± Standard Deviations		
	Attribute	Class 4	Class 5	Class 6
12	melanin incontinence	0	0	0
13	eosinophils in the infiltrate	0.07 ± 0.26	0.06 ± 0.24	0
14	PNL infiltrate	0	0.12 ± 0.33	0.15 ± 0.36
15	fibrosis of the papillary dermis	2.28 ± 0.72	0	0
16	exocytosis	0.84 ± 0.77	2.04 ± 0.70	1.5 ± 0.82
17	acanthosis	2.24 ± 0.68	1.44 ± 0.64	1.65 ± 0.58
18	hyperkeratosis	0.69 ± 0.80	0.30 ± 0.58	0.80 ± 0.61
19	parakeratosis	0.75 ± 0.86	0.75 ± 0.63	1.25 ± 0.63
20	clubbing of the rete ridges	0.07 ± 0.33	0	0.1 ± 0.30
21	elongation of the rete ridges	1.88 ± 0.83	0	0.1 ± 0.30
22	thinning of the suprapapillary epidermis	0.01 ± 0.13	0	0
23	spongiform pustule	0.01 ± 0.13	0	0.05 ± 0.22
24	munro microabcess	0	0.02 ± 0.14	0
25	focal hypergranulosis	0	0	0.05 ± 0.22
26	disappearance of the granular layer	0	0.38 ± 0.53	0
27	vacuolisation and damage of basal layer	0	0	0
28	spongiosis	0.34 ± 0.71	1.95 ± 0.70	1.2 ± 1.00
29	saw-tooth appearance of retes	0	0.02 ± 0.14	0
30	follicular horn plug	0.01 ± 0.13	0	1.75 ± 0.85
31	perifollicular parakeratosis	0	0	2.05 ± 0.60
32	inflammatory	1.82 ± 0.75	1.77 ± 0.62	1.60 ± 0.68

	monoluclear infiltrate			
33	band-like infiltrate	0	0	0

4.2.5 Glass Identification database

The study of classification of types of glass was motivated by criminological investigation. At the scene of the crime, the glass left can be used as evidence if it is correctly identified. The database contains 9 continuously valued attributes describing the chemical constituents of the glass. The glass originates from various objects: building window, vehicle window, container, tableware and headlamp. In terms of chemical constituents the glass can be divided into 3 classes (Table 4-23). Table 4-24 and Table 4-25 show the range of values of features, the means and the standard deviations of features in each class.

Table 4-23. Class distribution.

	Class Distribution	
Class code	Class name	Number of instances
1	Float processed	87
2	Non float processed	76
3	Non-window glass	51

Table 4-24. Attributes of the glass identification database: range of values.

	Attribute	Range of values		
#		Class 1	Class 2	Class 3
1	RI: refractive index	[1.51;1.52]	[1.51;1.53]	[1.51;1.52]
2	Na: Sodium	[12.16;14.77]	[10.73;14.86]	[11.03;17.38]
3	Mg: Magnesium	[2.71;4.49]	[0;3.98]	[0;3.34]
4	Al: Aluminum	[0.29;1.76]	[0.56;2.12]	[0.34;3.5]
5	Si: Silicon	[71.35;73.7]	[69.81;74.45]	[69.89;75.41]
6	K: Potassium	[0;0.69]	[0;1.1]	[0;6.21]
7	Ca: Calcium	[7.78;10.17]	[7.08;16.19]	[5.43;12.5]
8	Ba: Barium	[0;0.69]	[0;3.15]	[0;2.88]
9	Fe: Iron	[0;0.37]	[0;0.35]	[0;0.51]

Table 4-25. Attributes of the glass identification database: means and standard deviations.

#	Attribute	Means ± Standard Deviations		
		Class 1	Class 2	Class 3
1	RI: refractive index	1.51 0.00	1.51 ± 0.00	1.51 ± 0.00
2	Na: Sodium	13.28 ± 0.50	13.11 ± 0.66	14.06 ± 1.06
3	Mg: Magnesium	3.55 ± 0.23	3.00 ± 1.21	0.73 ± 1.10
4	Al: Aluminum	1.17 ± 0.28	1.40 ± 0.31	1.96 ± 0.59
5	Si: Silicon	72.57 ± 0.56	72.59 ± 0.72	72.85 ± 1.08
6	K: Potassium	0.43 ± 0.21	0.52 ± 0.21	0.55 ± 1.28
7	Ca: Calcium	8.79 ± 0.54	9.07 ± 1.92	9.06 ± 1.58
8	Ba: Barium	0.01 ± 0.07	0.05 ± 0.36	0.63 ± 0.74
9	Fe: Iron	0.05 ± 0.09	0.07 ± 0.10	0.02 ± 0.08

4.2.6 Thyroid gland data

Five laboratory tests are used to try to predict whether a patient's thyroid belongs to the class euthyroidism, hypothyroidism or hyperthyroidism. The diagnosis (the class) was based on a complete medical record, including anamnesis, scan, etc. The database contains 5 continuously valued attributes. The class distribution is presented in Table 4-26. Table 4-27 and Table 4-28 show range of values of features, means and standard deviations of features in each class.

Table 4-26. Class distribution.

Class Distribution		
Class code	Class name	Number of instances
1	Euthyroidism	150
2	Hyperthyroidism	35
3	Hypothyroidism	30

Table 4-27. Attributes of the glass identification database: range of values.

#	Attribute	Range of values		
		Class 1	Class 2	Class 3
1	T3-resin uptake test	[90;133]	[65;144]	[97;141]
2	Total Serum thyroxin	[4.2;16.1]	[11.1;25.3]	[0.5;6.8]
3	Total serum triiodothyronine	[0.4;3.1]	[1.6;10]	[0.2;2.5]
4	basal thyroid-stimulating hormone (TSH)	[0.3;3.7]	[0.1;1.8]	[1.2;56.4]
5	Maximal absolute difference of TSH value after injection of 200 micro grams of thyrotropin-releasing hormone	[-0.7;13.7]	[-0.6;0.6]	[1.4;56.3]

Table 4-28. Attributes of the glass identification database: means and standard deviations.

#	Attribute	Means ± Standard Deviations		
		Class 1	Class 2	Class 3
1	T3-resin uptake test	95.28 ± 18.76	121.70 ± 11.05	110.51 ± 8.09
2	Total Serum thyroxin	17.74 ± 4.16	3.60 ± 1.75	9.19 ± 2.04
3	Total serum triiodothyronine	4.26 ± 2.25	1.06 ± 0.55	1.73 ± 0.47
4	basal thyroid-stimulating hormone (TSH)	0.97 ± 0.40	12.91 ± 12.38	1.31 ± 0.49
5	Maximal absolute difference of TSH value after injection of 200 micro grams of thyrotropin-releasing hormone	-0.02 ± 0.26	17.53 ± 15.50	2.51 ± 1.97

4.3 Web Pages dataset

4.3.1 Description of the data structure

The Web pages dataset constitutes a collection of Web Pages related to various topics [3]. This set was created from the categories of Open Directory Project [2]. An internal concept of the data structure was based on thematic partitioning. In the first subset, the pages were taken from the categories significantly different in the content (e.g. *Top| Sports| Cycling| Clubs and Teams*; *Top| Business| Transportation and Logistics| Aviation| Airports*; *Top| Health| Medicine| Hospitals| North America| Canada*; etc.). Yet, in the second subset the categories of pages were somehow similar thematically to each other – they partially overlap because of having the same directory root to a certain depth of the tree (e.g. *Top| Sports| WinterSports| Curling*; *Top| Sports| WinterSports| Snowboarding*). The categories contain from about 20–100 web pages up to 200–300 pages. The structure of the dataset is presented in Fig. 4-1 and 4-2, and the details of each of category are in Table 4-29 and Table 4-30.

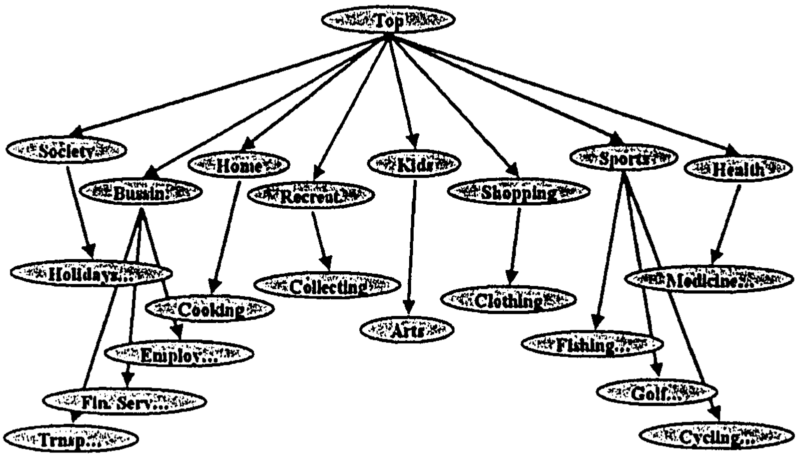


Figure 4-1. First three levels of thematically dissimilar (non-overlapping) categories.

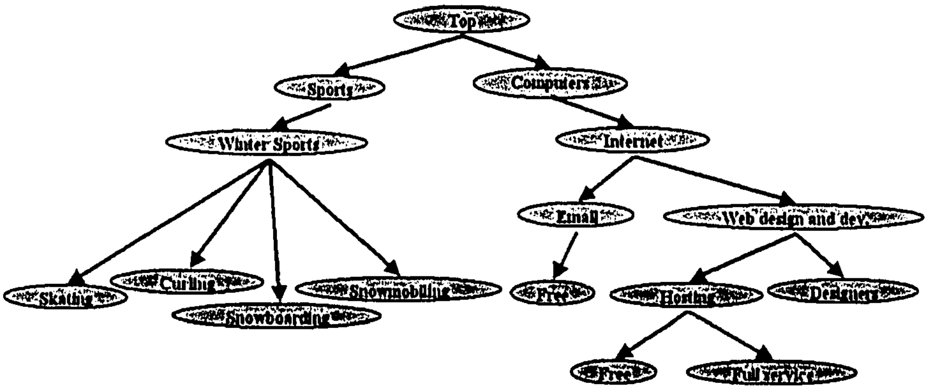


Figure 4-2. Thematically similar (overlapping) categories.

Table 4-29. Thematically dissimilar (non-overlapping) categories.

#	Name	Code	Cardinality	Open Directory Project Path
1	*Clothing*	*Cloth*	*17*	*Top| Shopping| Clothing*
2	*Kids art*	*Kart*	*24*	*Top| Kids and Teens| Arts*
3	*Collecting*	*Coll*	*27*	*Top| Recreation| Collecting*
4	*Cooking*	*Cook*	*34*	*Top| Home| Cooking*
5	*Fishing*	*Fish*	*44*	*Top| Sports| Fishing| Personal Pages*
6	*Halloween*	*Hall*	*48*	*Top| Society| Holidays| Halloween*
7	*Golf*	*Golf*	*49*	Top| Sports| Golf| Resources
8	*Careers*	*Carr*	*55*	*Top| Business| Employment| Careers*
9	*Gymnastics*	*Gymn*	*91*	*Top| Sports| Artistic| Clubs and Schools*
10	*Plants*	*Plan*	*202*	*Top| Home| Gardens| Plants*
11	*Mortgages*	*Mart*	*205*	*Top| Business| Financial Services| Mortgages*
12	*Airports*	*Airp*	*242*	*Top| Business| Transportation and Logistics| Aviation| Airports*
13	*Hospitals*	*Hosp*	*298*	*Top| Health| Medicine| Hospitals| North America| United States*
14	*Cycling*	*Cycl*	*300*	*Top| Sports| Cycling| Clubs and Teams*

Table 4-30. Thematically similar (overlapping) categories.

#	Name	Code	Cardinality	Open Directory Project Path
1	*Snowboarding*	*Snbd*	*60*	*Top| Sports| Winter Sports| Snowboarding*
2	*Snowmobiling*	*Snmb*	*60*	*Top| Sports| Winter Sports| Snowmobiling*
3	*Skating*	*Skat*	*62*	*Top| Sports| Winter Sports| Skating*
4	*Curling*	*Curl*	*64*	*Top| Sports| Winter Sports| Curling*
5	*Email*	*Mail*	*157*	*Top| Computers| Internet| E-mail| Free*
6	*Web hosting*	*Host*	*165*	*Top| Computers| Internet| Web Design and Development| Hosting| Free*
7	*Web design*	*Webd*	*180*	*Top| Computers| Internet| Web Design and Development| Designers| Full Service*

Additionally, using categories retrieved in the data collecting process there was formed an arbitrary dataset revealing similarity at different levels of hierarchy. In Fig. 4-3 there is presented the dataset with the schematic diagram of all its categories. In this dataset there exist 6 categories ('Curling', 'Fishing', 'Halloween', 'Golf', 'Skating', 'Snowboarding') gathering 326 web pages.

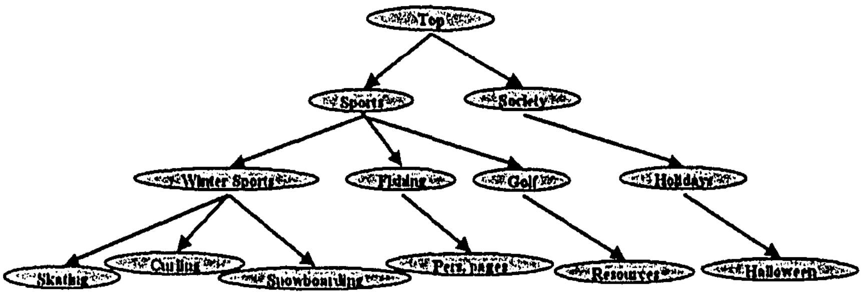


Figure 4-3. The dataset with nested (internal) categories, 326 pages in 6 categories.

Table 4-31. The number of pages in each category of dataset from Fig. 4-3.

	Curling	Fishing	Golf	Halloween	Skating	Snow- Boarding
Cardinality	64	44	49	48	62	59

4.3.2 Feature selection and datasets

For the features extracted from the Web pages, keywords were chosen from the title and the HTML metatags (keywords provided by the page designer in special <*meta*> tags). According to Pierre [4], rich meta-information constitutes the most authoritative source of information about the content of the web page and provides the most accurate results. From cleaned, stemmed keywords and elimination of the stop words (most frequently appearing keywords e.g. a, the, it, that, etc which are too frequent to be helpful in discrimination between pages), the feature vectors were created. Two weighting scheme were applied giving different weights to the corresponding keywords: (i) relative frequencies, and (ii) tf-idf approach. In this manner, the feature space based on a vector space model was constructed, in which every Web Page was represented by a normalized to [0,1] interval feature vector and each keyword is associated with a weight, which is greater if the descriptive ability of the keyword is larger.

Such collection poses a significant challenge in detecting the underlying structure of the data via clustering and classification. From the pre-processed pages in the several above-mentioned categories, the proper datasets were created (Table 4-32).

Table 4-32. Datasets constructed with the available categories.

#	Categories	Dimension	Cardinality (# of Pages)	# of Categories	Thematic overlapping
1	*Fishing; Halloween*	*1128*	*92*	*2*	*Non-overlapping*
2	*Fishing; Halloween; Golf*	*1849*	*141*	*3*	*Non-overlapping*
3	*Fishing; Halloween; Careers*	*1627*	*147*	*3*	*Non-overlapping*
4	*Fishing; Halloween; Gymnastics*	*1833*	*183*	*3*	*Non-overlapping*
5	*Fishing; Halloween; Career; Gymnastics;*	*2272*	*238*	*4*	*Non-overlapping*
6	*Clothing; Collecting; Cooking; Fishing; Golf; Halloween; Kids Art; Snowboarding*	*3660*	*302*	*8*	*Non-overlapping*
7	*Airports; Cycling; Plants*	*5450*	*744*	*3*	*Non-overlapping*
8	*Snowboarding; Snowmobiling; Skating; Curling*	*2002*	*244*	*4*	Overlapping
9	*Email; Web hosting; Web design*	*3249*	*501*	*3*	*Overlapping*

In order to predict how fast the dimensionality of the vectors increases, the relationship between dimensionality, the number of categories and the number of WWW pages was examined. Fig. 4-4 presents the graph. There exists a noticeable relationship between dimensionality and the number of categories within the data. The more categories there are in the data, the larger the dimension of the dataset.

The relationship between dimensionality and the number of pages seems to be weaker. This is because WWW pages in the same categories will most likely have the same keywords, so that after reaching a certain level of 'saturation' the dimension will not grow as fast as in the case of pages from different categories.

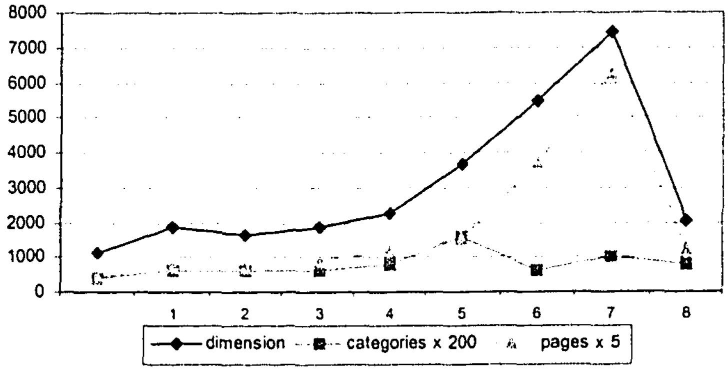


Figure 4-4. Relationships between dimensionality, the number of categories and the number of Web pages.

4.4 Bibliography

1. Blake, C. L, Merz, C. J, UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html], Irvine, CA: University of California, Department of Information and Computer Science, 1998.
2. Brzeminski, P., Pedrycz, W., Textual-based Fuzzy Clustering of Web Documents, *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, Vol. 12(6), 2004, pp.715-744.
3. ODP, Open Directory Project, http://domz.org.
4. Pierre, J.M., On the automated classification of Web Sites, *Linkoping Electronic Articles in Computer and Information Science*, Vol. 6(0), 2001.

5. Algorithms – Theory, Derivations and Illustration on Synthetic Datasets

5.1 Knowledge discovery with fuzzy clustering

Clustering is a fundamental method of unsupervised learning. More generally it can be referred to as a search for structure in data [1]. The data here can be virtually any data drawn from a physical process. The search enables the computers to transmit its findings to the researcher in usable forms. These forms depend not only on the data but also on the methods and models used, and upon the structure we expect to find. While data possibly carries the information about the process generating it, the structure is the fashion in which this information can be organized so that relationships between variables in the process can be identified. Representations of the organized structure depend on the data, the method of search and the model used. In terms of information, the data contains the information, the search recognizes it and the structure represents it.

More explicitly, the clustering seeks for data structure in datasets and generates a partitioning of a dataset $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ where $\mathbf{x}_k \in \mathbf{R}^p$ and $k = 1,2,...,N$ into $c \in \{2,...,N-1\}$ clusters (we do not consider trivial cases when $c = 1$ or $c = N$). Vectors are the representation of patterns from the dataset and are produced as a result of feature extraction and selection.

Classically, c clusters were disjoint, collectively exhaustive subsets of X. Let us consider a following problem. What is the possible partitioning of a set of three fruit $X=\{\mathbf{x}_1=peach,\ \mathbf{x}_2=plum,\ \mathbf{x}_3=nectarine\}$ into 2 clusters? Let us denote A_1 as the subset containing a peach, then $\mathbf{x}_1 \in A_1$. a plum, which is a different fruit and would belong to another subset, $\mathbf{x}_2 \in A_2$. Both sets exhibit different features of the contained objects so the following is needed: $A_1 \cap A_2 = \varnothing$. Now, we consider the assignment of a nectarine (a peach-plum hybrid). Clearly, $\mathbf{x}_3 \notin A_1 \wedge \mathbf{x}_3 \notin A_2$ and we encountered quite a disappointing mathematical constraint of this model resulting in the inability of a natural assignment $\mathbf{x}_3$ (a nectarine) to any of the subsets. More formally it means a failure of generating a 2-elements partition of the set X reflecting the features of objects from X. This limitation can be successfully eliminated by fuzzy sets. We can imagine a function $u_1 : X \to [0,1]$, whose values $u_1(x)$ give a grade of membership of x in the fuzzy set u_1. This is exactly what we needed in our example. A grade of membership of a nectarine can be expressed as $A_1(\mathbf{x}_3)$=0.51 and $A_2(\mathbf{x}_3)$=0.49 denoting that a nectarine is to the same extent as the peach and the plum.

A very convenient form of representation of a partitioning of X can be denoted as a fuzzy c-partition. X is a finite set, V_{cN} is the set of real matrices, c is an integer $2 \leq c < N$, $u_{ik} = u_i(\mathbf{x}_k)$. Then the fuzzy c-partition of X is the set:

$$\mathbf{M}_{fc} = \left\{\mathbf{U} \in \mathbf{V}_{cN} \mid u_{ik} \in [0,1] \forall i,k; \sum_{i=1}^{c} u_{ik} = 1 \forall k; 0 < \sum_{k=1}^{N} u_{ik} < N \forall i\right\} \tag{1}$$

Now, an example solution to the problem of finding a 2-partition of X may look as what follows:

$$U = \begin{array}{c} \\ \\ \end{array}\begin{array}{ccc} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 \\ \end{array}$$

$$U = \begin{bmatrix} 0.92 & 0.11 & 0.51 \\ 0.08 & 0.89 & 0.49 \end{bmatrix}$$

A membership function of $\mathbf{x}_1$ is equal to 0.92 in the first cluster and this means that $\mathbf{x}_1$ is more associated with cluster one (first row). Similarly, $\mathbf{x}_2$ clearly belongs to the second cluster. However, $\mathbf{x}_3$ (nectarine) exhibits features of both clusters, and this is apparent in its membership functions – it is equally similar to either one. For more detailed considerations the reader is referred to Bezdek [1].

5.2 Non-relational clustering algorithms

5.2.1 Fuzzy C-Means (FCM)

As we pointed out in the previous section, generating partitioning of a given dataset may be not trivial. The similarities and dissimilarities between objects easily recognized by humans might be challenging to grasp in terms of the features describing the objects. At this point we assume that the "best" features were extracted and each pattern is represented as a feature vector. The process of clustering can now be started. When clustering, it is necessary to distinguish between the different partitions and choose the desired one. In other words we need to quantify the quality of the produced clusters. The optimal solution should minimize the error of clustering which can be expressed as the minimization of the distances between the patterns, and the cluster centers to which these patterns belong. If the assumed representation of the clustering is the matrix $\mathbf{U} \in \mathbf{M}_{fc}$ as described above (1), each distance can be weighted by the elements of $\mathbf{U}$ and the expression to be minimized may look as follows:

$$J_m(\mathbf{U}, \mathbf{v}) = \sum_{k=1}^{N} \sum_{i=1}^{c} (u_{ik})^m (d_{ik})^2 \tag{2}$$

In fact this is the objective function of FCM algorithm [1] guiding the clustering process. In the objective function $\mathbf{U} \in \mathbf{M}_{fc}$ is the fuzzy c-partition of X; $u_{ik} \in [0,1]$ specifies the degree of membership of pattern $k = 1,2,\ldots,N$ in the cluster $i = 1,\ldots,c$; $\mathbf{v}$ is the set of prototypes (clusters centers) $\mathbf{v} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c\}$ with $\mathbf{v}_i \in \mathbf{R}^p$; d_{ik} being the distance between each data vector $\mathbf{x}_k$ and a fuzzy prototype $\mathbf{v}_i$:

$$d_{ik} = \|\mathbf{x}_k - \mathbf{v}_i\| \tag{3}$$

Notation of $\|\bullet\|$ refers to a norm, in particular it is assumed as L2 (Euclidean norm); $m \in (1,+\infty)$ is the fuzzification coefficient.

As it follows from (2) each term of J_m is proportional to $(d_{ik})^2$, thus J_m is a squared error-clustering criterion. The problem of generating partitions of X was reduced to finding in iterative steps the optimal partition matrix $\mathbf{U}$. Optimum in this case means with minimal squared error clustering criterion. The algorithm is performed in subsequent iterations (Table 5-1).

Table 5-1. The overall scheme of the Fuzzy C-Means (FCM).

I. Initialization:

Fix c, $2 \le c < N$ *; fix m,* $1 < m < \infty$*; Initialize randomly the partition matrix* $\mathbf{U} \in \mathbf{M}_{fc}$*. Proceed with* $l = 0,1,\ldots$

II. Computations of prototypes:

Compute the c prototypes $\{\mathbf{v}^{(l)}{}_i\}$ *with* $\mathbf{U}^{(l)}$ *according to:*

$$\mathbf{v}_i = \sum_{k=1}^{N}(u_{ik})^m \mathbf{x}_k \Big/ \sum_{k=1}^{N}(u_{ik})^m \ \forall i \qquad (4)$$

III. Computations of entries of partition matrix:

Compute $\mathbf{U}^{(l)}$ *using* $\{\mathbf{v}^{(l)}{}_i\}$ *if* $d_{ik} > 0$*:*

$$u_{ik} = 1 \Big/ \left[\sum_{j=1}^{c}\left(\frac{d_{ik}}{d_{jk}}\right)^{2/(m-1)}\right] \qquad (5)$$

Otherwise:

Set $u_{ik} = 0$ *and impose* $\sum_{i=1}^{c} u_{ik} = 1,\ \forall i \ \ 1 \le i \le c$ *that produces* $d_{ik} = 0$.

IV. Checking for termination condition or specifying maximal number of iterations:

Comparison in the convenient matrix norm if termination condition was reached $\left\|\mathbf{U}^{(l+1)} - \mathbf{U}^{(l)}\right\| \le \varepsilon$ *where* ε *is a very small positive constant or until* $l \le \max\_$ *iterations holds true.*

FCM is an excellent Data Mining tool for regular, distinguishable clusters of similar size. The algorithm computes the cluster centre using every pattern from the cluster. This feature reflects the mean behaviour and implies that FCM can be misguided in a noisy data environment. Another weak point of FCM is its sensitivity to the initial settings of the partition matrix. Depending on the different settings and the nature of the data, we can obtain different prototypes, and implicitly a different structure of clusters. In such cases it is advisable to repeat the experiments to get the best results. FCM allows convenient control of the "amount" of fuzziness in the partition matrix. This is done by the fuzzification parameter m. Increasing the value of m results in larger amount of fuzziness in the partition matrix – patterns tend to balance, equalize, the membership values in all clusters. Conversely, for $m \to 1$ the membership values of a pattern are increased and favour a single cluster. The standard value of m frequently assumed by researchers to be $m=2$.

5.2.2 Fuzzy C-Median (FCMED)

The FCM algorithm is based on the mean computations when calculating prototypes. As mean statistics can be easily influenced by even a small number of outliers, FCM cannot be successfully applied to noisy data. In this context it seems appropriate to suggest more sophisticated methods to deal efficiently with noisy, incomplete or distorted data. One of the possible approaches is a robust statistics that is more resistant to outliers. Kersten [7,8,9] suggested a Fuzzy C-Median algorithm founded on two robust statistics: the median estimating the center of the data and the median absolute deviation from the median (MAD). The fuzzy median is a robust statistic, which is able to accept almost 50% of outliers before it will lose its ability to generate meaningful results. How is the fuzzy median constructed? The process begins with the definition of median, which does not depend upon the ordering of the samples:

$$\min_{m_i \in \mathbf{R}} \sum_{k=1}^{N} \left| \mathrm{x}_k - m \right| \tag{6}$$

This definition is amenable to generalization for fuzzy sets. For u_{ik} being the membership of x_k in $i\text{-}th$ cluster $1 \le i \le c$, m solves:

$$\min_{m_i \in \mathbf{R}} \sum_{k=1}^{N} u_{\mathrm{ik}} \left| \mathrm{x}_k - m_i \right| \tag{7}$$

The derivative of this expression exists and is given by:

$$\sum_{k=1}^{N} u_{\mathrm{ik}} \operatorname{sgn}(\mathrm{x}_k - m_i) = 0 \tag{8}$$

Prototypes (v_i) for FCMED are constructed by finding the fuzzy median for each cluster. Each prototype (fuzzy median m_i) v_i is sought using the membership functions given by u_{ik} and the $p\text{-}th$ component x_k – can be computed from (8).

The FCMED algorithm is constructed on the general scheme of FCM. While both algorithms perform similarly for clean data, presence of outliers causes problems in convergence for FCM. FCMED handles this situation better. The possible drawback of FCMED is caused by non-continuity of the functional (8) and can make the search for medians troublesome. A numerical search is applied that may increase the computational complexity. For the purpose of this study the bisection method was used (applied initially by the author [9]). There exist some other methods that provide a trade-off between search cost and accuracy, and furthermore may reduce the search cost, e.g. the Remedian evaluation method [8].

FCMED forms a robust alternative to FCM with some computational penalty. The general flow of the algorithm is presented in Table 5-2.

Table 5-2. The overall scheme of the Fuzzy C-Median (FCMED).

I. Initialization:

Fix c, $2 \le c < N$ *; fix m,* $1 < m < \infty$*; Initialize randomly the partition matrix* $\mathbf{U} \in \mathbf{M}_{fc}$*; Choose* L_l *metric (Manhattan metric) for* d_{ik}*. Proceed with* $l = 0,1,...$

II. Computations of prototypes:

Compute c prototypes $\mathbf{v}_i \in \mathbf{R}^p$ $\{\mathbf{v}^{(l)}{}_i\}$ *by finding the fuzzy median with memberships* $u_{ik}{}^m$ *for each class. Each class prototype* $\mathbf{v}_i$ *is p-dimensional so* $\mathbf{v}_i(j)$ *must be found for each j={1,..., p}, using just the j-th component of* $\mathbf{x}_k$.

III. Computations of entries of partition matrix:

Compute $\mathbf{U}^{(l)}$ *using* $\{\mathbf{v}^{(l)}{}_i\}$ *if* $d_{ik} > 0$*:*

$$u_{ik} = 1 \Bigg/ \left[\sum\nolimits_{j=1}^{c} \left(\frac{d_{ik}}{d_{jk}} \right)^{2/(m-1)} \right]$$

Otherwise:

Proceed accordingly to Fuzzy C-Means (Table 5-1).

IV. Checking for termination condition or specifying maximal number of iterations:

Proceed accordingly to Fuzzy C-Means (Table 5-1).

5.2.3 Experimentation on synthetic data

The artificial data described in this section illustrate how the algorithms work. In particular this experiment compares the performance of FCM and FCMED in a noisy environment. The idea is to generate a regular structure from a distribution of well-known parameters and type, and to attempt to interfere with it using some noise. It is expected that there will be an observed continual deterioration of structure as more outliers are added to the dataset.

The original structure is more difficult to reveal by the method relying on mean statistics (FCM), which often converges to a non-optimal position. One way of overcoming the problem might be the application of more robust methods dealing efficiently with outliers (FCMED). It is expected that the robust FCMED would perform better with respect to FCM; more outliers exist in the dataset.

Initially, we constructed 3 clusters of 100 patterns per cluster, generated from normal distributions with the parameters, whose values are included in Table 5-3.

This structure was interfered with noise. 50 outliers from a heavy-tailed Cauchy distribution were additionally added to the initial dataset. The patterns are 4-dimensional. The 3rd and 4th feature are equal for all patterns to 0 so it is possible to present the structure on a plane (Fig. 5-1). The easily distinguishable groups of points are centered and surrounded by the outliers appearing in different locations.

Table 5-3. Mean and covariance matrices used to generate 3 clusters.

$N(\begin{bmatrix}2\\2\\0\\0\end{bmatrix},\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix})$	$N(\begin{bmatrix}-2\\-2\\0\\0\end{bmatrix},\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix})$	$N(\begin{bmatrix}6\\0\\0\\0\end{bmatrix},\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix})$

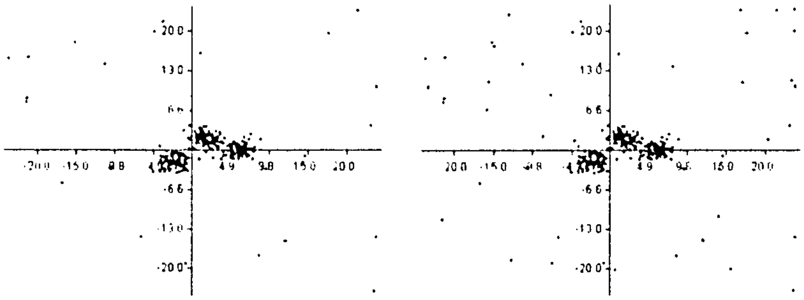


Figure 5-1. Three clusters from normal distribution and 25 outliers (left), 50 outliers (right).

As we can expect, in the dataset without outliers the prototypes obtained from the algorithms will be almost the same as the cluster centers calculated from data (a single mean vector for one cluster). The presence of noise influences the results of the algorithms in such a way that the original prototypes will change (move to another, less desired location). This "change" between cluster centers computed from the data and the prototypes obtained from the algorithms forms the criterion of quantifying the performance of the algorithms. This changing behaviour can be observed for the dataset without noise and after adding outliers to the dataset. The criterion of measuring the performance is the sum of distances between corresponding prototypes in their previous and new positions. The smaller the sum, the closer are the new prototypes to the original cluster centers and the better the resistance to noise (the algorithm is more robust).

Initially we recorded the results for three clusters. Knowing that FCMED is very sensitive to the fuzzifier m [8] we computed the sum of the distances for different parameters of m for both algorithms. The experiment was performed for (i) the undisturbed structure of clusters, (ii) adding 25 outliers and (iii) adding 50 outliers to the original structure. The sum of distances is recorded and presented in Fig. 5-2, Fig. 5-3 and Fig. 5-4. FCMED works similarly to FCM for smaller values of m but its performance deteriorates for m closer to value of 2. However FCMED is more stable than FCM in the dataset with outliers. The difference is considerable. The prototypes generated by FCMED seem to be more independent from the added outliers than the prototypes produced by FCM. It significantly outperforms FCM when noise appears in dataset (Fig. 5-2, Fig. 5-3 and Fig. 5-4).

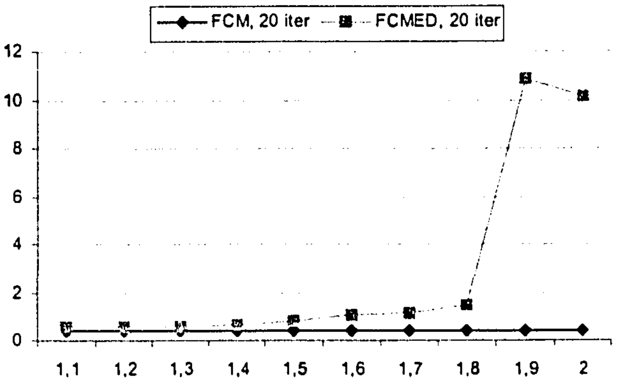


Figure 5-2. The sum of distances for the FCM and the FCMED for different values of m (3 clusters, no outliers).

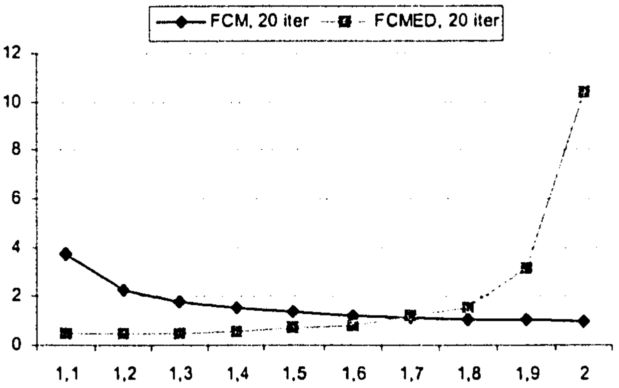


Figure 5-3. The sum of distances for the FCM and the FCMED for different values of m (3 clusters, 25 outliers).

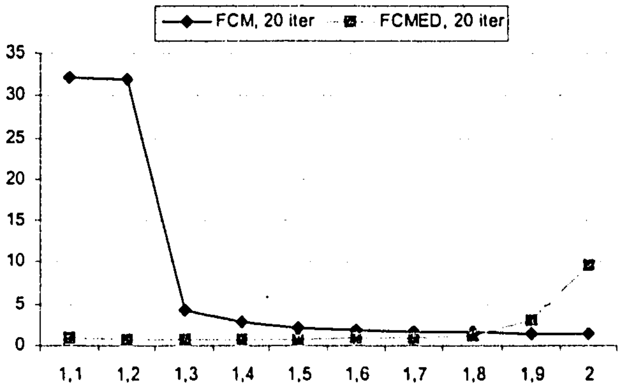


Figure 5-4. The sum of distances for the FCM and the FCMED for different values of m (3 clusters, 50 outliers).

The same experiment was run again using an increased number of clusters and outliers. In the new dataset there is 5 clusters and 100 outliers. The experiments were completed using a procedure similar to the previous one: (i) no outliers in dataset, (ii) 50 outliers and (iii) 100 outliers (Fig. 5-5). The results resemble the previous experiment but the effect is magnified. For the dataset with 100 outliers FCMED performs substantially better than FCM (for the smaller values of m).

Table 5-4. Mean and covariance matrices used to generate 5 clusters.

$$N(\begin{bmatrix}2\\2\\0\\0\end{bmatrix},\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}) \quad N(\begin{bmatrix}2\\-2\\0\\0\end{bmatrix},\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}) \quad N(\begin{bmatrix}-2\\-2\\0\\0\end{bmatrix},\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix})$$

$$N(\begin{bmatrix}-2\\2\\0\\0\end{bmatrix},\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}) \quad N(\begin{bmatrix}5\\0\\0\\0\end{bmatrix},\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix})$$

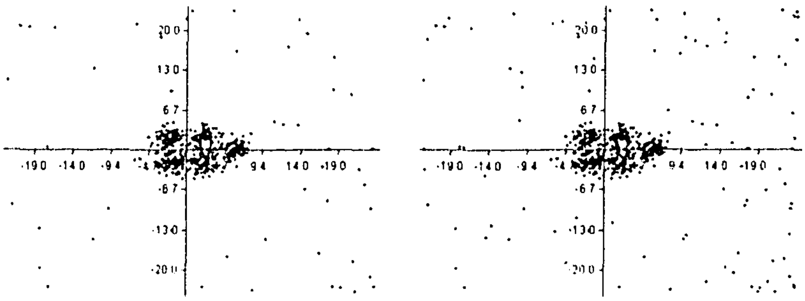


Figure 5-5. Three clusters from normal distribution and 50 outliers (left), 100 outliers (right).

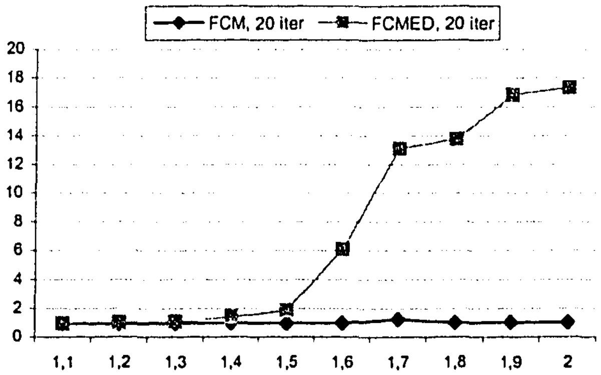


Figure 5-6. The sum of distances for the FCM and the FCMED for different values of m (5 clusters, no outliers).

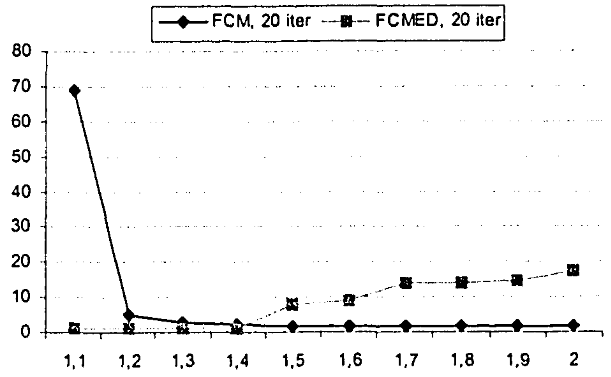


Figure 5-7. The sum of distances for the FCM and the FCMED for different values of m (5 clusters, 50 outliers).

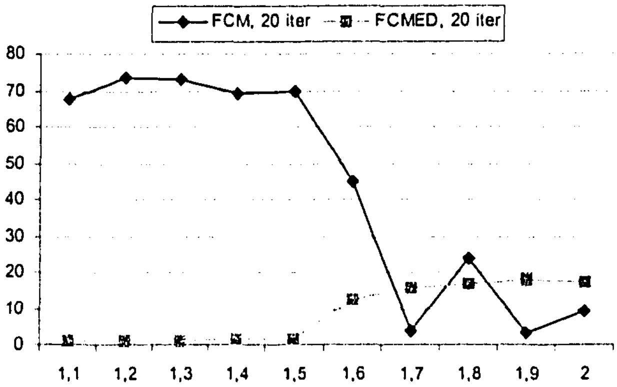


Figure 5-8. The sum of distances for the FCM and the FCMED for different values of m (5 clusters, 100 outliers).

These experiments reveal very important features of both algorithms. We could observe that even small presence of outliers may influence heavily the results of FCM (Fig. 5-3). In other words, it is possible that very few, large outliers may cause the production by the FCM the structure that does not conform to the physical state. The FCMED is more stable (Fig. 5-4 and Fig. 5-8) and the outliers are not influencing much its outcomes. This means it is more reliable method when dealing with unknown, unlabelled data.

5.3 Relational fuzzy clustering algorithms

5.3.1 Relational Fuzzy C-Means (RFCM)

The given set of patterns $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$ where $\mathbf{x}_k \in \mathbf{R}^p$ instead of being described by the numerical data, where a vector $\mathbf{x}_k$ gives measurements for every feature of a described pattern, may be represented by the numerical relational data. The numerical relational data describe the set of objects in less direct manner giving pair wise similarity (dissimilarity) between objects. The dissimilarities can be represented using a matrix $\mathbf{D}$, where $\mathbf{D}_{jk}, for\ 1 \leq j,k \leq N$ is the degree of dissimilarity between patterns. The matrix $\mathbf{D}$ has the following properties:

$$\mathbf{D}_{jk} \geq 0\ for 1 \leq j,k \leq N \tag{9a}$$

$$\mathbf{D}_{jk} = \mathbf{D}_{kj}\ for 1 \leq j,k \leq N \tag{9b}$$

$$\mathbf{D}_{jj} = 0\ for 1 \leq j,k \leq N \tag{9c}$$

The fuzzy c-means algorithms are methods, which in practice are very good at finding minimizing pairs of $(\mathbf{U}, \mathbf{v})$. Following the conditions imposed upon the matrix $\mathbf{U} \in \mathbf{M}_{fc}$, it is possible to restrict J_m in (2) to a surface in $(\mathbf{U}, \mathbf{v})$ space, which satisfies two properties: (i) $\mathbf{v}$ is a function of $\mathbf{U}$ on this surface; and (ii) this surface contains all optimal minimizing pairs of $(\mathbf{U}, \mathbf{v})$ of J_m. After some simplifications, the restricted J_m can be only the function of $\mathbf{U}$, written as:

$$K_m(\mathbf{U}) = \sum_{i=1}^{c} \left[\frac{\sum_{j=1}^{N} \sum_{k=1}^{N} (u_{ij}{}^{m} u_{ij}{}^{m} \mathbf{D}_{jk})}{(2\sum_{t=1}^{N} u_{it}{}^{m})} \right] \tag{10}$$

Where

$$\mathbf{D}_{jk} = (\|\mathbf{x}_j - \mathbf{x}_k\|)^2\ for\ 1 \leq j,k \leq N. \tag{11}$$

The minimization of K_m with usage of the relational data $\mathbf{D}_{jk}$ is equivalent to minimization of J_m with the numerical data provided that $\mathbf{D}_{jk}$ is derived from the Euclidean norm (11).
The general scheme of the algorithm is presented in Table 5-5.

Table 5-5. The overall scheme of the Relational Fuzzy C-Means (RFCM).

I. Initialization:

Fix c, $2 \leq c < N$ *; fix m,* $1 < m < \infty$*; Initialize randomly the partition matrix* $\mathbf{U} \in \mathbf{M}_{fc}$*; Proceed with* $l = 0,1,\ldots$

II. Computations of fuzzy prototypes:

Compute the c-mean vectors $\mathbf{v}_i = \mathbf{v}_i^{(l)}$ *using* $\mathbf{U} = \mathbf{U}^{(l)}$, *for* $1 \leq i \leq c$ *with:*

$$\mathbf{v}_i = \left(u_{i1}{}^{m}, u_{i2}{}^{m}, \ldots, u_{in}{}^{m}\right) \Big/ \sum_{k=1}^{N} (u_{ik}{}^{m}) \tag{12}$$

III. Computations of distances:

$$d_{ik} = (\mathbf{v}_i)_k - (\mathbf{v}_i{}'\mathbf{D}\mathbf{v}_i)/2\ for\ 1 \leq i \leq c,\ 1 \leq k \leq N \tag{13}$$

IV. Computations of entries of partition matrix:

Compute $\mathbf{U}^{(l)} = \mathbf{U}^{(l+1)}$. *If* $d_{ik} > 0$:

$$u_{ik} = 1 \Big/ \left(\sum_{j=1}^{c} \left[\frac{d_{ik}}{d_{jk}} \right]^{1/(m-1)} \right) \text{ for } 1 \le i \le c,\ 1 \le k \le N$$

Otherwise:
Proceed accordingly to Fuzzy C-Means (Table 5-1).

V. Checking for termination condition or specifying maximal number of iterations:
Proceed accordingly to Fuzzy C-Means (Table 5-1).

The assets of Relational Fuzzy C-Means are fairly obvious. It creates an alternative for clustering when there are not any numerical data available in a vector form but instead there exist a relational equivalent. Another benefit of the RFCM appears when the patterns' vectors have a very large dimensionality. The dissimilarity matrix can be computed prior running the algorithm that decreases computational complexity. This however is applicable for the datasets of smaller number of elements. While larger datasets are processed, the RFCM performs more slowly than FCM due to more costly distance computations. It is the third of the algorithm, in which the prototypes are multiplied by the dissimilarity matrix.

5.3.2 Non-Euclidean Relational Fuzzy C-Means (NERF)

The RFCM performs well on the numerical relational data, in which dissimilarity data **D** contains the squared Euclidian distances between each pair of the points – restrictions for the matrix **D**, (9a-b). The restriction of deriving relational data from the Euclidian norm is a strong limitation on the dissimilarity matrix. This makes RFCM inapplicable to most of relational clustering problems because there is no guarantee that arbitrary data provided by the physical process will conform to the Euclidean norm.

The problem of **D** not being Euclidean can be eliminated by applying a "beta-spreading" transformation [5]. The transformation chosen to convert a non-Euclidean matrix **D** into an Euclidean matrix $\mathbf{D}_\beta$ is the following:

$$\mathbf{D} = \mathbf{D}_0 \rightarrow \mathbf{D}_\beta = \mathbf{D} + \beta * (\mathbf{M} - \mathbf{I}) \qquad (14)$$

where β is a suitably chosen scalar, $\mathbf{I} \in \mathbf{R}^{N x N}$ is the identity matrix and $\mathbf{M} \in \mathbf{R}^{N x N}$ satisfies:

$$\mathbf{M}_{ij} = 1 \ \textit{for}\ 1 \le i,j \le N \qquad (15)$$

The optimization scheme of the Non-Euclidean Relational FCM (NERF) follows the RFCM flow with the only modification of incorporating the beta-transformation. The general scheme of the NERF is presented in Table 5-6.

Table 5-6. The overall scheme of the Non-Euclidean Relational Fuzzy C-Means (NERF).

I. Initialization:
Given the relational data **D** *satisfying (9a,b,c) fix c,* $2 \le c < N$ *; fix*

m, $1 < m < \infty$; Initialize $\beta = 0$; Initialize randomly the partition matrix $\mathbf{U} \in \mathbf{M}_{fc}$; Proceed with $l = 0,1,\ldots$.

II. Computations of fuzzy prototypes:

Compute the c-mean vectors $\mathbf{v}_i = \mathbf{v}_i^{(l)}$ using $\mathbf{U} = \mathbf{U}^{(l)}$, for $1 \le i \le c$ with:

$$\mathbf{v}_i = \left(u_{i1}^{\;m}, u_{i2}^{\;m}, \ldots, u_{in}^{\;m}\right) \Big/ \sum\nolimits_{k=1}^{N} (u_{ik}^{\;m})$$

III. Computations of distances:

$$d_{ik} = (\mathbf{v}_i)_k - (\mathbf{v}_i{}'\mathbf{D}\mathbf{v}_i)/2 \; \textit{for } 1 \le i \le c, 1 \le k \le N$$

If $d_{ik} < 0$ for any i and k, then:

Compute $\Delta\beta = \max\left\{-2 * d_{ik} / \left(\|\mathbf{v}_i - \mathbf{e}_k\|^2\right)\right\}$ *for* $1 \le i \le c,\; 1 \le k \le N$

Update $d_{ik} \leftarrow d_{ik} + (\Delta\beta/2) * \|\mathbf{v}_i - \mathbf{e}_k\|^2$ *for* $1 \le i \le c,\; 1 \le k \le N$

Update $\beta \leftarrow \beta + \Delta\beta$

IV. Computations of entries of partition matrix:

Compute $\mathbf{U}^{(l)} = \mathbf{U}^{(l+1)}$. *If* $d_{ik} > 0$*:*

$$u_{ik} = 1 \Big/ \left(\sum\nolimits_{j=1}^{c} \left[\frac{d_{ik}}{d_{jk}} \right]^{1/(m-1)} \right) \; \textit{for } 1 \le i \le c,\; 1 \le k \le N$$

Otherwise:

Proceed accordingly to Fuzzy C-Means (Table 5-1).

V. Checking for termination condition or specifying maximal number of iterations:

Proceed accordingly to Fuzzy C-Means (Table 5-1).

5.3.3 Experimentation on synthetic data

This experiment illustrates how the relational algorithms could be beneficial for data explorations. If we imagine that one or more components of **D** is missing we happen to deal with incomplete data. The incompleteness of data may arise as a result of missed observations, distorted data or simply an absence of complete relational knowledge about patterns. Aside from the origin of this problem, the algorithms for processing such data are desirable. In this section, the incomplete data will be simulated by elimination of some percentage of entries from **D**. The strategies for approximation of the missing values will be applied and experiment with the methods in such conditions will be performed. The RFCM as mentioned before is essentially a relational equivalent to the FCM and operates on dissimilarity matrix **D** derived from the Euclidean norm. What would happen if data were not Euclidean? The NERF algorithm could be used for such cases. This specific case focuses on the NERF because it is not correct to assume that the approximations will lead to a perfect recovery of the initial matrix **D** and as a result the RFCM is not applicable. To approximate the missing values from the matrix **D** the straightforward triangle inequality-based approximation (TIBA) [4] was used. When the similarities are distances besides conditions (9a, 9b and 9c), the following inequality holds true:

$$\mathbf{D}_{ij} \leq \mathbf{D}_{ik} + \mathbf{D}_{kj} \tag{16}$$

The element at the position *(i,j)* is denoted as $\mathbf{D}_{ij}$ if available and $\tilde{\mathbf{D}}_{ij}$ if missing.

Following (9b) we obtain $\tilde{\mathbf{D}}_{ij} = \tilde{\mathbf{D}}_{ji}$. Simple strategies of approximation for missing entries come from the triangle inequality (16). The first scheme is minimax TIBA. Minimax aims to obtain the upper bound for $\tilde{\mathbf{D}}_{ij}$ by *min*imizing the *max*imum possible value of $\mathbf{D}_{ij}$ over all possible *k* for which $\mathbf{D}_{ij}$ is available (16), that is:

$$\tilde{\mathbf{D}}_{ij} = \min\{\mathbf{D}_{ik} + \mathbf{D}_{kj}\} \; \mathit{and} \; \{k \mid \mathbf{D}_{ik} \; \mathit{and} \; \mathbf{D}_{kj} \; \mathit{are\ availabl}\} \tag{17}$$

The second TIBA scheme exploits the triangle inequality in the opposite manner. The conditions (9c) and (16) give a pair of inequalities:

$$\mathbf{D}_{ik} \leq \tilde{\mathbf{D}}_{ij} + \mathbf{D}_{jk} \qquad (18a)$$

$$\mathbf{D}_{jk} \leq \tilde{\mathbf{D}}_{ji} + \mathbf{D}_{ik} = \tilde{\mathbf{D}}_{ij} + \mathbf{D}_{ik} \qquad (18a)$$

Solving (18a) and (18b) yields:

$$\tilde{\mathbf{D}}_{ij} \geq \mathbf{D}_{ik} - \mathbf{D}_{jk} \qquad (19a)$$

$$\tilde{\mathbf{D}}_{ij} \geq \mathbf{D}_{jk} - \mathbf{D}_{ik} \qquad (19b)$$

The inequalities combined with each other constitute the lower bound for $\tilde{\mathbf{D}}_{ij}$:

$$\tilde{\mathbf{D}}_{ij} \geq \left|\mathbf{D}_{jk} - \mathbf{D}_{ik}\right| \tag{20}$$

The maximin TIBA was defined to be the *max*imum of the *min*imum possible values for $\tilde{\mathbf{D}}_{ij}$ given by (20), which is:

$$\tilde{\mathbf{D}}_{ij} = \max\{|\mathbf{D}_{ik} - \mathbf{D}_{jk}|\} \; \mathit{and} \; \{k \mid \mathbf{D}_{ik} \; \mathit{and} \; \mathbf{D}_{kj} \; \mathit{are\ availabl}\} \tag{21}$$

In cases where such *k* does not exist, the missing elements $\tilde{\mathbf{D}}_{ij}$ are defined to have the value 0.

The following experiments use four approximation modes: (i) filling the missing values with 0 (baseline for quality of other approximations), (ii) the *mini*max TIBA, (iii) the *max*imin TIBA, and (iv) is the average of the last two. The number of missing values was expressed via the percentage of possible values in **D**, in the lower or upper triangular matrix and was in the range: 10% - 90%. The quality of clustering was measured according to two criteria. The first one is the **U** error for each trial (10 averaged trials) and is computed by $\sum_{i=1}^{c}\sum_{j=1}^{N}\left|\tilde{\mathbf{U}}_{ij}-\mathbf{U}_{ij}\right|$. This criterion measures the difference between the matrix $\tilde{\mathbf{U}}_{ij}$ that is the final partition matrix obtained from the NERF on the incomplete data and matrix $\mathbf{U}_{ij}$ that is the final matrix produced by the RFCM on the original relational data. The second criterion is referred to as the training error and represents the averaged percentage of the misclassified data. The training error of misclassification is computed by matching the resulting class membership determined by the maximal terminal membership value and class membership obtained from the data (cluster membership).

The dataset used in this experimentation is a simple dataset of two clusters of 50 patterns per cluster generated from normal distribution, whose values of parameters are included in Table 5-7. The 3rd and 4th feature are equal for all patterns to 0. The opposite to each other groups of points are visible in Fig. 5-9.

Table 5-7. Mean and covariance matrices used to generate two clusters.

$$N(\begin{bmatrix}2\\2\\0\\0\end{bmatrix},\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}) \qquad N(\begin{bmatrix}-2\\-2\\0\\0\end{bmatrix},\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix})$$

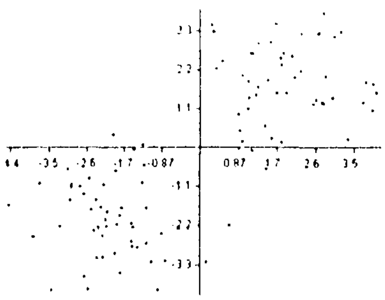


Figure 5-9. Two clusters from normal distribution.

There are two sets of graphs. In the first set the **U** error is presented for different beta parameter in four approximation modes (Fig. 5-10, Fig. 5-11 and Fig. 5-12). The second set embraces the sets of graphs for the training error and different values of beta parameter (Fig. 5-13, Fig. 5-14 and Fig. 5-15). The averaged values of beta parameter (10 trails) updated by the algorithm itself are presented in Fig.

5-16 and Fig. 5-17 (for the *maxi*min TIBA and the average mode, the beta value was not changed from initial 0 value).

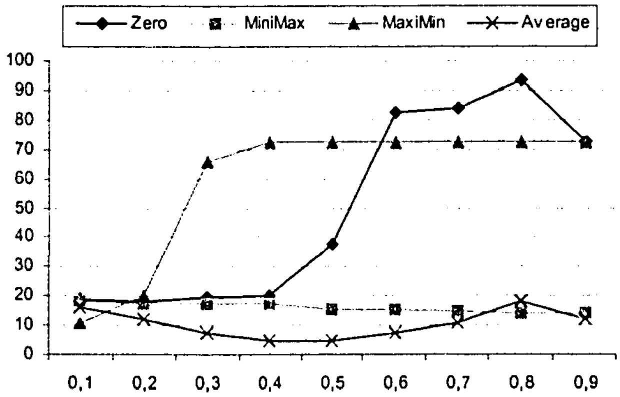


Figure 5-10. The averaged U error for four modes of approximation; beta parameter was equal to 0.

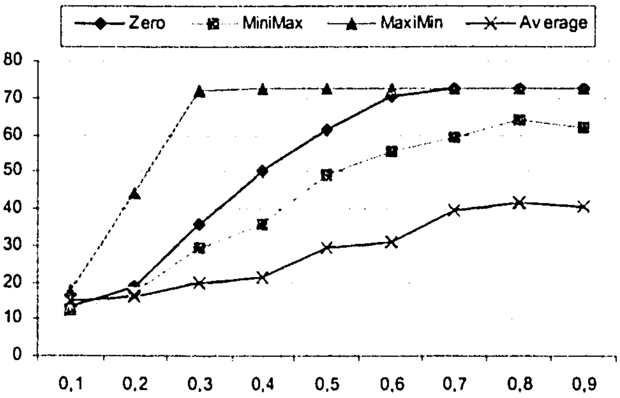


Figure 5-11. The averaged U error for four modes of approximation; beta=2.

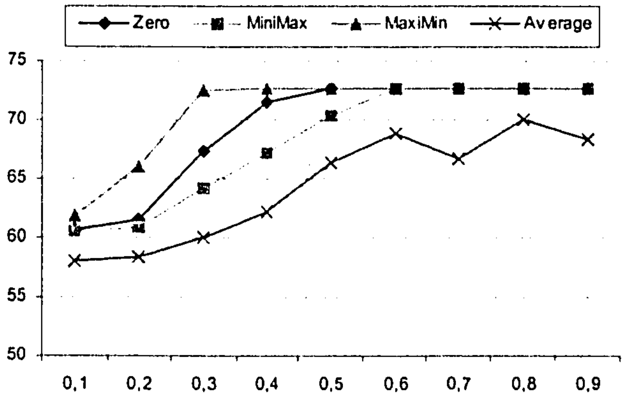


Figure 5-12. The averaged U error for four modes of approximation; beta=10.

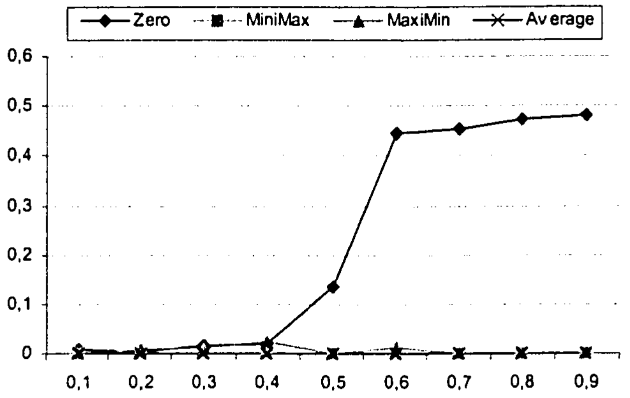


Figure 5-13. The averaged training error for four modes of approximation; beta=0.

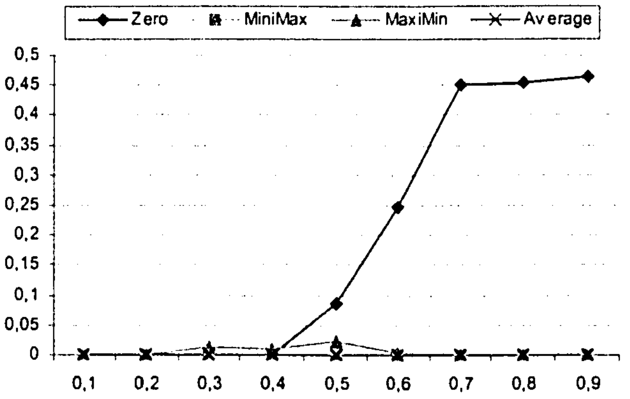


Figure 5-14. The averaged training error for four modes of approximation; beta=2.

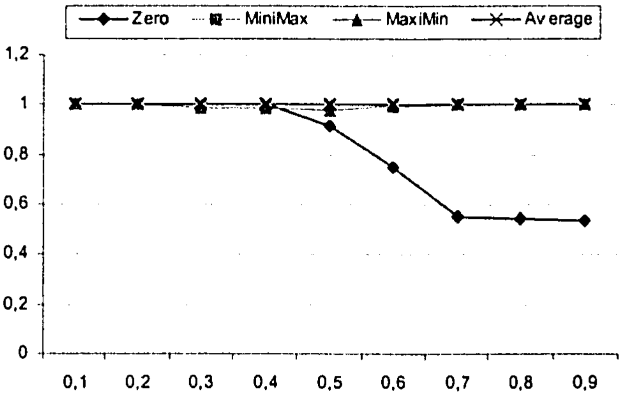


Figure 5-15. The averaged training error for four modes of approximation; beta=10.

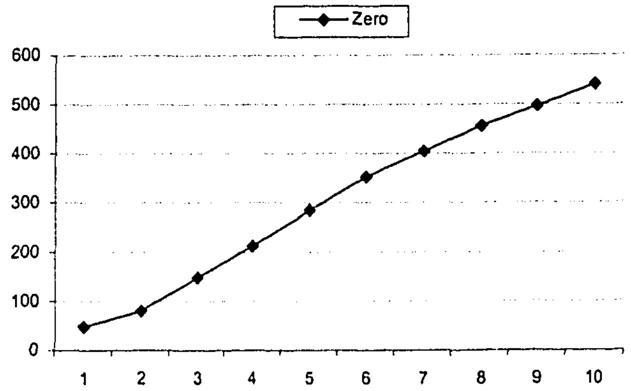


Figure 5-16. The averaged beta values for zero approximation mode; initial beta=0, percentage of missing values – 80%.

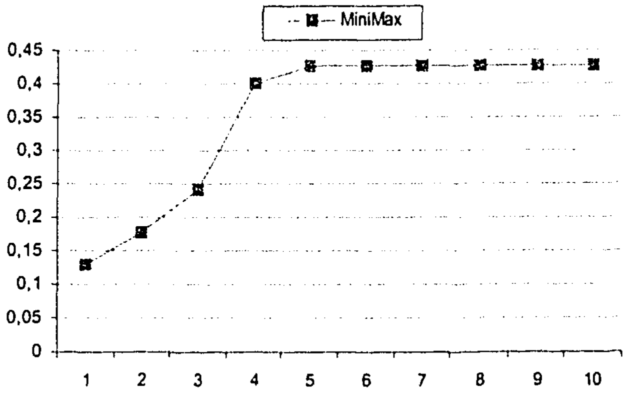


Figure 5-17. The averaged beta values for *mini*max approximation mode; initial beta=0, percentage of missing values – 80%.

The experiments' outcomes present the applicability of the NERF to cluster incomplete data. All approximation modes besides zero mode, produced in most cases low values of the training error. The partition matrix error was the lowest for the averaged TIBA (for initial beta=0). This may lead to the conclusion that all approximations worked fairly well but the closest one to the original matrix **D** was produced by the averaged TIBA. After increasing the value of initial beta parameter for the algorithm, the partition matrix error was increased too. Hence it is advisable not to increase the beta value initially and allow the algorithm to modify it accordingly. The example change of beta is presented in Fig. 5-16 and Fig. 5-17. This will lead to better results because the algorithm's mechanism has the option to modify the beta value "upwards" only if the computed distance is lower than 0 (Step III in NERF, Computations of distances). An unnecessary increase of beta and implicitly the dissimilarities in **D** will result in worse performance. Except of that, the NERF can be successfully applied to clustering of incomplete data.

5.4 *Partial supervision – algorithmic enhancements to FCM*

This section is focused on user supervision mechanisms. In the situations when the expert knowledge from the user is available, an exploitation of it may aid the unsupervised approaches and as a result may provide better performance. In this section the concepts of partial supervision are explored: partial supervision, a proximity-based approach and a combined approach of proximity-based partial supervision (a hybrid construction of these two).

5.4.1 Partially supervised FCM (PS-FMC)

The Partially Supervised FCM (PS-FCM) is founded on the concept of providing the user with a flexible way of incorporation of his domain knowledge by explicitly labelling patterns. This means the user will be involved in guidance of an algorithm. The idea relies on the user's knowledge that certain patterns belong to particular clusters. The user can label a selected pattern and in this way impose its assignment to a certain cluster. Importance lies in exploiting underlying information residing in these "known" patterns.

In a general sense the Partially Supervised Fuzzy C–Means [12] follows an alternating optimization scheme of the standard FCM algorithm. It is possible to accomplish the objective of enabling the human operator a significant involvement in the clustering process by extending the FCM objective function (2) by a supervision component. Let us consider the following objective function:

$$J_m(\mathrm{U},\mathrm{V}) = \sum_{k=1}^{N}\sum_{i=1}^{c}(u_{ik})^m (d_{ik})^2 + \alpha\sum_{k=1}^{N}\sum_{i=1}^{c}(u_{ik} - f_{ik})^m \mathbf{b}_k (d_{ik})^2 \qquad (22)$$

The parameters of the additive component are:

(a) The weight factor α, which models the impact coming from the labelled part.

(b) Information from the user about class membership f_{ik}. It has the form of hard c-partition, $f_{ik} \in \{0,1\}$ and both conditions $\sum_{i=1}^{c} f_{ik} = 1$ and $0 < \sum_{k=1}^{N} f_{ik} < N$ hold true.

(c) Boolean predicate b_k controlling if the user entered f_{ik} class membership information and assuming the following values:

$$\mathbf{b}_k = \begin{cases} 1 & \mathit{if}\ f_{ik} = 1\ \mathit{is\ given}\ (\mathit{for}\ k-\mathit{th\ pattern}) \\ 0 & \quad \mathit{if}\ f_{ik}\ \mathit{is\ not\ given}\ (f_{ik} = 0) \end{cases}$$

(d) Fuzzfier m. For the purpose of derivations of the formulas for the prototypes and u_{ik} the parameter m is equal to m*=2*. In a case of any other value (different from 2) conditions require an additional computational effort.

Ultimately we are interested in the minimization of:

$$J_2(\mathbf{U},\mathbf{V})=\sum_{k=1}^{N}\sum_{i=1}^{c}(u_{ik})^2(d_{ik})^2+\alpha\sum_{k=1}^{N}\sum_{i=1}^{c}(u_{ik}-f_{ik})^2\mathbf{b}_k(d_{ik})^2 \qquad (23)$$

The minimization of (23) is performed by the Lagrange multipliers method. The Lagrangian of (23) will be as follows:

$$F_k(\lambda,u_k)=\sum_{i=1}^{c}(u_{ik})^2(d_{ik})^2+\alpha\sum_{i=1}^{c}(u_{ik}-f_{ik})^2\mathbf{b}_k(d_{ik})^2-\lambda(\sum_{i=1}^{c}u_{ik}-1) \qquad (24)$$

Derivatives after u_k and λ give two equations:

$$\text{(i) } \frac{\partial F_k}{\partial u_{st}}(\lambda,u_k)=2u_{st}d_{st}^{\ 2}+2\alpha u_{st}\mathbf{b}_t d_{st}^{\ 2}-2\alpha f_{st}\mathbf{b}_t d_{st}^{\ 2}-\lambda=0$$

$$\text{(ii) } \frac{\partial F_k}{\partial \lambda}(\lambda,u_k)=\sum_{i=1}^{c}u_{ik}-1=0$$

Solving above equalities leads to the new formula for the entries u_k's of the partition matrix:

$$u_{ik}=\frac{(1-\frac{\alpha\mathbf{b}_k}{1+\alpha\mathbf{b}_k}\sum_{j=1}^{c}f_{jk})}{\sum_{j=1}^{c}\frac{d^2{}_{ik}}{d^2{}_{jk}}}+\frac{\alpha f_{ik}\mathbf{b}_k}{1+\alpha\mathbf{b}_k} \qquad (25)$$

Now remaining computations are to complete two stages alternating optimization design by new formulas for prototypes. After substitution $d_{ik}^{\ 2}=(x_{kj}-v_{ij})^2$ to (23) and taking the derivative of it after v_i, we obtain:

$$\frac{\partial F_k}{\partial v_{ij}}(\lambda,u_k)=u_{ik}^{\ 2}(2v_{ij}-2x_{kj})+\alpha(u_{ik}-f_{ik})^2\mathbf{b}_k(2v_{ij}-2x_{kj})=0 \qquad (26)$$

Reorganization of the terms provide us with final formulas for prototypes:

$$\mathbf{v}_i=\frac{\sum_{k=1}^{N}(u_{ik}+\alpha(u_{ik}-f_{ik})^2\mathbf{b}_k)\mathbf{x}_k}{\sum_{k=1}^{N}(u_{ik}+\alpha(u_{ik}-f_{ik})^2\mathbf{b}_k)} \qquad (27)$$

Influence of the additive component in (22) should be more significant; the larger the gap between the user input and the trend from dataset. When the values of u_{ik} and f_{ik} are similar, the impact of the additional part is minimal. If no information f_{ik} from the user is entered or if the learning coefficient is equal to $\alpha=0$, the supervision would have no effect and the algorithm will be reduced to the standard FCM. The optimization formulated in this way seems to fulfill the objective by allowing the user for guidance. The general scheme of the PS-FCM algorithm is presented in Table 5-8.

Table 5-8. The overall scheme of the Partially Supervised Fuzzy C-Means (PS-FCM).

I. Initialization:

Fix c, $2 \le c < N$ *; Initialize randomly the partition matrix* $\mathbf{U} \in \mathbf{M}_{fc}$*. Entering* f_{ik} *information by the user. Proceed with* $l = 0,1,\ldots$

II. Computations of prototypes:

Compute c prototypes $\{\mathbf{v}^{(l)}{}_{\mathrm{i}}\}$ *with* $\mathbf{U}^{(l)}$ *according to (27):*

$$\mathbf{v}_{\mathrm{i}} = \frac{\sum_{k=1}^{n}(u_{ik} + \alpha(u_{ik} - f_{ik})^2 \mathbf{b}_k)\mathbf{x}_{\mathrm{k}}}{\sum_{k=1}^{n}(u_{ik} + \alpha(u_{ik} - f_{ik})^2 \mathbf{b}_k)}$$

III. Computations of entries of partition matrix:

Compute $\mathbf{U}^{(l)}$ *using* $\{\mathbf{v}^{(l)}{}_{\mathrm{i}}\}$ *if* $d_{ik} > 0$ *; use (25):*

$$u_{ik} = \frac{(1 - \frac{\alpha \mathbf{b}_k}{1 + \alpha \mathbf{b}_k}\sum_{j=1}^{c} f_{jk})}{\sum_{j=1}^{c}\frac{d^2{}_{ik}}{d^2{}_{jk}}} + \frac{\alpha f_{ik}\mathbf{b}_k}{1 + \alpha \mathbf{b}_k}$$

Otherwise:

Proceed accordingly to Fuzzy C-Means (Table 5-1).

IV. Checking for termination condition or specifying maximal number of iterations:

Proceed accordingly to Fuzzy C-Means (Table 5-1).

5.4.2 Proximity-based FCM (P-FCM)

A proximity-based clustering explores another mode of partial supervision. An additional supervision process is augmenting the "standard" clustering performed in completely unsupervised manner. The source of supervision is enclosed in the user "hints" and is expressed in terms of resemblance of pairs of objects in the dataset. A data analyst enters proximity constraints in a way that any two patterns can be considered either similar or different. This paradigm can be perceived as reconciliation of two sources of information: (i) the first one resulting from data and intrinsically associated with data, (ii) another source is external to data and provided by the user (data expert), who gathers some general observations about data and is able to make use of them.

The general computing scheme of the P-FCM [10,11] is built upon the standard FCM's optimization procedure. The proximity optimization mechanism forms an internal optimization loop. These two procedures work in the interleaved manner and separately update the entries of the fuzzy partition matrix. The standard FCM phase is well known. It was described in detail in Section 5.2.1. The internal loop is optimized by a gradient method. The accommodation of proximity points is carried out by a certain objective function, which minimization leads to the optimal fuzzy partition matrix. The objective function is as follows:

$$V = \sum_{k_1=1}^{N} \sum_{k_2=1}^{N} (\hat{p}[k_1,k_2] - p[k_1,k_2])^2 b[k_1,k_2] d[k_1,k_2] \quad (28)$$

Where:

1) The notation of $\hat{p}[k_1,k_2]$ describes the proximity level produced by the partition matrix:

$$\hat{p}[k_1,k_2] = \sum_{i=1}^{c} (u_{ik_1} \wedge u_{ik_2}) \quad (29)$$

Where $\wedge$ denotes the minimum operation. It is not difficult to detect that if k_1=k_2 $\hat{p}[k_1,k_2] = \sum_{i=1}^{c} (u_{ik_1} \wedge u_{ik_2}) = \sum_{i=1}^{c} u_{ik} = 1$

2) $p[k_1,k_2]$ is the proximity value provided by the user

3) $b[k_1,k_2]$ receives the binary value corresponding to the situation if the user provided the proximity value (in this case $b[k_1,k_2] = 1$), or otherwise $b[k_1,k_2] = 0$ for each pair of patterns, for which the user did not enter the proximity hint

4) $d[k_1,k_2]$ is a distance between patterns

The objective function with proximity defined in (29) is as follows:

$$V = \sum_{k_1=1}^{N} \sum_{k_2=1}^{N} (\sum_{i=1}^{c} (u_{ik_1} \wedge u_{ik_2}) - p[k_1,k_2])^2 b[k_1,k_2] d[k_1,k_2] \quad (30)$$

The optimization of this performance index is done in an iterative gradient-based format:

$$u_{st}(iter+1) = \left[u_{st}(iter) - \alpha \frac{\partial V}{\partial u_{st}(iter)} \right]_{0,1} \quad (31)$$

Where $s = 1,2,...,c$, $t = 1,2,...,N$,

$[\ ]_{0,1}$ denotes clipping the result to the unit interval and α is a positive learning rate. The detailed derivations are straightforward (the successive iterations are referred to as "iter":

$$\begin{aligned} \frac{\partial V}{\partial u_{st}(iter)} &= \sum_{k_1=1}^{N} \sum_{k_2=1}^{N} \frac{\partial}{\partial u_{st}} (\sum_{i=1}^{c} (u_{ik_1} \wedge u_{ik_2}) - p[k_1,k_2])^2 = \\ &= 2\sum_{k_1=1}^{N} \sum_{k_2=1}^{N} (\sum_{i=1}^{c} (u_{ik_1} \wedge u_{ik_2}) - p[k_1,k_2]) \frac{\partial}{\partial u_{st}} \sum_{i=1}^{c} (u_{ik_1} \wedge u_{ik_2}) \end{aligned} \quad (32)$$

The derivative assumes binary values depending on satisfaction of the conditions:

$$\frac{\partial}{\partial u_{st}}\sum_{i=1}^{c}(u_{ik_1} \wedge u_{ik_2}) = \begin{cases} 1 & if\, t = k_1 \;\; and \;\; u_{sk_1} \le u_{sk_2} \\ 1 & if\, t = k_2 \;\; and \;\; u_{sk_2} \le u_{sk_1} \\ 0 & otherwise \end{cases} \qquad (33)$$

Both optimization schemas used in the P-FCM optimize the partition matrix. Learning rate α controls the impact of the internal optimization. The higher value of α is associated with more confidence that we have in the proximity hints provided by the user. This means that we allow the proximity hints to influence already formed structure of the partition matrix. The collaborative optimization effect can transform itself into competitive process if the entered proximity information does not correspond to the natural relationships in dataset. This happens because the two objective functions guiding optimization processes may lead to different optimal partition matrices. The optimal partition matrix for one objective function may not be optimal for the second one. This competitive behaviour is seen as oscillations during minimization of V.

A conceptual aspect of proximity guided clustering is somehow similar to the idea of relational clustering described in previous sections. However, there are important differences between these two approaches:

a) Relational clustering is motivated by an absence of the vector representation of single patterns while the relational data are only available. In proximity based clustering we still operate in the vector space of patterns' features. From this a computational advantage can arise because in P-FCM number of patterns is n and number of relational patterns is much higher – $n(n-1)/2$.

b) Unlike relational clustering, the P-FCM provides two independent optimization schemas. The number of hints entered and the learning coefficient can arbitrarily model the amount of user impact.

c) The number of clusters is to be specified at initialization of the relational algorithms. P-FCM relaxes this constraint dealing only with the number of proximities for pairs of objects. The actual number of clusters is entirely dependent upon the external optimization.

The overall scheme of the P-FCM optimizations is presented in Fig. 5-18.

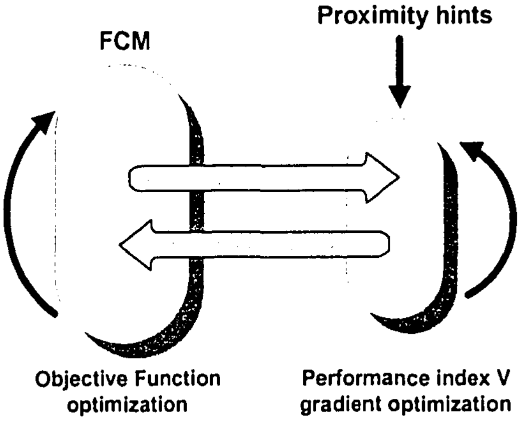

Proximity hints
FCM
Objective Function optimization
Performance index V gradient optimization

Figure 5-18. The overall scheme of Proximity-based FCM (P-FCM) optimization steps.

5.4.3 Partially Supervised Proximity–based FCM (PSP-FCM)

It is interesting to pursue the investigation of the consequences of availability for the algorithm two types of user supervision at the same time. Is it possible to combine both sources of the user input: partial supervision in the form of class memberships (PS-FCM) with proximity hints (P-FCM) into one method? What will be the clustering performance of such a method? Will it perform better than either one of the other methods separately?

We tried to answer these questions by construction of the Partially Supervised Proximity–based FCM (PSP-FCM) algorithm. The PSP-FCM attempts to combine in a collaborative way both sources of the user input: partial supervision in the form of class memberships and proximity hints. In this scenario the user will have the opportunity to explicitly impose the membership of some patterns to certain clusters, as well as enter the degree of similarity between them. This behaviour could be achieved by the application of the objective function of the Partially Supervised FCM (2) and extending it with the gradient optimization of the Proximity-based FCM (28). Following every iteration of the internal optimization of the PS-FCM algorithm's objective function, the external process starts optimizing the proximity-based performance index.

The overall flow of the combined optimizations is presented in Fig. 5-19.

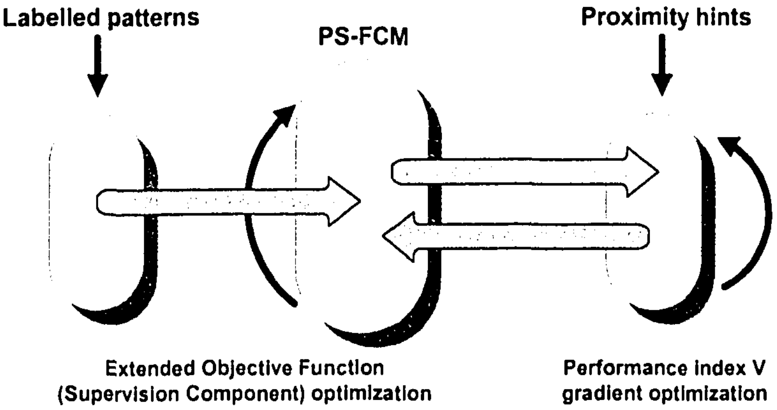

Labelled patterns
PS-FCM
Proximity hints
Extended Objective Function (Supervision Component) optimization
Performance index V gradient optimization

Figure 5-19. The overall scheme of Partially supervised Proximity-based FCM (PSP-FCM) optimization steps.

The initial concern of this hybrid integration is if the both schemes would not compete with each other or if some certain choice of the class membership information and proximity hints would make the competition possible. It is not difficult to show that these mechanisms would rather act in complimentary manner. Let us consider for instance two dissimilar patterns from different clusters (the opposite would be applicable for similar patterns). These dissimilar patterns would have class memberships for different clusters indicating the placement of them in different clusters. On the other hand, the induced proximity hint between these patterns would attain low value indicating that the patterns are dissimilar. It is easily seen from this situation that both mechanisms would rather support each other because they have the same objective of placing the patterns in separate clusters. In this situation both types of the user supervision for the patterns was entered and this is the model what was assumed for the experiments in this work.

However it is possible to envision other situations of specifying the user supervision elements. For instance they can be entered exclusively i.e. for some patterns the class membership information could be applied and for the other patterns the proximity hints could be given. Various experimental choices are possible especially in the constrained situations, in which only limited user knowledge is available.

In general the Partially Supervised Proximity-based FCM (PSP-FCM) gives more control the user who can decide what type of supervision is more natural to apply in a given context which depends right now on three sources of information with two of them being user supervision mechanisms: trend in data, user labelled patterns and proximity hints.

In this study the interest lies in gaining general knowledge about the influence of certain parameters (number and quality of f_{ik} and proximity hints) on the results of clustering and achieving the best performance.

5.4.4 Experimentation on synthetic data

The following experiment illustrates a user supervision effect induced by PS-FCM, P-FCM and PSP-FCM algorithms. The user will have the ability to impose assignment of particular patterns to certain clusters and/or to enter the proximity hints between patterns. The main interest is to investigate how the user input influences the performance and if it can improve it. Similarly as in Section 5.2.3 there will be assumed the criterion of comparison as the sum of the distances between original prototypes and the new prototypes obtained in noisy environment. It is expected that user supervision will help to revert them to the original positions.

The synthetic dataset used here contains 3 clusters of 50 patterns from normal distribution and 30 Cauchy outliers. The patterns are 4-dimensional (Fig. 5-20).

Table 5-9. Mean and covariance matrices used to generate three clusters.

$$N(\begin{bmatrix}2\\2\\0\\0\end{bmatrix},\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}) \qquad N(\begin{bmatrix}-2\\-2\\0\\0\end{bmatrix},\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}) \qquad N(\begin{bmatrix}6\\0\\0\\0\end{bmatrix},\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix})$$

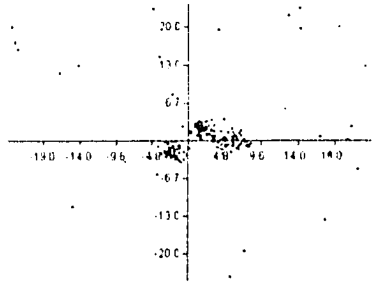


Figure 5-20. Three clusters from normal distribution and 30 outliers.

Fig. 5-21 shows the performance the of PS-FCM. The information *fik* was directly entered to impose assignment of data points to clusters they belong to. The case 0 *fik* denotes the sum of distances between the FCM on data with and without outliers and while *fik* information was not entered. The rest refers to the amount of user supervision entered and denotes the sum of distances computed for the FCM with outliers and the PS-FCM with user supervision. Evidently the sum of distances decreases and the PS-FCM performs better than the FCM. Increasing a parameter alpha strengthens this effect.

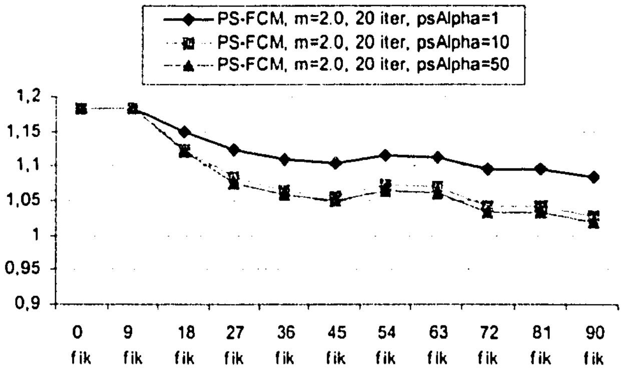


Figure 5-21. The averaged sum of distances of the PS-FCM for different values of alpha in respect to the FCM.

The performance of the P-FCM was measured in the same manner. The proximity hints were entered in two scenarios: (i) the similarity hints – entered for randomly chosen data points from the same cluster, (ii) the dissimilarity hints – introduced for randomly chosen points from different clusters. The experimentally determined constant equal to 0.5 modified proximity hints values and new proximity values were adjusted in the following manner: the actual proximity computed from the partition matrix plus 0.5 (for strengthening the similarity relationship between two data points), and the actual proximity minus 0.5 (for inhibiting the similarity relationship between two data points). Fig. 5-22a,b presents the results for the 2-10% of possible proximities to be entered (proximity matrix). The second figure skips the 0 *fik* case for clarity of the presentation.

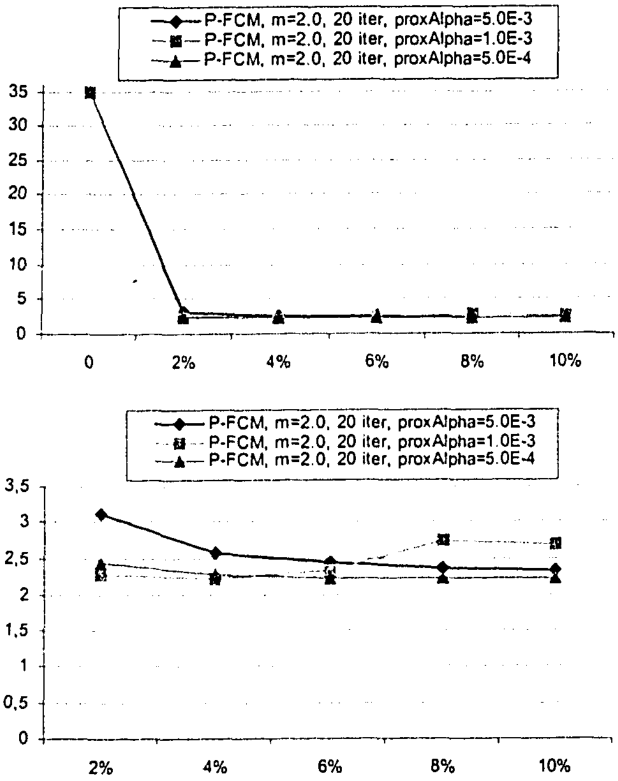


Figure 5-22a, b. The averaged sum of distances of the P-FCM for different values of alpha in respect to the FCM.

The combined effort of both modes of user supervision is combined by the PSP-FCM. This algorithm allows specifying *fik* class assignment of data points to certain clusters and proximity hints referring to the degree of similarity between data points. Fig. 5-23a,b shows the performance of the PSP-FCM in comparison to the single user supervision mode clustering: the PS-FCM and the P-FCM. In this very simple test case all three algorithms exhibit similar (very close) performance. PSP-FCM achieves the performance close to P-FCM. For both algorithms it is possible to obtain even better performance because the proximity hints were entered for selected points chosen in completely random manner. However the important fact is potentially promising, successful combination both sources of user supervision input into one method – PSP-FCM.

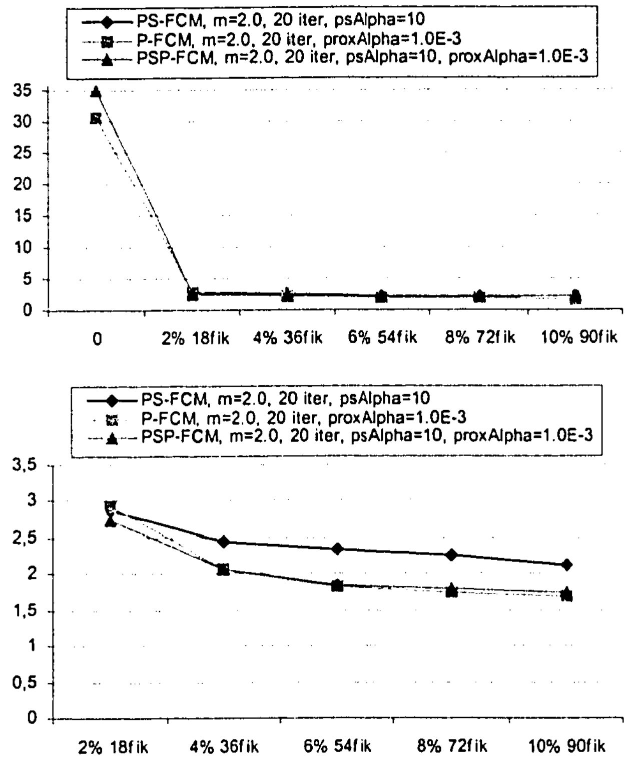


Figure 5-23a,b. The averaged sum of distances of the PS-FCM, the P-FCM, the PSP-FCM in respect to the FCM for the same dataset.

5.5 Bibliography

1. Bezdek J.C., "Pattern recognition with fuzzy objective function algorithms", Plenum Press, New York, 1981.

2. Bezdek, J.C., Hathaway, R.J., A note on two clustering algorithms for relational network data, *SPIE Vol. 1293 Applications of Artificial Intelligence VIII*, 1990, pp. 268-277.

3. Bezdek, J.C., Hathaway, R.J., Windham, M.P., Numerical comparison of the RFCM and AP algorithms for clustering relational data, *Pattern Recognition*, Vol. 24, No. 8, 1991, pp. 783-791.

4. Hathaway, R.J., Bezdek, J.C., Clustering incomplete relational data using non-Euclidean relational fuzzy c-means algorithm, *Pattern Recognition Letters 23*, 2002, pp.151-160.

5. Hathaway, R.J., Bezdek, J.C., Nerf c-means: Non-Euclidean relational fuzzy clustering, *Pattern Recognition*, Vol. 27, No. 3, 1994, pp. 429-437.

6. Hathaway, R.J., Bezdek, J.C., Davenport, W.J, On relational data versions of c-means algorithms, *Pattern Recognition Letters,* Vol. 17, 1996, pp. 607-612.

7. Kersten, P. R., Fuzzy Order Statistics and Their Application to Fuzzy Clustering, *IEEE Transactions on Fuzzy Systems*, Vol. 7, No. 6, 1999.

8. Kersten, P. R., Implementation issues in the fuzzy c-medians clustering algorithm, *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, Vol. 2, 1997.

9. Kersten, P. R., The Fuzzy Median and the Fuzzy MAD, *Proceedings of ISUMA-NAFIPS*, 1995.

10. Loia, V., Pedrycz, W., Senatore, S., P-FCM: a proximity-based fuzzy clustering for user-centered web applications, *International Journal of Approximate Reasoning*, Vol. 34, Issues 2-3, 2003, pp. 121-144.

11. Pedrycz, W., Loia, V., Senatore, S., P-FCM: proximity—based fuzzy clustering, *Fuzzy Sets and Systems,* Vol. 148, Issue 1, 16, 2004, pp. 21-41.

12. Pedrycz, W., Waletzky, J., Fuzzy clustering with partial supervision, *IEEE Trans. on Systems, Man, and Cybernetics 5*, 1997, pp. 787-795.

6 Data Exploration with User Supervision

6.1 Methodology

This section thoroughly develops the methodology of experimentation. The assumed model organizes data, performs the search and represents results. The constructed framework unifies data manipulation tools, clustering and classification, and representation of the results. Fuzzy clustering algorithms constitute the core of the model. FCM is used as the basis (ground truth) for comparison with other versions of the algorithm employing the user supervision enhancements. The experiments appearing in the later sections will be clustering experiments as well as classification experiments. In clustering we will be interested in partitioning the dataset – $X \subset S$ into subsets. While in the classification mode, we will attempt to divide the data space S into decision regions that will help in the classification of patterns from S instead of classifying the patterns only from the training set.

The assessment of the user input will be illustrated by the experiments run on the more advanced and complex data – the real world sets gathered from the Web and Machine learning repositories. The dataset D is to be divided into training set – D_1 and testing set – D_2. The split is performed for a specified ratio, in this case it will be 70% for the training set and 30% for testing set. It is done for every separate category, i.e. 70% of patterns from each category will remain in the training set and the rest will go to the testing set.

The dataset D_1 will be used to assess the performance of the clustering of FCM. On the same dataset we will run and compare algorithms with user supervision: P-FCM, PS-FCM and PSP-FCM. The user supervision mechanisms will allow for evaluation of a variety of scenarios depending on the method of incorporating user input, amount of it, and the strength of its impact. The foremost objective will be to detect the most successful scenarios of user interactions and provide guidelines for the application of these scenarios in any other experimentation. In the second part of this experiment we will be highly interested in the derivation of fuzzy classifiers. The prototypes obtained in the clustering mode will serve as the drivers for the classification of the data from testing set D_2.

In these experimental settings we will be doing a comparison between two modes of incorporating of the user input. The first mode will be the random mode and the user input will be applied in a random way. The second mode will focus on the misclassified patterns from the standard FCM method, and will manipulate the user supervision specifically to alleviate the problem of the misclassification.

In our examples we deal with labelled patterns, thus the assessment of the performance and simulation of user input is fairly straightforward.

Fig. 6-1 explains the logical flow of the experimentation methodology.

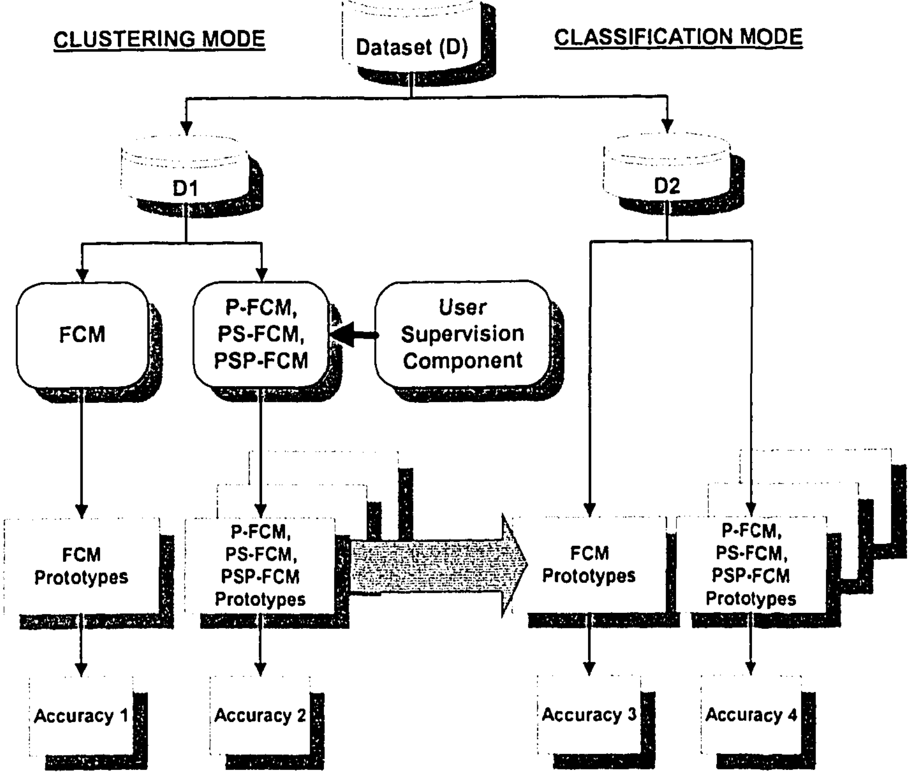


Figure 6-1. The logical flow of experimentation with user supervision.

6.2 Assessment of clustering performance

6.2.1 Accuracy formulation

Models provide a carrier for recognition, organization and representation of information. Obviously, it is possible to find more than one model for a physical process and immediately after this statement arises a question of finding the model that describes the given process in the best way. The best way is referred to as achieving the best performance. We need a criterion for assessment of its performance. In this section, the accuracy concept will be defined for the purpose of evaluation of results.

The accuracy is calculated according to membership values in the fuzzy c-partition matrix. A particular pattern will be assigned to the cluster with the final highest membership value for this cluster in the fuzzy partition matrix. The accuracy is calculated in the following manner. Having i clusters, where $i = 1...c$ and j classes, where $j = 1...p$ in data we formed the accuracy matrix $\mathbf{A}$.

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1p} \\ \cdots & \cdots & \cdots \\ a_{c1} & \cdots & a_{cp} \end{bmatrix} \tag{1}$$

Searching for the maximal value a_{ij} from given *i-th* cluster and dividing it by the number of patterns in the class, will give the accuracy for one class. The sum of accuracy for each class (the column in the matrix) will give the final accuracy.

Besides quantitative measurements of the accuracy such a matrix has another important asset. It enables the visual inspection of the dispersion of patterns over clusters. This allows the monitoring of a natural trend in the data, e.g. the most "unsure" patterns from one cluster will be spread over many classes and conversely, the most compact clusters will contain mainly patterns from one class.

6.2.2 Cluster validity

Accuracy calculations and performance measurement of algorithms is fairly easy while dealing with labelled data. However, evaluation of how "good" the structure is and what is the number of clusters is more complex.

The straightforward approach involves an objective function itself. This strategy leads to processing data through $c \in \{2,\ldots,n-1\}$ clusters and recording the optimal values of the performance index as a function of *c*. The significant problem arises in this situation because many objective functions are monotonic in *c* and unless the changes are considerable, it is almost impossible to detect pathological behaviour. For this reason, there were proposed heuristic validity indicants [2]. There exist two different approaches:

(a) the partition index

(b) the entropy index

The partition index of **U** is the scalar:

$$F(\mathrm{U};c) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^2 / n \tag{2}$$

The partition entropy of **U** is denoted as:

$$H(\mathrm{U};c) = -\sum_{k=1}^{n} \sum_{i=1}^{c} u_{ik} \log_a(u_{ik}) / n \tag{3}$$

Where the logarithmic base is assumed as $a \in (1,\infty)$ and $u_{ik}\log_a(u_{ik}) = 0$ whenever $u_{ik} = 0$, in this case a is base of the natural logarithm, $a = e$.

The illustration of cluster validity indicants applied to Web documents collections can be found in [4].

6.2.3 Cluster similarity

Aside from calculations of accuracy and estimation of the proper number of clusters, it is interesting to investigate another relationships residing in datasets. One of these relationships is the similarity between clusters. Besides revealing the relationships within the dataset based on the objects being grouped, the similarity between clusters could be used for instance to determine, which clusters should be merged or split when clustering the patterns for the number of clusters different than categories. We introduce the cluster similarity between cluster i and cluster j as a measure based on the entries of the final partition matrix:

$$ClusterSimilarity(i,j) = \frac{1}{N/2}\sum_{k=1}^{N}\frac{u_{ik} \wedge u_{jk}}{u_{ik} + u_{jk}} \qquad (4)$$

Where N is the number of patterns.

As we can see, the functional (4) reaches 1.0 value of similarity for $i = j$ (itself) and 0 for hard c-partitions, in which membership functions returns either 1 or 0. In the numerator we used the minimum operation because according to the operations on fuzzy sets, the set intersection is defined as the minimum operation of the values of characteristics functions for two fuzzy sets. If we assume the fact that the larger the sets intersection, the larger the similarity between them, the operation minimum for partition matrix membership functions of a pattern would express the similarity between clusters based on this pattern. Above assumption is straightforward because the sets will be more similar to each other, the greater their intersection, what in a particular case will be the highest for the same sets i.e. $u_{ik} \wedge u_{ik} = u_{ik}$. If we take a sum for all patterns, weight the functional by the sum of the characteristics functions of the sets and additionally weight by the number of patterns, we obtain the similarity between the rows of partition matrix, which in fact is the similarity between clusters. We can calculate the similarities for each pair of clusters and present it in the matrix form where the rows and columns would correspond to categories and the entries will be the similarities. Henceforth it will be referred to as cluster similarity matrix. Using the cluster similarity matrix enables easy inspection of the relationships in the dataset. In this work the cluster similarity matrices were calculated for the datasets with more than 3 categories, as it is not very interesting to investigate the cluster similarity for 2 or 3 categories.

The algorithms used do not associate the cluster number with category. Nevertheless, it is possible to derive this information from the accuracy matrix and tie the clusters to corresponding categories. This approach was used in the experiments.

6.3 *Machine Learning datasets experimentation*

6.3.1 Wine recognition database experiments

The experimental part starts with a simple dataset of wine recognition. The initial analysis of the dataset included the inspection and computations of the range, the mean and the standard deviation for every feature for each class. The range of feature values in vectors varied considerably. In order to minimize the impact of particular features we could (i) normalize data or (ii) introduce a weight component to our distance function: the Euclidean distance weighted by an inverse of the square of standard deviation from *i-th* feature:

$$d_{ik}^2 = \sum_{i=1}^{n} w_i^2 (x_k - v_i)^2 = \sum_{i=1}^{n} \frac{(x_k - v_i)^2}{\sigma_i^2} \qquad (5)$$

Fig. 6-2 shows the accuracy results of the FCM for the Euclidean distance of raw data, normalized data and weighted distance. While the results for the normalized data and the weighted distance are essentially the same, we decided that the rest of the experiments would be performed on normalized data. The FCM is converging fast and in a few iterations it is possible to obtain final solution. For the purpose of experimentation, 20 iterations were assumed as certainly guaranteeing to obtain the final partition matrix. The fuzzifier *m* was given the standard value of *m=2.0*. The accuracy for the FCM is presented in Table 6-1.

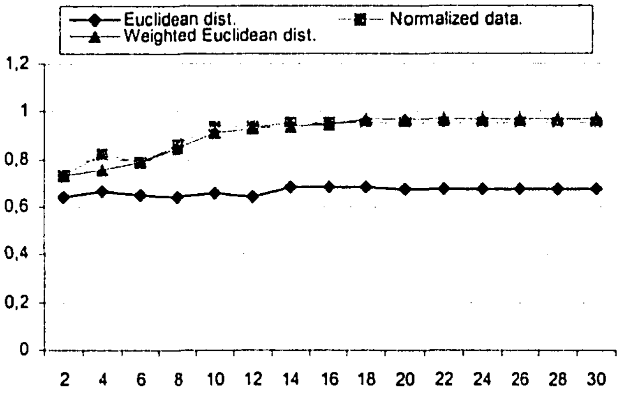


Figure 6-2. The averaged (10 trails) accuracy for 2-30 iterations.

Table 6-1. Accuracy results for clustering (10 trials average).

Algorithm and parameters	Euclidean distance	Weighted Euclidean distance	Euclidean distance - Normalized data
FCM m=2.0; 20 iterations; c=3	0.67 ± 0.001	0.97 ± 0.001	0.96 ± 0

6.3.1.1 Knowledge incorporation

The obtained accuracy result is very high, thus there is not much field for improvement. However, we are interested if the results still could be improved by application of user knowledge. The dataset was divided into training and testing set in the ratio 70%-30% respectively. The training set was used for the clustering mode to assess the methods' performance and to generate prototypes used later in the classification mode for testing set.

We investigated the possible improvement of accuracy when applying user input. The interaction with the user is provided via the mechanisms of the PS-FCM, the P-FCM and the PSP-FCM. There is a number of choices of incorporation user input. Two approaches of incorporating user supervision were compared. The first one, the Random mode, assumes random generation of class assignment information f_{ik}, based on the class memberships from categories' labels (PS-FCM) or the generation of the proximity hints between randomly chosen patterns (P-FCM). High proximity hints are for the patterns from the same class and low proximity hints for patterns originating from different classes. The user supervision for the PSP-FCM was a combination of the PS-FCM and the P-FCM supervision and was applied in the same manner as in the other algorithms.

In the second scenario, the Misclassification mode, instead randomly generating the class assignment information (PS-FCM) or proximity hints (P-FCM), or both (PSP-FCM), we calculated the misclassified patterns for each cluster and focused on reducing the problem of misclassification by attempting to improve the classification rate (accuracy) directly by the user input applied to the misclassified patterns. The second process as more complex deserves broader description. The PS-FCM allows imposing class membership of specific patterns to particular clusters. After determining the misclassified patterns for each class and finding out the correct class assignment, it was possible to demand the right class assignment of them. The algorithms do not associate the clusters numbers with the class numbers from data so to enforce that particular patterns belong to adequate clusters we experimented with adding more class membership information for correctly classified patterns in each cluster. This would cause to form clusters around these patterns. The amount of this additional information to enter varied in the range 3-9% of all patterns.

Unlike the PS-FCM, the P-FCM algorithm offers the designer another way of exploiting his or her knowledge. Namely the P-FCM permits the user to specify the degree of similarity between patterns expressed as a proximity hint. This is another kind of information, certainly much less specific than the rigid class assignment information. However, regarding the proximity hints as less quality information is wrong. This manner of incorporation of the knowledge can be very useful, because it does not demand from the user the specific knowledge of the particular class assignment but only the subjective judgement about the level of similarity between any of two patterns.

Similarly to the PS-FCM experiments, in P-FCM experiments the proximity for misclassified patterns was entered as well. To enter proximity hints it was necessary to make a distinction in perception of misclassified patterns. We regard the misclassified patterns of class *k* as patterns, which should be classified to *k-th* class but they were classified into remaining classes. The wrongly classified patterns of class *k* are patterns from other categories incorrectly classified to the category *k*. In fact misclassified patterns and wrongly classified patterns is the same set of patterns but the difference lies in the reference point (the class). The P-FCM proximity hints were entered in the following manner:

i) For misclassified patterns. A high proximity hint is entered for the specific misclassified pattern and the closest pattern (or *n* closest patterns) within the same cluster. The closest pattern is the pattern with the minimal distance to the misclassified pattern.

ii) For wrongly classified patterns. A low proximity hint is entered for the specific wrongly classified pattern and the closest pattern (or *n* closest patterns) from the cluster it was incorrectly classified to.

The proximity hints of different degree of similarity were incorporated in these two scenarios. High proximity hints were adjusted in such a manner that we calculated the proximity induced by the partition matrix (5-29) and added to it an experimentally determined values: (i) a constant 0.5 as proximity i.e. *highProx* = *actualProx + 0.5* or (ii) the value *1/c* where *c* is a number of classes. The low proximity hints were adjusted in the opposite manner i.e. *lowProx* = *actualProx - 0.5*. In case of overflow or underflow the results were clipped to [0,1] interval.

Another important aspect is the number of proximities. We experimented with the same number of hints introduced per one pattern. This number was determined as the percentage of the number of patterns per cluster i.e. *pattern_number_in_class*percentage_rate*. In practice the percentage of 10% brought a visible effect. The more the hints entered, the better the performance gain.

Class assignment information f_{ik} and the proximity hints entered for the first mode (the random mode) were entered in the same manner as for the misclassification mode. The only modification was that class assignment supervision was given as the percentage of patterns in the group for which f_{ik} was entered. The number of patterns for which the proximity hints were given was assumed as the average number of misclassified patterns calculated from the FCM. User supervision was set up in this manner to be able to compare both modes: the random mode and the misclassification mode. In other situation the number of user input would vary in these two modes and would bring into question the meaningful comparison between them.

The parameters of the PSP-FCM were combined from the parameters of the PS-FCM and the P-FCM. Thus it was easy to compare the relative performance of all three methods. In Table 6-2, Table 6-3, Table 6-4 are presented the results for clustering experiment (training set) and classification experiment (testing set).

Table 6-2. Average accuracy for clustering and classification (10 trials) of the FCM and the PS-FCM.

Algorithm and parameters	Form of user knowledge	Training dataset	Testing dataset
FCM m=2.0; 20 iter; c=3	———	0.96 ± 0.002	0.95 ± 0.008
	Random Mode		
PS-FCM α =1.0; 20 iter; c=3	(3%)*fik* randomly entered per group	0.96 ± 0.004	0.95 ± 0
	(6%)*fik* randomly entered per group	0.96 ± 0.005	0.95 ± 0.004
	(9%)*fik* randomly entered per group	0.97 ± 0.005	0.96 ± 0.006
	Misclassification Mode		

PS-FCM α =1.0; 20 iter; c=3	a) 0*fik* entered per group b)*fik* for misclassified patterns	0.95 ± 0.07	0.93 ± 0.01
	a) (3%)*fik* entered per group b)*fik* for misclassified patterns	0.98 ± 0.002	0.92 ± 0
	a) (6%)*fik* entered per group b)*fik* for misclassified patterns	0.99 ± 0.005	0.91 ± 0.004
	a) (9%) 12*fik* entered per group b)*fik* for misclassified patterns	0.99 ± 0	0.91 ± 0

Table 6-3. Average accuracy for clustering and classification (10 trials) of the P-FCM.

Algorithm and parameters	Form of user knowledge	Training dataset	Testing dataset
	Random Mode		
P-FCM α =5.0E-4; m=2.0; 20 iter; c=3	10% of randomly entered proximity hints	0.94 ± 0.005	0.95 ± 0
	20% of randomly entered proximity hints	0.95 ± 0.005	0.95 ± 0
	30% of randomly entered proximity hints	0.95 ± 0.007	0.95 ± 0
	Misclassification Mode		
P-FCM α =5.0E-4; m=2.0; 20 iter; c=3	10% of proximity hints for misclassified patterns	0.96 ± 0	0.95 ± 0
	20% of proximity hints for misclassified patterns	0.98 ± 0.003	0.95 ± 0
	30% of proximity hints for misclassified patterns	0.99 ± 0.003	0.95 ± 0
P-FCM α =5.0E-4; m=2.0; 20 iter; c=3 *prox = prox + 1/c (and prox= prox −1/c)*	10% of proximity hints for misclassified patterns	0.97 ± 0.004	0.95 ± 0
	20% of proximity hints for misclassified patterns	0.99 ± 0.004	0.95 ± 0
	30% of proximity hints for misclassified patterns	0.99 ± 0.003	0.95 ± 0

Table 6-4. Average accuracy for clustering and classification (10 trials) of the PSP-FCM.

Algorithm and parameters	Form of user knowledge	Training dataset	Testing dataset
	Random Mode		
PSP-FCM α =5.0E-4; m=2.0; 20 iter; c=3	a) (6%)*fik* randomly entered per group b) 10% of randomly entered prox. hints	0.94 ± 0.03	0.94 ± 0.01
prox = prox + 0.5 (and prox = prox - 0.5)	a) (6%)*fik* randomly entered per group b) 20% of randomly entered prox. hints	0.94 ± 0.03	0.93 ± 0.01
	Misclassification Mode		

PSP-FCM α =5.0E-4; m=2.0; 20 iter; c=3	a) (6%) *fik* entered per group *b) fik* for misclassified patterns c) 10% of proximity hints per pattern (for misclassified patterns)	0.99 ± 0	0.95 ± 0
prox = prox + 0.5 (and prox = prox - 0.5)	a) (6%) *fik* entered per group *b) fik* for misclassified patterns c) 20% of proximity hints per pattern (for misclassified patterns)	0.99 ± 0.005	0.95 ± 0

6.3.1.2 Observations

The classes of the dataset are well separable. Even though the maximal possible performance gain was small, the results of applying user supervision are very promising. The methods with user supervision are capable of improving the results, which had already been very high (the FCM produces 96% accuracy). All three methods provided improvement relatively to the FCM. The PS-FCM seems to be the most effective for the clustering mode (specific assignment of pattern to class). The P-FCM and the PSP-FCM perform well even with small percentage of proximity entered. The comparison between fully the random way of incorporating class assignment information, the proximity hints, and more advanced mode when the user input is provided with the focus on the misclassified patterns shows that the second scenario performs better. The user knowledge provided for the misclassified patterns is more effective. Smaller amount of class assignment information f_{ik} and smaller number of proximity hints record larger gain in the performance for the misclassified patterns mode. It is noticeable that the effort needed for the misclassified patterns mode is smaller and provides better results. For instance the PS-FCM and the PSP-FCM reach the maximal 99% accuracy while applying only 6% of class assignment information and 10% of proximity hints (PSP-FCM).

In the classification mode we did not observe any improvement from the user input. The accuracy remained on the level of the standard FCM.

6.3.2 Image segmentation dataset experiments

The dataset of image segmentation is more complex than the previous dataset. The number of patterns and the number of clusters were considerably increased. To explore the structure of the dataset we calculated the accuracy matrices for the FCM for different values of parameter *c* (Table 6-5). The mean and standard deviation values were computed for the accuracy matrices of the training dataset. The rows in the accuracy matrices were arranged in such a way that the largest values from categories were placed on the diagonal, thus it is convenient to observe dispersion of categories over clusters.

The clustering experiment was performed for the number of categories equal to the number of clusters (7). There was also examined how the categories merged for the number of clusters smaller than number of categories and how they split if there was more clusters than categories. Evidently, the most compact categories turned out to be 'Sky', 'Grass' and 'Window' (Table 6-5). Conversely, the most variable (uncertain) categories were 'Brickface' and 'Foliage'. An interesting situation occurred for number of clusters equal 6. Instead of the identification of new cluster centers, category 'Window' was allocated to other categories, while for the remaining categories we observed increased accuracy.

In this experiment it was used the 10 fold cross validation testing scheme. The training sets contain 210 patterns. For the testing set was used a different training set in every iteration.

Table 6-5a,b,c,d,e. The accuracy matrices exhibiting allocation of patterns from 7 categories to 5-9 clusters for testing dataset.

Cluster #	Brickface	Cement	Foliage	Grass	Path	Sky	Window
1	**15.92.3**	1.1 ± 0.6	6 ± 2.0	0.7 ± 0.7	0 ± 0	0 ± 0	4.6 ± 0.5
2	0 ± 0	**16.6 ± 2.5**	0.5 ± 0.1	0 ± 0	0.8 ± 0.2	0.1 ± 0	1.3 ± 0.4
3	5.9 ± 1.0	0.8 ± 0.3	**16.5 ± 0.1**	0.6 ± 0.1	0 ± 0	0 ± 0	5.6 ± 0.8
4	0 ± 0	0.1 ± 0	0.1 ± 0	**23.5 ± 0.8**	0 ± 0	0 ± 0	1 ± 0.3
5	0 ± 0	1.9 ± 0	0.3 ± 0.1	0.2 ± 0.6	**19.7 ± 0.4**	0 ± 0	0.2 ± 0.2
6	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	**25.9 ± 1.3**	0 ± 0
7	2.3 ± 3.2	2.8 ± 0	2.5 ± 0.8	2.6 ± 0.2	0 ± 0	1.2 ± 0.3	**12.2 ± 1.2**
8	0.7 ± 0.2	2.9 ± 1.7	1.3 ± 0.1	1.5 ± 0.5	7.0 ± 0.3	2.7 ± 0.9	2.6 ± 0.4
9	5.2 ± 1.7	3.5 ± 0.1	2.8 ± 0.9	0.8 ± 0.3	2.5 ± 0.5	0 ± 0	2.5 ± 0.1

Cluster #	Brickface	Cement	Foliage	Grass	Path	Sky	Window
1	**15.8 ± 1.9**	1.2 ± 0.9	6.4 ± 0.4	0.3 ± 0.1	0.2 ± 0	0 ± 0	3.7 ± 0.7
2	0.3 ± 0	**15.4 ± 0.1**	0.5 ± 0.1	0.9 ± 0.3	3 ± 1.0	0.1 ± 0	0.3 ± 0.5
3	4.5 ± 2.1	4.5 ± 0.1	**12.5 ± 3.8**	1.7 ± 0.2	0.2 ± 0	0 ± 0	6.6 ± 0.5
4	1.5 ± 0.5	0 ± 0	0 ± 0	**23.5 ± 0.8**	0 ± 0	0 ± 0	1.8 ± 0.6
5	0 ± 0	2.8 ± 0.2	0.5 ± 1.6	0.1 ± 0.3	**22.2 ± 2.0**	0 ± 0	0.1 ± 0
6	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	**27.2 ± 0.9**	0 ± 0
7	4.2 ± 1.5	4.3 ± 0.7	4.6 ± 1.5	1.3 ± 0.2	0.9 ± 0.3	0 ± 0	**14.3 ± 0.4**
8	3.6 ± 1.2	1.7 ± 2.3	5.4 ± 1.4	2.2 ± 0.7	3.5 ± 3.5	2.6 ± 0.8	3.2 ± 0.2

Cluster #	Brickface	Cement	Foliage	Grass	Path	Sky	Window
1	**16.8 ± 3.6**	2.3 ± 1.5	4.9 ± 1.3	2.1 ± 0	0.3 ± 0.9	0 ± 0	4.3 ± 0.2
2	0.6 ± 0.2	**18.8 ± 0.6**	1.3 ± 0.4	1.4 ± 0.4	4 ± 1	0 ± 0	0.8 ± 0.7
3	10 ± 0.3	1 ± 0.3	**20.1 ± 2.2**	2.2 ± 0.7	0.2 ± 0	0 ± 0	7.5 ± 1.8
4	0.1 ± 0	0.3 ± 0.5	0.1 ± 0	**23.9 ± 0.6**	0 ± 0	1.3 ± 0.4	1.3 ± 0.4
5	0 ± 0	2.3 ± 0.7	0.4 ± 0.1	0 ± 0	**23.7 ± 0.7**	0 ± 0	0.5 ± 0.8
6	0 ± 0	0.1 ± 0	0 ± 0	0 ± 0	0 ± 0	**25.9 ± 1.3**	0 ± 0
7	2.5 ± 3.5	5.2 ± 0.4	3.1 ± 0.7	0.4 ± 0.5	1.7 ± 0.5	2.7 ± 0.9	**15.59 ± 0.4**

Cluster #	Brickface	Cement	Foliage	Grass	Path	Sky	Window
1	**21.7 ± 2.5**	1.8 ± 1.0	9.4 ± 0.6	2.0 ± 0.6	0.1 ± 0	0 ± 0	6.5 ± 2.5
2	1.5 ± 0.5	**17.9 ± 0.2**	1.3 ± 0.5	1.5 ± 0.5	4.4 ± 0.8	0 ± 0	7.1 ± 0
3	6.5 ± 3.1	4.9 ± 2.6	**18.0 ± 0.7**	3.3 ± 0.7	0.6 ± 0.2	0 ± 0	10.3 ± 2.1
4	0.3 ± 0.1	1.3 ± 0.1	0.6 ± 0.5	**23.2 ± 0.5**	0.2 ± 0	0 ± 0	5.5 ± 1.8
5	0 ± 0	4.0 ± 0.6	0.6 ± 0	0 ± 0	**24.7 ± 1.1**	0 ± 0	0.6 ± 1.4
6	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	**30.0 ± 0**	0 ± 0

Cluster #	Brickface	Cement	Foliage	Grass	Path	Sky	Window
1	**12.5 ± 0.1**	6.4 ± 1.5	3.8 ± 0.6	2.2 ± 1.2	1.6 ± 0.5	9.0 ± 3.0	4.9 ± 5.0

2	1.5 ± 0.5	**9.3 ± 3.1**	1.3 ± 0.4	0 ± 0	1.7 ± 0.5	18.0 ± 4.0	2.2 ± 0.7
3	13.2 ± 1.2	3.4 ± 0.8	**22.6 ± 1.1**	3.0 ± 0.6	0.4 ± 0.1	0 ± 0	14.4 ± 3.1
4	2.7 ± 0.9	1.8 ± 0	1.8 ± 0.2	**24.8 ± 0.6**	0.9 ± 0.3	0 ± 0	7.0 ± 1.6
5	0 ± 0	9.1 ± 0.6	0.5 ± 0.1	0 ± 0	**25.4 ± 1.5**	3.0 ± 1.0	1.5 ± 0.5

From the accuracy matrices it is possible to observe the variability of the categories including the most variable and/or the most compact classes. It would be also interesting to investigate the similarity between clusters. Using cluster similarity measure (4) the mean (Table 6-6) and standard deviation (Table 6-7) values of the cluster similarity matrix were calculated. The classes most similar to each other are: Window to Brickface (0.752), Brickface to Foliage (0.762) and Foliage to Window (0.689). The least similar to other classes is the Sky class. These conclusions adhere to the observations derived from accuracy matrices. The wrongly classified patterns in the Foliage category are the patterns originating from Brickface and Window categories. The category Sky is also the least variable (most compact) category.

Table 6-6. Mean values of cluster similarity (upper triangular matrix equal to lower triangular matrix).

	Brickface	Cement	Foliage	Grass	Path	Sky	Window
Brickface	1.0	0.618	0.762	0.486	0.601	0.290	0.752
Cement		1.0	0.564	0.526	0.680	0.341	0.649
Foliage			1.0	0.574	0.587	0.331	0.689
Grass				1.0	0.488	0.449	0.438
Path					1.0	0.323	0.645
Sky						1.0	0.257
Window							1.0

Table 6-7. Standard deviation values of cluster similarity (upper triangular matrix equal to lower triangular matrix).

	Brickface	Cement	Foliage	Grass	Path	Sky	Window
Brickface	0	0.007	0.031	0.004	0.002	0.002	0.002
Cement		0	0.008	0.044	0.040	0.011	0.021
Foliage			0	0.046	0.021	0.024	0.052
Grass				0	0.001	0.000	0.000
Path					0	0.000	0.003
Sky						0	0.000
Window							0

6.3.2.1 Knowledge incorporation

The final accuracy obtained by standard FCM will be the subject of improvement in this section by the user input. The choice of parameters of incorporation the designer knowledge is chosen in a similar manner as for the previous dataset. The experimentation part was extended. Class membership information *fik* for the PSP-FCM was entered for 0,5-7,5% correctly classified patterns per class. These patterns impose forming clusters around themselves. The proximity hints for the

P-FCM were entered for 10-50% of the closest patterns in each cluster. Both modes were applied: the random mode and the misclassified mode.

The PSP-FCM algorithm parameters are adjusted according to the PS-FCM and the P-FCM parameters. The results are presented in Table 6-8, Table 6-9 and Table 6-10.

Table 6-8. Average accuracy for clustering and classification (10 trials) of the FCM and the PS-FCM.

Algorithm and parameters	User input	Training dataset	Testing dataset
FCM m=2.0; 20 iter; c=7	———	0.76 ± 0.07	0.71 ± 0.07
	Random Mode		
PS-FCM α =1.0; 20 iter; c=7	(0.5%)*fik* randomly entered per group	0.78 ± 0.06	0.73 ± 0.08
	(1,5%)*fik* randomly entered per group	0.82 ± 0.03	0.75 ± 0.07
	(3%)*fik* randomly entered per group	0.85 ± 0.05	0.77 ± 0.10
	(4,5%)*fik* randomly entered per group	0.89 ± 0.03	0.79 ± 0.10
	(6%)*fik* randomly entered per group	0.92 ± 0.02	0.77 ± 0.10
	(7,5%)*fik* randomly entered per group	0.93 ± 0.02	0.77 ± 0.11
	(9%)*fik* randomly entered per group	0.93 ± 0.02	0.79 ± 0.11
	Misclassification Mode		
PS-FCM α =1.0; 20 iter; c=7	a) 0*fik* entered per group b)*fik* for misclassified patterns	0.78 ± 0.09	0.70 ± 0.07
	a) (0.5%)*fik* entered per group b)*fik* for misclassified patterns	0.84 ± 0.04	0.74 ± 0.07
	a) (1.5%)*fik* entered per group b)*fik* for misclassified patterns	0.86 ± 0.05	0.74 ± 0.08
	a) (3%)*fik* entered per group b)*fik* for misclassified patterns	0.89 ± 0.03	0.75 ± 0.08
	a) (4,5%)*fik* entered per group b)*fik* for misclassified patterns	0.95 ± 0.02	0.75 ± 0.09
	a) (6%)*fik* entered per group b)*fik* for misclassified patterns	0.96 ± 0.02	0.75 ± 0.10
	a) (7,5%)*fik* entered per group b)*fik* for misclassified patterns	0.97 ± 0.01	0.74 ± 0.11
	a) (9%)*fik* entered per group b)*fik* for misclassified patterns	0.99 ± 0	0.80 ± 0.11

Table 6-9. Average accuracy for clustering and classification (10 trials) of the P-FCM.

Algorithm and parameters	User input	Training dataset	Testing dataset
	Random Mode		
P-FCM α =5.0E-4; m=2.0; 20 iter; c=7 *prox = prox + 0.5 (and prox = prox - 0.5)*	10% of randomly entered proximity hints	0.75 ± 0.08	0.67 ± 0.08
	20% of randomly entered proximity hints	0.74 ± 0.08	0.67 ± 0.08
	30% of randomly entered proximity hints	0.74 ± 0.08	0.67 ± 0.09
	40% of randomly entered proximity hints	0.74 ± 0.08	0.67 ± 0.07
	50% of randomly entered proximity hints	0.75 ± 0.10	0.67 ± 0.09
	Misclassification Mode		
P-FCM α =5.0E-4; m=2.0; 20 iter; c=7 *prox = prox + 0.5 (and prox = prox - 0.5)*	10% of proximity hints for misclassified patterns	0.80 ± 0.05	0.69 ± 0.09
	20% of proximity hints for misclassified patterns	0.79 ± 0.12	0.71 ± 0.09
	30% of proximity hints for misclassified patterns	0.84 ± 0.06	0.68 ± 0.10
	40% of proximity hints for misclassified patterns	0.87 ± 0.06	0.72 ± 0.11
	50% of proximity hints for misclassified patterns	0.84 ± 0.07	0.71 ± 0.08
P-FCM α =5.0E-4; m=2.0; 20 iter; c=7 *prox = prox + 1/c (and prox = prox – 1/c)*	10% of proximity hints for misclassified patterns	0.80 ± 0.05	0.71 ± 0.08
	20% of proximity hints for misclassified patterns	0.78 ± 0.09	0.71 ± 0.10
	30% of proximity hints for misclassified patterns	0.75 ± 0.09	0.67 ± 0.10
	40% of proximity hints for misclassified patterns	0.79 ± 0.09	0.72 ± 0.08
	50% of proximity hints for misclassified patterns	0.76 ± 0.11	0.69 ± 0.10

Table 6-10. Average accuracy for clustering and classification (10 trials) of the FCM and the PSP-FCM.

Algorithm and parameters	User input	Training dataset	Testing dataset
	Random Mode		
PSP-FCM α =5.0E-4; m=2.0; 20 iter; c=7	a) (3%)*fik* randomly entered per group b) 30% of randomly entered prox. hints	0.74 ± 0.10	0.69 ± 0.09

prox = prox + 0.5 *(and prox = prox - 0.5)*	a) (3%)*fik* randomly entered per group b) 40% of randomly entered prox. hints	0.73 ± 0.07	0.72 ± 0.07
	a) (4,5%)*fik* randomly entered per group b) 30% of randomly entered prox. hints	0.72 ± 0.09	0.54 ± 0.70
	a) (4,5%)*fik* randomly entered per group b) 40% of randomly entered prox. hints	0.73 ± 0.08	0.71 ± 0.07
	Misclassification Mode		
	a) (3%)*fik* entered per group b)*fik* for misclassified patterns c) 30% of proximity hints per pattern	0.91 ± 0.05	0.86 ± 0.06
PSP-FCM α =5.0E-4; m=2.0; 20 iter; c=7	a) (3%)*fik* entered per group b)*fik* for misclassified patterns c) 40% of proximity hints per pattern	0.93 ± 0.05	0.87 ± 0.07
prox = prox + 0.5 *(and prox = prox - 0.5)*	a) (4,5%)*fik* entered per group b)*fik* for misclassified patterns c) 30% of proximity hints per pattern	0.95 ± 0.03	0.88 ± 0.06
	a) (4,5%)*fik* entered per group b)*fik* for misclassified patterns c) 40% of proximity hints per pattern	0.95 ± 0.03	0.88 ± 0.06

6.3.2.2 Observations

The final result from the standard FCM for clustering was 76%. We investigated if it was possible to improve the classification rate with user input. The maximum obtained accuracy for the PS-FCM was 99% correctly classified patterns for 9% additional class membership information *fik* entered. This is a very good result. The results for the P-FCM were less spectacular. The highest recorded accuracy was 87% but it is still a considerable improvement in comparison to the FCM. The modification of the value of proximity hints influenced the results. The value modified by the *1/c* expression was less successful than the modification by constant 0.5. The first option worked well for 3 clusters (Wine recognition dataset) but it turned out to be too small for larger number of clusters (7 in this case). The number of proximity hints was also important. Generally, the accuracy was increasing proportionally to the number of proximity hints entered. The highest accuracy was obtained for 30-40%. This may be explained by the fact that after incorporating too many hints the structure of the dataset starts to deteriorate. If we consider the values of the high proximity hints (the same would apply to low proximity hints as well), we will notice that too many hints entered with the same (similar) value for a misclassified pattern and the large number of the patterns from the class of this pattern (e.g. 80% of patterns) will cause that even the patterns with large distance from the misclassified pattern would have very high similarity hint associated with each other. This certainly might not be true and would affect the relationships within the dataset. As a result of it the accuracy might drop.

The parameters for the PSP-FCM were constructed from the most successful parameters' combinations of two previous algorithms. The clustering results were high. In comparison the PSP-FCM with the P-FCM and the PS-FCM the results obtained were higher than each of the two methods would provide separately for the same choice of parameters. This is an important conclusion because it shows that combining both sources of knowledge is possible and provides higher results than using any of the methods alone.

The results based on this dataset show more clearly that there exists a considerable difference between the random and the misclassification mode of incorporation of the user input. Focusing on the misclassified patterns allows for obtaining higher results. Moreover the results are obtained with smaller effort i.e. smaller number of incorporated user knowledge For instance the PS-FCM for 9% class assignment information per group records 93% for the random mode but 99% for the classification mode. This effect is even more visible for the P-FCM. The random mode did not improve the results in respect to the FCM (76%) for any number of hints used in the experiment (10%-50%) but the misclassification mode obtained 87% correctly classified patterns.

The accuracy for the testing set (classification mode) initially oscillated around the values obtained from the FCM for smaller amount of user supervision and subsequently improved for the PS-FCM and the PSP-FCM. The accuracy of the classification mode for the P-FCM was worse than the FCM results. This behaviour may lead to conclusions that the class membership assignment is less specific (the pattern belongs to a cluster or not) than the level of proximity between two patterns and the first case can be generalized to other patterns better then the second approach. Usually the relationships (degree of similarity) between patterns are very specific to the certain dataset and particular patterns existing in it. They cannot be applied to the other datasets because other patterns would most probably have different kind of relationship between each other.

6.3.3 Wisconsin Diagnostic Breast Cancer (WDBC) and Wisconsin Prognostic Breast Cancer (WPBC) experiments

The next two datasets come from the University of Wisconsin Hospitals, Madison. The classification task of the WDBC dataset is to correctly classify a given patient to either benign or malignant class. There are 569 cases in this dataset. The second learning problem is defined for the WPBC dataset with 194 patterns. It is to predict the recurrence of the illness. If it recurs within 24 months the case will be marked as positive, otherwise negative.

6.3.3.1 Knowledge incorporation

The parameters for the algorithm were constructed in a similar manner as in the previous datasets. We divided the dataset into training and testing set (70%-30%). Class membership information *fik* for the PSP-FCM was entered for the misclassified patterns and for 1,5%-18% correctly classified patterns per class. The proximity hints for the P-FCM were entered for the misclassified and wrongly classified patterns for 10-30% of the closest patterns in each cluster. The PSP-FCM algorithm's parameters merged PS-FCM and P-FCM parameters. The results for WDBC and WPBC are presented in Table 6-11, Table 6-12, Table 6-13, and Table 6-14, Table 6-15, Table 6-16 respectively.

Table 6-11. Average accuracy for clustering and classification (10 trials) of the FCM and the PS-FCM for the WDBC dataset.

Algorithm and parameters	User input	Training dataset	Testing dataset
FCM m=2.0; 20 iter; c=2	———	0.91 ± 0	0.92 ± 0.02
	Random Mode		
PS-FCM α =1.0; 20 iter; c=2	(1,5%)*fik* randomly entered per group	*0.91 ± 0.004*	*0.95 ± 0.001*
	(4,5%)*fik* randomly entered per group	0.92 ± 0	0.95 ± 0
	(9%)*fik* randomly entered per group	0.93 ± 0	0.95 ± 0
	(18%)*fik* randomly entered per group	0.95 ± 0	0.95 ± 0
	Misclassification Mode		
PS-FCM α =1.0; 20 iter; c=2	a) 0 *fik* entered per group; b) *fik* for misclassified patterns	0.93 ± 0.04	0.95 ± 0
	a) (1,5%)*fik* entered per group b) *fik* for misclassified patterns	0.94 ± 0.05	0.95 ± 0
	a) (4,5%)*fik* entered per group b) *fik* for misclassified patterns	0.99 ± 0	0.95 ± 0

Table 6-12. Average accuracy for clustering and classification (10 trials) of the FCM and the PS-FCM for the WDBC dataset.

Algorithm and parameters	User input	Training dataset	Testing dataset
	Random Mode		
P-FCM α =5.0E-4; m=2.0; 20 iter; c=2 *prox = prox + 0.5 (and prox = prox - 0.5)*	10% of randomly entered proximity hints	0.95 ± 0.005	0.95 ± 0
	20% of randomly entered proximity hints	0.98 ± 0.003	0.95 ± 0
	30% of randomly entered proximity hints	0.99 ± 0.002	0.95 ± 0
	Misclassification Mode		
P-FCM α =5.0E-4; m=2.0; 20 iter; c=2 *prox = prox + 0.5 (and prox = prox - 0.5)*	10% of proximity hints for misclassified patterns	0.98 ± 0	0.95 ± 0
	20% of proximity hints for misclassified patterns	1.0 ± 0	0.95 ± 0
	30% of proximity hints for misclassified patterns	1.0 ± 0	0.95 ± 0

Table 6-13. Average accuracy for clustering and classification (10 trials) of the PSP-FCM for the WDBC dataset.

Algorithm and parameters	Form of user knowledge	Training dataset	Testing dataset
	Random Mode		
PSP-FCM α =5.0E-4; m=2.0; 20 iter; c=2	a) (1,5%)*fik* randomly entered per group b) 10% of randomly entered prox. hints	0.89 ± 0.09	0.87 ± 0.08
prox = prox + 0.5 (and prox = prox - 0.5)	a) (3%)*fik* randomly entered per group b) 20% of randomly entered prox. hints	0.89 ± 0.09	0.87 ± 0.08
	Misclassification Mode		
PSP-FCM α =5.0E-4; m=2.0; 20 iter; c=2	a) 6*fik* entered per group (1,5%) b)*fik* for misclassified patterns c) 10% of proximity hints for misclassified patterns	0.97 ± 0.04	0.95 ± 0
prox = prox + 0.5 (and prox = prox - 0.5)	a) 18*fik* entered per group (4,5%) b)*fik* for misclassified patterns c) 20% of proximity hints for misclassified patterns	0.95 ± 0.04	0.95 ± 0

Table 6-14. Average accuracy for clustering and classification (10 trials) of the FCM and the PS-FCM for the WPBC dataset.

Algorithm and parameters	User input	Training dataset	Testing dataset
FCM m=2.0; 20 iter; c=2	———	0.58 ± 0	0.41 ± 0.02
	Random Mode		
PS-FCM α =1.0; 20 iter; c=2	(3%)*fik* randomly entered per group	0.57 ± 0.02	0.43 ± 0.005
	(6%)*fik* randomly entered per group	0.70 ± 0	0.41 ± 0
	(12%)*fik* randomly entered per group	0.77 ± 0.002	0.44 ± 0.004
	(24%)*fik* randomly entered per group	0.83 ± 0	0.40 ± 0
	Misclassification Mode		
PS-FCM α =1.0; 20 iter; c=2	0*fik* entered per group; *fik* for misclassified patterns	0.56 ± 0.002	0.49 ± 0.01
	a) (3%)*fik* entered per group b)*fik* for misclassified patterns	0.51 ± 0.001	0.49 ± 0.01
	a) (6%)*fik* entered per group b)*fik* for misclassified patterns	0.60 ± 0	0.49 ± 0.01
	a) (12%)*fik* entered per group b)*fik* for misclassified patterns	0.74 ± 0.002	0.50 ± 0.01
	a) (24%)*fik* entered per group b)*fik* for misclassified patterns	0.96 ± 0.04	0.50 ± 0.01

Table 6-15. Average accuracy for clustering and classification (10 trials) of the P-FCM for the WPBC dataset.

Algorithm and parameters	User input	Training dataset	Testing dataset
	Random Mode		
P-FCM α =5.0E-4; m=2.0; 20 iter; c=2	10% of randomly entered proximity hints	0.65 ± 0.01	0.38 ± 0.009
	20% of randomly entered proximity hints	0.70 ± 0.01	0.37 ± 0.004
prox = prox + 0.5 (and prox = prox - 0.5)	25% of randomly entered proximity hints	0.74 ± 0.02	0.37 ± 0.007
	30% of randomly entered proximity hints	0.66 ± 0.049	0.39 ± 0.01
	Misclassification Mode		
P-FCM α =5.0E-4; m=2.0; 20 iter; c=2	10% of proximity hints for misclassified patterns	0.75 ± 0.02	0.43 ± 0.04
	20% of proximity hints for misclassified patterns	0.93 ± 0.01	0.44 ± 0.02
prox = prox + 0.5 (and prox = prox - 0.5)	25% of proximity hints for misclassified patterns	0.95 ± 0.01	0.44 ± 0.003
	30% of randomly entered proximity hints	0.58 ± 0.10	0.46 ± 0.003

Table 6-16. Average accuracy for clustering and classification (10 trials) of the PSP-FCM for the WPBC dataset.

Algorithm and parameters	Form of user knowledge	Training dataset	Testing dataset
	Random Mode		
PSP-FCM α =5.0E-4; m=2.0; 20 iter; c=2	a) (12%)*fik* randomly entered per group b) 20% of randomly entered prox. hints	0.63 ± 0.03	0.50 ± 0.09
prox = prox + 0.5 (and prox = prox - 0.5)	a) (24%)*fik* randomly entered per group b) 25% of randomly entered prox. hints	0.64 ± 0.04	0.50 ± 0.09
	Misclassification Mode		
PSP-FCM α =5.0E-4; m=2.0; 20 iter; c=2	a) (12%)*fik* entered per group b)*fik* for misclassified patterns c) 20% of proximity hints for misclassified patterns	0.90 ± 0.02	0.44 ± 0.003
prox = prox + 0.5 (and prox = prox - 0.5)	a) (24%)*fik* entered per group b)*fik* for misclassified patterns c) 25% of proximity hints for misclassified patterns	0.91 ± 0.08	0.41 ± 0.006

6.3.3.2 Observations

The results for the WDBC and the WPBC datasets show that the user supervision plays an important role in recovering the structure of the dataset. The standard

FCM algorithm obtained 91% accuracy for the WDBC dataset. This rate was successfully increased to 95% by the PS-FCM and 99% the P-FCM in the random mode and to 99% by the PS-FCM and 100% the P-FCM in the misclassification mode. The PSP-FCM accuracy was equal to 97% in the misclassification mode. It is worth noting that the P-FCM algorithm in this experiment obtains comparable or even better results than PS-FCM. The classification results for all algorithms were stable and provided us with 95% of correct classification rate.

The accuracy of FCM for the second dataset (WPBC) was 58%. The PS-FCM raised the accuracy to 83% in the random mode and to 96% in the classification mode. The performance of the P-FCM showed 74% and 95% accuracy for the random and the classification mode respectively. The PS-FCM obtained 91% accuracy.

The results obtained after applying user supervision were much better. It was observed also very significant improvement of the misclassification mode over the random mode, especially for the WPBC dataset.

6.3.4 Dermatology database

The dermatology database contains 366 patterns. The learning task is to correctly assign patterns to 6 groups of erythemato-squamous diseases based on clinical and histopathological features. The cluster similarity measure was calculated for all clusters using the partition matrix obtained from FCM. The cluster similarity matrix (Table 6-17) reveals that the categories are very difficult to separate because they exhibit very high similarity in respect to each other. The categories 1 and 3, 1 and 6, 3 and 6 are the most similar to each other attaining almost the maximal value of similarity. The least similar category to all others is the second category (Seboreic dermatitis).

Table 6-17. Average values of cluster similarity (upper triangular matrix equal to lower triangular matrix).

	Psoriasis	Seboreic dermatitis	Lichen planus	Pityriasis rosea	Cronic dermatitis	Pityriasis rubra pilaris
Psoriasis	1.0	0.704	0.992	0.943	0.962	0.990
Seboreic dermatitis		1.0	0.701	0.753	0.733	0.709
Lichen planus			1.0	0.940	0.964	0.986
Pityriasis rosea				1.0	0.927	0.951
Cronic dermatitis					1.0	0.960
Pityriasis rubra pilaris						1.0

Table 6-18. Standard deviations values of cluster similarity (upper triangular matrix equal to lower triangular matrix).

	Psoriasis	Seboreic dermatitis	Lichen planus	Pityriasis rosea	Cronic dermatitis	Pityriasis rubra pilaris

Psoriasis	0	0.076	0.001	0.001	0.008	0.000
Seboreic dermatitis		0	0.076	0.080	0.087	0.076
Lichen planus			0	0.001	0.009	0.001
Pityriasis rosea				0	0.024	0.002
Cronic dermatitis					0	0.005
Pityriasis rubra pilaris						0

6.3.4.1 Knowledge incorporation

We followed the experimentation manner and parameters for the algorithms assumed for previous experiments. The dataset was divided into training and testing set (70%-30%). Class membership information *fik* for the PSP-FCM was entered for the misclassified patterns and for 3%-12% correctly classified patterns per class. The proximity hints for the P-FCM were entered for the misclassified and wrongly classified patterns for 10-30% of the closest patterns in each cluster. The parameters of the PSP-FCM algorithm merged the PS-FCM and the P-FCM parameters. The results are presented in Tables 6-19, 6-20, 6-21.

Table 6-19. Average accuracy for clustering and classification (10 trials) of the FCM and the PS-FCM.

Algorithm and parameters	Form of user knowledge	Training dataset	Testing dataset
FCM m=2.0; 20 iter; c=6	———————————	0.43 ± 0.06	0.40 ± 0.04
	Random Mode		
PS-FCM α =1.0; 20 iter; c=6	(3%)*fik* randomly entered per group	0.87 ± 0.005	0.79 ± 0.003
	(6%)*fik* randomly entered per group	0.91 ± 0.004	0.81 ± 0.003
	(9%)*fik* randomly entered per group	0.93 ± 0	0.82 ± 0
	(12%)*fik* randomly entered per group	0.92 ± 0	0.83 ± 0
	Misclassification Mode		
PS-FCM α =1.0; 20 iter; c=6	a) 0*fik* entered per group b)*fik* for misclassified patterns	0.99 ± 0.005	0.96 ± 0.006
	a) (3%)*fik* entered per group b)*fik* for misclassified patterns	0.99 ± 0.005	0.95 ± 0.001

Table 6-20. Average accuracy for clustering and classification (10 trials) of the P-FCM.

Algorithm and parameters	Form of user knowledge	Training dataset	Testing dataset
	Random Mode		
P-FCM α =5.0E-4; m=2.0; 20	10% of randomly entered proximity hints	0.57 ± 0.11	0.61 ± 0.05

iter; c=6	20% of randomly entered proximity hints	0.70 ± 0.09	0.69 ± 0.11
	30% of randomly entered proximity hints	0.76 ± 0.14	0.77 ± 0.07
	40% of randomly entered proximity hints	0.64 ± 0.11	0.73 ± 0.14
	Misclassification Mode		
P-FCM α =5.0E-4; m=2.0; 20 iter; c=6	10% of proximity hints for misclassified patterns	0.65 ± 0.06	0.67 ± 0.04
	20% of proximity hints for misclassified patterns	0.62 ± 0.06	0.65 ± 0.02
	30% of proximity hints for misclassified patterns	0.59 ± 0.05	0.63 ± 0.05
	40% of randomly entered proximity hints	0.64 ± 0.09	0.63 ± 0.03

Table 6-21. Average accuracy for clustering and classification (10 trials) of the PSP-FCM.

Algorithm and parameters	Form of user knowledge	Training dataset	Testing dataset
	Random Mode		
PSP-FCM α =5.0E-4; m=2.0; 20 iter; c=6	a) (3%)*fik* randomly entered per group b) 10% of randomly entered prox. hints	0.70 ± 0.12	0.74 ± 0.12
prox = prox + 0.5 (and prox = prox - 0.5)	a) (6%)*fik* randomly entered per group b) 20% of randomly entered prox. hints	0.66 ± 0.09	0.96 ± 0.004
	Misclassification Mode		
PSP-FCM α =5.0E-4; m=2.0; 20 iter; c=6	a) (3%)*fik* entered per group *b) fik* for misclassified patterns c) 10% of proximity hints per pattern (for misclassified patterns)	0.99 ± 0.001	0.96 ± 0.007
prox = prox + 0.5 (and prox = prox - 0.5)	a) (6%)*fik* entered per group *b) fik* for misclassified patterns c) 20% of proximity hints per pattern (for misclassified patterns)	0.99 ± 0.003	0.95 ± 0.008

6.3.4.2 Observations

The results of applying user knowledge proved that user supervision methods are very powerful. The initial 43% accuracy from the standard FCM was improved to 76% by the P-FCM (30% of randomly entered proximity hints), 99% by the PS-FCM (*fik* for the misclassified patterns and 3% *fik* entered per group) and the PSP-FCM (*fik* for the misclassified patterns and 3% *fik* entered per group and 10% proximity hints per pattern for the misclassified patterns). It is evident that the misclassification mode performs much better than the random mode. It requires smaller amount of user supervision and obtains better final accuracy (this is especially visible for the PS-FCM; in the random mode it achieved 92% accuracy for 12% *fik* and 99% accuracy for only 3% *fik* in the misclassification mode and the PSP-FCM – 70% and 99% respectively). The random mode obtained comparable or better results for the P-FCM in two situations.

The accuracy in the classification mode was significantly improved over the standard FCM accuracy (from 40% to 96% for the PS-FCM and the PSP-FCM). This demonstrates that the user supervision may help in constructing fuzzy classifiers where the prototypes can be used to classify patterns outside of the training set.

6.3.5 Glass identification database

The glass identification database contains 214 patterns. The classification task is to correctly assign patterns to 3 groups of different kinds of glass based on the chemical constituents as features.

6.3.5.1 Knowledge incorporation

We followed the experimentation manner and parameters' values for the algorithms assumed for previous experiments. The dataset was divided into training and testing set (70%-30%). Class membership information *fik* for the PSP-FCM was entered for the misclassified patterns and for 3%-18% correctly classified patterns per class. The proximity hints for the P-FCM were entered for the misclassified and the wrongly classified patterns for 10-30% of the closest patterns in each cluster. The PSP-FCM algorithm parameters merged the PS-FCM and the P-FCM parameters. The results for are presented in Tables 6-22, 6-23, 6-24.

Table 6-22. Average accuracy for clustering and classification (10 trials) of the FCM and the PS-FCM.

Algorithm and parameters	Form of user knowledge	Training dataset	Testing dataset
FCM m=2.0; 20 iter; c=3	————————	0.56 ± 0.004	0.70 ± 0
	Random Mode		
PS-FCM α =1.0; 20 iter; c=3	(3%)*fik* randomly entered per group	0.58 ± 0.01	0.70 ± 0.004
	(6%)*fik* randomly entered per group	0.59 ± 0.01	0.70 ± 0.004
	(9%)*fik* randomly entered per group	0.60 ± 0	0.70 ± 0.007
	(12%)*fik* randomly entered per group	0.60 ± 0.03	0.70 ± 0
	(15%)*fik* randomly entered per group	0.61 ± 0.02	0.70 ± 0
	(18%)*fik* randomly entered per group	0.63 ± 0	0.69 ± 0.01
	Misclassification Mode		
PS-FCM α =1.0; 20 iter; c=3	a) 0*fik* entered per group b)*fik* for misclassified patterns	0.65 ± 0.003	0.70 ± 0
	a) (3%)*fik* entered per group b)*fik* for misclassified patterns	0.64 ± 0.003	0.68 ± 0.003

a) (6%)*fik* entered per group b)*fik* for misclassified patterns	0.70 ± 0	0.68 ± 0
a) (9%)*fik* entered per group b)*fik* for misclassified patterns	0.73 ± 0.002	0.68 ± 0
a) (12%)*fik* entered per group b)*fik* for misclassified patterns	0.75 ± 0.003	0.67 ± 0
a) (15%)*fik* entered per group b)*fik* for misclassified patterns	0.77 ± 0.005	0.68 ± 0.005
a) (18%)*fik* entered per group b)*fik* for misclassified patterns	0.83 ± 0.002	0.61 ± 0.003

Table 6-23. Average accuracy for clustering and classification (10 trials) of the P-FCM.

Algorithm and parameters	Form of user knowledge	Training dataset	Testing dataset
	Random Mode		
P-FCM α =5.0E-4; m=2.0; 20 iter; c=3	10% of randomly entered proximity hints	0.59 ± 0.01	0.70 ± 0.001
	20% of randomly entered proximity hints	0.60 ± 0.02	0.70 ± 0.007
	30% of randomly entered proximity hints	0.61 ± 0.05	0.70 ± 0.006
	40% of randomly entered proximity hints	0.65 ± 0.02	0.69 ± 0.03
	50% of randomly entered proximity hints	0.65 ± 0.01	0.71 ± 0.01
	Misclassification Mode		
P-FCM α =5.0E-4; m=2.0; 20 iter; c=3	10% of proximity hints for misclassified patterns	0.61 ± 0.009	0.70 ± 0
	20% of proximity hints for misclassified patterns	0.60 ± 0.007	0.70 ± 0.004
	30% of proximity hints for misclassified patterns	0.64 ± 0.01	0.70 ± 0
	40% of randomly entered proximity hints	0.65 ± 0.02	0.69 ± 0.02
	50% of randomly entered proximity hints	0.65 ± 0.008	0.71 ± 0.02

Table 6-24. Average accuracy for clustering and classification (10 trials) of the PSP-FCM.

Algorithm and parameters	Form of user knowledge	Training dataset	Testing dataset
	Random Mode		
PSP-FCM α =5.0E-4; m=2.0; 20 iter; c=3	a) (9%)*fik* randomly entered per group b) 20% of randomly entered prox. hints	0.58 ± 0.06	0.63 ± 0.09

prox = prox + 0.5 *(and prox = prox - 0.5)*	a) (12%)*fik* randomly entered per group b) 20% of randomly entered prox. hints	0.59 ± 0.06	0.62 ± 0.10
	Misclassification Mode		
PSP-FCM α =5.0E-4; m=2.0; 20 iter; c=3	a) (9%)*fik* entered per group *b) fik* for misclassified patterns c) 20% of proximity hints per pattern (for misclassified patterns)	0.77 ± 0.03	0.67 ± 0.004
prox = prox + 0.5 *(and prox = prox - 0.5)*	a) (12%)*fik* entered per group *b) fik* for misclassified patterns c) 20% of proximity hints per pattern (for misclassified patterns)	0.80 ± 0.03	0.68 ± 0.01

6.3.5.2 *Observations*

The results obtained seem to confirm that the user supervision mechanisms are useful. The standard FCM obtained 56% accuracy in the clustering mode. The best improvement was observed for the PS-FCM in the misclassification mode – 83% accuracy (the random mode obtained only 63% accuracy). The difference between the performance of the random mode and the misclassification mode for the P-FCM was smaller. In the misclassification mode the P-FCM recorded 64% accuracy (over 61% in the random mode). The PSP-FCM proved to perform better in the misclassification mode: 80% accuracy against 59% accuracy in the random mode.

The classification accuracy was oscillating around the accuracy obtained from the standard FCM and we did not observe any significant improvement or deterioration of the accuracy. The accuracy for the FCM is 70%, which is unexpectedly high in respect to the clustering experiment (56%).

6.3.6 Thyroid gland database

The thyroid gland database contains 215 patterns. The learning task is to classify patterns to 3 groups of different kind of diseases based on the results of five laboratory tests.

6.3.6.1 *Knowledge incorporation*

We followed the experimentation manner and the parameters for assumed for other datasets. The dataset was divided into training and testing set (70%-30%). Class membership information *fik* for the PSP-FCM was entered for the misclassified patterns and for 3%-15% correctly classified patterns per class. The proximity hints for *fik* P-FCM were entered for the misclassified and wrongly classified patterns for 10-30% of the closest patterns in each cluster. The PSP-FCM algorithm parameters merged the PS-FCM and the P-FCM parameters. The results for are presented in Tables 6-25, 6-26, 6-27.

Table 6-25. Average accuracy for clustering and classification (10 trials) of the FCM and the PS-FCM.

Algorithm and parameters	Form of user knowledge	Training dataset	Testing dataset

FCM m=2.0; 20 iter; c=3	—	0.82 ± 0.03	0.62 ± 0.08
	Random Mode		
PS-FCM α =1.0; 20 iter; c=3	(3%)*fik* randomly entered per group	0.82 ± 0.05	0.65 ± 0.11
	(6%)*fik* randomly entered per group	0.84 ± 0.08	0.64 ± 0.05
	(9%)*fik* randomly entered per group	0.82 ± 0.14	0.67 ± 0.06
	(12%)*fik* randomly entered per group	0.87 ± 0	0.64 ± 0.009
	(15%)*fik* randomly entered per group	0.90 ± 0	0.71 ± 0
	Misclassification Mode		
PS-FCM α =1.0; 20 iter; c=3	a) 0 *fik* entered per group b)*fik* for misclassified patterns	0.96 ± 0.03	0.72 ± 0.07
	a) (3%)*fik* entered per group b)*fik* for misclassified patterns	0.85 ± 0.11	0.66 ± 0.09
	a) (6%)*fik* entered per group b)*fik* for misclassified patterns	0.95 ± 0.08	0.87 ± 0.14
	a) (9%)*fik* entered per group b)*fik* for misclassified patterns	0.92 ± 0.11	0.75 ± 0.01
	a) (12%)*fik* entered per group b)*fik* for misclassified patterns	0.98 ± 0.008	0.73 ± 0.009
	a) (15%)*fik* entered per group b)*fik* for misclassified patterns	0.99 ± 0.007	0.73 ± 0.01

Table 6-26. Average accuracy for clustering and classification (10 trials) of the P-FCM.

Algorithm and parameters	Form of user knowledge	Training dataset	Testing dataset
	Random Mode		
P-FCM α =5.0E-4; m=2.0; 20 iter; c=3	10% of randomly entered proximity hints	0.84 ± 0.01	0.63 ± 0.01
	20% of randomly entered proximity hints	0.84 ± 0.009	0.64 ± 0
	30% of randomly entered proximity hints	0.86 ± 0.01	0.64 ± 0
	40% of randomly entered proximity hints	0.87 ± 0.01	0.64 ± 0
	50% of randomly entered proximity hints	0.90 ± 0.02	0.64 ± 0
	Misclassification Mode		
P-FCM α =5.0E-4; m=2.0; 20 iter; c=3	10% of proximity hints for misclassified patterns	0.85 ± 0.005	0.67 ± 0.02
	20% of proximity hints for misclassified patterns	0.82 ± 0.09	0.61 ± 0.10

30% of proximity hints for misclassified patterns	0.88 ± 0.03	0.62 ± 0.08
40% of randomly entered proximity hints	0.78 ± 0.12	0.57 ± 0.11
50% of randomly entered proximity hints	0.85 ± 0.14	0.63 ± 0.08

Table 6-27. Average accuracy for clustering and classification (10 trials) of the PSP-FCM.

Algorithm and parameters	Form of user knowledge	Training dataset	Testing dataset
	Random Mode		
PSP-FCM α =5.0E-4; m=2.0; 20 iter; c=3	a) (9%)*fik* randomly entered per group b) 10% of randomly entered prox. hints	0.85 ± 0.007	0.74 ± 0.10
prox = prox + 0.5 (and prox = prox - 0.5)	a) (12%)*fik* randomly entered per group b) 10% of randomly entered prox. hints	0.98 ± 0.01	0.73 ± 0.009
	Misclassification Mode		
PSP-FCM α =5.0E-4; m=2.0; 20 iter; c=3	a) (9%)*fik* entered per group *b) fik* for misclassified patterns c) 10% of proximity hints per pattern (for misclassified patterns)	0.99 ± 0.009	0.73 ± 0.01
prox = prox + 0.5 (and prox = prox - 0.5)	a) (12%)*fik* entered per group *b) fik* for misclassified patterns c) 10% of proximity hints per pattern (for misclassified patterns)	0.98 ± 0.01	0.72 ± 0.01

6.3.6.2 Observations

The experiments on the thyroid gland database show that the user supervision helped to improve the results. In the clustering experiment the PS-FCM improved from initial 82% of the standard FCM's accuracy to 90% in the random mode and to 99% in the clustering mode. The P-FCM was able to achieve 86% and 88% accuracy in the random and clustering mode respectively. Very close performance in both modes revealed the PSP-FCM obtaining 98% and 99% of accuracy.

In the classification experiments we obtained good results too. Each algorithm improved the accuracy level of the standard FCM (62%). The highest accuracy obtained the PSP-FCM – 74%.

It is evident that the random mode of applying user supervision was able to improve the accuracy over the standard FCM results. Even better performance for all algorithms was obtained the misclassification mode. The user input was valuable in classification experiment. There was a visible improvement over standard FCM.

6.3.7 Summary and protocol for other experimentation

This section summarizes the results obtained and suggests the procedure to be undertaken in a particular experimental setting. The experiments were performed on 7 Machine Learning databases. The overall quality of the results achieved with the aid of user supervision is impressive. To compare the performances, a table was created to summarize the average percentage of accuracy improvement in the clustering mode (Table 6-28). The values were entered if the difference between the standard FCM accuracy and the accuracy of PS-FCM, P-FCM or PSP-FCM was larger than 2%. When the improvement was smaller than 2% the field was left blank. In the clustering experiment it was observed that there was improvement for every dataset in the misclassification mode and in most cases for the random mode. Only one case was recorded when the accuracy was smaller in the clustering mode than in the accuracy of FCM. This accuracy drop of 3% was observed only for the random mode of the Wisconsin Diagnostics Breast Cancer database. Evidently if we compared the results, the misclassification mode would prove to be better than the random mode. The performance gain in some cases was larger than 100%.

Table 6-28. The improvement of the average accuracy in the clustering mode.

#	Dataset	PS-FCM		P-FCM		PSP-FCM	
		Random mode	Misclass. mode	Random mode	Misclass. Mode	Random mode	Misclass. mode
1	Wine recognition		3%		3%		3%
2	Image segmentation	17%	21%		11%		21%
3.1	Wisc. Diag. Breast Cancer	4%	8%	8%	9%		6%
3.2	Wisc. Prog. Breast Cancer	25%	38%	16%	37%	6%	33%
4	Dermatology	50%	56%	33%	22%	27%	56%
5	Glass Identification	7%	27%	9%	9%		24%
6	Thyroid gland	8%	17%	8%	6%	40%	41%

We analyzed the results of the classification experiment in the same way. This time two tables were created; one for improved accuracy and the other one for the results where the performance deteriorated, and was worse than the results from FCM (Table 6-29 and Table 6-30). The data were entered only when the difference (improvement or drop) was larger than 2% and the rest of the fields were left blank. The values for the deteriorated table were taken as the highest accuracy achieved for any of the parameters (user input) for a particular algorithm and mode (random or misclassification).

Table 6-29. The improvement of the average accuracy in the classification mode.

#	Dataset	PS-FCM		P-FCM		PSP-FCM	
		Random mode	Misclass. mode	Random mode	Misclass. mode	Random mode	Misclass. mode
1	Wine recognition						
2	Image segmentation	8%	9%				7%

3.1	Wisc. Diag. Breast Cancer	3%	3%	3%	3%		3%
3.2	Wisc. Prog. Breast Cancer	3%	9%		5%	9%	3%
4	Dermatology	43%	55%	37%	27%	56%	56%
5	Glass Identification						
6	Thyroid gland	9%	25%		5%	12%	11%

Table 6-30. The deterioration of the average accuracy in the classification mode.

#	Dataset	PS-FCM		P-FCM		PSP-FCM	
		Random mode	Misclass. mode	Random mode	Misclass. mode	Random mode	Misclass. mode
1	Wine recognition						
2	Image segmentation			4%			
3.1	Wisc. Diag. Breast Cancer					5%	
3.2	Wisc. Prog. Breast Cancer			3%			
4	Dermatology						
5	Glass Identification					7%	
6	Thyroid gland						

The results of the classification experiment are better than the accuracy of FCM for all datasets in the misclassification mode. There was only one scenario when the accuracy dropped, and this resulted in the random mode. Random improvement of the structure does not provide the desired results, and it is better to proceed in an organized manner. This makes the fact stronger in that the misclassification mode is the preferred application mode of user knowledge. It is seen in the results that the gain of accuracy is smaller in the classification experiment than in the clustering experiment, because the user does not have a direct influence on the patterns in the testing set. However, if the initial prototypes were somehow disturbed and the applied user knowledge brought about the structure closer to the original one, we can expect a high performance boost – e.g. in the Dermatology database the performance gain in the classification experiment was more than 50% accurate. From this it follows that the user input could be beneficial in forming fuzzy classifiers, as it is a generalization from a particular dataset to the domain of patterns being classified, and it is very probable that the classification decisions made with the aid of user supervision would achieve higher accuracy.

The other important factor to consider is the number of class membership information or proximity hints to enter. The accuracy was plotted for the random and misclassification mode for the Image Segmentation database. Although the plots will be different for each dataset, it is possible to depict some recurring behaviours. The accuracy for PS-FCM will grow continuously with the increase in number of class membership information in the misclassification mode. The accuracy in the random mode will grow more slowly and tend to stabilize at some value (usually not reaching 99% accuracy as in the other mode) (Fig. 6-3).

The number of patterns for which we would enter the proximity hints will improve the accuracy only to a certain number of patterns – in most cases less than 50% of the patterns for a group. After reaching this number the accuracy starts to decrease. This is associated with the larger number of hints that will modify the original structure, but it is important to note that the modification is not always desirable. In our experiments we specified the proximity between two patterns by adjusting it with a 0.5 constant. The degree of proximity varies greatly inside the cluster, and it can be correct to assume that a few of the closest patterns to our misclassified pattern would have the same, very high, level of proximity. But this unnecessarily might be the case for the other patterns – especially for those located far in another part of the cluster. This effect is even more visible in the random mode when we choose any two patterns within the cluster and specify the level of proximity between them. In the end, the results for the random mode for P-FCM provide worse results than standard FCM. It is obvious in the random mode that because the patterns are being chosen in a fully random way, it is easy to skip the patterns likely to be misclassified and enter the proximity for the patterns well associated with correct clusters. This is an obvious fact but important to remember when comparing between the modes.

The user supervision for the PSP-FCM algorithm is a hybrid of two previous techniques. Typically for mergers of different techniques, it shares the advantages and disadvantages of the methods. The accuracy of PSP-FCM exhibits more stable behaviour than P-FCM, but usually obtains lower performance than PS-FCM. However in some cases (Thyroid Gland) it may obtain equally high accuracy for smaller amount of user supervision than any of the other methods alone.

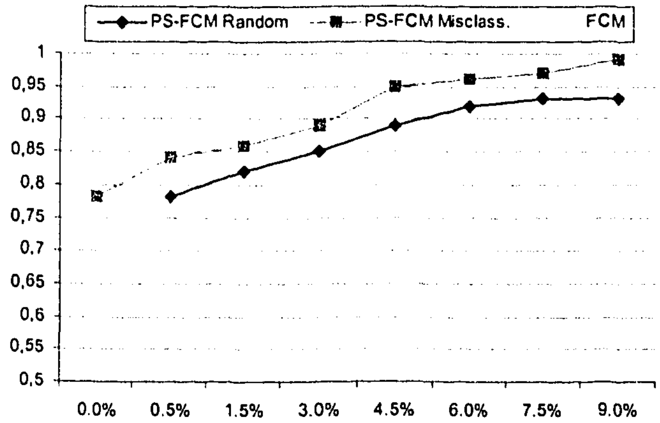


Figure 6-3. Random and classification mode comparison for the PS-FCM.

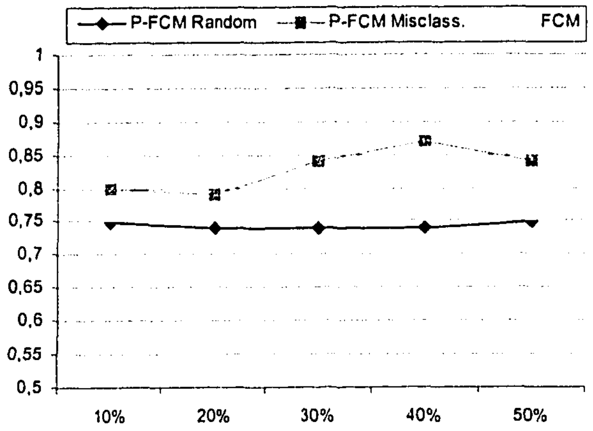


Figure 6-4. Random and classification mode comparison for the P-FCM.

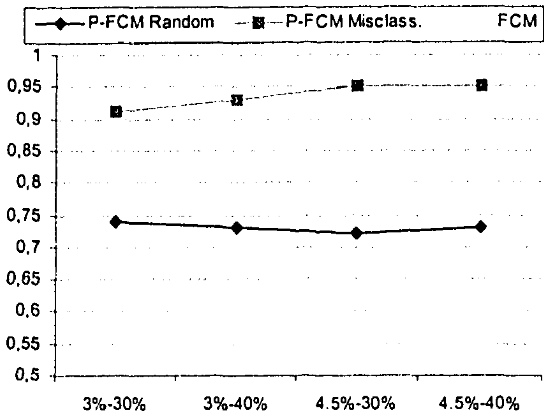


Figure 6-5. Random and classification mode comparison for the PSP-FCM.

There is another advantage of the misclassification mode over the random mode that has not yet been mentioned: it is that there only needs to be a smaller amount of user supervision to enter in order to improve the results, and only the application of the user input in the places where it is necessary. For instance, in the Dermatology database the results reached 99% for only misclassified patterns whereas the random mode obtained only 92% having 12% of all patterns entered.

In the experiments we also made a comparison between the values by which the proximity hints were adjusted for P-FCM algorithm. The following was evaluated: the modification by the constant (*0.5*) and the value *1/c*, where c is the number of clusters. The first approach worked well in the experiments. The second one performed better for a smaller number of clusters achieving, in fact, the value *0.5* for *c=2*. However, for a larger number of clusters the change was too small to be significant.

Based on the above discussion and the experimental part, it is advisable to proceed with the misclassification mode as it performs better, exhibits more desirable features, and is more efficient. These assumptions will be evaluated by applying them for the next dataset – Web pages collection. All above conclusions and observations form a well-defined protocol of actions to be taken in a certain experimental schema. The protocol is presented as a simplified decision tree in the Fig. 6-6.

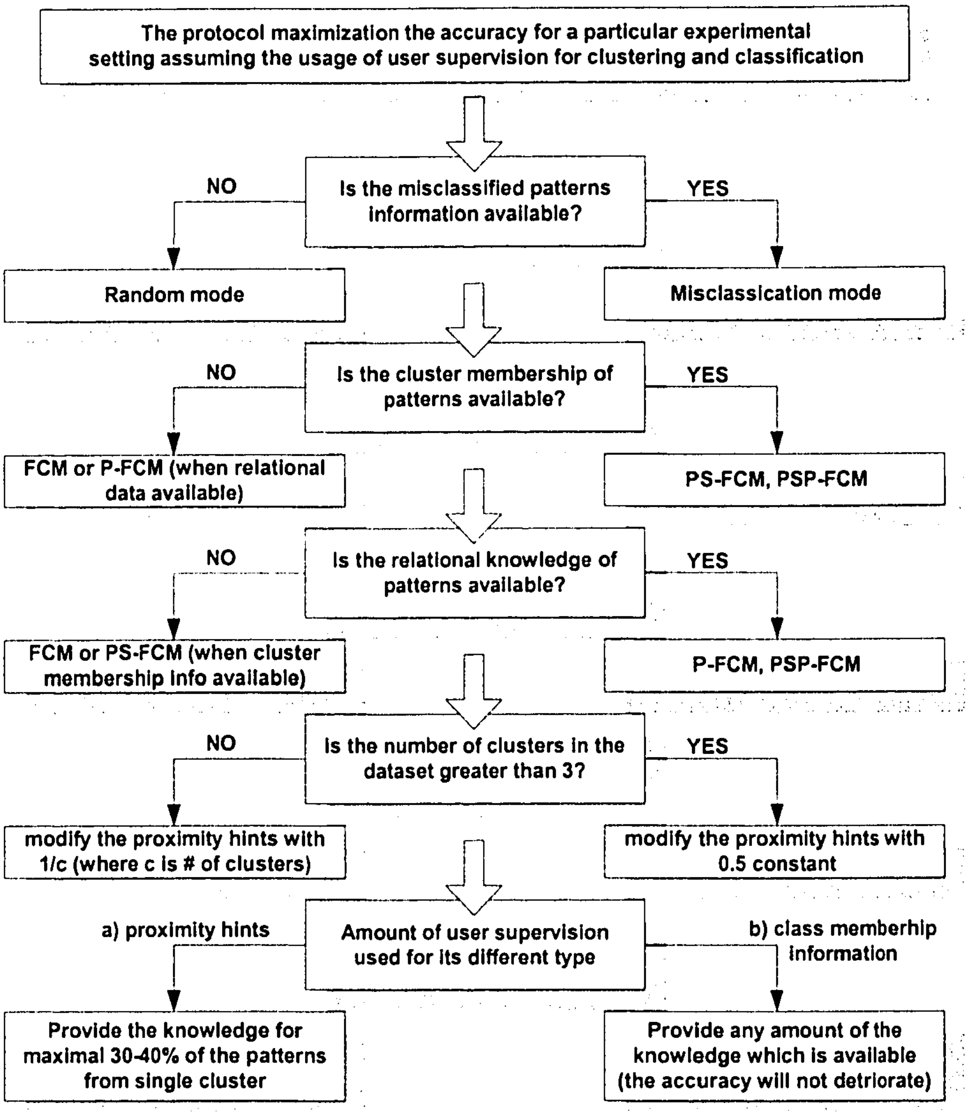

The protocol maximization the accuracy for a particular experimental setting assuming the usage of user supervision for clustering and classification
Is the misclassified patterns information available?
NO
YES
Random mode
Misclassication mode
Is the cluster membership of patterns available?
NO
YES
FCM or P-FCM (when relational data available)
PS-FCM, PSP-FCM
Is the relational knowledge of patterns available?
NO
YES
FCM or PS-FCM (when cluster membership info available)
P-FCM, PSP-FCM
Is the number of clusters in the dataset greater than 3?
NO
YES
modify the proximity hints with 1/c (where c is # of clusters)
modify the proximity hints with 0.5 constant
Amount of user supervision used for its different type
a) proximity hints
b) class memberhip information
Provide the knowledge for maximal 30-40% of the patterns from single cluster
Provide any amount of the knowledge which is available (the accuracy will not detriorate)

Figure 6-6. The protocol of application of user supervision.

6.4 Web Pages dataset experiment

Partitioning of a set of Web pages into clusters refers to formation of groups in dataset of "similar' Web documents. In such cases a user would expect to find in the same cluster similar pages in terms of their semantic (thematic) content. For instance, pages describing windsurfing are more similar to sailing pages than to e.g. basketball pages (or other sports related pages) and should be assigned to the more general (broader) category of sailing. The proper grasping of the concept of similarity between the pages plays the key role here.

This study concerned with Web pages dataset starts with an extensive part analysing the information content of Web pages, which provide interesting and valuable insights about the structure [4]. This is followed by the experimentation with the user supervision, where user input will be used to improve the cluster structure.

6.4.1 Web Pages dataset analysis with FCM

6.4.1.1 Feature space and dimensionality

We analyzed the first Web pages dataset. Dimensionality of the feature vectors varies significantly from 2 dimensions up to 263 in the longest vector. The average dimensionality is equal 26 keywords per vector. The number of occurrences of keywords (relative frequencies) in the vectors used in the experiments varies from 1 up to 129. The average number of occurrences of keywords was equal to 6. The average distribution of keywords in the documents reveals some regular patterns. Usually, only few keywords have higher and significantly larger number of occurrences and the rest has only two and one occurrences (Fig. 6-6.).

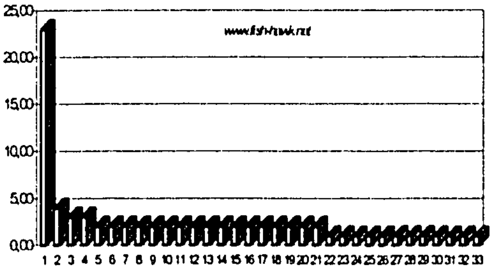


Figure 6-7. The histogram of the number of occurrences (relative frequencies) of the keywords for the Web page: www.fish-hawk.org - fishing online resources.

As we can observe in Fig. 6-7, flat distribution of numbers of occurrences of keywords was equalized by tf-idf scheme (the set of Web pages from the 'fishing' category only was taken to obtain the tf-idf weights). The modified weights gained different distribution of values and this is the reason why the order of the keywords was slightly changed. Table 6-31 presents first 15 out of 33 keywords.

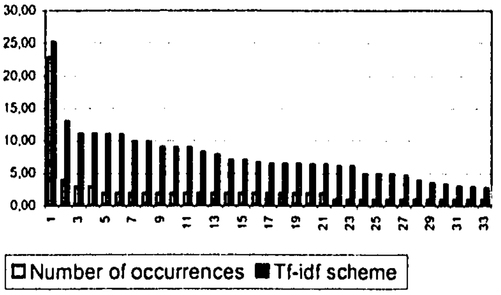


Figure 6-8. Comparison between weighting schemas for the Web page: www.fish-hawk.org.

Table 6-31. Comparison between weighting schemas for the http://www.fish-hawk.org Web page.

#	Relative frequency		Tf-idf scheme	
	Keyword	Weight	Keyword	Weight
1	*fish*	*23*	*fish*	*25.24*
2	*bass*	*4*	*manufactur*	*13.05*
3	*river*	*3*	*river*	*11.15*
4	*tip*	*3*	*tip*	*11.15*
5	*canada*	*2*	*map*	*11.05*
6	*hawk*	*2*	*ottawa*	*11.05*
7	*lake*	*2*	*hawk*	*9.88*
8	*largemouth*	*2*	*smallmouth*	*9.88*
9	*link*	*2*	*largemouth*	*9.05*
10	*manufactur*	*2*	*muski*	*9.05*
11	*map*	*2*	*ontario*	*9.05*
12	*muski*	*2*	*bass*	*8.26*
13	*ontario*	*2*	*canada*	*7.88*
14	*ottawa*	*2*	*site*	*7.05*
15	*pictur*	*2*	*walley*	*7.05*

6.4.1.2 Construction of feature space for FCM

A set of feature vectors accepted by the FCM algorithm is denoted by: $X = \{x_1, x_2, ..., x_n\}$ where $x_i \in R^p$. Apparently, the vectors constructed from keywords have various dimensions thus we need a process, which would adopt these vectors to the algorithm's constraints. Therefore the concept of the universe vector was introduced. The universe vector gathers all terms from entire Web document collection taken to clustering process. In the next step, the length of every vector from the Web document set is extended to the dimension of the universe vector filling the extended spaces with '0' frequency. Finally, all the weights in the vectors are normalized to the interval [0,1] and sorted in the descending order.

6.4.1.3 Fuzzification coefficient

Higher values of the fuzzification coefficient (fuzzifier m) support the trend to spread the degrees of 'fuzzy' membership of the vectors over all clusters. Alternatively, when assumed $m \to 1$, the fuzzy partition matrix will consist only of 0's and 1's and in this case entire value of the fuzzy membership would appear only in one cluster (hard c-partitions).

The experiments were performed with the standard value of the fuzzifier $m=2$. This value turned out to be too large for two largest sets and the fuzzy membership values for the vectors in the partition matrix had the same values for each cluster. The value equal to $m=1.1$ was assumed in this case.

Fig. 6-8 and Fig. 6-9 show the partition matrix for the parameter m equal 2.0. For simplicity and clarity of figures we presented the results of clustering of the dataset consisting of two categories 'Fishing' and 'Halloween' category, there were gathered 92 pages in the whole dataset (Section 3. Datasets description). Evidently, the algorithm formed two easily distinguishable clusters.

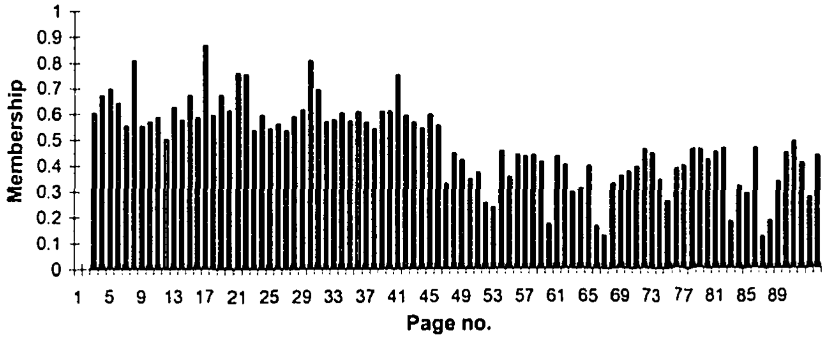


Figure 6-9. Fuzzy partition matrix membership grades graph: Halloween category.

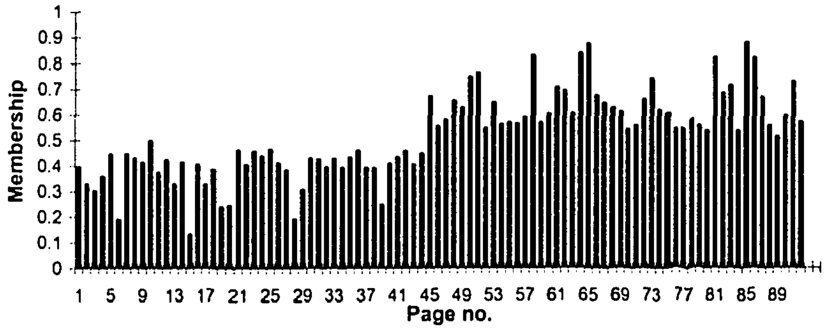


Figure 6-10. Fuzzy partition matrix membership grades graph: Fishing category.

For the same dataset we tested the convergence of the algorithm assuming different values of the parameter *m*. The algorithm converged very fast and in a few iterations (4-7) we obtained the final partition matrix. Thus we assumed 10 iterations as the number of iterations throughout the experimentation. In Fig. 6-10 values of the objective function for m equal 1.5 and 1.1 are presented. In general, decreasing the value of *m* eliminates the amount of 'fuzziness' from the partition matrix and the membership values tend to be closer to 1. We set up the value m=1.1. In practice we obtained very close results for *m* in the range 1.1–2.0 but higher values generated undesired amount of fuzziness in the partition matrix and FCM wasn't able to determine proper number of clusters for larger datasets.

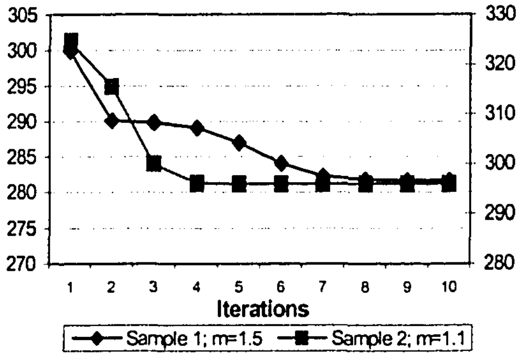


Figure 6-11. The objective function values for 10 iterations.

6.4.1.4 Prototypes analysis

In order to evaluate clustering process the prototypes of the clusters were examined more carefully. The results seem to confirm the intuition. There were investigated terms and weights of the prototypes from both clusters with the vectors having the largest degree of membership in the cluster. The results make perfect sense. Clearly, in the cluster centers and selected pages the largest weights gained keywords the mostly associated with the category i.e. the keyword 'fish' in 'Fishing' category and the keyword 'Halloween' in 'Halloween' category. In the first 20 terms with the largest weights the keywords appearing in both cluster centre and Web page were marked with shaded area (Table 6-32).

Table 6-32. Comparison between clusters centers and Web pages with the highest membership values in the clusters (Relative frequencies used).

Cluster 1 (Fishing)				Cluster 2 (Halloween)					
Prototype		Page with highest membership degree = 0.999		Prototype		Page with highest membership degree = 0.999		Misclassified page with membership degree = 0.663	
Terms	*Weights*	*Terms*	*Weights*	*Terms*	*Weights*	*Terms*	*Weights*	*Terms*	*Weights*
fish	*0.924*	***fish***	*1.000*	***halloween***	*0.816*	***halloween***	*1.000*	*catfish*	*1.000*
bass	*0.188*	***bass***	*0.174*	*haunt*	*0.241*	***costum***	*0.111*	*blue*	*0.111*
trout	*0.162*	*river*	*0.130*	***costum***	*0.116*	*movi*	*0.111*	*channel*	*0.111*
pike	*0.108*	*tip*	*0.130*	*hous*	*0.109*	*adult*	*0.056*	*flathead*	*0.111*
angl	*0.106*	*canada*	*0.087*	***game***	*0.106*	*clipart*	*0.056*	*link*	*0.111*
carp	*0.097*	*hawk*	*0.087*	*ghost*	*0.104*	*craft*	*0.056*	*pictur*	*0.111*
hunt	*0.093*	***lake***	*0.087*	*horror*	*0.099*	*curti*	*0.056*	*stori*	*0.111*
boat	*0.090*	*largemouth*	*0.087*	*trivia*	*0.087*	***decor***	*0.056*	*aaron*	*0*
tackl	*0.089*	*link*	*0.087*	*treat*	*0.082*	***game***	*0.056*	*abend*	*0*
sport	*0.088*	*manufactur*	*0.087*	*witch*	*0.078*	*jami*	*0.056*	*abigail*	*0*
bait	*0.084*	*map*	*0.087*	*parti*	*0.078*	*lee*	*0.056*	*abroad*	*0*
lure	*0.084*	*muski*	*0.087*	*trick*	*0.077*	*mask*	*0.056*	*accommod*	*0*
fly	*0.084*	*ontario*	*0.087*	*holidai*	*0.075*	*pictur*	*0.056*	*activ*	*0*
angler	*0.082*	*ottawa*	*0.087*	*hallow*	*0.071*	*prop*	*0.056*	*adam*	*0*
lake	*0.070*	*pictur*	*0.087*	*stori*	*0.071*	***spooki***	*0.056*	*addon*	*0*
outdoor	*0.067*	***pike***	*0.087*	***spooki***	*0.068*	*aaron*	*0*	*adult*	*0*
rod	*0.063*	*site*	*0.087*	*pumpkin*	*0.066*	*abend*	*0*	*advertis*	*0*
page	*0.063*	*smallmouth*	*0.087*	*recip*	*0.065*	*abigail*	*0*	*affili*	*0*
photo	*0.055*	*stori*	*0.087*	*scari*	*0.065*	*abroad*	*0*	*air*	*0*
guid	*0.055*	***trout***	*0.087*	***decor***	*0.059*	*accommod*	*0*	*alabama*	*0*

It is easily noticeable that the first terms in the vectors, which have the largest impact in the clustering process (the largest weights) appeared within first 20 keywords of 1128-dimensional vectors in both cluster centre and Web page. The only miss-classified page was *http://www.catfished.com* (Fig. 6-11), which contains a very small number of keywords.

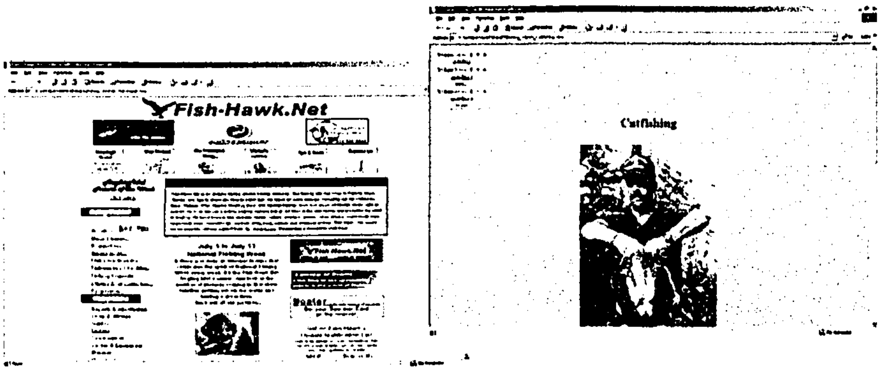


Figure 6-12. The 'best' representative page from Fishing category (left) and miss-classified page (right).

The keyword 'catfish' with the largest weight appear in both prototypes only after first 80 keywords, while other keywords less associated with fishing category like 'pictur' (after stemming) appeared earlier (with larger weight) in the 'Halloween' category prototype than 'catfish' keyword. This could be the reason for miss-classification. Moreover, the membership degree of this page to 'Halloween' category is not very high (equal to 0.663) what explains that this page is not strongly associated with the category it was assigned to. As presented in Table 6-32 prototypes from both categories do not have any terms in common within the first 20 terms.

In Fig. 6-12 are presented the prototypes from the dataset 1 and their weights' distribution. It is remarkable that few dominant keywords have distinctly higher values of weights.

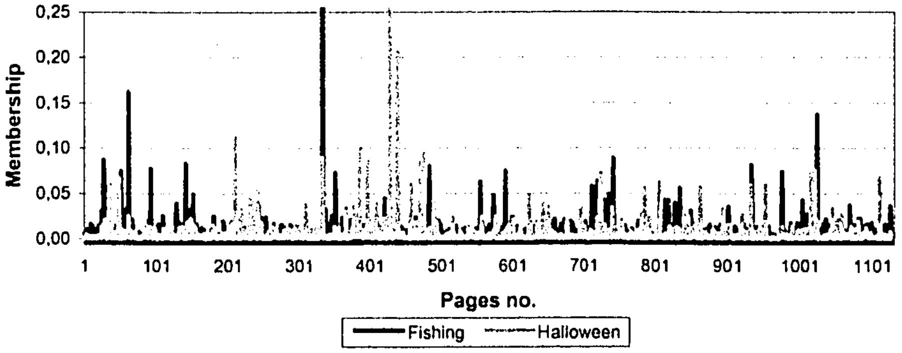


Figure 6-13. Prototypes and their weights' distribution. (Relative frequencies used).

6.4.1.5 Results

This section is concerned with extensive numeric experimentation that will lead to some general observations as to the behaviour of the FCM in the highly dimensional textual space of the Web pages. The accuracy is calculated for relative frequencies and tf-idf scheme for all datasets. In this experiment it is introduced also another similarity measure. Besides the Euclidean distance it was used another measure of similarity, derived from well known in Information Retrieval cosine similarity measure.

$$sim(u,v) = 1 - \cos sim(u,v) = 1 - \frac{u * v}{|u||v|} \text{ for } \cos \in < 0, \frac{\pi}{2} > \tag{6}$$

Where *cossim(u,v)* is the cosine similarity measure. The cosine of the angle between two vectors does not fulfill metric properties, in fact does not meet the criteria of triangle equality and *d(x,x) = 0*. Above derived similarity still doesn't fulfill triangle equality but meets the second condition. The obtained results are very good. This similarity measure produced better results and is more stable for both weighting schemas then the Euclidean distance.

Relative frequencies scheme performed better than tf-idf. Tf-idf scheme provided worse results and was sensitive to initialization. According to the observations this is due to equalization of weights in tf-idf method where a few predominant keywords from the previous scheme (relative frequencies) lost their inevitable prevalence. The second reason is the relatively rare vector space. In some cases it was easy to foresee the drop of the accuracy. This happened for the 9th dataset. The categories of *Email, Web hosting* and *Web design* used very similar and general set of keywords describing them, e.g. 'web', 'free', 'host', 'site', 'page'. For this dataset, the accuracy was considerably lower then in other ones.

Table 6-33. Accuracy results for Euclidean distance and 1–cosine similarity, m=2.0.

#	Dataset	Dimension	Relative frequency		Tf-idf weighting scheme	
			Euclidean distance	1- cosine	Euclidean distance	1- cosine
1	*Fishing; Halloween*	*1128*	0.99 ± 0.0	0.99 ± 0.0	0.92 ± 0.005	0.99 ± 0.0
2	*Fishing;Halloween; Golf*	*1849*	0.98 ± 0.0	0.95 ± 0.127	0.97 ± 0.008	0.98 ± 0.045
3	*Fishing;Halloween; Careers*	*1627*	0.97 ± 0.0	0.99 ± 0.0	0.96 ± 0.014	0.99 ± 0.0
4	*Fishing; Halloween; Gymnastics*	*1833*	0.95 ± 0.111	0.94 ± 0.112	0.60 ± 0.014	0.94 ± 0.115
5	*Fishing; Halloween; Gymnastics; Career;*	*2272*	0.94 ± 0.086	0.89 ± 0.136	0.60 ± 0.136	0.91 ± 0.121
6	*Clothing;Collecting; Cooking; Fishing; Golf; Halloween; Kids Art; Snowboarding*	*3660*	0.78 ± 0.075	0.88 ± 0.073	0.76 ± 0.041	0.85 ± 0.090
7	*Airports; Cycling; Plants*	*5450*	0.99 ± 0.0	0.99 ± 0.0	0.63 ± 0.038	0.98 ± 0.0
8	*Snowboarding; Skating;*	*1983*	0.96 ± 0.075	0.96 ± 0.076	0.67 ± 0.145	0.95 ± 0.104

	Snowmobiling; *Curling*					
9	*Email; Web hosting; Web design*	*3249*	0.83 ± 0.023	0.85 ± 0.005	0.60 ± 0.026	0.81 ± 0.067

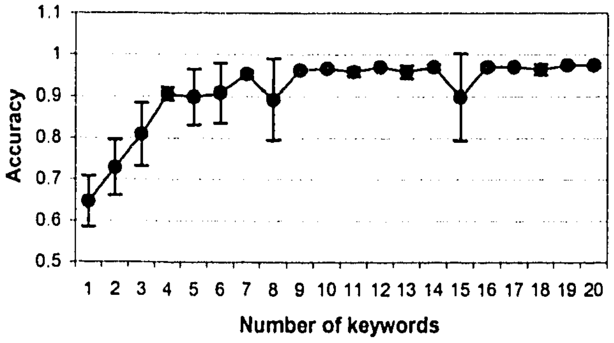
6.4.1.6 Feature space improvement

However the accuracy of the classification is quite impressive, this approach has one significant disadvantage. A large number of keywords in Web pages causes the dimensionality explosion. In order to overcome this problem each vector was truncated. Only the most important keywords with the largest weights remained. After such an operation the dimensionality increase linearly according to the number of pages and categories.

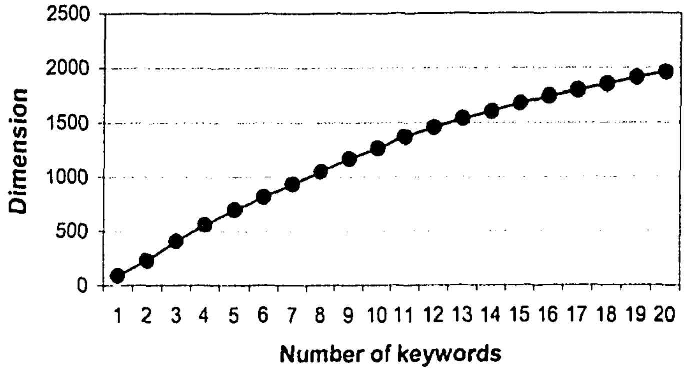
The accuracy was evaluated for the vectors' lengths from 1 up to 20 for the dataset with nested categories (Fig. 4-3). It is easily seen that the accuracy increases with the number of keywords included in a vector (Fig. 6-13a). In Fig. 6-13a we present growth of the dimensionality of the vectors. At the very small vectors' length the accuracy can behave in not a very stable way (can vary depending on a specific keyword) but after reaching the length of 17-18, accuracy stabilizes at the level of maximal accuracy. Based on the results of this experiment the results for the vectors' length were truncated to 5 and 20 keywords. With the length of 5 only the most dominant keywords are used, which is sufficient to obtain reasonable results and considerably decrease dimensionality.

The results with the vectors' length of 20 proved that it is not necessary to include all keywords in the vectors. The accuracy will not be further improved after including more keywords. There is an interesting fact that can be observed with the vectors truncated to the length equal to 20. The classification results after such operation were improved (for most of datasets) in comparison to full vector length and are the best from all our experiments. This fact allows arguing that at the beginning of the vector appear the most significant keywords. Less important keywords or keywords accidentally included in the vector appear on further positions in the vector. These keywords are eliminated after truncating of the vector and do not lower the accuracy.

The accuracy results are presented in Table 6-34.



(a)



(b)

Figure 6-14. The accuracy evaluation. For each length there is shown standard deviation for 5 experiments. (a) The growth of the dimensionality, (b) Relative frequencies used.

Table 6-34. Accuracy results after truncating the vectors lengths calculated using the relative frequencies scheme, m=2.0, 10 iterations.

		Accuracy (max vector length = 5)			Accuracy (max vector length = 20)		
#	Dataset	Dimension	Euclidean distance	1 – cosine	Dimension	Euclidean distance	1 – cosine
1	*Fishing; Halloween (92 pages)*	251	0.96 ± 0.0	1.0	733	1.00	1.00
2	*Fishing; Golf; Halloween; (141)*	353	0.96 ± 0.002	0.98 ± 0.0	1031	0.99 ± 0.0	0.99 ± 0.0
3	*Fishing; Halloween; Careers (147)*	356	0.95 ± 0.002	0.98 ± 0.01	1059	0.97 ± 0.0	0.99 ± 0.0
4	*Fishing; Halloween; Gymnastics (183)*	426	0.90 ± 0.005	0.89 ± 0.14	1142	0.92 ± 0.14	0.88 ± 0.1
5	*Fishing; Halloween; Gymnastics; Career; (238)*	523	0.90 ± 0.005	0.85 ± 0.12	1440	0.94 ± 0.08	0.88 ± 0.1
6	*Clothing; Collecting; Cooking; Art; Fishing; Golf; Halloween; Kids Snowboarding (302)*	722	0.76 ± 0.041	0.77 ± 0.07	2083	0.79 ± 0.03	0.88 ± 0.0
7	*Airports; Cycling; Plants (744)*	1332	0.93 ± 0.055	0.95 ± 0.05	3496	0.94 ± 0	0.99 ± 0.0
8	*Snowboarding; Snowmobiling; Skating; Curling (244)*	531	0.92 ± 0.003	0.86 ± 0.12	1462	0.97 ± 0	0.94 ± 0.0
9	*Email; Web hosting; Web design (501)*	676	0.67 ± 0.096	0.76 ± 0.02	2102	0.78 ± 0.09	0.83 ± 0.0

The truncation of keywords to 20 turned out to be a suitable trade-off in reducing greatly the dimensionality while preserving the accuracy on very high level. However, the number of features contained in the vectors is still large. We tried to overcome this problem with a well-known technique in information retrieval – Latent Semantic Indexing (LSI).

6.4.1.7 Latent Semantic Indexing (LSI)

In the vector space model, a vector is used to represent a Web page. Each feature of the vector corresponds to a keyword and reflects the importance of this particular keyword in the semantics of the document. A set of documents can be described in terms of a t x d term-by-document matrix $\mathbf{A}$. The t-terms are all keywords from all documents. The d-vectors represent the d-documents and form the columns of matrix $\mathbf{A}$. Thus, we can perceive the a_{ij} element of $\mathbf{A}$ as a magnitude of importance of *i-th* term in the document *j*. The straightforward approach assumes the number of occurrences of term *i* in the document *j*. In the algebraic meaning the document vectors span the semantic content contained in the document set. Meanwhile, the geometrical interpretation visualizes the documents vectors in the vector space model and allows for computing the distance between them. The power of the LSI is apparent and important in two aspects: (i) a significant dimension reduction based on rank-k approximation, (ii) revealing the hidden (latent) semantic information of the documents' set.

It is common for vector space models of Web documents to suffer from high dimensionality due to a large number of keywords in documents. To reduce dimensionality the LSI uses the SVD method (Appendix 2). After decomposition of the t x d term-by-document matrix $\mathbf{A}$ into $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$, we can obtain the new vectors reduced to $k = \min(t, d)$ by multiplying $\mathbf{V}^T$ by Σ (we can interpret this operation as rescaling of the axes in k dimensional space, by singular values of Σ) [5].

In terms of revealing the latent information LSI technique overcomes two important problems in the information retrieval: synonymy and polysemy. This is due to the replacement of the keyword weights with other set of entities, which are more reliable indicants. The detailed description and explanations how LSI overcomes these problems can be found in [1], [5], [6].

The feature vectors' dimension was reduced with LSI. The accuracy of the FCM is presented in Table 6-35. The obtained accuracy decreased considerably for the Euclidean distance and the similarity based on the cosine similarity. Yet, the second similarity measure recorded much better results than the Euclidean distance.

Table 6-35. Accuracy results after reducing the dimensionality with the LSI, m=2.0, 10 iterations.

		Accuracy (max vector length = 20)		
#	Dataset	LSI Reduced Dimension	Euclidean distance	1 – cosine
1	*Fishing; Halloween (92 pages)*	92	0.65 ± 0.005	0.99 ± 0.001

2	*Fishing; Golf; Halloween; (141)*	141	0.74 ± 0.0	0.99 ± 0.004
3	*Fishing; Halloween; Careers (147)*	147	0.63 ± 0.003	0.94 ± 0.09
4	*Fishing; Halloween; Gymnastics (183)*	183	0.61 ± 0.01	0.93 ± 0.02
5	*Fishing; Halloween; Gymnastics; Career; (238)*	238	0.41 ± 0.03	0.49 ± 0.06
6	*Clothing; Collecting; Cooking; Fishing; Golf; Halloween; Kids Art; Snowboarding (302)*	302	0.44 ± 0.03	0.35 ± 0.06
7	*Airports; Cycling; Plants (744)*	744	0.68 ± 0.01	0.68 ± 0.004
8	*Snowboarding; Snowmobiling; Skating; Curling (244)*	244	0.61 ± 0.02	0.79 ± 0.08
9	*Email; Web hosting; Web design (501)*	501	0.49 ± 0.01	0.72 ± 0.05

6.4.1.8 User supervision for clustering and classification

In this section we will investigate the possible improvement of accuracy when applying user supervision. The user input would be applied via mechanisms of the PS-FCM, the P-FCM and the PSP-FCM. The experiments were performed for the random mode and the misclassification mode. Because of a number of datasets to test we fixed the amount of user supervision available to the same amount for all datasets of Web documents choosing the parameters based on previous experiments. For the PS-FCM we assumed 10% of class assignment information per group in the random mode. For the misclassification mode we entered information for misclassified patterns. The parameters for the P-FCM were constructed in the following way. The number of the proximity hints entered for misclassified patterns per group was equal 30% of patterns from clusters. In the random mode we introduced the average misclassified patters – this was 30% of patterns from each cluster. Assumption of theses values for the P-FCM allowed for comparing the relative performance of the random with the misclassification mode. The parameters for the PSP-FCM were combined from the PS-FCM and the P-FCM.

In Table 6-36 there are presented the results for clustering experiment for the random mode, and in Table 6-37 for the misclassification mode. The clustering results are followed by the classification results in Table 6-38 and Table 6-39. The training and testing set were adjusted according to 70%-30% ratio. The outcomes were averaged in 10 trails.

Table 6-36. Random mode accuracy results for clustering after reducing the dimensionality with the LSI, m=2.0 (for sets 6,7 m=1.1), 10 iterations.

#	Dataset	Accuracy (max vector length = 20)			
		1 – cosine			
		FCM	PS-FCM	P-FCM	PSP-FCM
1	*Fishing; Halloween (92 pages)*	1.0	1.0	1.0	1.0
2	*Fishing; Golf; Halloween; (141)*	0.98 ± 0.01	0.99 ± 0	0.96 ± 0	0.87 ± 0.08
3	*Fishing; Halloween; Careers (147)*	0.99 ± 0.002	1.0	1.0	0.78 ± 0.20
4	*Fishing; Halloween; Gymnastics (183)*	0.92 ± 0.10	0.92 ± 0	0.89 ± 0.006	0.89 ± 0.008

5	*Fishing; Halloween; Gymnastics; Career; (238)*	0.83 ± 0.08	0.91 ± 0.001	0.88 ± 0.04	0.74 ± 0.15
6	*Clothing; Collecting; Cooking; Fishing; Golf; Halloween; Kids Art; Snowboarding (302)*	0.83 ± 0.04	0.92 ± 0	0.88 ± 0.03	0.66 ± 0.19
7	*Airports; Cycling; Plants (744)*	0.78 ± 0.01	0.98 ± 0	0.79 ± 0.03	0.73 ± 0.14
8	*Snowboarding; Snowmobiling; Skating; Curling (244)*	0.81 ± 0.11	0.89 ± 0	0.96 ± 0.003	0.79 ± 0.16
9	*Email; Web hosting; Web design (501)*	0.86 ± 0.02	0.90 ± 0	0.88 ± 0.001	0.70 ± 0.17

Table 6-37. Misclassification mode accuracy results for clustering after reducing the dimensionality with the LSI, m=2.0 (for sets 6,7 m=1.1), 10 iterations.

		Accuracy (max vector length = 20)			
#	Dataset		1 – cosine		
		FCM	PS-FCM	P-FCM	PSP-FCM
1	*Fishing; Halloween (92 pages)*	1.0	1.0	1.0	1.0
2	*Fishing; Golf; Halloween; (141)*	0.98 ± 0.01	1.0	0.98 ± 0.01	1.0
3	*Fishing; Halloween; Careers (147)*	0.99 ± 0.002	1.0	1.0	1.0
4	*Fishing; Halloween; Gymnastics (183)*	0.92 ± 0.10	0.93 ± 0.01	0.92 ± 0.07	0.93 ± 0.02
5	*Fishing; Halloween; Gymnastics; Career; (238)*	0.83 ± 0.08	0.99 ± 0.01	0.78 ± 0.06	0.99 ± 0.007
6	*Clothing; Collecting; Cooking; Fishing; Golf; Halloween; Kids Art; Snowboarding (302)*	0.83 ± 0.04	0.99 ± 0.002	0.90 ± 0.04	0.99 ± 0.002
7	*Airports; Cycling; Plants (744)*	0.78 ± 0.01	0.98 ± 0	0.79 ± 0.01	0.98 ± 0
8	*Snowboarding; Snowmobiling; Skating; Curling (244)*	0.81 ± 0.11	0.98 ± 0.004	0.98 ± 0.003	0.98 ± 0.005
9	*Email; Web hosting; Web design (501)*	0.86 ± 0.02	0.99 ± 0.01	0.89 ± 0.007	0.99 ± 0.003

Table 6-38. Random mode accuracy results for classification after reducing the dimensionality with the LSI, m=2.0 (for sets 6,7 m=1.1), 10 iterations.

		Accuracy (max vector length = 20)			
#	Dataset		1 – cosine		
		FCM	PS-FCM	P-FCM	PSP-FCM
1	*Fishing; Halloween (92 pages)*	1.0	1.0	1.0	1.0
2	*Fishing; Golf; Halloween; (141)*	0.87 ± 0	0.87 ± 0	0.87 ± 0	0.87 ± 0.08
3	*Fishing; Halloween; Careers (147)*	0.94 ± 0	0.98 ± 0	0.92 ± 0	0.79 ± 0.18
4	*Fishing; Halloween; Gymnastics (183)*	0.74 ± 0	0.60 ± 0	0.59 ± 0.008	0.51 ± 0.08
5	*Fishing; Halloween; Gymnastics; Career; (238)*	0.59 ± 0.16	0.59 ± 0	0.82 ± 0.03	0.71 ± 0.12
6	*Clothing; Collecting; Cooking; Fishing; Golf; Halloween; Kids Art; Snowboarding (302)*	0.79 ± 0.09	0.70 ± 0.002	0.71 ± 0	0.68 ± 0.21
7	*Airports; Cycling; Plants (744)*	0.77 ± 0.04	0.97 ± 0	0.74 ± 0.01	0.77 ± 0.17
8	*Snowboarding; Snowmobiling; Skating; Curling (244)*	0.82 ± 0.15	0.97 ± 0	0.97 ± 0.005	0.80 ± 0.16
9	*Email; Web hosting; Web design (501)*	0.77 ± 0.009	0.78 ± 0	0.78 ± 0	0.65 ± 0.12

Table 6-39. Misclassification mode accuracy results for classification after reducing the dimensionality with the LSI, m=2.0 (for sets 6,7 m=1.1), 10 iterations.

#	Dataset	Accuracy (max vector length = 20)			
		1 – cosine			
		FCM	PS-FCM	P-FCM	PSP-FCM
1	*Fishing; Halloween (92 pages)*	1.0	1.0	1.0	1.0
2	*Fishing; Golf; Halloween; (141)*	0.87 ± 0	0.98 ± 0	0.98 ± 0.006	0.98 ± 0
3	*Fishing; Halloween; Careers (147)*	0.94 ± 0	0.98 ± 0	0.98 ± 0	0.98 ± 0
4	*Fishing; Halloween; Gymnastics (183)*	0.74 ± 0	0.77 ± 0.01	0.80 ± 0.07	0.78 ± 0
5	*Fishing; Halloween; Gymnastics; Career; (238)*	0.59 ± 0.16	0.93 ± 0.01	0.61 ± 0.03	0.79 ± 0
6	*Clothing; Collecting; Cooking; Fishing; Golf; Halloween; Kids Art; Snowboarding (302)*	0.79 ± 0.09	0.72 ± 0.002	0.71 ± 0	0.71 ± 0
7	*Airports; Cycling; Plants (744)*	0.77 ± 0.04	0.95 ± 0	0.76 ± 0.04	0.94 ± 0
8	*Snowboarding; Snowmobiling; Skating; Curling (244)*	0.82 ± 0.15	0.98 ± 0.006	0.96 ± 0.01	0.98 ± 0.006
9	*Email; Web hosting; Web design (501)*	0.77 ± 0.009	0.80 ± 0.004	0.77 ± 0.01	0.81 ± 0.007

6.4.2 Conclusions

The clustering results of the standard FCM are based on textual information only. Although it is possible to group similar, thematic pages and obtain satisfactory, and sometimes even good results, as the experiment's results showed, another source of input is immensely beneficial. The user guidance may rely on other, non-textual characteristics of Web pages, such as multimedia, links and layout, and improve the performance. We can also observe that user supervision improved the results for both modes, although the results of the random mode were very variable. For some datasets we observed better results, but for the other datasets the results were worse than those of the FCM. Very meaningful insights can give us standard deviation values for the random mode. They were relatively high. For every set the values were much higher than the values from the misclassification mode.

Equally good performance was obtained from the PS-FCM and the PSP-FCM. In the clustering experiment we obtained the results of 99% for nearly of all datasets. The P-FCM also performed well.

The incorporation of the user knowledge in the classification mode provided us with good results. In fact there was a noticeable improvement of the accuracy for the patterns from the testing set. Higher accuracy values were obtained for all datasets besides the 6th dataset. This dataset contained 8 different categories, and the amount of user input that was applied might not have been sufficient.

6.5 Bibliography

1. Berry M. W., Dramac Z., Jessup E. R., Matrices, vector spaces, and information retrieval, *SIAM Rewiev*, Vol. 41, No. 2, 1999, pp. 335-362.

2. Bezdek J.C., "Pattern recognition with fuzzy objective function algorithms", Plenum Press, New York, 1981.
3. Blake, C. L, Merz, C. J, UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html], Irvine, CA: University of California, Department of Information and Computer Science, 1998.
4. Brzeminski, P., Pedrycz, W., Textual-based Fuzzy Clustering of Web Documents, *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, Vol. 12(6), 2004, pp.715-744.
5. Deerwester S., Dumais S. T., Furnas G. W., Landauer T. K., Harshman R., Indexing by latent semantic indexing, Journal of the American Society for Information Science, 41(6), 1990, pp. 391-407.
6. Lochbaum K. E., Streeter L. A., Comparing and combining the effectiveness of latent semantic indexing and the ordinary vector space model for information retrieval, *Information Processing & Management*, Vol. 25, No. 6, 1989, pp. 665-676.

7 Conclusions

7.1 General Discussion

The user supervision can be beneficial in many practical applications. It can immensely help in recovering the distorted structure of the dataset containing a number of outliers, model the relationships between the complex objects and their features when it comes to Web exploration, significantly aid the work of the clinical researcher trying to classify new cases of illnesses in combination with accumulated knowledge, or allow the meteorologist to more efficiently classify the elements of a tornado based on satellite weather maps. As we can see, the possible applications of user input are various and broad. Interestingly, it is surprising that the algorithms and the enhancements to existing methods allowing the user to interact with the learning process are narrow.

In this work we were concerned with the possible user supervision algorithmic extensions to the FCM algorithm. In particular, the experiments allowed examination of, in a detailed way, the available means for adding user supervision to FCM (Fuzzy C-Means) in a form for class assignment information provided by PS-FCM (Partially Supervised Fuzzy C-Means), and the proximity hints offered by P-FCM (Proximity-based Fuzzy C-Means). Moreover, the combined effort of these two approaches in PSP-FCM (Partially Supervised Proximity-based Fuzzy C-Means) was examined. Specifically, by taking into account the synthetic data, Machine Learning datasets, and the Web documents datasets, the PS-FCM approach exhibited the best average performance. The approach of P-FCM seemed to be more variable and usually performed worse than PS-FCM, but still considerably better than standard FCM without any user input. The performance of PSP-FCM provided high accuracy, close to the PS-FCM accuracy, and in some cases it proved to perform better than any of the other methods alone for the same number of parameters. It follows that both sources of user supervision can be successfully applied together.

The P-FCM does not obtain information about the class membership, but instead it allows for the specification of similarities between patterns. This kind of information is not directly related to class membership, but it carries the implicit information that the patterns that are very similar to each other should be placed in the same cluster. The patterns that are quite different should in the end appear in different clusters. This kind of information is definitely very useful, although the expected and actual performance of it is less spectacular than the PS-FCM algorithm's performance. The variability of the results of P-FCM is visible only in the random mode of knowledge incorporation, and this becomes obvious when the results are much more stable in the misclassification, and even more so when we make a comparison between the random mode and the misclassification mode.

As it follows from all the experiments, the random mode improves the quality of the clustering over standard FCM. However, this way of incorporating the user input is not stable and may unexpectedly provide higher or lower values than the expected results. The other significant disadvantage of this approach is the effort from the user side. In comparison with the misclassification mode the random mode requires a greater amount of user supervision. On the contrary, the misclassification mode is applied in a specific context only – to the misclassified patterns. This approach reduces the effort needed and immensely improves the accuracy. In almost all of the datasets, PS-FCM, nearly 100% of accuracy in the misclassification mode was achieved. Thus, as it is apparent from this study, the misclassification mode performs better, and because of that, we should focus on this way of incorporating user input. Depending on our knowledge we can decide between the following choices: using the PS-FCM class membership information, specifying the similarity between the objects (P-FCM), or combining these types of knowledge in PSP-FCM. The considerable improvement in the PS-FCM will be visible for even a small amount of supervision. In order to observe substantial improvement for proximity hints we should enter values for at least 10% of all patterns from the group. The possible scenarios of experimentation from this work could be extensively applied to any other dataset. The scheme of the experimentation we assumed could be extended as it could be worth examining, for instance, an application with only one component of proximity hints (e.g. high similarity hints or low similarity hints only). The user supervision can be narrowed to selected clusters only, or the amount of the user input for the PSP-FCM algorithm could vary in respect to its type or given ratio of these types (class membership information and proximity hints).

The other interesting practical aspect that was examined and analyzed here is the potential abilities of fuzzy clustering framework to construct fuzzy classifiers. In the classification experiment that was created, the decision regions can be used for classification of other patterns from the same domain. The prototypes obtained from the clustering experiment of the training set from FCM, PS-FCM, P-FCM and PSP-FCM were used for classification of the patterns from the testing set. While user supervision has a positive effect on reverting the structure (e.g. noise interference) in the clustering experiment, it uses the information about particular patterns from particular datasets. Then, if we want to apply the prototypes constructed with the use of specific knowledge for specific patterns, it may not be very effective for other patterns from the same domain. However, the classification rate could possibly be improved in the classification mode with user input, if the training set, and the revealed structure there, is a reasonably good approximation of the testing set. The performance gain was compared to the accuracy of the classification obtained from standard FCM. In the misclassification mode we depicted the improvement with respect to the accuracy obtained for the prototypes generated by standard FCM. It is usually smaller than in the clustering experiment but significant, for the Dermatology database the improvement was over 50%. The results showed that user supervision is valuable for constructing fuzzy classifiers, and the performance of classifiers constructed in this way is better than the accuracy obtained from the standard FCM in the classification experiment.

This work presented and examined the algorithmic enhancements to Fuzzy C-Means (FCM) as class assignment information (PS-FCM), proximity hints (P-

FCM), and class assignment information together with proximity hints (PSP-FCM). The experiments were conducted in a detailed way, and that used synthetic datasets, Machine Learning sets, and a Web Documents dataset, by quantifying the amount of user supervision in two modes: random and misclassification mode. The results presented here can serve as a practical guide for any other applications in terms of the way user knowledge is incorporated, the amount that is available, and the expected results. As FCM becomes more widely known, and more widely used, any possible improvements are valuable not only for the new users but also for current users. The improvements will allow them to enrich the way they can explore and interact with data. Nevertheless, the concepts and approaches applied in this work might be beneficial and transferable for other methods used for Data Mining tasks.

7.2 Future research

This thesis embraces the concepts of algorithmic enhancements of user supervision mechanisms for the Fuzzy C-Means algorithm. We evaluated the Partially Supervised FCM, the Proximity-based FCM and proposed a hybrid method of both techniques: the Partially Supervised Proximity-based FCM successfully combining in collaborative way class assignment information along with proximity hints.

While these techniques extensively cover the types of possible user supervision to be applied for the standard FCM, the next steps of this research can be more focused on the importance of each feature in the feature vectors used to describe the patterns. We could envision the weight vectors, which will have a certain weight value for each feature in the feature vector. In this way it would be possible to assign higher weights to more descriptive features i.e. these ones, which better distinguish between different objects. The key challenge would be to introduce the mechanism for optimization of the weight vectors and as a result finding the best (most optimal) combination of weights allowing for obtaining the highest performance in the given set of objects and set of features describing them.

Another important aspect in the clustering is the cluster validity problem. The estimation of the proper number of clusters would have a enormous practical value for the researchers and practitioners. Based on the contained here material it would be worth to investigate the creation of the hierarchical fuzzy clustering algorithm with usage the cluster similarity measure introduced in the Section 6.2.3. The cluster similarity allows for deciding at each step which clusters should be merged with each other. It would be reasonable that the clusters with the highest similarity would be combined because we could expect that the patterns they contain are the most similar to each other. The top-down strategy applied would result in creating the dendrogram from which it would follow the number of clusters at each level of split.

Besides aforementioned ideas it would be interesting to pursue research with other distance or similarity measures, which might be better suited for particular applications (as an example it can serve the cosine measure for Information

Retrieval) or the similarity measures that would be able to provide us with the similarity between vectors of different dimensions. Finally, the alternative ways of dimension reduction would be a definite candidate to consider while continuing to research topics gathered in this material.

Appendices

Appendix A: Description of the software tools. Testing Framework and DataMiner.

The constructed framework is a set of Java applications unifying data manipulation tools, clustering, classification, and presentation of the results to users. The data manipulation tools span the data generation, gathering, pre-processing, transformation and are specific to the particular type of a dataset. The graphical user interface (GUI) components monitor the execution of the algorithms and measure their performance (Fig. A-2, Fig. A-3 and Fig. A-4). Moreover they provide the user with data analysis and interpretation (Fig. A-1).

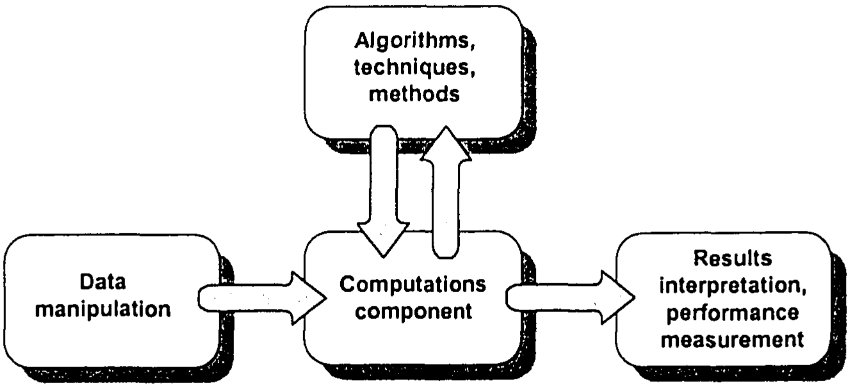
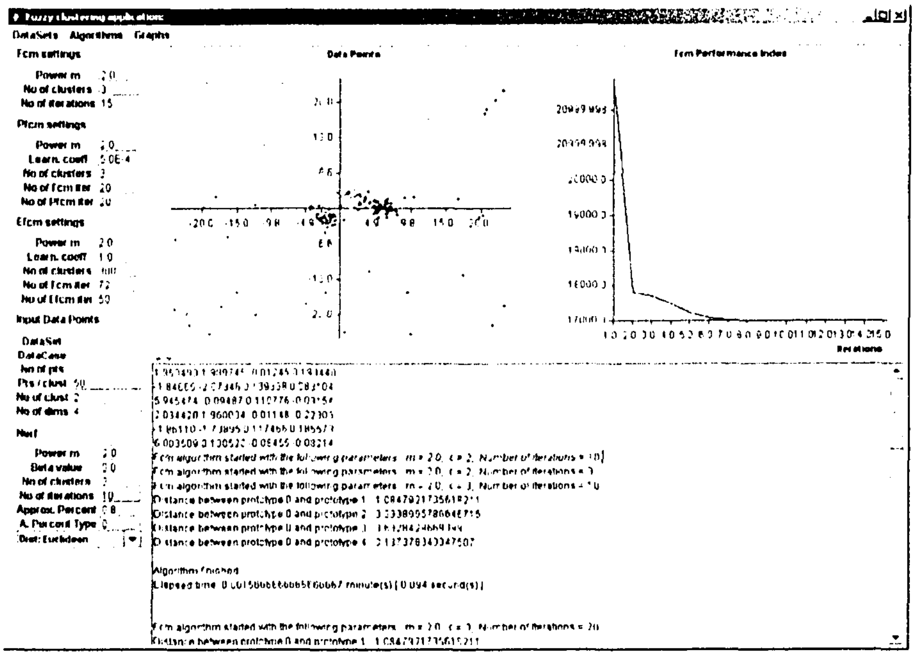

Algorithms, techniques, methods
Data manipulation
Computations component
Results interpretation, performance measurement

Figure A-0-1. The main components and logical flow of the framework.



Fcm settings

Power m 2.0
No of clusters 3
No of iterations 15

Pfcm settings

Power m 2.0
Learn. coeff 5.0E-4
No of clusters 3
No of Fcm iter 20
No of Pfcm iter 20

Efcm settings

Power m 2.0
Learn. coeff 1.0
No of clusters 100
No of Fcm iter 72
No of Efcm iter 50

Input Data Points

DataSet
DataCase
No of pts
Pts / clust 50
No of clust 2
No of dims 4

Nef

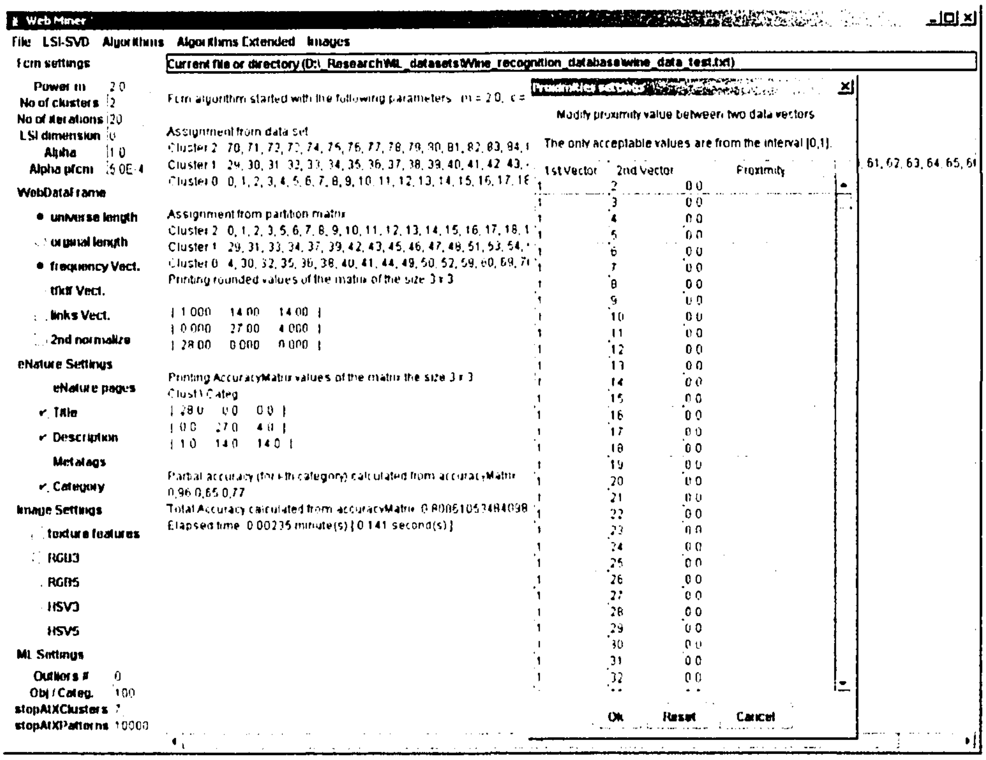
Power m 2.0
Beta value 2.0
No of clusters 2
No of iterations 10
Approx. Percent 0.8
A. Percent Type 0
Dist: Euclidean

```
1.953493 1.939745 0.01245 0.131440
-1.84665 -2.07346 0.139338 0.083104
5.945474 0.09487 0.110776 -0.03154
2.034420 1.960034 0.01148 0.22305
-1.86110 -1.73895 0.117466 0.185577
6.003509 0.130522 -0.05455 -0.03214
Fcm algorithm started with the following parameters   m = 2.0,  c = 2,  Number of iterations = 10
Fcm algorithm started with the following parameters   m = 2.0,  c = 2,  Number of iterations = 3
Fcm algorithm started with the following parameters   m = 2.0,  c = 3,  Number of iterations = 10
Distance between prototype 0 and prototype 1   1.0847921735618211
Distance between prototype 0 and prototype 2   3.3338995780648715
Distance between prototype 0 and prototype 3   16.12842466914
Distance between prototype 0 and prototype 4   3.1373783403347507

Algorithm finished
Elapsed time  0.001566666666666667 minute(s) [ 0.094 second(s) ]


Fcm algorithm started with the following parameters   m = 2.0,  c = 3,  Number of iterations = 20
Distance between prototype 0 and prototype 1   1.0847921735618211
```

Figure A-0-2. The main window of the testing framework.



Current file or directory (D:_Research\ML_datasets\Wine_recognition_database\wine_data_test.txt)

Fcm settings

- Power m: 2 0
- No of clusters: 2
- No of iterations: 20
- LSI dimension: 0
- Alpha: 1 0
- Alpha pfcm: 5 0E-4

WebDataFrame

- ● universe length
- ○ original length
- ● frequency Vect.
- tfidf Vect.
- links Vect.
- 2nd normalize

eNature Settings

- eNature pages
- ✓ Title
- ✓ Description
- Metatags
- ✓ Category

Image Settings

- texture features
- RGB3
- RGB5
- HSV3
- HSV5

ML Settings

- Outliers #: 0
- Obj / Categ.: 100
- stopAtXClusters: 7
- stopAtXPatterns: 10000

```
Fcm algorithm started with the following parameters  m = 2 0,  c =

Assignment from data set
Cluster 2  70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 8
Cluster 1  29, 30, 31  32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42  43, ·
Cluster 0  0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18

Assignment from partition matrix
Cluster 2  0, 1, 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 1
Cluster 1  29, 31, 33, 34, 37, 39, 42, 43, 45, 46, 47, 48, 51, 53, 54, ·
Cluster 0  4, 30, 32, 35, 36, 38, 40, 41, 44, 49, 50, 52, 59, 60, 69, 70
Printing rounded values of the matrix of the size 3 x 3

| 1 000    14 00    14 00 |
| 0 000    27 00    4 000 |
| 28 00    0 000    0 000 |

Printing AccuracyMatrix values of the matrix the size 3 x 3
Clust\Categ
| 28 0    0 0    0 0 |
| 0 0    27 0    4 0 |
| 1 0    14 0    14 0 |

Partial accuracy (for i-th category) calculated from accuracyMatrix
0,96 0,65 0,77
Total Accuracy calculated from accuracyMatrix 0 8005105348409B
Elapsed time  0 00235 minute(s) [0 141 second(s)]
```

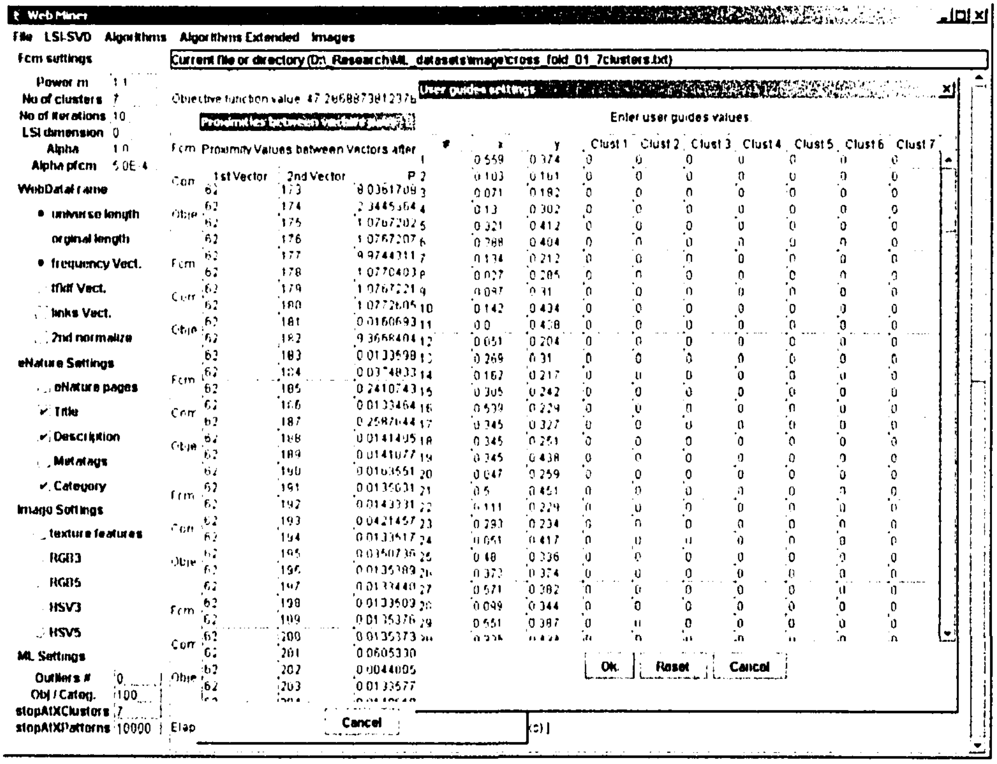
, 61, 62, 63, 64, 65, 6

Modify proximity value between two data vectors

The only acceptable values are from the interval [0,1].

1st Vector	2nd Vector	Proximity
1	2	0 0
1	3	0 0
1	4	0 0
1	5	0 0
1	6	0 0
1	7	0 0
1	8	0 0
1	9	0 0
1	10	0 0
1	11	0 0
1	12	0 0
1	13	0 0
1	14	0 0
1	15	0 0
1	16	0 0
1	17	0 0
1	18	0 0
1	19	0 0
1	20	0 0
1	21	0 0
1	22	0 0
1	23	0 0
1	24	0 0
1	25	0 0
1	26	0 0
1	27	0 0
1	28	0 0
1	29	0 0
1	30	0 0
1	31	0 0
1	32	0 0

Figure A-0-3. The main window of the DataMiner Application. Proximity hints settings.


Web Miner
File LSI-SVD Algorithms Algorithms Extended Images
User guides settings
Proximities between vectors
Ok
Reset
Cancel
Cancel

Current file or directory (D:_Research\ML_datasets\image\cross_fold_01_7clusters.txt)

Fcm settings

- Power m: 1 1
- No of clusters: 7
- No of iterations: 10
- LSI dimension: 0
- Alpha: 1 0
- Alpha pfcm: 5 0E-4

WebDataFrame

- ● universe length
- orginal length
- ● frequency Vect.
- tfidf Vect.
- links Vect.
- 2nd normalize

eNature Settings

- eNature pages
- ✓ Title
- ✓ Description
- Metatags
- ✓ Category

Image Settings

- texture features
- RGB3
- RGB5
- HSV3
- HSV5

ML Settings

- Outliers #: 0
- Obj / Categ.: 100
- stopAtXClusters: 7
- stopAtXPatterns: 10000

Objective function value 47 2868873812376

Fcm Proximity Values between Vectors after

	1st Vector	2nd Vector	P
Con	62	173	8 0361709
Obje	62	174	2 3445364
	62	175	1 0767202
	62	176	1 0767207
Fcm	62	177	9 9744311
	62	178	1 0770403
Corr	62	179	1 0767221
	62	180	1 0772605
Obje	62	181	0 0160693
	62	182	9 3668404
	62	183	0 0133598
Fcm	62	184	0 0374833
	62	185	0 2410743
Corr	62	186	0 0133464
	62	187	0 2587644
Obje	62	188	0 0141495
	62	189	0 0141077
	62	190	0 0163551
Fcm	62	191	0 0135031
	62	192	0 0143331
Con	62	193	0 0421457
	62	194	0 0133517
Obje	62	195	0 0350736
	62	196	0 0135389
	62	197	0 0133440
Fcm	62	198	0 0133503
	62	199	0 0135376
Corr	62	200	0 0135373
	62	201	0 0605330
Obje	62	202	0 0044005
	62	203	0 0133577

Elap

(s)]

Enter user guides values.

#	x	y	Clust 1	Clust 2	Clust 3	Clust 4	Clust 5	Clust 6	Clust 7
1	0 559	0 374	0	0	0	0	0	0	0
2	0 103	0 161	0	0	0	0	0	0	0
3	0 071	0 182	0	0	0	0	0	0	0
4	0 13	0 302	0	0	0	0	0	0	0
5	0 321	0 412	0	0	0	0	0	0	0
6	0 788	0 404	0	0	0	0	0	0	0
7	0 134	0 212	0	0	0	0	0	0	0
8	0 027	0 205	0	0	0	0	0	0	0
9	0 097	0 31	0	0	0	0	0	0	0
10	0 142	0 434	0	0	0	0	0	0	0
11	0 0	0 438	0	0	0	0	0	0	0
12	0 051	0 204	0	0	0	0	0	0	0
13	0 269	0 31	0	0	0	0	0	0	0
14	0 167	0 217	0	0	0	0	0	0	0
15	0 305	0 242	0	0	0	0	0	0	0
16	0 539	0 229	0	0	0	0	0	0	0
17	0 345	0 327	0	0	0	0	0	0	0
18	0 345	0 251	0	0	0	0	0	0	0
19	0 345	0 438	0	0	0	0	0	0	0
20	0 047	0 259	0	0	0	0	0	0	0
21	0 5	0 451	0	0	0	0	0	0	0
22	0 111	0 229	0	0	0	0	0	0	0
23	0 293	0 234	0	0	0	0	0	0	0
24	0 051	0 417	0	0	0	0	0	0	0
25	0 48	0 336	0	0	0	0	0	0	0
26	0 372	0 374	0	0	0	0	0	0	0
27	0 571	0 382	0	0	0	0	0	0	0
28	0 099	0 344	0	0	0	0	0	0	0
29	0 551	0 387	0	0	0	0	0	0	0

Figure A-0-4. The main window of the DataMiner Application. Class membership settings.

Appendix B: Singular Value Decomposition (SVD). Description of the method.

The Singular Value Decomposition (SVD) is a powerful computational method for analyzing matrices and solving problems involving matrices. One of its great assets is a dimension reduction in a vector space model. The decomposition is defined as follows [2]:

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T \tag{1}$$

Where:

$\mathbf{U}$ - m-by-n orthogonal matrix (i.e. $\mathbf{U}^T\mathbf{U} = \mathbf{I}$) and the its columns contain the left singular vectors of $\mathbf{A}$

$\mathbf{V}$ - n-by-n orthogonal matrix and the its columns contain the right singular vectors of $\mathbf{A}$

Σ - m-by-n diagonal matrix with singular values σ_{ij}, if $i \neq j$ then $\sigma_{ij} = 0$ and $\sigma_{ij} \geq 0$

Above factorization can be simplified by noting that the rank of a diagonal matrix Σ is equal to the number of non-zero singular values σ_{ij}. This feature of the SVD allows for finding a rank-k approximation to a matrix $\mathbf{A}$ with minimal change to that matrix for a given value of k (where k is the rank of Σ) [1], [2]. After approximation the dimensions of the matrices change as follows: $\mathbf{U}$ - m-by-k, $\mathbf{V}^T$ - k-by-n, Σ - k-by-k.

Bibliography

1. Berry M. W., Dramac Z., Jessup E. R., Matrices, vector spaces, and information retrieval, *SIAM Rewiev*, Vol. 41, No. 2, 1999, pp. 335-362,.
2. Forsythe G. E., Malcolm M. A., Moler C.B., "Computer Methods for Mathematical Computations", Englewood Cliffs, Prentice Hall, 1977.