University of Alberta

Faculty of Graduate Studies and Research ©

Design and Development of Dynamic Collaborative Frameworks Using Concepts of Knowledge-Based Networks

by

Partab Rai

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

Department of Electrical and Computer Engineering

Edmonton, Alberta

Spring 2008

Your file Votre référence
ISBN: 978-0-494-45585-2
Our file Notre référence
ISBN: 978-0-494-45585-2

Abstract

We have developed a suite of dynamic frameworks for clustering, classification, and regression problems of knowledge-based networks using collaborative approaches. For solving clustering problems, we provide a formulation of the model-integration problem using the principles of sharing prototypes and membership-functions and describe iterative algorithms that converge to an optimal solution. We show that the measure of proximity-distance is a suitable vehicle for quantifying the consensus of collaborative data sites.

For classification and regression problems, we present a new experience-consistent framework. By extending the performance index, we show that the domain knowledge captured by regression and classification models plays a regularization role in system identification problems. We demonstrate that the achieved consistency between collaborative sites can be quantified through fuzzy sets related to the parameters of the model.

In the development of an approach to fuzzy rule-based model identification realized in a collaborative framework of experiential evidence (data) and knowledge evidence (past experience), we demonstrate how to reconcile these two essential sources of guidance in the form of local regression models.

Using a radial-basis function neural networks approach, consistency is achieved using a connection value framework to reconcile data with past experience by considering gradient-based neural networks method.

The study provides architectural considerations, elaborates on essential communication mechanisms, and covers underlying algorithmic aspects of knowledge-

based networks. We explain how the collaboration mechanism gives rise to higher order granular constructs such as type-2 fuzzy sets that emerge in a highly legitimate manner in distributed fuzzy modeling. We evaluate our methods with type-2 fuzzy sets.

The theoretical and algorithmic approaches to collaborative frameworks investigated in this study can be used as a foundation for further research in the area of distributed fuzzy modeling.

Dedications

I dedicate this work to my wife Vimla Devi and my children Roheet Rai, Ameet Rai, and Shilpa Rai. Thank you for being patient during my four long years of study.

Acknowledgment

I thank my supervisor Dr. Witold Pedrycz for offering me a chance to work under his guidance, for believing in me, and for encouraging my research efforts. I express my gratitude and appreciation for his active supervision throughout this research.

Support from the Canada Research Chair (CRC) program (W. Pedrycz), the Natural Sciences and Engineering Research Council (NSERC), and Mehran University of Engineering & Technology Jamshoro is gratefully acknowledged.

I acknowledge numerous referees whose scientific reviews have enriched my research contributions.

I also thank the examination committee members: Dr. M. Reformat, Dr. P. Musilek, Dr. A. Robinson, and Dr. A. Braga (external examiner) for insightful suggestions to improve the thesis.

Table of contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

Over the last ten years the progress in data acquisition technology has resulted in the growth of huge distributed databases. Extracting useful knowledge from such databases is often challenging due to technical or nontechnical constraints related to dimensionality, privacy, computation, or communication. Such constraints may prevent one from directly integrating the distributed data into a single dataset at a central site. This has led to the emergence of distributed data computing techniques which are used to extract high quality information from distributed database sources with limited interactions among associated data sites. In particular, informational-privacy concerns have resulted in an increased focus on privacy-preserving distributed techniques. We study scenarios in which the data is either horizontally or vertically partitioned. Vertically partitioned sites are heterogeneous in that they contain different attributes of a common set of objects. Horizontally partitioned sites are homogeneous in that objects are distributed among different sites, but have the same set of features.

Distributed computing is a problem based on the following three factors:

- Distribution of data (horizontal or vertical partitioned),
- Type of data analysis (supervised or unsupervised learning), and
- Restrictions placed on information sharing.

Acknowledging such diverse objectives of data analysis and the need for highly collaborative pursuits, we develop frameworks for knowledge-based networks. We consider such networks to be composed of highly autonomous nodes which operate at a certain level of abstraction. In general, a node is inherently engaged in processing numeric data coming from the environment, and considers the knowledge provided by other network nodes (communicated in the form of information granules). The three collaborative knowledge-based approaches explored in our studies are shown in Figure 1.1.

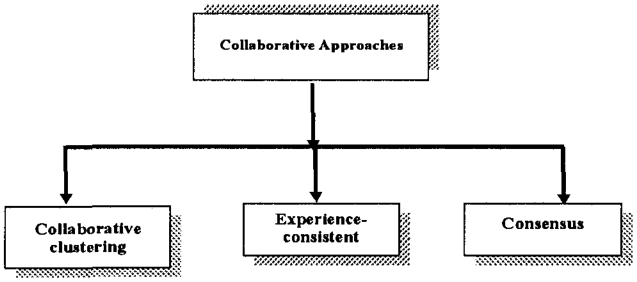


Figure 1.1 Collaborative knowledge-based approaches.

We propose new collaborative frameworks based on the concepts of knowledge-base networks. We also introduce methods to quantify and evaluate our collaborative frameworks. In a collaborative clustering framework we extend the study of Pedrycz's collaborative model. We also develop a new collaborative model (experience-consistent) for labelled data.

1.2 Collaborative clustering frameworks

In a collaborative clustering framework, data access is available at a certain level of granularity rather than in numeric values. Various communication links can be efficiently established among datasets if the levels of granularity are similar[3]. Fuzzy sets are one among several key vehicles of granular computing.

Pedrycz[2] proposed a model of collaborative clustering over a collection of databases in which computing agents carry out clustering in a distributed environment.

Collaborative clustering is a new concept in the area of data analysis in which several subsets of a pattern can be processed together with the objective of finding global or similar structures. Collaborative clustering is realized over a collection of databases when results of individual (local) clustering processes collaborate in a distributed environment [4]. A network of databases viewed in an environment of a collaborative clustering, as in Figure 1.2.

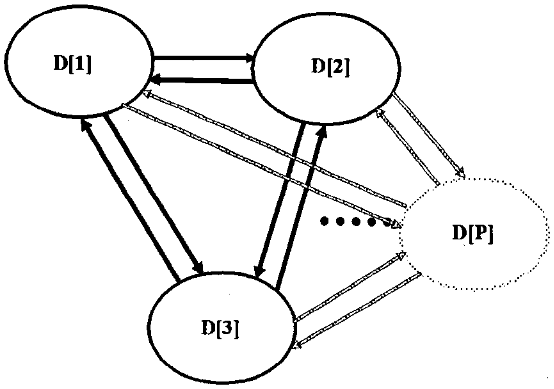


Figure 1.2 Collaborative based clustering: arrows show interactions between collaborating datasites (D[1], D[2], ..., D[P]).

We formulate the model-integration problem using the collaborative principles of sharing prototypes and membership-function and describe the iterative algorithms that converge to an optimal solution. Next, we identify the main difficulties and show that the partitions proximity-distance index could be a suitable vehicle for quantifying the consensus of the collaboration datasites. We also elaborate on the role of the underlying optimization criterion and its components that both guide the development of the partition matrices and lead to a collaboration.

1.2.1 Experience-consistent framework

The experience-consistent model interacts in the centralized mode. In this approach we are concerned with system modeling which involves data and reconciles the developed model with some previously acquired domain knowledge being captured in the format of some already constructed models which were based on auxiliary datasets. To emphasize the nature of modeling being guided by the reconciliation mechanisms, we refer to this mode of identification as experience-consistent modeling. A network of databases viewed in an experience-consistent model as in Figure 1.3.

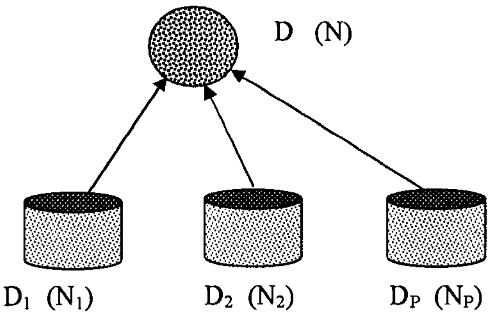


Figure 1.3 Interactive links between D and Di realized indirectly through passing the parameters of the models available at D1, D2, …, DP.

For such models the assessment of consistency is realized by making use of the dataset D. First, the model is constructed on the basis of D. Second, the consistency is expressed on a basis of differences between the constructed model and those models from D_i (i=1,2,…,p) where the differences are assessed with the use of data D. Further details of the model is provided in chapters 7-9.

1.2.2 Consensus framework

Consensus based clustering is concerned with a reconciliation of data structures discovered in different feature spaces realized at the level of information granules[1]. The Figure 1.4 shows a consensus based clustering scenario in which a collection of fuzzy partition matrices U[1], U[2], …, U[P] (each with dimension $U[ii] = c_{ii} \times N, ii = 1,2,...,p$) reconciles their knowledge to a fuzzy partition matrix U of dimensionality $c \times N$, where $c \in \{c_1, c_2, c_3, ..., c_p\}$.This requirement reflects the fundamental notion of consensus.

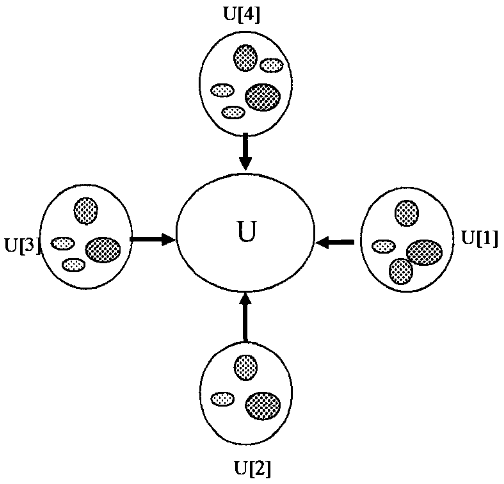


Figure 1.4 Knowledge-based networks using a consensus based approach; merging partition matrices, where p = 4.

While looking at several alternatives for formulating the problem and optimal solutions of distributed data clustering, we observe two fundamental difficulties.

- A lack of correspondence between rows of partition matrices; we are interested in the correspondence between all partition matrices. There is no guarantee that the rows of U[ii] correspond in a straightforward way with the rows of U[jj], that is, that the first row of U[ii] corresponds to the first row of U[jj]. What makes the problem even more difficult is that we are interested in the correspondence that holds for all partition matrices.

- Different dimensionalities of the partition matrices are involved in the formation of the consensus outcome to find global structure in distributed clustering.

These two challenges must be addressed as an integral part of an overall solution. A justifiable way of handling them would be to abstract from the partition matrices and move to the next level of abstraction by mapping over partition matrices so that the resulting constructs do not exhibit dependence over the order of clusters and the dimensionality does not require the number of clusters in any explicit manner. In the literature such correspondence between partition matrices is tackled through logic transformation as discussed in [3] and also by fuzzy proximity described in [1]. A fuzzy proximity approach is a viable alternative.

In the existing literature, mapping of the consensus function is performed in two phases. In the first phase base-partitions mapping defines a new representation of the base-partitions outputs (as the voting structure); another way is through an associated cluster label matching method. In [8] the cluster relabelling problem is formulated as a weighted bipartite matching problem and is solved using the Hungarian method[6]. In phase 2 of consensus clustering, all base-partitions generated in phase 1 are combined. If the cluster label mismatch problem is resolved and the number of clusters in the base clustering is equal to the number of clusters in the combined clustering, the majority voting or maximum likelihood classification[8] can be readily applied. Otherwise, co-association based consensus functions can be used[7][5][13][15].

1.3 Motivations

As the Internet is becoming a part of our daily life, we have unlimited information and there is a need to develop new robust tools to organize it. Dynamic distributed environments are in great demand for applications like e-medicine, web mining, and e-business.

The standard approach today in such distributed computing environments is centralized. Data which might have been collected in multiple locations are gathered to a single location. On the basis of this data, a model is computed. This process is easy to understand and constructing such an algorithm is straightforward, but there are a number of limitations to the centralized approach:

- All of the data is visible at the central location, so there is no way for some or all of the data-collection locations to limit access to their local data; that is, privacy is not preserved. It may not be possible to gather data to a single location because of legal restrictions arising from privacy concerns, so some forms of analysis cannot be done in a centralized fashion.

- There are substantial performance requirements at the central site: the centralized model must look at all of the data at least once, and often in an iterative fashion; there is little exploitable locality in the access patterns. The time performance is limited by the memory hierarchy, so the need to fetch records repeatedly from the bottom of the memory hierarchy is a substantial performance issue.

- When data locations are geographically separate, the movement of data to a central location requires bandwidth and takes time.

As the dataset at the central location grows and the model becomes more complex, it is important to consider replacing centralized computation approaches by distributed techniques. For example, business organizations collect information about their customers via physical stores, websites, and call centers which are typically located in different geographical locations. Thus we are motivated to pursue this study to assist the business organization:

Knowledge reuse:
In grouping customers for direct marketing campaigns, several legacy customer segmentations might already exist based on demographics, credit ratings, geographical regions, or purchasing history.

Privacy considerations:
Real-world applications often involve databases distributed due to organizational or operational constraints. For example, a health department may want to use data mining to identify trends and patterns of a particular disease in different age groups or in ethnic groups. Insurance companies have considerable data that would be useful but are unwilling to release this due to privacy concerns. An alternative possibility is to have insurance companies provide abstracts of their data that cannot be traced to individuals, but can be utilized to identify the trends and patterns of interest to the health department.

1.4 Research objectives

We aim to build a suite of collaborative frameworks for a distributed computing environment using the concepts of knowledge-based networks. Such collaborative frameworks will take into account privacy restrictions and will be applicable to scenarios where the different sites have diverse and overlapping subsets of features.

The intended collaborative approach is especially attractive if there is enough processing power and sophistication to build local models at the locations where data are collected. The local models can then be moved to a central location and merged to produce a coherent global model. It is expected that such an approach will compensate for the weaknesses of centralized computational models:

- As only models leave the local sites, details of the private raw data remain hidden, at least in principle. Hence, such a distributed approach handles privacy issues effectively.

- As processing takes place at each local site, the computational load is spread over many processors and data accessing is efficient.

- Higher communication bandwidth is not required as there is no longer a need to move raw data from the remote sites; only models or pieces of models need to be moved, and these elements are of much smaller size than the data itself.

We study the following collaborative frameworks in order to produce a global model from collaborating sites:

- Design and development of multiphase collaborative clustering frameworks based on the horizontal and vertical clustering concepts introduced by Pedrycz[10].

- Introduction of a novel system modeling that utilizes a collaborative framework of the data-driven experience using methods like regression, classification, fuzzy rule-based modeling and radial basis neural networks.

1.5 Problem formulation

In this section, we introduce the concept of collaborative pursuits and elaborate on algorithmic developments. We are given a finite collection of datasets D[1], D[2], ..., D[P], where each pattern is represented as a vector in n-dimensional space $\mathbf{R}^n$. We also refer to sites to underline the distributed character of the datasets under consideration; hence, D[i] is located at D_i. The objective is to find structure in the data using collaborative activities. When data cannot be freely communicated between individual sites, a careful analysis is performed of possible communication capabilities that could be established.

We revise and update the findings obtained locally at specific datasites. Data cannot always be shared between sites. Reasons for this include (a) privacy and security of data. Each data site is treated as a separate computing entity when we are restricted from sharing data outside the site, and (b) technical constraints and their feasibility. Quite often transfer of huge masses of highly dimensional data is not allowed. For instance, in wireless sensor networks which are highly distributed architectures, computing is restricted to individual nodes (sensors), limiting potential communication overhead.

To deal with these constraints while still allowing communication, we exchange abstract findings found at the individual nodes. The datasites exchange local findings and then use them in further development of the structures. It is necessary to implement the granularity scheme to establish interaction while preserving privacy and security constraints. The level of abstraction itself is inherently associated with the notion of granularity of information. In a nutshell, information granules are a manifestation and an effective realization of the concept of abstraction. Given these characteristics of the problem and its anticipated solution, we are talking about *collaboration* between datasites and *collaborative clustering*.

There are two fundamental modes of interaction:

Centralized mode: In this mode, we consider one dataset D_i for which we are going to reconcile the findings (its local model) with the modeling results available at all remaining datasets D_1, D_2, …, D_i, D_{i+1}, D_p.

Distributed mode: Here we allow all datasites to interact and the resulting local models are shared. Each data site affects all other data sites when optimizing.

It is possible to combine these two modes; that is, each data site acts as a central site but does not interact with other datasites when optimizing.

In the collaborative clustering framework we concentrate on a distributive mode of interaction whereas in experience-consistent modeling we focus on a centralized model. The optimization details are explained in later chapters.

1.5.1 Methodology

In the clustering problem, we introduce a new framework of collaborative fuzzy clustering—a conceptual and algorithmic machinery for the collective discovery of a common structure (relationships) within a finite family of data residing at individual data sites. There are two fundamental features of the proposed optimization environment. First, given existing constraints which prevent individual sites from exchanging detailed numeric data, any communication has to be realized at the level of information granules. The specificity of these granules impacts the effectiveness of ensuing collaborative activities. Second, the fuzzy clustering realized at the level of the individual datasite has to constructively consider the findings communicated by other sites and act upon them while running the optimization confined to the particular datasite.

Following these two general guidelines, we develop a comprehensive optimization scheme and discuss its two-phase character in which the communication phase of the granular findings intertwines with the local optimization being realized at the level of the individual site and exploits the evidence collected from other sites. The proposed augmented form of the objective function is essential in the navigation of the overall optimization that has to be completed on a basis of the data and available information granules. The intensity of collaboration is optimized by choosing a suitable trade-off between the two components of the objective function. The objective function-based clustering used here concerns the well-known fuzzy c-means (FCM) algorithm. Moreover, in new developed collaborative framework where collaborative phases are iterated multiple times enabling the framework to be more suitable for applications of distributed spatial datasets.

When direct access to numeric data is unavailable at individual sites, a viable alternative to establish meaningful collaborative links is to form communication at the level of information granules viz. communicate and exchange the findings at a higher level of abstraction, that is, prototypes and partition matrices[2][3][12][14][16]. Given the fact that all datasets are defined in the same feature space, exchanging prototypes becomes a meaningful alternative. As the equivalence of prototype-partition representation has been identified, we could translate the prototypes into the corresponding partition matrices. Once the prototypes obtained at the remote site are communicated to the active site, they are translated into the corresponding partition matrix. As stressed before, the transformation is obvious given the prototype-partition mapping. The advantage of having the induced partition matrix is that it is defined for the same dataset—that of the active central site. By forming the induced partition matrices for all sites, collaboration can take place.

Collaboration relies on the knowledge of the findings (structure) revealed at the other sites. We can envision the following way of sharing these findings:

a. Initial phase: At each site the FCM is run independently, forming a structure at the local level. Hence we arrive at the partition matrices and the prototypes.

b. Collaborative phases: Here we iteratively communicate the findings formed at each site to the remaining ones. They impact the way in which clustering at a specific site is developed. Notably, data at this site is combined with the induced structures produced at the other sites involved in the collaboration.

In the experience-consistent approach, we partition datasets into different portions, each having similar attributes; this should result in regions of similar target value. Therefore, using the relevant features, a clustering algorithm is used to partition each dataset independently into "similar" regions. A clustering algorithm is applied in an unsupervised manner (ignoring the target attribute value). As a result a number of partitions (clusters) on each data site are obtained. Assuming similar data distributions of the observed datasets, the number of clusters on each dataset is usually the same. If this is not the case, by choosing appropriate clustering parameters, an identical number of clusters on each dataset can be easily enforced.

The next step is to match the clusters among the distributed sites, i.e., by matching clusters from one dataset that are the most similar to clusters in another dataset. This is followed by building local regression models on identified clusters at sites with known target attribute values. Finally, learned models are transferred to the remaining sites where they are integrated and applied to estimate unknown target values at the appropriate clusters.

We also evaluate collaboration approaches using a consensus clustering method. We compute proximity matrices of partition matrices of each base datasite and then find closeness between the sites by aggregating the proximity matrices' differences among all the collaborating sites.

1.6 Organization of the thesis

Chapter 1 introduces the collaborative schemes studied during the course of our research. Chapter 2 describes research related to distributed computing for clustering and classification problems similar to the ones we are dealing with. Chapter 3 introduces the diversity of datasets used in this thesis to illustrate and demonstrate the various algorithms investigated. Chapter 4 briefly reviews the fundamentals of fuzzy sets, proximity, and type-2 fuzzy sets. We describe classification, hierarchical clustering, and partition-based clustering methods. This is followed by a discussion of fuzzy rule-based method and the techniques of radial basis function networks. Chapter 5 describes fuzzy vertical collaborative clustering and Chapter 6 covers fuzzy horizontal collaborative clustering. Chapter 7 introduces regression and two-class experience-consistent models. Chapter 8 presents the experience-consistent fuzzy rule-based system. Chapter 9 introduces the experience-consistent radial-basis neural networks model.

For all approaches described in chapters 5–9, we first describe the general flow of the model, then follow with an explanation of the mechanism of communication between datasites. Then we evaluate the quality of collaboration and quantify the collaboration using type-2 fuzzy sets. Finally, we simulate the model using a series of datasets derived from synthetic and machine learning.

Chapter 10 outlines the conclusions of this research and suggests ways to expand it. Ideas for future research are proposed.

Details of methods used in Chapter 7 and chapter 9 are presented in the Appendices.

1.7 Conclusions

The objectives of this research are presented in this chapter. Frameworks of collaborative approaches based on knowledge-based networks are described, the basic formulation of the problem is explained, and different interaction modes are presented. Finally, the organization of the thesis is described.

References

1. H. Ayad, M. Kamel, Finding natural clusters using multi-clusterer combiner based on shared nearest neighbors in Multiple classifier Systems, *Fourth International Workshop, MCS 2003, UK Proceedings*, 2003, 166-175.

2. P. Bradley, U. M. Fayyad, C. A. Reina, *Scaling Clustering to Large Databases*, KDD98, 1998.

3. P. Devijver, J. Kittler, *Pattern Recognition: A statistical Approach*, Prentice Hall, London, 1982

4. S. Dudoit, J. Fridlyand, Bagging to improve the accuracy of a clustering procedure, *Bioinformatics*, 19, 2003, 1090-1099.

5. B. Fischer, J. M. Buhmann, Path-based clustering for grouping of smooth curves and segmentation, IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 25(4), 2003, 513-518.

6. A. Fred, A. K. Jain, Data clustering using evidence accumulation, *In Proceedings of the 16th International conference on Pattern Recognition*, 4, 2002, 276-280.

7. K. Fukunaga, *Satistical Pattern Recognition*, Second Ed. Acadamic Press, 1990.

8. A. K. Jain, R. C. Dubes, *Algorithms for Clustering Data* , Prentice Hall, 1977.

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

10. W. Pedrycz, Collaborative fuzzy clustering, *Pattern Recognition Letters*, 23, 2002, 675-686.

11. W. Pedrycz, Knowledge-*Based Clustering: From Data to Information Granules*, John Wiley, 2005.

12. W. Pedrycz, G. Vukovich, Clustering in the Framework of Collaborative Agent, *Proceedings of the 2002 IEEE, International Conference on Fuzzy Systems*, 1, 2002, 134-138.

13. W. Pedrycz, J. Waletzky, Neural network front-ends in unsupervised learning, *IEEE Trans. On Neural Networks*, 8, 390-401.

14. A. Topchy, A. K. Jain, W. Punch, Clustering Ensembles: Models of Consensus and Weak Partitions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 2005, 1866-1881.

15. Q. Zhang, J. Sun, E. Tsang, and J. Ford, Hybrid estimation of distribution algorithm for global optimization, *Eng. Comput.*, 21(1), 2004, 91-107.

16. T. Zhang, R. Ramakrishnan, and M. Livny, BIRCH: an efficient data clustering method for very large databases, ACM SIGMOD Record, 25(2), 1996,103-114.

Chapter 2

Literature Review

In the last few years the problems of distributed computing have been studied by many researchers. Proposed algorithms based on problems of clustering and classification methods and related to the research reported in this thesis are reviewed in this chapter.

2.1 Distributed computing approaches

Modern computational techniques in information science are often adapted to construct distributed computing frameworks. Recently, researchers proposed hybrid data mining[1][21][42][43][46][46] in the design of effective distributed computing models. Computing models integrate a number of base learning models to improve the performance of individual base learning models.

Several approaches were found in the literature: ensemble[47][48][49][51][54][63], distributed clustering[1][21][22], consensus[13][22][55], collaborative[45][51], and multiview[3][8][25] and classification ensemble[5][32][39]. Such computing models are distributed, centralized, or consensus (static) in nature. Figure 2.1 shows different levels of complexity handled by distributed computing approaches.

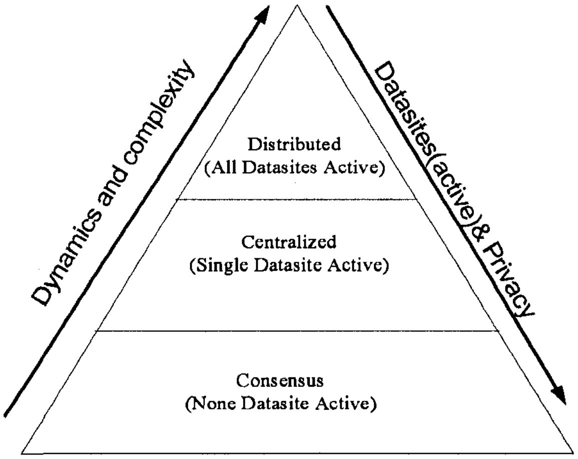


Figure 2.1 Pyramid presenting different issues handled by distributed computing approaches.

Techniques related to clustering and classification in distributed computing modeling are reviewed in the following sections.

2.2 Distributed clustering models

Cluster coordination is closely related to consensus [40] models of clustering. It is widely recognized that combining multiple-classification or multiple-regression models typically provides superior results compared to using a single and well-tuned model; however there are no well known approaches to combining multiple clustering.

The aim of combining partitions is to improve the quality and robustness of results. There is no single clustering algorithm which performs best for all datasets. Choosing a single clustering algorithm for the problem at hand requires expertise and insight, and the choice might be crucial to success of the study. The difficulty is that no ground truth is available to match the results. Instead of running the risk of choosing an unsuitable clustering algorithm, consensus clustering (discussed below) can be used.

2.2.1 Consensus clustering

One use of consensus clustering is to exploit multiple existing groupings of the data. Several analogous approaches exist in supervised learning scenarios where class labels are known, but we have not seen these applied in totally unsupervised settings. Johnson & Kargupta[22] proposed a feasible approach to combining distributed agglomerative clusterings in which dendrograms are collected generated from each local site, and then pairwise similarities for all objects are created. The combined clustering is then derived from the similarities.

Topchy et al.[55] proposed a consensus function based on informative-theoretic principles. This study shows that the consensus function is related to the classical intra-class variance criterion. The authors describe approaches for combining weak clusterings discuss and analyze how accurate consensus can be obtained from an unreliable component.

A different consensus function was developed by Dimitriadou et. al.[13] based on a voting/merging method that combines clusterings pairwise and iteratively; the cluster mapping problem handled in this study is not unique. Generally, fuzzy-membership decisions are accumulated during merging. The final clustering is obtained by assigning each object to a derived cluster with the highest membership value.

2.2.2 Multiview clustering

Multiaspect or multiview clustering algorithms train two independent hypotheses which bootstrap by providing each other with labels for the unlabeled data. The training algorithms tend to maximize agreement between the two independent hypotheses. Yarowsky[62] proposes multiview learning in a semisupervised setting for word sense disambiguation. One classifier is based on the local context of a word (view one) and a second classifier senses other occurrences of that word in the same document (view two); classifiers iteratively bootstrap each other.

Blum and Mitchell[3] suggest a concept of co-training in which two hypotheses are trained on distinct views. Their co-training algorithm augments the training set of two classifiers with the negative and positive highest confidence examples from the unlabeled data in iteration for each view. The two classifiers work on different views and a new training example is exclusively based on the decision of unlabeled data.

Collins and Singer[8] propose an improvement on the co-training algorithm which explicitly optimizes an objective function that measures the degree of agreement between the rules in distinct views.

Other related clustering algorithms that work in a multiview setting include a multiview version of DBSCAN[33] and a reinforcement clustering[58].

2.2.3 Distributed clustering

Johnson & Kargupta[22] propose a feasible approach in combining distributed agglomerative clusterings. First, each local site generates a dendrogram. After the dendrograms are collected, pairwise similarities for all objects are created from them. The combined clustering is then derived from the similarities. Agglomerative clusterings are hierarchical and is a static model; patterns assigned to a cluster are difficult to merge.

There have been some attempts at nonapproximated distributed clustering. In literature Kantabutra and Couch apply a parallel version of k-means and their algorithm rebroadcasts the datasets to all sites on each iteration which leads to heavy network loading. Also, their analytical and empirical analysis estimates 50% utilization of processors. Such an algorithm becomes impractical in a distributed environment.

2.2.4 Clustering ensemble model

The objective of ensemble[55] learning is to integrate a number of base-learning models in an ensemble so that performance of the ensemble is better than any of the individual-base learning models. If the base learning models are created using the same learning algorithms, the ensemble-learning schema is homogeneous; otherwise, it is heterogeneous.

The idea behind an ensemble system is to exploit each constituent model's unique features so as to capture different patterns that exist in the dataset. Both theoretical and practical works indicate ensemble can be an effective and efficient way to improve accuracies. Bates and Granger[2] showed that a linear combination of different techniques gives a smaller error variance than any of the individual techniques working in standalone mode. Since then, many researchers have worked on ensembles.

Several ensemble clustering algorithms do not use data summarization but attempt to directly restructure the clusters in order to adapt to the dynamic changes in the dataset(s).

Cluster ensemble methods differ in the following respects: the way the base clusterings are obtained and the procedure by which they are combined [55].

Strehl and Ghosh[53] proposed three different ensemble clustering models based on a consensus method. All models use various hypergraph operations to search for solutions. The cluster-based similarity partitioning algorithm (CSPA)[53] induces a graph from a coassociation matrix and clusters using the MEITS algorithm[53]. The hypergraph partitioning algorithm (HGPA)[53] represents each cluster with a hyperedge in a graph where the nodes correspond to a given set of patterns. The hyperbased metaclustering algorithm (MCLA)[54] uses hypercollapsing operations to determine soft cluster-membership values for each object. The optimization details of the model are:

The cluster ensemble (CE) is based on a consensus clustering function where in each component the learner tries to solve the same task heuristically. This is a multilearner system as illustrated in Figure 2.2.

A clusterer consists of a particular clustering algorithm with a specific view of the data. A clustering is the output of a clusterer and consists of group labels for some or all objects. Consensus clustering provides a tool for the consolidation of results from a portfolio of individual clustering results. In the following section optimization of a Strehl & Ghosh cluster ensemble model is described and the reader can find more details in [53].

To better explain some efficacies of the method, we adhere to the following notations. Let a dataset D = $\{\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\ldots,\mathbf{x}_N\}$ in $\mathbf{R}^n$ be divided into c clusters. This can be represented as a set of c sets of objects $\{C_i \mid i=1, \ldots, c\}$ or as a label vector $\lambda \in \mathbf{R}^n$. A clusterer Φ is a function that generates a label vector. A set of r labelling $\lambda^{(1,\ldots,r)}$ is combined into a single labelling λ using a consensus function r: $\{\lambda^{(q)} \mid q \in (1,\ldots,r)\} \rightarrow \lambda$, where q is an index.

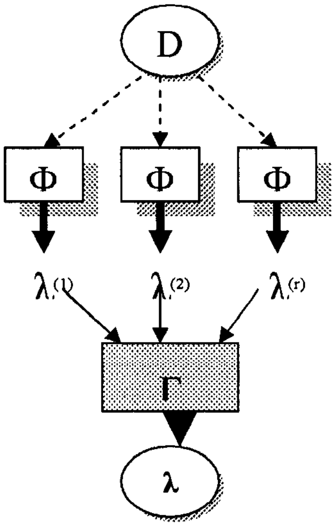


Figure 2.2 A cluster ensemble model based on consensus function r combines clustering λ(q) from different sources without visiting D.

The main idea of CE methodology is to proceed to a combination of several clustering results obtained in a distributed environment from different clusters simultaneously in a parallel fashion; such a combination needs to take into account the label of each clustering.

Objective function for cluster ensembles

Given r groupings with the q-th grouping $\lambda^{(q)}$ having $C^{(q)}$ clusters, a consensus function r is defined as a function $R^{n\times r} \rightarrow R^n$ mapping a set of clustering to an integrated clustering: $\{\lambda^{(q)} \mid q \in (1,\dots,r)\} \rightarrow \lambda$. The optimal combined clustering should share the most information with the original clusterings. In information theory, mutual information is a symmetric measure to quantify the statistical information shared between two distributions. Suppose there are two labelings, $\lambda^{(a)}$ having $k^{(a)}$ groups and $\lambda^{(b)}$ having $k^{(b)}$ groups. Let $n^{(h)}$ be the number of objects in cluster C_h according to $\lambda^{(a)}$, and n_l the number of objects in cluster C_l according to $\lambda^{(b)}$. Let $n_l^{(h)}$ denote the number of objects that are in cluster h according to $\lambda^{(a)}$ as well as in group l according to $\lambda^{(b)}$, then a normalized mutual information criterion $\phi^{(NMI)}$ is computed as shown below[53]:

$$\Phi^{(NMI)}(\lambda^{(a)},\lambda^{(b)}) = \frac{2}{n}\sum_{l=1}^{k^{(a)}}\sum_{h=1}^{k^{(b)}} n_l^h \log_{k^{(a)}.k^{(b)}}\left[\frac{n_l^{(h)} n}{n^{(h)} n_l}\right]. \tag{1}$$

Strehl and Ghosh proposed that the optimal combined clustering $\lambda^{(k\text{-opt})}$ be defined as the one that has maximal average mutual information with all individual labelings $\lambda^{(q)}$ given that the number of consensus clusters desired is k. Thus, the objective function is the average normalized mutual information (ANMI) and is given as:

$$\lambda^{(k-opt)} = \arg\frac{\max}{\hat{\lambda}}\sum_{q=1}^{r}\varphi^{(NMI)}(\hat{\lambda},\lambda^{(q)}), \tag{2}$$

where $\hat{\lambda}$ goes through all possible k-partitions. In equation (1) the sum represents the ANMI. For finite populations, the trivial solution is to exhaustively search through all possible clusterings with k labels for the one with maximum ANMI which is computationally prohibitive.

2.2.5 Collaborative fuzzy clustering

Pedrycz[45] introduced a “Fuzzy collaborative clustering” (FCC) based on the FCM algorithm. The author discusses the applicability of such an approach to the privacy issue of data confidentiality in a collaborative framework.

In this proposed approach, all data patterns reside on the sites processed together and the objective is to find structures that are common to all datasites. Clustering is carried out by first applying FCM on all individual sites and then by exchanging information from local clustering results based on partition matrices. This objective function is proposed in [45]. The model is able to communicate high-level information (i.e., prototypes and partition matrices), which makes the model more suitable for the distributed environment. There are two fundamental collaborative clustering modes:

Horizontal collaboration clustering mode

In horizontal clustering (feature based), collaboration happens at a more abstract level as the partition matrices are being provided to other collaborators. In such clustering we search the data structure with the same objects characterized by different attributes.

Vertical clustering algorithm

Vertical clustering is pattern based and collaboration happens at the prototype level. In such clustering we are faced with different objects being characterized by the same attributes[45]. The minimization criteria for collaboration between datasites based on the communicating partition matrix or on prototypes, is detailed in [45].

Computational flow of collaborative clustering

Collaborative clustering generally consists of two-step scenarios. In the first referential scenario we search for an independent structure in all collaborating sites. In this step the FCM algorithm is used for all the datasets to establish some preliminary structure in the data that will help us proceed with the next collaboration process, where global structures are discovered.

In the horizontal collaborative mode, the mechanism of collaboration is invoked by partition matrices U[1], U[2], ..., U[P]. In the vertical mode we are concerned with communication realized by passing prototypes as shown in Figure 2.3.

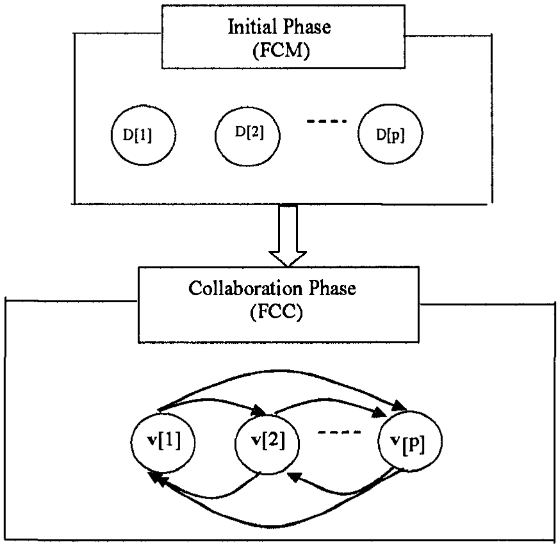


Figure 2.3 A flow diagram of collaborative clustering: In the initial phase FCM is applied to all collaborating sites; in the collaboration phase datasites communicate on a basis of cluster labels (produced in an initial phase) to reveal global structures in datasites.

2.3 Distributed classification models

Discriminant analysis was first proposed by Fisher in 1936 as a classification technique. To date, it has been reported as the most commonly utilized data-mining technique in handling classification problems[40][7][16][26][28][29][39][41][57]. Discriminant analysis has been utilized in a wide range of applications in areas such as medicine, business, education, marketing research, finance, engineering[14][38].

Regression is a widely used statistical modeling technique in which the probability of a dichotomous outcome is related to a set of potential independent variables[9]. The regression model does not necessarily require the assumptions of discriminant analysis; however, Harrell and Lee[27] found that logistic regression is as efficient and accurate as discriminant analysis even when the assumptions of discriminant analysis are satisfied. Regression models have been widely discussed in medical research, social research, market segmentation, and customer behaviour[17].

Several techniques combine multiple classifiers: sum, product, minimum, maximum, median, and majority [15][6][12]. The advantage of these techniques is that they are simple and do not require training [24]. This classifier fusion, however, requires careful selection of base classifiers [60] to achieve acceptable misclassification rates. Experience-consistent modeling requires further training for incorporating the decision of different classifiers [44][61][23].

This thesis presents a suite of "experience-consistent" identification models for different classification problems assuming the same base classifiers for all collaborating sites. A review of related studies from the literature is as follows.

2.3.1 Classification ensemble models

The ensemble method is a natural next step to simple model averaging for class predictions. An ensemble uses the predictions of multiple base classifiers typically through majority vote or average prediction, to produce a final ensemble-based decision [4][20].

Dudin and Tax[14] present ensemble architectures in three categories: (1) ensembles that combine classifiers of the same type trained on different types of features (parallel combining); (2) ensembles that combine classifiers of different types that train on the same set of features (stacked combination); (3) ensembles that combine classifiers of the same type trained on the same type of set (subset of entire set) of features (weak combining).

Recently, three ensemble models of voting approaches from the third category have received attention: boosting[19][52], bagging[5], and random subspaces[32].

Boosting changes adaptively the distribution of the training set based on the performance of previously created classifiers. The final classifier takes a weighted majority vote of the predictions. The bagging algorithm uses bootstrap samples to build the base classifier. The final classification produced by the ensemble using base classifiers is obtained using equal weight voting. The random subspaces approach combines multiple classification trees constructed in randomly selected subspaces. The final classification is obtained by an equal weighting of the base trees. Breiman[6] developed a random forest by combining classification trees such that each tree is generated by bagging and a random subspace of the predictors is used at each node.

The work in this thesis investigates the experience-consistent model using an approach similar to the ensemble architecture in the third category.

2.3.2 Regression ensemble models

Ensemble predictions can be used to provide probabilistic distributions of a future scenario (probabilistic predictions). They can also be used to provide best estimates of the future state of the atmosphere (deterministic predictions). The deterministic prediction[35] is of interest in the study of linear regression.

In meteorology, Kirshanamurti et al.[30] used multiple linear regressions to improve weather forecasts by specifying the weights for members of an ensemble of models.

Pagowski et al.[44] used ensemble forecasts[11][22] of surface ozone over the eastern USA and southern Canada to show that overall statistics of ensemble forecasts can be improved compared to averaging through linear regressions.

2.3.3 Distributed fuzzy rule-based model

Ravi and Zimmermann's[51] fuzzy rule-based classifier generates fuzzy 'if-then' rules from a numerical dataset in the training phase. In the test phase, these fuzzy rules make predictions in the test dataset as to which class they belong. In the case of the very high dimensional dataset, the fuzzy rule based classifier[51] cannot predict directly because of a large number of predictor variables. Hence they employed TreeNet[38] as a preprocessing module to perform feature selection. Further, after the fuzzy classifier generated fuzzy 'if-then' rules, a combinatorial global optimization algorithm was invoked to solve a multiobjective optimization problem with two objective functions, maximization of the classification rate and minimization of the size of the rules base. The result of this optimization stage is a compact set of fuzzy rules with a very high classification rate; but this severely affects the interpretability of the rule base. Lack of clear interpretation is a disadvantage of a distributed representation of fuzzy rules.

In this study, fuzzy rules corresponding to local sites are first computed, then rule correspondence between central and remote sites is established. Finally, rules are combined using a collaborative scheme at the central site to increase their interpretability.

2.3.4 Distributed neural networks model

Hansen and Salaman[25] provide some of the first research results to demonstrate that the generalization error of a neural network[10][17][31][56][59][64] can be significantly reduced by using an ensemble of similar networks, all trained with the same data[25]. They referred to this strategy as a crossvalidation(CV)[25] ensemble. Hansen and Salaman explain that the CV has a performance advantage because the multitude of local minima encountered in the training of individual neural networks results in errors occurring in different regions of input space. The collective decision of the ensemble is therefore less likely to be in error than a decision made by an individual network[39].

Fish, Barnes, and Aiken[18] proposed a new methodology for industrial market segmentation by integrated neural networks.

Lee, Chiu, Lu, and Chen[37] explored the performance of credit scoring by integrating backpropagation neural networks with traditional discriminate analysis.

2.4 Conclusions

Historical fundamentals of work relevant to our research is presented. Research in this area is mainly related to consensus distributed computing—which is static, with no training required at the aggregating stage. Consensus models are standalone, knowledge-reusable models. Collaborating models are distributed computing models in which standalone models integrate knowledge reusable models in an active manner. We extend the concepts presented in [45] to solve different collaborative clustering issues and to introduce a new experience-consistent collaborative model for classification problems. We discuss newly developed approaches in Chapters 5–9.

References

1. D. E. Bakken, R. Parameswaran, D. M. Blough, A. A. Franz, T. J. Palmer, Data obfuscation: anonymity and desensitization of usable data sets, *IEEE Security and Privacy,* 2, 2004, 34–41.

2. J. M. Bates, C. W. J. Granger, The combination of forecasts, *Operations Research Quaterly*, 20, 1969, 451-468.

3. A. Blum and T. Mitchell, Combining labeled and unlabeled data with co-training, *In Proceedings of the Conference on Computational Learning Theory*, 1998, 92-100.

4. L. Breiman, Arcing classifers, *Ann. Statist.* 26, 1998, 801–849.

5. L. Breiman, Bagging predictors, *Mach. Learning*, 24, 1996, 123–140.

6. L. Breiman, J. H. Friedman, R. A. Olshen, C. J. Stone, *Classifcation and Regression Trees*, Wadsworth, Belmont, 1984.

7. S. G. Cao, N. W. Rees, G. Feng, Analysis and design for a class of complex control systems, Part I and Part II, *Automatica*, 33, 6, 1997, 1017-1028 and 33, 6, 1997, 1029-1039.

8. M. Collins and Y. Singer, Unsepervised models for named entity classification. In *EMNLP*,6 , 1999, 13-49.

9. D. R. Cox, E. J. Snell, *Analysis of Binary Data*, Chapman & Hall, London, 1989.

10. P. Cunningham, J. Carney, S. Jacob, Stability problems with artificial neural networks and the ensemble solution, *Artificial Intelligence in Medicine*, 20, 2000, 217–225.

11. M. Delle, et al., Ozone ensemble forecasts A Kalman filter predictor bias correction, *J. Geophys. Res.,* 111, D05308, doi:10.1029/2005D006311.

12. T. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, *J. Artificial Intell. Res.,* 1995, 263-286.

13. E. Dimitriadou, A. Weingessed, K. Hornik, Voting-merging: An ensemble method for clustering, *In Proc. Int. Conf. on Artificial Neural Networks,* Vienna, 2001, 217-224.

14. R. P. W. Duin, D. M. J. Tax, Experiments with classifier combining rules. In: J. Kittler, F. Roli, (Eds.), *Lecture Notes in Computer Science Springer,* 1857, 2000, 16–29.

15. M.C. Fairhurst, A. F. R. Rahman, Generalised approach to the recognition of structurally similar handwritten characters using multiple ability of expert classifiers, *IEE Proc. Vision, Image Signal Process.,* 144, 1997, 15-22.

16. K. H. Fasol, H. P. Jörgl, Principles of model building and identification, *Automatica,* 16, 5, 1980, 505-518.

17. L. Fausett, *Fundamentals of neural networks: architectures, algorithms and applications,* Prentice Hall, 1994.

18. K. E. Fish, J. H. Barnes and M. W. Aiken, Artificial neural networks- a new methodology for industrial market segmentation, *Industrial Marketing Management,* 24, 1995, 431-438.

19. Y. Freund, R. Schapire, A decision-theoretic generalization of online
learning and an application to boosting, *J. Comput. System Sci.,* 55, 1997, 119–139.

20. Y. Freund, R. Schapire, Experiments with a new boosting algorithm. In: *Machine Learning: Proceedings of the Thirteenth International Conference,* 1996, 148–156.

21. L. S. Jha, et al., Privacy preserving clustering, *in: Proceedings of the 10th European Symposium on Research in Computer Security,* 2005, 397–417.

22. E. Johnson, H. Kargupta, Collective, hierachical clustering from distributed heterogeneous data Large-Scale Parallel KDD Systems (Editors, Zaki,M.,Ho,C.), *Lecture Notes in Computer Science, Springer-Verlag,* 1759, 1999, 221-244.

23. M. I. Jordan, R. A. Jacobs, Hierarchical mixture of experts and the EM algorithm. *Neural Comput.,* 6, 1994, 181-214.

24. T. K. Ho, J. J. Hull, J. N. Srihari, Decision combination in multiple classifier systems, *IEEE Trans. Pattern Anal. Machine Intell.,* 16, 1994, 66-75.

25. L. K. Hansen, P. Salamon, Neural network ensemble, *IEEE, Transactions on Pattern Analysis and Machine Intelligence*, 12, 1990, 993-1001.

26. W. Hardle, *Applied Nonparametric Regression*, Cambridge University Press, 1990.

27. F. E. Harrell, K. L. Lee, A *comparison of the discrimination of discriminant analysis and logistic regression*, North-Holland, Amsterdam, 1985.

28. T. Hastie, R. Tibshirani, Generalized additive models, *Statistical Science*, 1, 1986, 297–318.

29. N. Indurkhya, S. M. Weiss, Solving regression problems with rule-based ensemble classifiers, *In ACM International Conference Knowledge Discovery and Data Mining (KDD01)*, 2001, 287–292.

30. T. N. Krishnamurti, et al. , Improved weather and seasonal climate forecasts from multi-model superensemble, *Science*, 1999, 1548-1550.

31. A. Krogh, J. Vedelsby, Neural Networks Ensembles, Cross validation, and Active Learning, *in: Advances in Neural Information Processing Systems, MIT Press Cambridge*, 1995, 231–238.

32. T. K. Ho, The random subspace method for constructing decision forests, *IEEE Trans. Pattern Anal. Mach. Intelligence*, 20, 1998, 832–844.

33. K. Kailing, H. Kriegel, A. Pryakhin, and M. Schubert, Clustering mluti-represented objects with noise. In Proc. *of the Pacific-Asia Conf. on Knowl. Disc. and Data Mining*, 2004, 394-403.

34. G. Karypis, V. Kumar, A fast and high quality multilevel scheme for portioning irregular graphs, *SIAM Journal of Scientific Computing*, 20, 1998, 359-392.

35. E. K. Laitinen, Predicting a corporate credit analyst's risk estimate by logistic and linear models, *Int. Rev. Financial Anal.*, 8, 1999, 97-121.

36. J. C. Kim, D. H. Kim, J. J. Kim, J. S. Ye, H. S. Lee, Segmenting the Korean housing market using multiple discriminant analysis, *Constr. Manage. Econ.*, 18, 2000, 45-54.

37. T. S. Lee, C. C. Chiu, C. J. Lu, I. F. Chen, Credit scoring using the hybrid neural discriminate technique, *Expert Systems with Applications*, 23, 2002, 245-254.

38. H. Lee, H. Jo, I. Han, Bankruptcy prediction using case-based reasoning, neural networks, and discriminant analysis, *Expert Systems Appl.*, 13, 1997, 97-108.

39. G. Lee, T. K. Sung, N. Chang, Daynamics of modeling in datamining: interpretive approach to bankruptcy prediction, *J. Manag. Inform. Systems,* 16, 1999, 63-85.

40. S. Merugu, J. Ghosh, A privacy-sensitive approach to distributed clustering, *Pattern Recognition Letters*, 26, 2005, 399-410.

41. A. Miller, *Subset Selection in Regression*, Chapman & Hall, Los Angeles, 2002.

42. S. R. M. Oliveira, O. R. Zaiane, Achieving privacy preservation when sharing data for clustering, *in: Proceedings of the International Workshop on Secure Data Management in a Connected World*, 2004, 67–82.

43. S. R. M. Oliveira, O. R. Zaiane, Privacy preserving clustering by data transformation, *in: Proceedings of the 18th Brazilian Symposium on Databases*, 2003, 304–318.

44. M. Pagowski, et al. A simple method to improve ensemble-based ozone forecasts, Geophys. Res. Lett., 32, Lo7814, doi:10.1029/2004GL022305.

45. W. Pedrycz, Collaborative Fuzzy Clustering, *Pattern Recognition Letters*, 23, 2002, 675-1686.

46. W. Pedrycz, Distributed fuzzy systems modeling, *IEEE Transactions on Systems, Man, and Cybernetics*, 25, 1995, 769-780.

47. W. Pedrycz, *Knowledge-Based Clustering: From Data to Information Granules*, J. Wiley, Hoboken, NJ, 2005.

48. W. Pedrycz, Conditional fuzzy clustering in the design of radial basis function neural networks, *IEEE Transactions on Neural Networks*, 9, 1998, 601-612.

49. W. Pedrycz, M. Reformat, Evolutionary fuzzy modeling, *IEEE Transactions on Fuzzy Systems*, 11, 2003, 652-665.

50. W. Pedrycz, G. Vukovich, Clustering in the framework of collaborative agents, *Proc. 2002 IEEE Int. Conference on Fuzzy Systems*, 1, 2002, 134-.

51. V. Ravi H. J. Zimmermann, Fuzzy rule based classification with FeatureSelector and modified threshold accepting, *European Journal of Operational Research, Elsevier*, 123, 2000, 16-28.

52. R. E. Schapire, The strength of weak learnability, *Mach. Learning,* 5, 1990, 197–227.

53. A. Strehl, J. Ghosh, Cluster ensembles: a knowledge reuse framework for combining multiple partitions, *Journal of Machine Learning Research*, 3, 2002, 583-617.

54. L. Sweeney, Anonymity: a model for protecting privacy, *International Journal*

on Uncertainty, Fuzziness and Knowledge-based Systems, 10, 2002, 557–570.

55. A. Topchy, K. Jain, W. Punch, Clustering ensembles: models of consensus and weak partitions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 2005, 1866-1881.

56. N. Ueda, Optimal linear combination of neural networks for improving classification performance, *IEEE Trans. Pattern Anal. Machine Intelligence,* 22, 2000, 207–215.

57. V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, NewYork, 1995.

58. J. Wang, H. Zeng, Z. Chen, H. Lu, L. Tao, and W. Ma, Reinforcement clustering of multi-type interrelated data objects. In Proc. *of the 33rd Annual Meeting of the Association for Comp. Linguistics*, 1995, 297-301.

59. D. Wolpert, Stacked generalization, *Neural Networks*, 5, 1992, 241–259.

60. K. Woods, W.P. Kegelmeyer, K.Bowyer, Combination of multiple experts using local accuracy estimates, *IEEE Trans. Pattern Anal. Machine Intell.*, 19, 1997, 405-410.

61. L. Xu, A. Krzyzak, C. Y. Suen., Methods of combining multiple classifiers and their applications to handwriting recognition., *IEEE Trans. Systems Man Cybernet,* 22, 1992, 418-435.

62. D. Yarowsky, Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. of the 33rd Annual Meeting of Association for Comp. Linguistics*, 1995, 454-460.

63. L. A. Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*, 90, 1997, 111-117.

64. Z. H. Zhou, J. X. Wu, W. Tang, Z. Q. Chen, Combining regression estimators: GA-based selective neural Intelligence and Applications, 1, 341–356, 2001.

Chapter 3

Experimental Datasets

This chapter discusses details of the different datasets analyzed in this study. Several synthetic 2 dimensional datasets are used to illustrate the working of the algorithms and to assess their performance. In addition to synthetic datasets, a variety of datasets are also selected from UCI[1], StatLib[6], and Weather Canada repositories[5].

We evaluate a fuzzy collaborative clustering method with real-world datasets such as Boston, Abalone, Wine, and Weather Canada. We assess algorithms using linear regression methods through Abalone, Auto-mpg, Machine-CPU, Breast Cancer, Auto Car Price, California Housing, Friedman Synthetic and Boston Housing datasets. The linear classifier model is demonstrated through a suite of datasets such as Breast Cancer, Ionosphere, Contraception, Ringnorm, Twonorm, and Liver disorder. We demonstrate a fuzzy rule based model using datasets such as Abalone, Boston, Auto-mpg and California Housing. Finally, Radial-basis function networks model is evaluated through Auto-mpg and Boston Housing datasets.

3.1 Methodology

To use a statistical approach in designing and analyzing an experiment, it is necessary to have a clear idea in advance of exactly how the data are to be collected and at least a qualitative understanding of how these data are to be analyzed. The following are some considerations we exercise in conducting experiments with our collaborative approaches.

In experimenting with vertical collaborating clustering, the entire dataset is divided into parts of equal size. All parts contain the same feature space.

In horizontal collaborating clustering, the entire dataset is divided into different parts with features containing the same instances.

Using a rule-based experience-consistent approach we build the model at the central site with 5% of instances; the balance of collaborating sites have the remaining 95% instances in equal sized parts. All parts contain the same feature space.

3.2 Synthetic datasets

The synthetic datasets generated in this study have a normal distribution with a mean vector and covariance matrix $N(\mu,\Sigma)$. Synthetic datasets with different topologies are generated to demonstrate the concept of two-class classification horizontal and vertical clustering. The details of such synthetic datasets are described in later chapters.

In experiments with synthetic datasets, the primary interest lies in simulating real-world scenarios. The challenge is to reconstruct with available methods the 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 allow us to tune the algorithms to our needs.

3.3 Machine learning datasets

The UCI repository of machine learning databases[4] provided several datasets for this study and we have selected those which are widely used in the literature to benchmark algorithms. They are used by the machine learning community for empirical analysis of algorithms. The UCI repository has a large collection of datasets covering a range of topics from the chemical analysis of wine to biological analyses, therefore, a variety of classifications (linear regressions, predictions) and clustering tasks can be performed on them.

Boston Housing

Boston Housing is the dataset derived from information collected by the U.S Census Services concerning housing in the Boston area. The aim is to predict the median value of a house. The dataset contains 506 records of real estate prices and 13 related characteristics of the houses as described in [4]. Quinlan in [7] used this dataset to combine instance-based and model-based learning methods.

Abalone

The original owner of the Abalone database is the Marine Research Laboratories in Tasmania, Australia. An abalone is a marine crustacean. The age in years of an abalone can be obtained by adding 1.5 to the number of rings. The number of rings varies between 1 and 29. The dataset contains 4,177 records of marine crustaceans, 8 input attributes, and 1 output variable. The task is to predict the age of the abalone from physical measurements. This dataset was used by Waugh[9] for a cascade-correlation problem and Clark and Schreter[3] used it in a 3-category classification problem.

Auto-mpg

The Auto-mpg dataset contains a set of mileage records in miles per gallon (mpg) for different cars. The aim is to predict the fuel consumption of different car models in miles per gallon in terms of 3 multivalued discrete and 5 continuous attributes. The dataset contains 398 records and a total of 9 attributes. Quinlan in [7] used this dataset to combine instance-based and model-based learning methods.

Wine

The Wine dataset contains the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents (attributes) found in each of three types of wine. In total there are 178 records; 59 records are of class 1, 71 belong to class 2, and 48 to class 3. Aeberhard et al.[1] used this dataset for a classification problem.

Breast Cancer

The Breast Cancer dataset contains 198 records and 34 (ID, outcome, 32 real-valued input features) attributes as described in [1]. Class distributions are 151 nonrecur (for nonrecurring disease) and 47 recur (where disease recurs). We have removed four cases with unknown values of the last attribute. In the problem of a classification we remove the dependent attribute as it is the class attribute. Wolberg[10][11] used the Breast Cancer dataset to analyze images in bioinformatics.

Ionosphere

The Ionosphere dataset uses data from radar sensing. It has 351 instances and all 34 attributes are continuous. There were 17 pulse numbers for the Goose Bay system. Each instance is described by 2 attributes per pulse number. Each pulse number independently represents the instance. This radar data was collected by a system in Goose Bay, Labrador. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not; their signals pass through the ionosphere. The 35th attribute is either "good" or "bad" and this is a binary classification task. Sigillito et al.[8] used this dataset for classification using a neural network model.

Contraception

The Contraception dataset is a subset of the National Indonesia Contraceptive Prevalence Survey (1987). The instances are married women who were either not pregnant or did not know if they were pregnant at the time of interview. The task is to predict the current contraceptive method of choice (no use, long-term methods, short-term methods) by a woman based on her socio-economic and demographic characteristics. This dataset has 1,473 instances and 10 attributes.

Liver disorder

BUPA Liver Disorders contains 7 attributes of 345 instances of excessive alcohol use by male individuals. The first 5 variables are blood tests which are thought to be sensitive to liver disorders that might arise from excessive alcohol consumption. The description of attributes is in [4].

Machine-CPU

The Machine-CPU dataset is concerned with relative CPU performance. There are a total of 209 cases in the dataset and 7 attributes. More information can be obtained in the UCI Machine Learning repository[4].

Auto Car Price

The original data from the UCI repository[4] for Auto Car Price has 205 instances described by 26 attributes: 15 continuous, 1 integer, 10 nominal. The original data has some missing attribute values. We have changed the original data by removing all unknown cases leaving 159 instances. Also, all nominal attributes (10) were removed.

3.4 StatLib repository dataset

California House

We have considered a California House high dimensional dataset from the Statlib repository[6]. We collected information on the variables using all the block groups in California from the 1990 Census. In this sample a block group on average includes 1425.5 individuals living in a geographically compact area. This dataset contains 20,640 observations with 9 attributes. The dependent variable is in (median house value).

3.5 Canada Weather network data

This weather network site provides direct access to the Canadian National Climate Archive, operated and maintained by Environment Canada. It contains official climate and weather observations. In this study we collect the Alberta and British Columbia weather data available at this weather network site[5].

Alberta

We selected 10 data sites geographically distributed in the province of Alberta: (1) Beaver Mines, (2) Calgary INT A, (3) Kananaskis, (4) Bindloss East, (5) Big stone, (6) Alliance South, (7) Cold Lake A, (8) Athabasca-2, (9) Brule Black, and (10) Ballater. The collected data comprises 801 weather records collected over the winter seasons (December, January, and February) of 1991–2000. Each site is described by four features: maximum temperature, minimum temperature, average temperature, and precipitation.

British Columbia

Following the same scheme as discussed for the Alberta weather data, we consider 10 data sites: (1) Chemainus, (2) Black Creek, (3) Albeni Robertson Creek, (4) Boat Bluff, (5) Gibson's Gower Point, (6) Langara, (7) Bella Coola A, (8) Babine Lake Pinkut Creek, (9) Hixon, and (10) Penticton A. Each datasite is described by four features: maximum temperature, minimum temperature, average temperature, and precipitation.

3.6 Other sources

Friedman Synthetic

The Friedman Synthetic dataset has 40,768 cases and 10 attributes (all continuous). The cases are generated using the following method: the values of 10 attributes, X1, …, X10, are generated independently; each is uniformly distributed over [0,1]. The value of the target variable Y is then obtained. The detailed description of the dataset is in [2].

Ringnorm

Leo Breiman's ringnorm example[2] is a classification of two normal distributions, one within the other; origin: artificial. It is a 20 dimensional, 2 class classification example. Each class is drawn from a multivariate normal distribution. Class 1 has a mean of zero and a covariance of 4 times the identity. Class 2 has a mean of (a,a,..a) and a unit covariance a = 2/sqrt(20). This dataset contains 7,400 observations with 21 attributes.

Twonorm

The Twonorm dataset contains 7,400 observations. It is a 20 dimensional, 2 class classification example. Each class is drawn from a multivariate normal distribution with unit variance. Class 1 has a mean of (a,a,..a) while class 2 has a mean of (-a,-a,..-a), where a = 2/sqrt(20).

3.7 Conclusions

This chapter describes the datasets explored in this study. To demonstrate collaborative approaches, the entire dataset is splitted into several parts (P), each part becoming a separate datasite in a collaborative model.

References

1. S. Aeberhard, D. Coomans and O. de Vel, *Comparison of Classifiers in High Dimensional Settings*, *Tech. Rep. no. 92-02*, Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, 1992.

2. L. Breiman, Bagging predictors, *Machine Learning* , 24(3), 1996, 123-140.

3. D. Clark, Z. Schreter, A. Adams, A Quantitative Comparison of Dystal and Backpropagation, *submitted Australian Conference on Neural Networks (ACNN),* 1996.

4. http://www.ics.uci.edu/~mlearn/MLRepository.html

5. http://www.climate.weatheroffice.ec.gc.ca/prods_servs/cdcd_iso_e.html.

6. http://lib.stat.cmu.edu/

7. R. Quinlan, Combining Instance-Based and Model-Based Learning. *In Proceedings on the Tenth International Conference of Machine Learning*, University of Massachusetts, Amherst. Morgan Kaufmann, 1993, 236-243

8. V. G. Sigillito, S. P. Wing, L. V. Hutton, and K. B. Baker, Classification of radar returns from the ionosphere using neural networks, *Johns Hopkins APL Technical Digest*, 10, 1989, 262-266.

9. S. Waugh, *Extending and benchmarking Cascade-Correlation*, PhD. thesis, Computer Science Department, University of Tasmania., 1995.

10. W. H. Wolberg, W. N. Street, and O. L. Mangasarian, Image analysis and machine learning applied to breast cancer diagnosis and prognosis, *Analytical and Quantitative Cytology and Histology*, 17(2), 1995, 77-87.

11. W. H. Wolberg, W. N. Street, D. M. Heisey, and O. L. Mangasarian, Computer-derived nuclear 'grade' and breast cancer prognosis, *Analytical and Quantitative Cytology and Histology*, 17, 1995, 257-264.

Chapter 4

Selected Fundamentals

This chapter introduces the fundamental concepts and models used at each collaborating site. Fuzzy sets, triangular fuzzy sets, type-2 fuzzy sets, a fuzzy rule based model, and radial-basis function neural networks are introduced. Data analysis issues with respect to clustering, classifications, and prediction models are discussed.

4.1 Fuzzy sets

In many situations the Boolean membership of an object x to set A is too restrictive. For example, we can express the set of middle-aged persons as a collection of persons who fall in the range of 21–45 years of age. A classical set classifies a person of 44.99 years but not a person of 45.1 years as a middle-aged person. This sharp transition between inclusion and exclusion in a set is intuitively inconsistent.

Zadeh[28] proposed a new type of set theory, the fuzzy set, that allows the representation of concepts that are not well defined. Fuzzy sets allow an element to belong to a set with a degree of membership. The membership degree of an object to a fuzzy set takes values between 0 and 1, where 1 means entirely in the set, 0 means fully excluded from the set, and other values mean partial membership in the set. The degree of membership of an object in a fuzzy set is defined as a function where the universe of discourse is the domain, and the interval [0, 1] is the range. The higher the membership grade the stronger the association of the given elements to the concept. This simple concept is more appropriate than the classical concept of a set for capturing semantic, linguistic, and real-world vagueness.

Generally, a fuzzy set A defined in the domain of X is characterised by membership. We can specify a fuzzy set as follows:

$$A : \mathbf{X} \rightarrow [0,1]. \tag{1}$$

Thus, a fuzzy set A in X may be represented as a set of ordered pairs of a generic element $x \in X$ and its grade of membership can be represented as $A = \{(A(x)/x) \mid x \in X\}$.

A fuzzy set A in X is directly specified by the function A(x). This represents the value of the "grade of membership" of each x in A. In practice, the form of membership functions should reflect the problem at hand (semantics) for which we are constructing fuzzy sets. The functions should also reflect our perception of the concept to be represented and used in problem solving, the level of detail we intend to capture, and the context in which the fuzzy sets are going to be used[19]. An important class of membership grade is the triangular as explained in the next section. References[15][19][3] provide more information on fuzzy sets and membership functions.

4.1.1 Triangular fuzzy sets

A triangular MF is specified by three parameters {a,m,b} by two piecewise linear segments as follows:

$$\text{triangle}(x;a,m,b)=\begin{cases}0, x\le a.\\ \dfrac{x-a}{m-a}, a\le x\le m\\ \dfrac{b-x}{b-m}, m\le x\le b.\\ 0, b\le x.\end{cases} \tag{2}$$

By using min and max, we have an alternative expression for the preceding equation:

$$\text{triangle}(x;a,m,b)=\max(\min(\frac{x-a}{m-a},\frac{b-x}{b-m}),0). \tag{3}$$

The meaning of parameters {a,m,b} is straightforward: m denotes a modal (typical) value of the fuzzy set while a and b denote lower and upper bounds, respectively. Triangular fuzzy sets (membership functions) are the simplest possible models of grades of membership as they are fully defined by only three parameters. Figure 4.1 illustrates a triangular membership function defined by triangle (x;0.3,0.5,0.7).

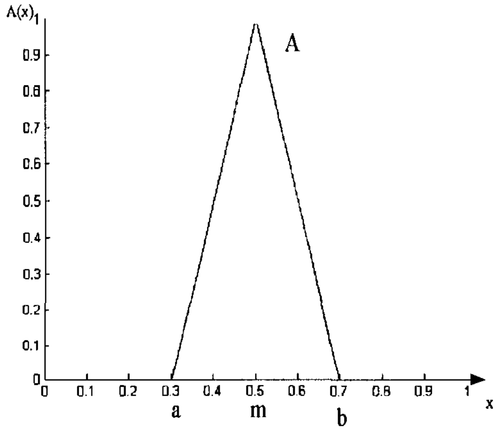


Figure 4.1 Triangular membership function: m is the modal value, a is the left bound, and b is the right bound of the fuzzy set.

A collection of fuzzy sets, called fuzzy space, describes fuzzy classes that an object can belong to. In this way, fuzzy sets allow an object to belong to different classes at the same time with different grades of membership, as shown in Figure 4.2.

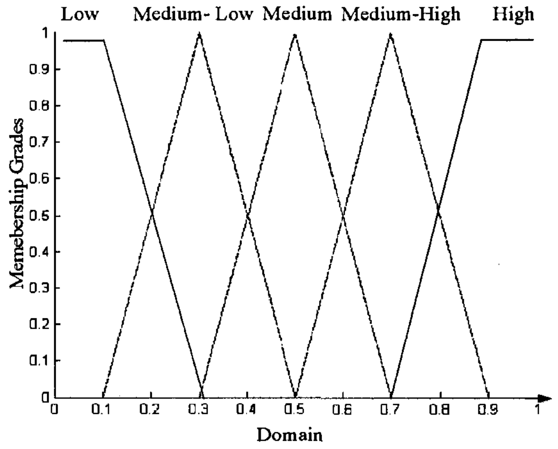


Figure 4.2 Triangular fuzzy sets.

4.2 Type-2 fuzzy sets

Type-2 fuzzy sets were introduced by Zadeh[29] as an extension of the concept of an ordinary, type-1 fuzzy set. Membership functions (MF) for type-2 fuzzy sets are characterized by more parameters than MFs for type-1 fuzzy sets. In this way type-2 fuzzy sets[19][10] generalize the concept of fuzzy sets and are defined in the form:

$$A : X \rightarrow F([0,1]) \qquad (4)$$

This transformation means that for each element x in the domain X we have a fuzzy set of membership grades defined in the unit interval. In such a construct, A is treated as a function of two variables, that is, A(x, u), $x \in X$, $u \in [0,1]$. As the grades of membership are fuzzy sets themselves, one can view type-2 fuzzy sets as fuzzy sets with linguistic membership grades. For example, type-2 fuzzy set A = {young, middle-aged, senior} where all linguistic variables young, middle-aged, and senior are also fuzzy sets in the unit interval with membership functions. From this point of view type-2 fuzzy sets are "fuzzy fuzzy" sets and are more expressive.

4.2.1 Type-2 fuzzy set membership estimation

Type-2 fuzzy sets have grades of membership that are themselves fuzzy. At each value of a primary variable (e.g., pressure, temperature), membership is a function (not just a point value)—the secondary MF—whose domain—the primary membership—is in the interval [0, 1]. Hence, the MF of a type-2 fuzzy set is three-dimensional, and it is the third dimension that provides new design degrees of freedom for handling uncertainties. Thus, in uncertain environments type-2 fuzzy sets have the potential to outperform type-1 fuzzy sets.

The membership function in a type-2 fuzzy set is computed[3] as follows:
In collaborative clustering we estimate the membership function on a basis of collection of membership grades available in different partition matrices. Consider what is known about cluster membership of pattern x in D[ii] given the results of collaborative clustering. The membership in the i-th cluster is computed using prototypes of D[ii] and is denoted as u = u_i. The prototypes optimized for the jj-th data site, jj = 1, 2, ..., ii-1, ii+1, ..., P give rise to the membership of x to the same i-th cluster. Denote them by z_1, z_2, ..., z_{P-1}. We obtain a collection of membership grades which are now captured in the form of a type-2 fuzzy set. The corresponding membership function is determined by solving a certain optimization problem[3]. As discussed below this realizes a principle of *justifiable* granularity.

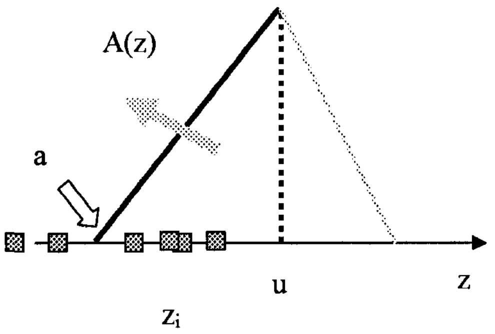


Figure 4.3 Computation of a membership function of a type-2 fuzzy set; note that in order to maximize the performance index, we rotate the linear segment of the membership function around the modal value of the fuzzy set. Small dark boxes denote available experimental data. The same estimation procedure applies to the right-hand side of the fuzzy set.

Two requirements guide the design of the fuzzy set, namely:

(a) The experimental evidence of the fuzzy set is maximized to "cover" as many numeric data as possible, viz. the coverage has to be made as high as possible. In the graphically in the optimization of this requirement, we adjust the parameters of the membership function that makes it shrink or expand so that more data are embraced . The sum of the membership grades $A(z_i)$ is $\sum_i A(z_i)$, where A is the linear membership function to be optimized with respect to its slope and z_i is located to the left of the modal value u. $\sum_i A(z_i)$ has to be maximized.

(b) Simultaneously, we would like to make the fuzzy set as specific as possible so that it represents some well defined semantics. This requirement is met by making the support of A as small as possible, that is, $\min_a|u - a|$.

To accommodate the two conflicting requirements, we have to combine a and b into a single scalar index which in turn becomes maximized. Two alternatives are:

$$\max_{a \neq u} \frac{\sum_{i} A(z_i)}{|u-a|}, \tag{5}$$

or

$$\sum_{i} (1 - A(z_i))(u-a). \tag{6}$$

We exclude a trivial solution of a = u in which case the type-2 fuzzy set collapses to a type-1 fuzzy set (with numeric values of membership functions).

4.3 Data clustering

Data clustering is used in pattern recognition, exploratory data analysis, computer vision, machine learning, and many other related fields[25][24]. Data clustering is the process of identifying natural grouping or clusters within multidimensional data based on some similarity measures, for instance, a cluster is usually identified by a cluster center (or prototype). Data clustering is a difficult problem in unsupervised pattern recognition as the clusters in data may have different shapes and sizes[1].

A pattern (or feature vector) **x** is a single object or data item used by the clustering algorithm. A feature (or attribute) is an individual component of a pattern. A cluster is a set of similar patterns; patterns from different clusters are not similar. Clustering algorithms assign each pattern to one and only one cluster. Fuzzy clustering algorithms assign each pattern to each cluster with some degree of membership. A distance measure is a metric used to evaluate the similarity of patterns as discussed below:

Distance functions

Clustering is the process of identifying natural groupings or clusters within multidimensional data based on dissimilarity or similarity measures. Hence, dissimilarity measures are fundamental components in most clustering algorithms[10][3].

The lower the distance between two patterns, the higher the level of their similarity. Initially, the family of Minkowski distances[30] is discussed:

$$d(x,y) = \sqrt[p]{\sum_{i=1}^{n} |x_i - y_i|^p}, \ p > 0, \tag{7}$$

where x,y$\in\mathbf{R}^n$; depending upon different values of p, we have different forms of the distance function.

When p = 2, we compute the Euclidean distance as follows:

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{n} \left|\mathbf{x}_i - \mathbf{y}_i\right|^2} \; . \tag{8}$$

More specifically, for any two patterns **x** and **y** in X, $||\mathbf{x}\text{-}\mathbf{y}||^2 = \sum_{j=1}^{n} \frac{(\mathbf{x}_j - \mathbf{y}_j)^2}{\sigma_j^2}$, where σ_j^2 is the sample variance of the j-th coordinate (variable) of the feature space.

A measure of the size is the variance, so, the variance of the distance to the prototype of all the elements in a cluster (σ_j^2) is calculated.

When p = 1 the measure is referred to as the Hamming distance (city block):

$$d(\mathbf{x},\mathbf{y}) \;\; = \sum_{i=1}^{n} \left|\mathbf{x}_i - \mathbf{y}_i\right| \; .$$

When p = ∞ the measure is referred to as the Tschebychev distance:

$$d(x,y) = \max i = 1,2,\ldots\ldots,n\left|x_i - y_i\right| .$$

Another commonly used distance measure is the Mahalanobis, defined as:

$$d(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^T \Sigma^{-1} (\mathbf{x}-\mathbf{y}) , \tag{9}$$

where $\sum^{-1}$ is the covariance matrix of patterns. The Mahalanobis distance gives different features different weights based on their variances and pair-wise linear correlations. Thus, this metric implicitly controls the geometry of potential clusters[25].

4.4 Clustering techniques

The two main categories of clustering algorithms are based on two popular techniques known as hierarchical and objective function-based clustering[17]. In general we think of clustering[20][21][22] as a vehicle of forming information granules. Fuzzy set theory offers an adequate framework that requires interpretation of the input and output of the clustering model. Fuzzy clustering has been widely studied and applied in a variety of substantive areas[17]. The following is an overview of hierarchical and objective function-based techniques.

4.4.1 Hierarchical clustering techniques

The clustering techniques in this category generate a cluster tree (or graphic representation) by using a heuristic splitting or a merging method[8]. A cluster tree shows the sequence of clustering with each clustering being a partition of the dataset[21]. The construction of the graph is done in two ways: bottom-up and top-down. In the bottom-up mode known as an agglomerative approach, we treat each pattern as a single-element cluster and then successively merge the closet cluster. The process is repeated until we get a single cluster[17]. In the top-down construction the algorithms that use splitting to generate the cluster tree are called divisive. Divisive hierarchical algorithms start with all patterns assigned to a single cluster. Then splitting is applied to a cluster at each stage until each cluster consists of one pattern[17].

Several agglomerative hierarchical algorithms are proposed in the literature; these differ in the way the two most similar clusters are calculated. An important issue is how to measure the distance between two clusters[17]. The two most popular agglomerative hierarchical algorithms are the single link[1][25] and the complete link[2] algorithms. Single link algorithms merge the clusters with the smallest distance between their closest patterns. Complete link algorithms merge the clusters whose distance between their most distant patterns is the maximum[17]. In general, complete link algorithms generate compact clusters while single link algorithms generate elongated clusters as illustrated in Figure 4.5 and Figure 4.4 respectively. The results of hierarchical clustering are usually represented in the form of dendrograms. Hierarchical clustering techniques have the following advantages:

- The number of clusters need not to be specified a priori, and
- They are independent of the initial conditions.

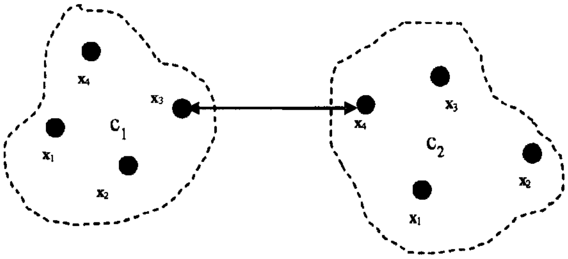


Figure 4.4 Single link algorithm: the distance between x_3 and x_4 (the closest patterns in clusters C_1 and C_2) is the smallest

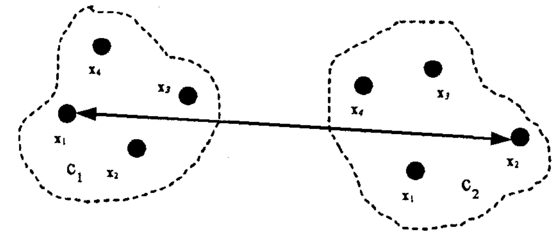


Figure 4.5 Complete link algorithm: the distance between x_1 and x_2, the most distant patterns, is the maximum in clusters C_1 and C_2.

The group average link method considers the average among distances computed between all pairs of patterns, one from each cluster. This method is more computationally intensive, but reflects the general distance computed between individual pairs of patterns.

Hierarchical clustering (HC) techniques suffer from the following drawbacks:

- HC techniques are computationally expensive in terms of time complexity and space complexity[1]. Hence, they are not suitable for very large datasets.

- HC techniques may fail to separate overlapping clusters due to a lack of information about the global shapes or sizes of the clusters.

- HC techniques are static, i.e., patterns assigned to a cluster are difficult to merge.

4.4.2 Objective function-based clustering techniques

The second general category of clustering deals with building partitions (i.e., clusters) of data on the basis of objective function.

Objective function-based clustering techniques are more frequently used than hierarchical techniques in pattern recognition[9]. Hence in our study, we concentrate on objective function-based clustering techniques.

Fuzzy c-means (FCM) is one of the best known clustering techniques; it is based on the minimization of an objective function[15][19].

Fuzzy clustering algorithms are objective function-based, i.e., division into clusters is determined by optimizing an objective function. Each cluster is represented by a cluster prototype. This prototype consists of a cluster centre (whose name indicates its purpose) and may contain additional information about the sizes and the shapes of the clusters. The cluster centre is computed by the clustering algorithm and may or may not appear in the dataset. The shape and size parameters determine the extension of the cluster in different directions. The degrees of membership to which a given data point belongs to different clusters are computed from the distances of the data point to the cluster centres with respect to the size and shape of the cluster as stated by the additional prototype information. The closer a data point lies to the centre of a cluster (with respect to size and shape), the higher is its degree of membership to this cluster. Hence the problem of dividing a dataset D= $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_N\}$ lying in $\mathbf{R}^n$ into c clusters can be stated as the task to minimize the distances of the data points to the cluster centers. This technique tries to minimize the objective function. The disadvantages of hierarchical algorithms are advantages of objective function-based clustering algorithms and vice versa. The design challenge lies in formulating an objective function capable of reflecting the nature of the problem and whose minimization reveals a meaningful structure in the data space [17].

$$Q = \sum_{k=1}^{N} \sum_{i=1}^{c} u_{ik}^{m} d_{ik}^{2}, \; m>1,$$

where k=1,2,…,N, and i=1,2,…,c

(10)

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} \sum_{i=1}^{c} u_{ik}^{m} \mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^{m}},$$

(11)

$$d_{ik}^{2} = \sum_{j=1}^{n} \frac{(x_{kj} - v_{ij})^2}{\sigma_j^2},$$

(12)

where σ_j^2 is variance of the j-th feature x_{kj}:

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

(13)

Clustering algorithms use a Boolean membership function (i.e., $u_{ik} \in \{0, 1\}$) while fuzzy clustering algorithms use a degree of membership function (i.e., $u_{ik} \in [0, 1]$).

Different stopping criteria can be used in an iterative clustering algorithm, for example:

- When the change in prototype values is smaller than Є (very small user-specified value),
- When the changes in membership values are smaller than Є (very small user-specified value),
- When a maximum number of iterations have been exceeded.

The use of fuzzy sets in the fuzzy model is normally created by categorized results (i.e., knowledge) from fuzzy clustering methods. We use and integrate such knowledge in constructing collaborative models.

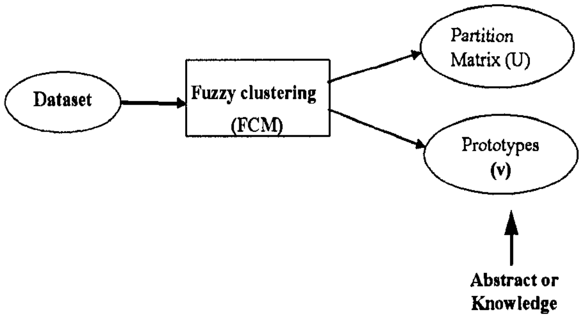


Figure 4.6 Data clustering resulting in abstract data (or knowledge).

4.5 Proximity measures

For any partition matrix U = [u_{ik}], i = 1, 2,..., c, k = 1, 2, ..., N, an induced proximity matrix[13], that is Prox = [prox(k, l)], k, l = 1, 2,..., N, comes with entries which satisfy the following properties:

(a) symmetry $\mathrm{prox}(k_1, k_2) = \mathrm{prox}(k_2, k_1)$
(b) reflexitivity $\mathrm{prox}(k_1, k_1) = 1.0$

The proximity values are based on the corresponding membership degrees occurring in the partition matrix:

$$\mathrm{prox}(k_1, k_2) = \sum_{i=1}^{c} \min(u_{ik_1}, u_{ik_2}). \tag{14}$$

Note that the proximity matrix is more abstract than the original partition matrix it is based upon. It "abstracts" the clusters themselves and this is what we need in the construct of collaborative clustering. Given the proximity matrix, we cannot "retrieve" the original entries of the partition matrix from which it was generated.

4.6 Data classification

Classification is the process of using the feature vector to assign the pattern to a category. A very basic classifier for assigning a feature vector to 1 to 2 classes is the linear classifier[16] in which the output is computed as follows:

$$y = \sum_{j=1}^{n} a_j x_j + a_0, \qquad a_0, a_1, \ldots a_j \in R, j = 1,2,\ldots,n, \tag{15}$$

where **a** denotes a set of weights, **a** = $[a_0\ a_1\ a_2 \ldots a_n]^T$; a_0 is usually referred to as bias and **x** represents the set of input features, **x** = $[1\ x_1\ x_2\ \ldots\ x_n]^T$, and the target y is a linear combination of the input features. We determine optimal weights through the pseudoinverse method found in many statistics text books.

4.7 Classification for system identification

The problem of determining a mathematical model for an unknown system or a target system by observing its input-output data pairs is generally referred to as system identification. The purposes of system identification are multiple:

- To predict a system's behaviour, as in time series predictions and weather forecasting.
- To explain interactions and relationships between inputs and outputs of a system. For example, a mathematical model can be used to examine whether demand varies proportionally to supply in an economic system.
- To design a controller (ship, aircraft control) based on the model of a system. Also, a model is required to do a computer simulation of a system under control.

System identification generally involves two top-down steps[14].

Step-1: Structure identification:

We need to apply a priori knowledge about the target system to determine a class of models within which the search for the most suitable model is to be conducted. Usually this class of models is denoted by a parameterized function y = f(**x**, **a**), where y is the model's output, **x** is the input vector, and **a** is the parameter vector. Determination of the function f is problem dependent; the function is based on the designer's experience and intuition and the laws of nature governing the target system.

Step-2: Parameter identification:

If the structure of the model is known, all we need to do is apply optimization techniques to determine the parameter vector $a = \hat{a}$ such that the resulting model $\hat{y} = f(\mathbf{x};\hat{\mathbf{a}})$ can describe the system appropriately.

If we do not have a priori knowledge about the target system, then structure identification is a difficult problem and we have to select the structure by trial and error. Fortunately, we know a great deal about the structures of most engineering systems and industrial processes. Consequently, the system identification problem is usually reduced to that of parameter identification. The problem of parameter identification is thus of great importance.

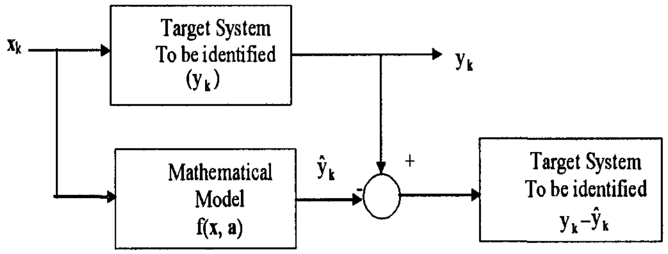


Figure 4.7 Parameter identification method.

Figure 4.7 is a block diagram of parameter identification, where an input $\mathbf{x}_k$ is applied to both system and model, and the difference between the target system's output and the model's output is used in an appropriate manner to update a parameter vector **a** to reduce this difference. Note that the dataset composed of N desired input-output pairs ($\mathbf{x}_k$, $\mathbf{y}_k$), k = 1, 2, …., N is called the training dataset.

Least squares methods are powerful and well-developed mathematical tools useful in a variety of areas including statistics, adaptive control, and signal processing. Nowadays they are essential and indispensable tools for constructing linear mathematical models[14]. The same fundamental concepts can be extended to nonlinear models as well. Thus it can be said that linear least-squares methods provide the most basic and important mathematical foundation for solving modeling problems.

Linear models are linear in their parameters; but a linear model may be nonlinear in its inputs. By static (memory less) systems, we mean that the output of the target system depends on its current inputs only; it does not depend on the history of inputs. The output of a dynamic system can be treated as a static mapping of its current inputs and several previous states, assuming they are available.

4.7.1 Linear regression and classification

Prediction methods[14][16] fall into two categories of statistical problems: classification and regression. For classification the predicted output is a discrete number, a class, and performance is typically measured in terms of error rates. For regression, the predicted output is a continuous variable, and performance is typically measured in terms of distance, for example, absolute distance or mean square error. The main difference is that regression values have a natural ordering, whereas class values are unordered in the classification method. This affects the measurement of error. For classification, predicting the wrong class is an error no matter which class is predicted. For regression, the error in prediction varies depending on the distance from the correct value.

In linear regression for horizontally partitioned data, where participating data sites have databases that contain the same numerical attributes for the same sets of data patterns. We call them participating data sites even though in some settings they might be corporations or other data holders. In vertically partitioned[17] data, the database holds different attributes for the same set of data subjects—for example, one has employment information, another has health data, and a third has information about education. Regression methods are well suited for prediction problems.

In the statistics literature, regression papers predominate, whereas in the machine-learning literature, classification plays a dominant role. For classification, it is not unusual to apply a regression method, such as neural nets trained by minimizing squared error distance for zero or one outputs. In that restricted sense, classification problems might be considered a subset of regression methods.

4.7.2 Rule-based methods

A static or dynamic system that makes use of fuzzy sets or fuzzy logic and the corresponding mathematical framework is called a fuzzy system[25][7]. There are a number of ways fuzzy sets can be involved in a system. Fuzzy systems defined by means of "if-then" rules are known as rule-based fuzzy systems. Fuzzy systems can serve different purposes, such as modeling, data analysis, prediction, or control. A fuzzy rule-based system is called a fuzzy model for simplicity, regardless of its eventual purpose. Fuzzy models can be seen as logical models which use "if–then" rules and logical operators to establish qualitative relationships among the variables in the model. Fuzzy sets serve as a smooth interface between qualitative variables involved in the rules and numerical domains of the inputs and outputs of the model[19][20][21].

Fuzzy rules can be defined in many different ways[4]. In general, a fuzzy conditional rule is made up of a premise and a conclusion. It can be written in the following form:

IF premise THEN conclusion,

where premise is a complex fuzzy expression, i.e., a logic expression that uses fuzzy logic operators and atomic fuzzy expressions, and the conclusion is an atomic expression. The truth values of a fuzzy rule are given by the truth value of its premise part. Therefore, for a given object x, and fuzzy rule R:

$$\text{Truth value}(R, x) = \text{Truth value}(\text{premise}R, x). \tag{16}$$

For rule-based models the topology of the model is based upon fuzzy sets; in input and output variables we require well-developed interfaces[19]. The generic models in this category are formulated as follows:

Rule-based fuzzy models exploit the calculus of rule-based structures and, in general, can be structured as a series of "IF-THEN" conditional statements of the form:

$$\text{IF } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ and } \ldots \text{ and } x_n \text{ is } A_{in} \text{ THEN } y \text{ is } B_i, \tag{17}$$

where $i = 1, 2, \ldots, c$, $x_1, x_2, \ldots, x_n$ are input variables, and y is an output variable. A_{i1}, A_{i2}, …, A_{in} and B are fuzzy sets (linguistic labels) of corresponding systems variables being defined. Any logic processing carried out by the rule-based inference mechanism requires that any input be transformed to its numeric format.

Rule-based models endowed with local regression models forming their conclusions are commonly referred to as Takagi-Sugeno fuzzy models[19]; they are regulated by the following formula:

$$\text{IF } x_1 \text{ is } A_{i1} \text{ and } x_2 \text{ is } A_{i2} \text{ and } \ldots \text{ and } x_n \text{ is } A_{in} \text{ THEN } y \text{ is } L_i(\mathbf{x}, \mathbf{a}_i), \tag{18}$$

where $L_i(\mathbf{x}, \mathbf{a}_i)$ represents a local regression model equipped with some vector of parameters $\mathbf{a}_i$. In particular, one can envision a linear form of the model in which the local model becomes a linear function of its parameters such that $L_i(\mathbf{x}, \mathbf{a}_i) = a_i^T x$. Obviously, depending upon the specificity of the problem and the structure of available data, regression models could be made nonlinear[19][31].

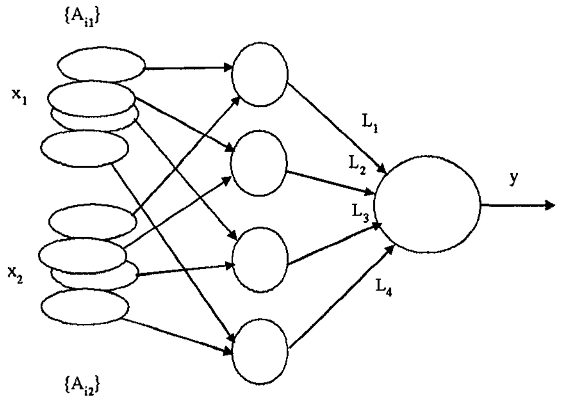


Figure 4.8 Takagi-Sugeno model with local regression models; the connections of the output unit realize processing through the local regression models (L_i).

4.7.3 Radial-basis function networks (RBFN)

RBFN have increasingly been used in many practical areas such as control, signal processing, pattern recognition, and time series prediction. RBFN are rooted in the interpolation of a high dimensional space to solve the curve-fitting problem[17]. According to this viewpoint, learning is equivalent to finding a surface in a multidimensional space that provides a best fit to the training data, with the criterion for "best fit" being measured in some statistical sense. In the context of a neural network, the hidden units provide a set of "functions" that constitute an arbitrary "basis" for the input patterns (vectors) when they are expanded into the hidden-unit space; these functions are called *radial-basis functions*. Radial-basis functions were first introduced in the solution of the real multivariate interpolation problem. The early work on this subject is surveyed by Powell[23]. It is now one of the main fields of research in numerical analysis.

The construction of a radial-basis function network in its most basic form involves three entirely different layers. The input layer is made up of source nodes (sensory units). The second layer is a hidden layer of high enough dimension. The output layer supplies the response of the network to the activation patterns applied to the input layer. The transformation from the input space to the hidden-unit space is nonlinear, whereas the transformation from the hidden-unit space to output space is linear. A justification of this rationale may be traced to Cover[6]. A pattern classification problem cast in a high dimensional space nonlinearly is more likely to be linearly separable than one cast in a low dimensional space—hence the reason for making the hidden dimensional space high in an RBFN.

Computational methods

Radial-basis function networks (RBFN)[20][27] are structures that use locally tuned and overlapping receptive field units (neurons) to perform function mappings and approximations. Figure 4.9 shows an RBFN with three neurons in a hidden layer. The activation level of the i-th neuron is expressed as:

$$R_i = H_i(\mathbf{x}) = H_i(\|\mathbf{x} - \mathbf{v}_i\| / \sigma_i),\ i = 1, 2, \ldots, C, \tag{19}$$

where $\mathbf{x}$ is a multidimensional input vector, $\mathbf{v}_i$ is a vector with the same dimension as $\mathbf{x}$, C is the number of radial basis functions (neurons), and $H_i(\cdot)$ is the i-th radial basis function, also called the kernel function. Typically, $H_i(\cdot)$ is a Gaussian function of the form:

$$H_i(x) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{v}_i\|^2}{2\sigma_i^{\ 2}}\right), \tag{20}$$

where σ_i^2 is a normalization parameter that represents a measure of the spread of data associated with each neuron. The activation level of a radial basis function R_i computed by the i-th hidden neuron is the largest when the input vector $\mathbf{x}$ is at the center $\mathbf{v}_i$ of that neuron, and it decreases as the distance between the two vectors increases.

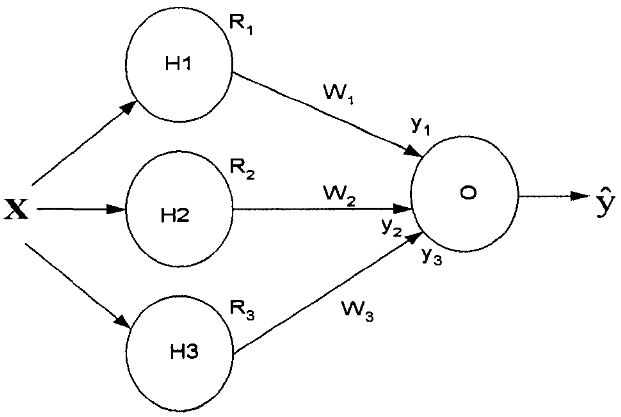


Figure 4.9 Radial basis function neural network model.

The output of RFBN can be computed as the weighted sum of the output values generated by each hidden neuron:

$$\hat{y}_k = \sum_{i=1}^{C} y_i = \sum_{i=1}^{C} R_i w_i\,. \tag{21}$$

To identify the receptive field parameters $\mathbf{v}_i$ and σ_i^2, and weights w_i in the output layer of each function, a two stage training process is typically needed. Accordingly, to determine parameters $\mathbf{v}_i$ and σ_i^2 of each neuron in a hidden layer, we use the

FCM[17][16] clustering technique in the first stage. To obtain weights w_i for a single neuron in an output layer, the gradient method was employed in the second stage. Because parameters $\mathbf{v}_i$ and σ_i^2 of the kernel functions are fixed during the second stage, the linear weight values w_i can be trained very efficiently.

4.8 Conclusions

This chapter gives an overview of fuzzy sets and their generalized form type-2 fuzzy sets. Clustering and different clustering models are discussed. Algorithms directly applicable to the implementation and realization of multiphase collaborative clustering [16] and basics of linear regression and classification models [26] are described. A fuzzy rule-based and radial-basis function networks models are also introduced.

References

1. J. Allan, A. Feng, A. Bolivar, Flexible Intrinsic Evaluation of Hierarchical Clustering for TDT, *ACM 12th International Conference Information and Knowledge Management (CIKM'03)*, 2003, 263-270.

2. M. Anderberg, Custer analysis for Applications, *Academic Press*, 1973.

3. A. Bargiela, W. Pedrycz, *Granular Computing, An Introduction*, Kluwer Academic Publishers, 2003.

4. J. C. Bezdek, Pattern Recognition With Fuzzy Objective Function Algorithms, *PlenumPress*, 1981.

5. D. S. Broomhead and Lowe D, Multivariable functional interpolation and adaptive networks, *Complex Sys.*, 2, 1988, 321-355.

6. T. M. Cover, Geometrical and statistical properties of systems of linear inequilities with applications in pattern recognition, *IEEE Transactions on Electronic Computers*, 14, 1965, 326-334.

7. M. Delgado, A. F. Gomez-Skarmets, F. Martin, A Methodology to Model Fuzzy Systems using Fuzzy Clustering in a Rapid Prototyping Approach, *IEEE Transactions on Fuzzy Sets and Systems*, 97, 1998, 287-302.

8. R. O. Duda, P. E. Hart, D. G. Stroke, *Pattern Classification*, John Wiley, 2001.

9. A. Jain, R. Duin, J. Mao, Statistical Pattern Recognition: Review, *IEEE Transactions on Pattern Analysis and Machine Intelligence*,22 ,2000, 4-37.

10. A. Jain, M. Murt, P. Flynn, Data Clustering: A Review, *ACM Computing Surveys*, 31, 1999, 264-323.

11. B. Kosko, Fuzzy Systems as Universal Approximators, *IEEE Transactions on Computers*, 43, 1994, 1329–1333.

12. Y. Leung, J. Zhang, Z. Xu, Clustering by Space-Space Filtering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 2000, 1396-1410.

13. V. Loia, W. Pedrycz, S. Senatore, P-FCM: a proximity-based fuzzy clustering for user-centered web applications, International Journal of Approximate Reasoning, 34, 2003, 121-144.

14. D. C. Montgomery, *Design and Analysis of Experiments*, Wiley, 2005.

15. W. Pedrycz, Fuzzy control and fuzzy systems, 2nd extended edition, *Research Studies Press, New York, Wiley*, 1993.

16. W. Pedrycz, Collaborative Fuzzy Clustering, *Pattern Recognition Letters*, 23, 2002, 675-1686.

17. W. Pedrycz, *Knowledge-Based Clustering: From Data to Information Granules*, John Wiley, 2005.

18. W. Pedrycz, F. Gomide, An Introduction to Fuzzy Sets, Analysis and Design, *MIT Press*, 1998.

19. W. Pedrycz , F. Gomide, *Fuzzy Systems Engineering, Towards Human-Centric Computing*, IEEE Press, 2007

20. W. Pedrycz, V.O. Jose, Optimization of Fuzzy Models, *IEEE Trans. On Systems Man and Cybernetics-Part B: Cybernetics*, 26, 1996, 627-636.

21. W. Pedrycz, M. Reformat, Evolutionary Fuzzy Modeling, *IEEE Transactions on Fuzzy Systems*, 11, 2003, 652-665.

22. W. Pedrycz, G. Vukovich, Clustering in the Framework of Collaborative Agent, *Proceedings IEEE, International Conference on Fuzzy Systems,*12-17 May 2002,1, 2002 134-138.

23. M. J. D. Powel, Radial basis functions for multivariable interpolation: A review, *In IMA Conference on Algorithms for the Approximation of Functions and Data*, 1985, 143-167.

24. D. B. Skillicorn, S. M. McConnell, Distributed prediction from vertically partitioned data, *Journal of Parallel and Distributed computing*, 2007.

25. P. Sneath, R. Sokal, *Numerical Taxonomy: The principles and practice of numerical classification*, Freeman, 1973.

26. A. Strehl, J. Ghosh, Cluster Ensembles: A Knowledge Reuse Framework for Combining Multiple Partitions, *Journal of Machine Learning Research*, 3, 2002, 583-617.

27. L. X. Wang, Fuzzy Systems are Universal Approximators, *Proc. IEEE Int. Conf. on Fuzzy Systems,* 1992, San Diego,USA,1992 , 1163–1170.

28. L. A. Zadeh, Fuzzy sets, Inf. Control, 8, 1965, 338–353.

29. L. A. Zadeh, The concept of a linguistic variable and its applications to approximate reasoning -1, *Information Sciences*, 8, 1975, 199-249.

30. P. Zezula, G. Amato, V. Dohnal, M. Batko, *Similarity Search: The Metric Space Approach*, Springer, 2006.

31. X. J. Zeng, M. G. Singh, Approximation Theory of Fuzzy Systems – MIMO Case, *IEEE Transactions on Fuzzy System*, 3, 1995, 219–235.

Chapter 5

Vertical Fuzzy Collaborative Clustering

In this chapter the framework of vertical collaborative fuzzy clustering is introduced—a conceptual and algorithmic machinery for the collective discovery of a common structure (relationships) within a finite family of data residing at individual datasites. The proposed optimization environment has two fundamental features. First, given existing constraints which prevent individual sites from exchanging detailed numeric data, any communication has to be realized at the level of information granules. The specificity of these granules impacts the effectiveness of ensuing collaborative activities. Second, the fuzzy clustering realized at the level of the individual datasite has to constructively consider the findings communicated by other sites and act upon them while running the optimization confined to the particular datasite.

Experimental studies presented include some synthetic data, selected datasets from the Machine Learning Repository, and weather data from Environment Canada.

5.1 Introduction

An important extension of clustering is the combination of several clustering results. Different outcomes of clustering can result from several runs of the same clustering algorithm, several clustering methods being applied to the same dataset, or the use of clustering for several datasets. These developments come under the rubric of multicluster combinations [1], bagging [4], collective clustering pursuits [9][10][11][16][13], and cluster ensembles [22][23][24][25]. The collaborative aspects of fuzzy clustering [2][6][7][8][10][12][22][23] were originally studied in [17][20]. There are a number of highly motivating factors that stand behind the developments occurring in this realm of investigation. In parallel to what is very much visible in supervised learning and pattern classifiers, we investigate ways of improving the quality of unsupervised learning, and clustering in particular. We encounter scenarios in which data are generated in a distributed manner and have to be handled separately (we are not allowed to treat the data together given existing restrictions of security, privacy [14] or for other technical reasons). In this case, the clustering results produced at the individual datasites have to be reconciled.

An important feature of collaborative fuzzy clustering is the level of communication realized at the granular level. Information granules [3][19][21][27] play an active role in the development (clustering) of structures at the local level of individual datasites.

Collaborative clustering [17] activities involve discovering and sharing knowledge. To derive global relationships common to several databases, we must allow the databases to collaborate at the level of patterns. We commonly are not permitted to have access to all databases but eventually could be provided with some abstract information such as mean, median, or synthetic indexes describing the data. Two modes of collaborative clustering, horizontal and vertical [23][21], are implemented to deal with such abstract information. Horizontal clustering mode is covered in Chapter 6. This Chapter focuses on vertical clustering mode.

The vertical mode of collaboration clustering is concerned with a collection of databases involving different patterns defined in the same feature space, where sites interact using prototypes, as shown in Figure 5.1.

The algorithmic issue of collaboration dwells on the well-known fuzzy c-means (FCM) [17]. Generally we think of clustering [26][27] as a vehicle for forming information granules. As in the FCM, in collaboration clustering, we also require that the partition matrix satisfies standard requirements of membership grades summing to 1 unit intervals.

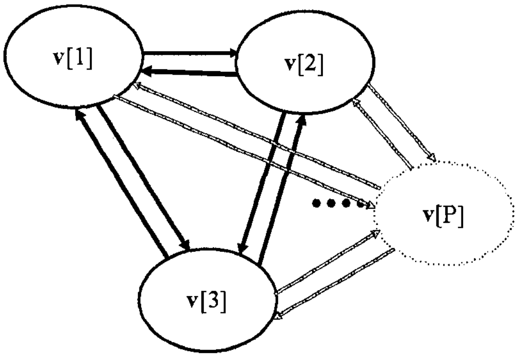


Figure 5.1 Vertical collaboration between databases at a local level; in each database objects are located in the same data space but have different patterns.

Clustering algorithms interact by exchanging partition matrices. In this way communication links are established at the level of information granules instead of at the data level; technical details can be found in [1][23]. The collaborative clustering presented in this thesis may be regarded as a special algorithmic model of knowledge reuse and knowledge integration.

We encounter situations where data is inherently distributed and can be accessed and processed only locally. Sharing the data is not feasible considering existing constraints of a technical or regulatory nature. At the same time, it becomes evident that when searching for a structure it would be highly desirable to establish some interaction when running data analysis at individual datasets (datasites or databases) and to allow all processes to engage in communication and reconciliation of the findings. It is anticipated that knowledge of these relationships will help in the discovery of relationships and in making these dependencies more stable and general. Several points are important in the translation of these preliminary observations:

- As the data cannot be transferred between datasites and therefore cannot be processed explicitly, a viable alternative is to communicate the local findings at a higher level of abstraction such as information granules.
- Reconciliation of findings between datasites needs to benefit all parties involved. In other words, our actions have to enhance the quality of the results at the global level of all datasites
- When reconciling the results of data analysis between individual datasites, the findings must be driven by the local data. The results of communication should be supportive but they should not supersede the search for structure implied by the local data.
- It is desirable to quantify the results of collaboration in the language of information granules. This is demonstrated in the study.

This procedure is called *collaborative* clustering as collaboration is crucial to the globally developed process of searching for structure in data.

To realize the above requirements, we consider fuzzy clustering, and fuzzy c-means as a vehicle of granulation of information cf. [2][10][18] for several compelling reasons. Fuzzy clustering is commonly used and is associated with a wealth of algorithmic developments and experimental evidence. The formation of clusters at each datasite is driven by minimization of an augmented objective function which takes into account the data being available locally and involves a term expressing difference between the local structure and data produced at other sites. Information granules in the form of type-2 fuzzy sets are used to quantify the results of clustering in terms of the consistency and diversity being recognized across datasites.

The concept of collaborative fuzzy clustering and its realization was introduced in [17]. This thesis adds a number of novel and essential components to the work in [17]. First, the effect of collaboration is quantified and an effective way of determining an optimal level of collaboration is provided. Second, this study brings forward an interesting and practically relevant generalization in which different levels of granularity (number of clusters) are allowed at local datasets. Third, when we are concerned with numeric results, findings are expressed as information granules of higher order, for instance as type-2 fuzzy sets. Aggregation of this type is governed by the principle of justifiable granularity, cf. [14].

The chapter begins by formulating the problem which is to use collaborative clustering to seek structure in data (Section 5.2). In section 5.3, we elaborate on a general optimization scheme of collaborative clustering. Sections 5.4–5.6 focus on the optimization process; ways of producing interaction between datasites using prototypes and induced partition matrices are discussed, a form of the optimization problem (objective function) is presented, and a mechanism to evaluate the quality of collaboration is proposed. The emergence of granular prototypes becomes a result of collaboration. The quantification of collaboration is discussed in section 5.7. Section 5.8 discusses a significant generalization in which different numbers of clusters at individual datasites is considered. Numerical experiments are reported in section 5.9. Concluding observations are offered in section 5.10. Detailed derivations are covered in the text for the case of collaborative fuzzy clustering realized in the presence of the same and different number of clusters.

5.2 Problem statement

Let us consider P datasites, D[1], D[2], ..., D[P] consisting of N[1], N[2], ..., N[P] patterns (data) defined in the same feature space **X**. At each datasite we are interested in revealing a structure by forming c clusters; we assume in the first case same number of information granules (clusters) for each datasite. While for a particular datasite we are interested in the structure at this local level, we want to take into account the results of clustering reported at other datasites.

In fuzzy clustering, and fuzzy c-means in particular, there are two fundamental facets of granularity: one is conveyed by prototypes while the other is captured by partition matrices. Given the prototypes, we can produce partition matrices. Conversely, given partition matrices, we can develop the corresponding prototypes. Either of these manifestations of granular information can be used as a communication vehicle, depending on the problem at hand. Here, the datasets at the datasites all reside in the same feature space, so communication can be realized by exchanging the prototypes produced at each datasite.

The problem of collaborative clustering can be briefly defined as follows:

> Given a finite number of disjoint datasites with patterns defined in the same feature space, develop a scheme of collective development and reconciliation of a fundamental cluster structure across the sites that is based upon exchange and communication of local findings where the communication needs to be realized at some level of information *granularity*. The development of the structures at the local level exploits the communicated findings in an *active* manner through minimization of the corresponding objective function augmented by the structural findings developed outside the individual datasite. We also allow for retention of key individual (specific) findings that are essential (unique) to the corresponding datasite.

The essence of collaborative clustering is presented in Figure 5.2.

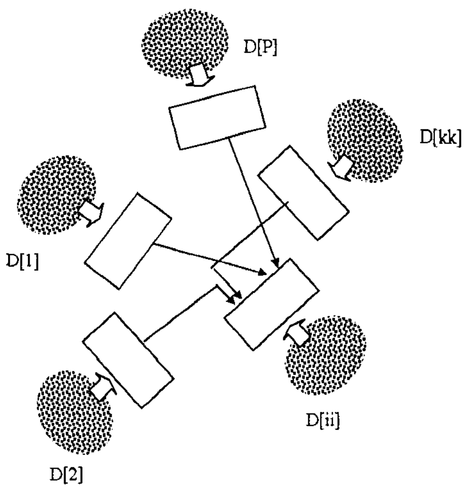


Figure 5.2 The essence of collaborative clustering. The goal is to build a global characterization of the data by striking a balance between local findings (produced at the level of locally available data) and findings coming from other datasites (sensors). The arrows show communication links between datasite D[ii] and all other datasites.

Alluding to Figure 5.2, we can offer another important and visible category of application which deals with wireless sensor networks. In such networks, we envision a collection of randomly scattered sensors whose communication is established on an ad hoc basis. Each node (sensor) collects the data available in its neighbourhood and realizes its processing. This leads to a determination of the *local* characteristics of the data (formulated as a collection of clusters being observed at a particular local level of the given sensor). At the same time it is recognized that the local processing could benefit from collective activities between sensors. This need for a *global* and collective style of processing is motivated by the limited amount of data available locally and the need to establish a global view of the data collected from the overall network. Each sensor formulates a very limited and localized perception of the environment that has to be augmented by local findings formed by other sensors.

There are essential differences between the proposed approach and the concepts which have been encountered in the literature under the umbrella of distributed clustering, cf. [3][14][15][26]. In distributed clustering it is assumed that the clusters are the same across all datasites. In particular, it is assumed that at each datasite there are exactly the same clusters being modeled by Gaussian distributions $N(\mathbf{m}_i, \Sigma_i)$ described by mean vectors $\mathbf{m}_i$ and covariance matrices Σ_i and put together in the form of a linear combination, cf. [15]. More specifically, we encounter a relationship of the form $\sum_{i=1}^{c} \lambda_{ji} N(\mathbf{m}_i, \Sigma_i)$, $j = 1, 2, \ldots, P$, where the values of the mixing parameters λ_{ji} are potentially unique for each datasite. In contrast, in this study no specific assumptions are being made. The only assumption here concerns the same granularity of the findings (viz. number of clusters at each datasite). As a result the structure at each datasite makes an attempt to reconcile differences and retains and quantifies those of particular relevance to the given datasite. The findings are expressed in fuzzy sets of the prototypes or in the case of membership degrees in type-2 fuzzy sets.

Concepts of cluster ensemble found in the literature are based on concepts different from what is proposed in this thesis. Cluster ensemble methods differ in the way the generic clustering procedure is developed and the way in which results are combined [25]. Topchy et. al. [25] proposed a consensus function based on informative-theoretic principles and generalized mutual information. A different consensus function was developed in [5] based on a voting/merging method which provides a pairwise iterative scheme of combination. Strehl and Ghosh [24] proposed three different ensemble clustering models based on a consensus method, all of which use hypergraph operations to construct solutions.

Here, we follow a standard notation encountered in pattern recognition. The patterns (data) are treated as vectors in $\mathbf{X} \subset \mathbf{R}^n$ and the distance between two elements in this space $\|.\|$ is realized as a weighted Euclidean distance. The standard FCM is used as a clustering vehicle, although any objective function-based clustering could be a sound alternative. All results produced at a given site are clearly identified by the index of this site. Thus for the ii-th datasite, the partition matrix is denoted by $U[ii] = [u_{ik}[ii]]$, $i = 1, 2, \ldots, c$; $k = 1, 2, \ldots, N[ii]$, while the corresponding prototypes are given as $\mathbf{v}_1[ii]$, $\mathbf{v}_2[ii], \ldots, \mathbf{v}_c[ii]$.

Optimization details of vertical clustering with same level of granularity

Here we present pertinent derivations of the collaborative scheme. D[1], D[2], …, D[P] denote the datasites involved in the collaboration. The objective function guiding the formation of the clusters at the ii-th datasite comes as an augmented version of the one being used in the "standard" FCM, that is:

$$Q[ii] = \sum_{k=1}^{N[ii]} \sum_{i=1}^{c} u_{ik}^2[ii] d_{ik}^2 + \beta \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} \sum_{k=1}^{N[ii]} \sum_{i=1}^{c} (u_{ik}[ii] - u_{ik}^{\sim}[ii \mid jj])^2 d_{ik}^2 . \quad (1)$$

The minimization of Q[ii] is carried out with respect to the fuzzy partition U[ii] and the prototypes $\mathbf{v}_i[ii]$. The distance d_{ik} concerns the k-th data (pattern) in D[ii] and the i-th prototype $d_{ik}^2 = \| \mathbf{x}_k - \mathbf{v}_i \|^2$. $U^{\sim}[ii \mid jj]$ stands for a partition matrix induced by prototypes obtained for the datasite D[jj] which is being presented at the ii-th datasite. The partition matrix belongs to the family of matrices satisfying the conditions:

$$U = \{u_{ik} \in [0,1] \mid \sum_{i=1}^{c} u_{ik}[ii] = 1, \forall k, \text{and } 0 < \sum_{k=1}^{N[ii]} u_{ik}[ii] < N[ii], \forall i\} . \quad (2)$$

Given the standard identity constraint imposed on the partition matrix, viz. $\sum_{i=1}^{c} u_{ik}[ii] = 1$, in the optimization of (1) we confine ourselves to the use of Lagrange multipliers. For any data point k, k = 1, 2, …, N[ii], we reformulate the objective function to be in the form:

$$V[ii] = \sum_{i=1}^{c} u_{ik}^{2}[ii]d_{ik}^{2} + \beta\sum_{\substack{jj=1 \\ jj\neq ii}}^{P}\sum_{i=1}^{c}(u_{ik}[ii] - u_{ik}^{\sim}[ii \mid jj])^{2}d_{ik}^{2} - \lambda(\sum_{i=1}^{c} u_{ik}[ii] - 1). \tag{3}$$

The necessary conditions for the minimum of V[ii] are expressed as:

$$\frac{\partial V[ii]}{\partial u_{rs}} = 0, \quad \frac{\partial V[ii]}{\partial \lambda} = 0. \tag{4}$$

After computing the derivative with respect to the elements of the partition matrix we obtain:

$$\frac{\partial V}{\partial u_{rs}} = 2u_{rs}[ii]d_{rs}^{2} + 2\beta\sum_{\substack{jj=1 \\ jj\neq i}}^{P}(u_{rs}[ii] \text{-} u_{rs}^{\sim}[ii \mid jj]d_{rs}^{2}) - \lambda = 0, \tag{5}$$

where r = 1, 2, …, c, s = 1, 2, …, N[ii]. The detailed calculations are shown below:

$$2u_{rs}[ii]d_{rs}^{2} + 2\beta(P\,\square)u_{rs}[ii]d_{rs}^{2} - 2\beta d_{rs}^{2}\sum_{\substack{jj=1 \\ jj\neq ii}}^{P} u_{rs}^{\sim}[ii \mid jj] - \lambda = 0,$$

$$u_{rs}[ii]\left(2d_{rs}^{2} + 2\beta(P-1)d_{rs}^{2}\right) = \lambda + 2\beta d_{rs}^{2}\sum_{\substack{ii=1 \\ jj\neq ii}}^{P} u_{rs}^{\sim}[ii \mid jj],$$

$$u_{rs}[ii] = \frac{\lambda + 2\beta d_{rs}^{2}\sum_{\substack{jj=1 \\ jj\neq ii}}^{P} u_{rs}^{\sim}[ii \mid jj]}{2d_{rs}^{2}[1 + \beta(P\text{-}1)]}.$$

Finally,

$$u_{rs}[ii] = \frac{\lambda + 2\beta d_{rs}^{2}\sum_{\substack{jj=1 \\ jj\neq ii}}^{P} u_{rs}^{\sim}[ii \mid jj]}{2d_{rs}^{2}[1 + \beta(P-1)]} = \frac{\lambda}{2d_{rs}^{2}[1 + \beta(P-1)]} + \beta\sum_{\substack{jj=1 \\ jj\neq ii}}^{P} u_{rs}^{\sim}[ii \mid jj] \times \frac{1}{1 + \beta(P-1)}. \tag{6}$$

Given the constraint of the form $\sum\limits_{j=1}^{c} u_{js}[ii] = 1$, we obtain:

$$\sum_{j=1}^{c} \frac{\lambda + 2\beta d_{js}^{2} \sum\limits_{\substack{jj=1 \\ jj \neq ii}}^{P} u_{js}^{\sim}[ii \mid jj]}{2d_{js}^{2}[1+\beta(P-1)]} = 1,$$

$$\sum_{j=1}^{c} \frac{\lambda}{2d_{js}^{2}[1+\beta(P-1)]} + \sum_{j=1}^{c} \frac{2\beta d_{js}^{2} \sum\limits_{\substack{jj=1 \\ jj \neq ii}}^{P} u_{js}^{\sim}[ii \mid jj]}{2d_{js}^{2}[1+\beta(P-1)]} = 1. \tag{7}$$

In the sequel,

$$\lambda \sum_{j=1}^{c} \frac{1}{2d_{js}^{2}[1+\beta(P-1)]} = 1 - \sum_{j=1}^{c} \frac{\beta \sum\limits_{\substack{jj=1 \\ jj \neq ii}}^{P} u_{js}^{\sim}[ii \mid jj]}{[1+\beta(P-1)]},$$

$$\lambda = \sum_{j=1}^{c} 2d_{js}^{2}[1+\beta(P-1)] \left(1 - \sum_{j=1}^{c} \frac{\beta \sum\limits_{\substack{jj=1 \\ jj \neq ii}}^{P} u_{js}^{\sim}[ii \mid jj]}{[1+\beta(P-1)]} \right),$$

$$\lambda = \frac{1 - \sum\limits_{j=1}^{c} \dfrac{\beta \sum\limits_{\substack{jj=1 \\ jj \neq ii}}^{P} u_{js}^{\sim}[ii \mid jj]}{[1+\beta(P-1)]}}{\dfrac{1}{2\sum\limits_{j=1}^{c} d_{js}^{2}[1+\beta(P-1)]}}. \tag{8}$$

Hence,

$$\lambda = 2\frac{1-\frac{\beta\sum\limits_{j=1}^{c}\sum\limits_{\substack{jj=1\\ jj\neq ii}}^{P} u_{js}^{\sim}[ii \mid jj]}{[1+\beta(P-1)]}}{\sum\limits_{j=1}^{c}\frac{1}{d_{js}^{2}}}\langle 1+\beta(P-1)\rangle . \tag{9}$$

Plugging (9) into (7) gives:

$$u_{rs}[ii] = \frac{2\frac{1-\frac{\beta\sum\limits_{j=1}^{c}\sum\limits_{\substack{jj=1\\ jj\neq ii}}^{P} u_{js}^{\sim}[ii \mid jj]}{[1+\beta(P-1)]}}{\sum\limits_{j=1}^{c}\frac{1}{d_{js}^{2}}}[1+\beta(P-1)]}{2d_{rs}^{2}[1+\beta(P-1)]} + \frac{2\beta d_{rs}^{2}\sum\limits_{\substack{jj=1\\ jj\neq ii}}^{P} u_{rs}^{\sim}[ii \mid jj]}{2d_{rs}^{2}[1+\beta(P-1)]} . \tag{10}$$

Further simplifications lead to:

$$u_{rs}[ii] = \frac{1}{\sum\limits_{j=1}^{c}\frac{d_{rs}^{2}}{d_{js}^{2}}}\left[1-\sum_{j=1}^{c}\frac{\beta\sum\limits_{\substack{jj=1\\ jj\neq ii}}^{P} u_{js}^{\sim}[ii \mid jj]}{[1+\beta(P-1)]}\right] + \frac{\beta\sum\limits_{\substack{jj=1\\ jj\neq ii}}^{P} u_{rs}^{\sim}[ii \mid jj]}{[1+\beta(P-1)]} . \tag{11}$$

In order to optimize the objective function with regard to the prototypes, the Euclidean distance (its weighted version is handled in the same manner) is considered. Given the form of the distance, the objective function is written as:

$$Q[ii] = \sum_{k=1}^{N[ii]}\sum_{i=1}^{c} u_{ik}^{2}[ii]\sum_{j=1}^{n}(x_{kj}-v_{ij}[ii])^{2} + \beta\sum_{\substack{jj=1\\ jj\neq ii}}^{P}\sum_{k=1}^{N[ii]}\sum_{i=1}^{c}(u_{ik}[ii]-u_{ik}^{\sim}[ii \mid jj])^{2}\sum_{j=1}^{n}(x_{kj}-v_{ij}[ii])^{2} . \tag{12}$$

The necessary condition leading to the minimization of Q[ii] comes in the form:

$$\frac{\partial Q[ii]}{\partial v_{rt}[ii]} = 0 .$$

Then we obtain:

$$\begin{aligned}\frac{\partial Q[ii]}{\partial v_{rt}[ii]} &= -2\sum_{k=1}^{N[ii]} u_{rk}^{2}[ii](x_{kt} - v_{rt}[ii]) \\ &\quad - 2\beta\sum_{\substack{ii=1 \\ jj\neq ii}}^{P}\sum_{k=1}^{N[ii]}(u_{rk}[ii] - \tilde{u}_{rk}[ii \mid jj])^{2}\,(x_{kt} - v_{rt}[ii]) = 0\end{aligned} \tag{13}$$

After further simplifications,

$$\begin{aligned} v_{rt}[ii]&\left\{\sum_{k=1}^{N[ii]} u_{rk}^{2}[ii] + \beta\sum_{\substack{jj=1 \\ jj\neq ii}}^{P}\sum_{k=1}^{N[ii]}(u_{rk}[ii] - \tilde{u}_{rk}[ii \mid jj])^{2}\right\} . \\ &= \sum_{k=1}^{N[ii]} u_{rk}^{2}[ii]x_{kt} + \beta\sum_{\substack{jj=1 \\ jj\neq ii}}^{P}\sum_{k=1}^{N[ii]}(u_{rk}[ii] - \tilde{u}_{rk}[ii \mid jj])^{2}x_{kt}\end{aligned}$$

Finally,

$$v_{rt}[ii] = \frac{\displaystyle\sum_{k=1}^{N[ii]} u_{rk}^{2}[ii]x_{kt} + \beta\sum_{\substack{jj=1 \\ jj\neq ii}}^{P}\sum_{k=1}^{N[ii]}(u_{rk}[ii] - \tilde{u}_{rk}[ii \mid jj])^{2}x_{kt}}{\displaystyle\sum_{k=1}^{N[ii]} u_{rk}^{2}[ii] + \beta\sum_{\substack{jj=1 \\ jj\neq ii}}^{P}\sum_{k=1}^{N[ii]}(u_{rk}[ii] - \tilde{u}_{rk}[ii \mid jj])^{2}} , \tag{14}$$

where r = 1, 2, ..., c, t = 1, 2, ..., n.

In the next section it is explained how optimization activities are completed at individual datasites and how communication is realized in terms of the granular constructs of the prototypes.

5.3 General flow of collaborative processing

Collaborative clustering aims to develop structures at individual datasites using information obtained from other datasites. Two phases occur in a fixed sequence: an optimization of the structures at the individual sites and an interaction between sites when findings are exchanged. This process is depicted in Figure 5.3.

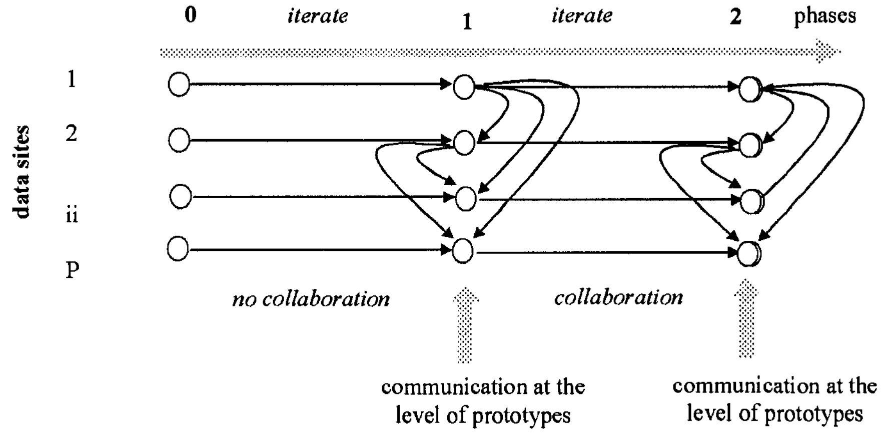

0
iterate
1
iterate
2
phases
data sites
1
2
ii
P
no collaboration
collaboration
communication at the level of prototypes
communication at the level of prototypes

Figure 5.3 A functional view of the processing realized in collaborative clustering.

Initially, the FCM algorithm is run independently at each datasite (this happens without any communication). After the FCM has terminated at each site, processing stops and the datasites communicate their findings. This communication needs to be realized at some level of information granularity. The effectiveness of the interaction depends on the way in which one datasite "talks" to others in terms of what has been discovered so far. Once communication has been established and the nodes are informed about structural findings at other sites, each site proceeds with optimization by focusing on the local data while at the same time taking into consideration the findings communicated by other datasites. Optimization for each site is run independently. Once all sites have declared termination of computing, they are ready to engage in the communication phase. Again, they communicate the findings and set up new conditions for the next phase of FCM optimization. Optimization and communication comprise the collaboration phase. The overall collaboration takes a finite number of collaboration phases, terminating when no further significant change in the revealed structure is reported.

Two criteria must be satisfied for a successful collaboration. First, we have to specify a way of communicating and representing findings at some level of granularity (recall that we are not allowed to communicate at the level of individual data but have to establish communication at the higher level of abstraction by engaging the exchange of the granular constructs). Second, we have to create an augmented objective function whose minimization embraces both the structures at the local level of the individual datasites and reconciles them with the structures communicated by other datasites.

5.4 Induced partition matrices as a mechanism of granular communication

The structural findings at individual datasites come in the form of prototypes and partition matrices. These are the two possible communication mechanisms. Since different datasites have datasets of possibly different cardinalities, sharing knowledge about partition matrices is not helpful. The prototypes, on the other hand, form a viable alternative. Communicating a limited number of prototypes is attractive since no significant communication overhead is built in this manner. As the FCM optimization focuses on the partition matrices as one of its components to be adjusted, we introduce a concept of so-called *induced* partition matrices. Consider the ii-th datasite. The prototypes produced at the jj-th datasite $\mathbf{v}_1[jj]$, $\mathbf{v}_2[jj]$, …, $\mathbf{v}_c[jj]$ are communicated to the ii-th datasite. Given this collection of prototypes, we induce a partition matrix over the datasite D[ii] and denote it by $U^{\sim}[ii|jj]$, where the two indexes (ii and jj) emphasize datasites taking part in this interaction. Its entries are determined in the standard way encountered in FCM computing [2], that is:

$$u^{\sim}_{ik}[ii|jj]=\frac{1}{\sum_{j=1}^{c}\left(\frac{\|\mathbf{x}_k[ii]-\mathbf{v}_i[jj]\|}{\|\mathbf{x}_k[ii]-\mathbf{v}_j[jj]\|}\right)^2}, \tag{15}$$

where I = 1, 2, …, c; k = 1, 2, …, N[ii], and $\mathbf{x}_k \in D[ii]$. Refer also to Figure 5.4 which highlights the essence of this mechanism of the collaboration.

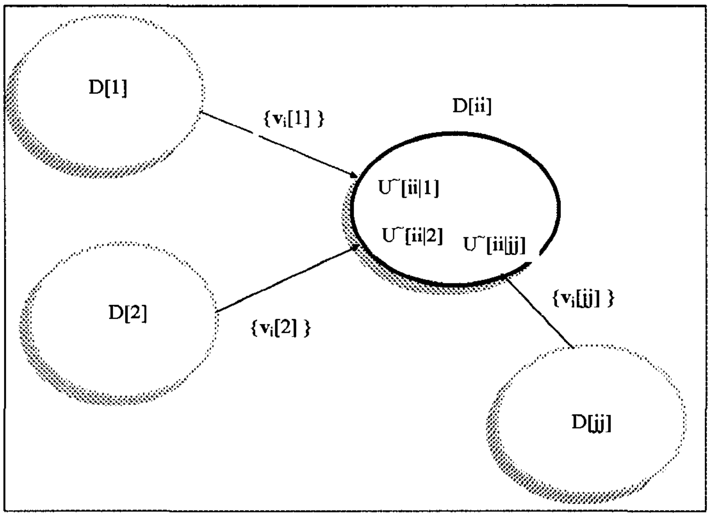


Figure 5.4 Datasites and communication realized through communication of prototypes and the consecutive generation of the induced partition matrices $U^{\sim}[ii|jj]$.

Proceeding similarly with all other datasites, D[1], …, D[ii-1], D[ii+1], …, D[P], we end up with P – 1 induced partition matrices, $U^{\sim}[ii|1]$, $U^{\sim}[ii|2]$, …, $U^{\sim}[ii|ii\text{-}1]$, $U^{\sim}[ii|ii+1]$, …,

$U^{\sim}[ii|P]$. The minimization of difference between U[ii] and $U^{\sim}[ii|jj]$ is used to establish collaborative activities between the datasites.

5.5 An augmented objective function

At the ii-th site, the clustering is guided by the augmented objective function assuming the following form:

$$Q[ii] = \sum_{k=1}^{N[ii]} \sum_{i=1}^{c} u_{ik}^2[ii] \|x_k - v_i\|^2 + \beta \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} \sum_{k=1}^{N[ii]} \sum_{i=1}^{c} (u_{ik}[ii] - u_{ik}^{\sim}[ii \mid jj])^2 d_{ik}^2 , \tag{16}$$

where β is a nonnegative number. The objective function Q[ii] consists of two components. The first one is nothing but a standard sum of weighted distances between the patterns in D[ii] and their prototypes $d_{ik}^2 = \| \mathbf{x}_k - \mathbf{v}_i[ii] \|^2$. In this sense, it is just the objective function encountered in the standard FCM being applied to D[ii] with the fuzzification coefficient m = 2. The second component reflects an impact coming from the structures formed at all remaining datasites. The distance between the optimized partition matrix and the induced partition matrices is to be minimized—this requirement is captured by this part of the objective function (16). The scaling coefficient β strikes a balance between the optimization guided by the structure in D[ii] and the already developed structures available at the remaining sites. The value of β implies a certain level of intensity of collaboration; the higher its value, the stronger the collaboration. For β = 0 no collaboration occurs and the problem is reduced to the collection of P independently run clustering tasks being confined to the corresponding datasites. The overall scheme of the collaborative clustering is covered in Table 5.1.

Table 5.1 The flow of collaborative clustering showing the main optimization phases and underlining the mechanism of communication in the form of exchange of prototypes obtained at each datasite.

Given: datasites D[1], D[2], ..., D[P].
Choose the number of clusters c to be looked for in the collaborative clustering; set up a termination criterion of the FCM and establish a level of collaboration (interaction) by choosing a nonnegative value of β.

Initial phase: Carry out clustering (FCM) for each datasite producing a collection of prototypes $\{\mathbf{v}_i[ii]\}$, i = 1, 2, ..., c for each datasite.

Collaboration
Iterate {successive phases of collaboration}

Communicate the results about the structure determined at each datasite.

```
    For each datasite (ii)
    {
    Minimize (2) at each datasite by iteratively proceeding with the iterative calculations of the partition matrix ($u_{rs}[ii]$) and the prototypes ($v_{rt}[ii]$), using respectively (11) and (14).

    $r = 1, 2, \ldots, c; t = 1, 2, \ldots, n; s = 1, 2, \ldots, N[ii]$
    } for datasite

until termination condition of the collaboration activities has been satisfied.
```

5.6 Evaluation of the quality of collaboration

Evaluation of the quality of results of collaboration between datasites requires careful assessment. The distance between partition matrices associated with each of the D[ii]'s could be computed and treated as a measure of the quality of the ongoing process. However, a direct comparison of two partition matrices is not feasible if there is no direct correspondence between rows (respective clusters). This is a well-known problem identified in the literature, cf. [13]. We use the concept of proximity and a proximity matrix induced by a given partition matrix to avoid this problem.

Consider the ii-th datasite with its partition matrix U[ii] and the induced partition matrices $U^{\sim}[ii|jj], jj = 1, 2, \ldots, ii-1, ii + 1, \ldots, P$. Structure revealed at the ii-th datasite is compared with structures at remaining sites by computing the following expression:

$$W[ii] = \frac{1}{(N^2[ii]/2)} \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} \left\| \mathrm{Prox}(U[ii]) - \mathrm{Prox}(U^{\sim}[ii|jj]) \right\| . \tag{17}$$

More specifically, the distance between the corresponding proximity matrices is realized in the form of the Hamming distance. In other words,

$$\| \mathrm{Prox}(U[ii]) - \mathrm{Prox}(U^{\sim}[ii \mid jj]) \| = \sum_{k_1=1}^{N[ii]} \sum_{k_2>k_1}^{N[ii]} | \mathrm{prox}(k_1, k_2)[ii] - \mathrm{prox}(k_1, k_2)^{\sim}[ii \mid jj] | , \tag{18}$$

where $\mathrm{prox}(k_1,k_2)[ii]$ denotes the (k_1, k_2) entry of the proximity matrix U[ii]. Similarly, $\mathrm{prox}(k_1,k_2)^{\sim}[ii|jj]$ is the corresponding (k_1, k_2) entry of the proximity matrix produced by the induced partition matrix $U^{\sim}[ii|jj]$. In a nutshell, rather than working at the level of comparing the individual partition matrices (which requires knowledge of the explicit correspondence between rows of the partition matrices), we generate their corresponding proximity matrices. This allows us to carry out comparison at this more abstract level. Summing up the values of W[ii] over all datasites, we arrive at the global level of consistency of the structure discovered collectively through the collaboration:

$$W = W[1] + W[2] + \ldots + W[P]. \tag{19}$$

The lower the value of W, the higher is the consistency between P structures. Likewise, the value of W being reported during successive phases of the collaboration can indicate the progress and quality of the collaborative process and serve as a suitable termination criterion (refer to Table 5.1), that is, one could stop the collaboration once no further change in the value of W is reported. Use of the above consistency measure is essential when gauging the intensity of collaboration and adjusting its level through changes of β. This parameter shows up in the minimized objective function and shows how much other datasites impact the formation of the clusters at the given site. Higher values of β imply stronger collaborative linkages established between the sites. By reporting W as a function of β, that is W = W(β), we can experimentally optimize the intensity of collaboration. One may anticipate that while for low values of β no collaboration occurs and the values of W tend to be high, large values of β might lead to competition and subsequently the values of W(β) may tend to be high. Under some conditions, no convergence of the collaboration process is reported. There might be some regions of optimal values of β. Obviously, the optimal level (intensity) of collaboration depends upon a number of parameters, in particular, the number of clusters and the number of datasites involved in the collaboration. It could also depend on the data.

5.7 Quantification of collaboration using Type-2 fuzzy sets

It is advantageous to assess the quality of the results by evaluating their consistency and expressing a level of differences. Here, the quantification of results constitutes an interesting alternative—that of prototypes being treated as granular constructs. In the experiments reported here, results are computed in terms of type-2 fuzzy sets rather than numeric entities. Type-2 fuzzy sets are granular constructs; they are fuzzy sets whose membership functions do not assume numeric membership grades but instead are defined in a unit interval. Interestingly, type-2 fuzzy sets are discussed in various settings, but very little has been said about determination of their membership functions.

We estimate the membership of the i-th cluster using prototypes of D[ii] denoted by u = u_i. The prototypes optimized for the jj-th datasite, jj = 1, 2, ..., ii-1, ii+1, ..., P, give rise to the membership of **x** to the same i-th cluster. They are denoted by $z_1, z_2, \ldots, z_{P-1}$. We obtain a collection of membership grades captured in a type-2 fuzzy set. The corresponding membership function is determined by solving a certain optimization problem [18] which realizes a principle of*justifiable* granularity.

We consider a triangular fuzzy set to be one of the simpler versions of a membership function. Its use is legitimate when limited experimental evidence is available. The modal value u of the fuzzy set is the membership value obtained with the use of the prototypes present at D. The values of z_i that are lower than u, $z_i < u$, are used in the formation of the left-hand side of the linear portion of the membership function. The linearly decreasing portion of the membership function positioned at the right-hand side of the modal value u is optimized in the same manner. Computation of a type-2 fuzzy set membership function is already covered in Chapter 4 (Section 4.2.1).

5.8 Levels of information granularity during collaboration

So far we have assumed that the number of clusters at each of collaborating datasites is the same. In general this assumption is quite restrictive and not realistic. A more flexible scenario is one in which each party considers its own number of clusters (this could be quite legitimate considering that data structure could vary from site to site). This situation is well covered in existing literature and is supported by various algorithmic means including an extensive suite of cluster validity indexes.

Given this, the algorithmic settings in the present study have to be augmented. The major step is to present information granules at each datasite at the level of granularity that has been accepted before collaboration. There are several possible ways of doing this. Here we consider the one which uses clusters of prototypes. Consider the ii-th datasite. Before each phase of collaboration, we cluster the prototypes of this datasite $\{\mathbf{v}_i[ii]\}$, i = 1, 2, ..., c[ii], and the prototypes from all remaining datasites are communicated, $\{\mathbf{v}_i[jj]\}$, i = 1, 2, ..., c[jj], jj = 1, 2, ii-1, ii+1, ..., P. In such phase of clusters of prototypes,the number of clusters is kept the same as the number of clusters at this datasite. The results are denoted by $\mathbf{v}_i^{\sim}$, I = 1, 2, ..., c[ii]. The new prototypes are used in the next steps of collaborative clustering. More specifically, the minimized objective function is in the form:

$$Q[ii] = \sum_{i,k} u_{ik}^2[ii] \|\mathbf{x}_k - \mathbf{v}_i[ii]\|^2 + \beta \sum_{i=1}^{c[ii]} u_{ik}^2[ii] \|\mathbf{v}_i[ii] - \mathbf{v}_i^{\sim}[ii]\|^2 . \tag{20}$$

The optimization of (20) is described in the following section.

Optimization details of vertical clustering with different levels of granularity

We present all pertinent derivations of the collaborative scheme that pertains when participating datasites come with a different number of clusters. As before, datasites involved in the collaboration process are denoted by D[1], D[2], ..., D[P]. The objective function guiding the formation of the clusters at the central datasite comes as an augmented version of the one being used in the "standard" FCM, that is:

$$Q[ii]=\sum_{i,k}u_{ik}^{2}[ii]\left\|\mathbf{x}_{k}-\mathbf{v}_{i}[ii]\right\|^{2}+\beta\sum_{i=1}^{c[ii]}u_{ik}^{2}[ii]\left\|\mathbf{v}_{i}[ii]-\mathbf{v}_{i}^{\sim}[ii]\right\|^{2}. \tag{21}$$

Given the standard identity constraint imposed on the partition matrix, the optimization of (21), we confine ourselves to the use of the technique of Lagrange multipliers. For any data point k, k = 1, 2, ..., N, we expand the objective function to include the constraints:

$$V[ii]=\sum_{i,k}u_{ik}^{2}[ii]\left\|\mathbf{x}_{k}-\mathbf{v}_{i}[ii]\right\|^{2}+\beta\sum_{i=1}^{c[ii]}u_{ik}^{2}[ii]\left\|\mathbf{v}_{i}[ii]-\mathbf{v}_{i}^{\sim}[ii]\right\|^{2}-\lambda\Big(\sum_{i=1}^{c[ii]}u_{ik}[ii]-1\Big). \tag{22}$$

The necessary conditions for the minimum of V[ii] are expressed as:

$$\frac{\partial V[ii]}{\partial u_{rs}}=0,\ \frac{\partial V[ii]}{\partial \lambda}=0. \tag{23}$$

After computing the derivative with respect to the elements of the partition matrix we obtain:

$$\frac{\partial V[ii]}{\partial u_{rs}}=2u_{rs}[ii]\left\|\mathbf{x}_{s}-\mathbf{v}_{r}[ii]\right\|^{2}+2\beta\big(u_{rs}^{2}[ii]\left\|\mathbf{v}_{r}[ii]-\mathbf{v}_{r}^{\sim}[ii]\right\|^{2}\big)-\lambda=0, \tag{24}$$

where r = 1, 2, ..., c; s = 1, 2, ..., N.

The detailed calculations are shown below:

$$u_{rs}[ii]\Big(2\left\|\mathbf{x}_{s}-\mathbf{v}_{r}[ii]\right\|^{2}+2\beta\left\|\mathbf{v}_{r}[ii]-\mathbf{v}_{r}^{\sim}[ii]\right\|^{2}\Big)=\lambda$$

$$u_{rs}[ii]=\frac{\lambda}{2\left\|\mathbf{x}_{s}-\mathbf{v}_{r}[ii]\right\|^{2}+2\beta\left\|\mathbf{v}_{r}[ii]-\mathbf{v}_{r}^{\sim}[ii]\right\|^{2}}. \tag{25}$$

Given the constraint of the form $\sum_{j=1}^{c[ii]}u_{js}[ii]=1$, we obtain:

$$\sum_{j=1}^{c[ii]} \frac{\lambda}{2\left\|\mathbf{x}_s - \mathbf{v}_j[ii]\right\|^2 + 2\beta\left\|\mathbf{v}_j[ii] - \mathbf{v}_j^{\sim}[ii]\right\|^2} = 1. \tag{26}$$

In the sequel,

$$\lambda = \sum_{j=1}^{c[ii]} 2\left\|\mathbf{x}_s - \mathbf{v}_j[ii]\right\|^2 + \sum_{j=1}^{c[ii]} 2\beta\left\|\mathbf{v}_j[ii] - \mathbf{v}_j^{\sim}[ii]\right\|^2. \tag{27}$$

Plugging (27) into (25) gives:

$$u_{rs}[ii] = \frac{\sum_{j=1}^{c[ii]} 2\left\|\mathbf{x}_s - \mathbf{v}_j[ii]\right\|^2 + \sum_{j=1}^{c[ii]} 2\beta\left\|\mathbf{v}_j[ii] - \mathbf{v}_j^{\sim}[ii]\right\|^2}{2\left\|\mathbf{x}_s - \mathbf{v}_r[ii]\right\|^2 + 2\beta\left\|\mathbf{v}_r[ii] - \mathbf{v}_r^{\sim}[ii]\right\|^2}. \tag{28}$$

Further simplifications lead us to the following expression:

$$u_{rs}[ii] = \frac{1}{\sum_{j=1}^{c[ii]} \frac{\left\|\mathbf{x}_s - \mathbf{v}_r[ii]\right\|^2 + \beta\left\|\mathbf{v}_r[ii] - \mathbf{v}_r^{\sim}[ii]\right\|^2}{\left\|\mathbf{x}_s - \mathbf{v}_j[ii]\right\|^2 + \beta\left\|\mathbf{v}_j[ii] - \mathbf{v}_j^{\sim}[ii]\right\|^2}}. \tag{29}$$

Proceeding with the optimization of the objective function (21) with regard to the prototypes, we consider now the Euclidean distance (its weighted version is handled in the same manner). Given the form of the distance, the objective function is written as:

$$Q[ii] = \sum_{i,k} u_{ik}^2[ii] \sum_{j=1}^{n} \left(x_{kj} - v_{ij}[ii]\right)^2 + \beta \sum_{i=1}^{c[ii]} u_{ik}^2[ii] \sum_{j=1}^{n} \left(v_{ij}[ii] - v_{ij}^{\sim}[ii]\right)^2. \tag{30}$$

The necessary condition leading to the minimization of Q comes in the form:

$$\frac{\partial Q[ii]}{\partial v_{rt}} = 0.$$

Then we obtain:

$$\frac{\partial Q[ii]}{\partial v_{rt}} = -2\sum_{k=1}^{N} u_{rk}^{2}[ii](x_{kt} - v_{rt}[ii]) + 2\beta\sum_{i=1}^{c[ii]} u_{rk}^{2}[ii](v_{rt}[ii] - v_{rt}^{\sim}[ii]) = 0. \tag{31}$$

After further simplifications we derive:

$$\sum_{k=1}^{N} u_{rk}^{2}[ii]x_{kt} = \sum_{k=1}^{N} u_{rk}^{2}[ii]v_{rt}[ii] + \beta\sum_{k=1}^{N} u_{rk}^{2}[ii]v_{rt}[ii] - \beta\sum_{k=1}^{N} u_{rk}^{2}[ii]v_{rt}^{\sim}[ii].$$

Finally, we arrive at the expression:

$$v_{rt}[ii] = \frac{\sum_{k=1}^{N} u_{rk}^{2}[ii]x_{kt} + \beta\sum_{k=1}^{N} u_{rk}^{2}[ii]v_{rt}^{\sim}[ii]}{\sum_{k=1}^{N} u_{rk}^{2}[ii](1+\beta)}, \tag{32}$$

where r = 1, 2, ..., c; t = 1, 2, ..., n.

The overall flow of processing can be presented as follows:

Given: datasites D[1], D[2], ..., D[P] with different structures,
Select a number of clusters (c[ii]) for each datasite, set up some termination criterion, and establish a level of collaboration (interaction) by choosing some nonnegative value of β.

Initial phase
Carry out clustering (FCM) for each datasite producing a collection of prototypes $\{v_i[ii]\}$, i = 1, 2, ..., c[ii] for each datasite.

Collaboration
Iterate {successive phases of collaboration}

    Communicate the results about the structure determined at each datasite.

    For each datasite (ii)
    {

```
    Collect all prototypes from other sites at datasite (ii) and run FCM on that
    collection by selecting the same number clusters at that site to generate new
    prototypes $\tilde{\mathbf{v}}[ii]$. Minimize (20) at each datasite by iteratively proceeding with
    the iterative calculations of the partition matrix ($u_{rs}[ii]$) and the prototypes
    ($v_{rt}[ii]$) using respectively (29) and (32).

    $r = 1, 2, \ldots, c[ii]; t = 1, 2, \ldots, n; s = 1, 2, \ldots, N$
    } for the datasite

*until* termination condition of the collaboration activities has been satisfied.
```

The quality of collaboration is optimized by choosing a suitable value of β (which minimizes the performance index W given by (19)).

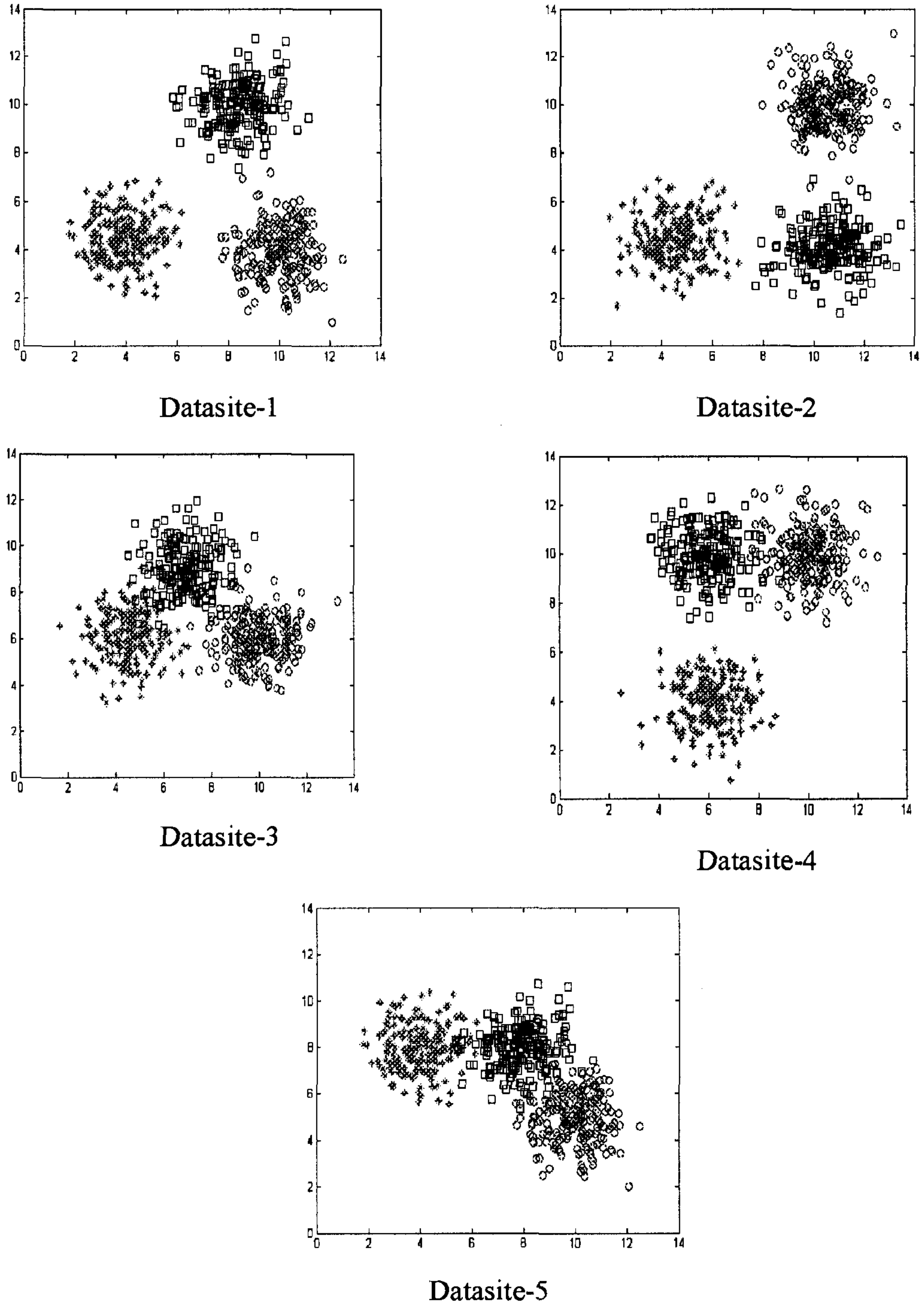
5.9 Experimental studies

The collaborative clustering mechanisms presented in this chapter are illustrated through a series of experimental studies. Here we use synthetic datasets and datasets from the Machine Learning Repository and the Canada Weather network. By experimenting with essential parameters of the environment, we gain a better sense of the impact of the granularity (number of clusters), the number of collaborating datasites, and the intensity of collaboration on the dynamics of interaction and the quality of results. The performance of the collaborative discovery of structure in datasites is expressed through the values of the sum of distances between the proximity matrices induced by the partition matrices obtained at the individual datasites. As discussed, meaningful collaboration is reflected by the reduction of differences between these partition matrices (and their proximity matrices).

Synthetic two-dimensional data. We consider two-dimensional synthetic datasets with five datasites (datasite-1, datasite-2, …, datasite-5) as shown in Figure 5.5. Each datasite is comprised of 600 patterns. These data were generated using a mixture of three Gaussian distributions with following mean vectors:

Datasite1: $\mathbf{v}_1$= [4.0 4.5]	$\mathbf{v}_2$= [8.5 10.0]	$\mathbf{v}_3$ = [10.0 4.0]
Datasite2: $\mathbf{v}_1$= [4.5 4.5]	$\mathbf{v}_2$= [10.5 4.0]	$\mathbf{v}_3$ = [10.5 10.0]
Datasite3: $\mathbf{v}_1$= [4.5 6.0]	$\mathbf{v}_2$= [7.0 9.0]	$\mathbf{v}_3$ = [10.0 6.0]
Datasite4: $\mathbf{v}_1$= [6.0 4.0]	$\mathbf{v}_2$= [6.0 10.0]	$\mathbf{v}_3$ = [10.0 10.0]
Datasite5: $\mathbf{v}_1$= [4.0 8.0]	$\mathbf{v}_2$= [8.0 8.0]	$\mathbf{v}_3$ = [10.0 5.0]

The covariance matrix is equal to $\begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$ and is the same for all groups across the datasites. Different mean vectors resulted in quite different topology of data.



Datasite-1

Datasite-2

Datasite-3

Datasite-4

Datasite-5

Figure 5.5 Two-dimensional synthetic datasets used in the collaboration process; the mean vectors vary across datasites.

We consider the number of clusters to be equal to 3. Running the collaborative clustering for different values of β and reporting the values of W in successive numbers of phases, we have determined an optimal combination where the lowest value of the performance index W equal to 2.05 was achieved after 17 phases of collaboration. The initial value of W was equal to 4.41 so the collaboration has led to a substantial reduction of the values of W and in this way demonstrates the effectiveness of the collaboration process. The optimal value of β was found to be equal to 0.25. Figure 5.6 displays the convergence process of the values of W.

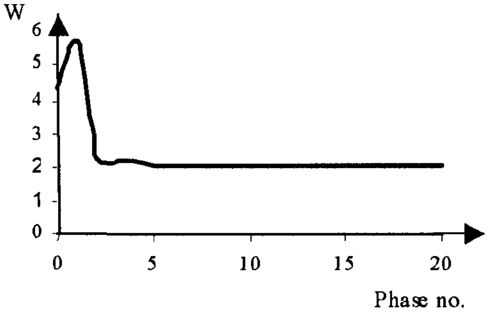


Figure 5.6 Values of the performance index obtained in successive phases of collaboration for the optimal value of β (0.25). Note that the values of W have been substantially reduced with respect to the original value (W = 4.41) when no collaboration has been realized.

The tangible and visually appealing outcome of collaboration comes in the form of the prototypes of the clusters. Table 5.2 shows these results obtained for the optimal value of β (0.25).

Table 5.2 Prototypes at individual datasites before and after collaboration.

Before collaboration	After collaboration
Datasite-1	
v_1=[8.40 10.02]	v_1=[9.85 4.63]
v_2=[3.94 4.43]	v_2=[4.46 4.55]
v_3=[10.00 3.94]	v_3=[8.24 9.72]
Datasite-2	
v_1=[4.50 4.41]	v_1=[10.47 5.21]
v_2=[10.64 4.09]	v_2=[4.97 4.59]
v_3=[10.48 9.97]	v_3=[10.08 9.65]
Datasite-3	
v_1=[9.97 5.91]	v_1=[9.65 6.03]
v_2=[4.47 5.93]	v_2=[4.78 5.85]
v_3=[7.07 9.18]	v_3=[7.28 9.15]
Datasite-4	
v_1=[5.93 4.09]	v_1=[9.09 7.97]
v_2=[10.09 10.01]	v_3=[5.58 5.18]
v_3=[5.89 10.05]	v_3=[8.14 9.97]
Datasite-5	
v_1=[7.96 8.06]	v_1=[9.38 5.79]
v_2=[9.94 4.94]	v_2=[5.46 6.73]
v_3=[4.01 8.21]	v_3=[6.89 8.66]

Collaboration can also be quantified in the form of granular prototypes emerging through the collaboration process. Prototypes produced when datasites are processed are individually numeric; it is the collaborative reconciliation of the findings that gives rise to the granular character of the prototypes. The form of the resulting triangular fuzzy sets (the result of optimization of (19) reflects the level of consistency obtained between various structures in the datasites. For instance, in the case of datasite-4, the prototype of the first cluster exhibits a substantial spread for the second variable. A similar effect is observed in the case of the second cluster. Computing the Cartesian product of the corresponding membership functions, we obtain the fuzzy sets of the prototypes shown in Figure 5.7.

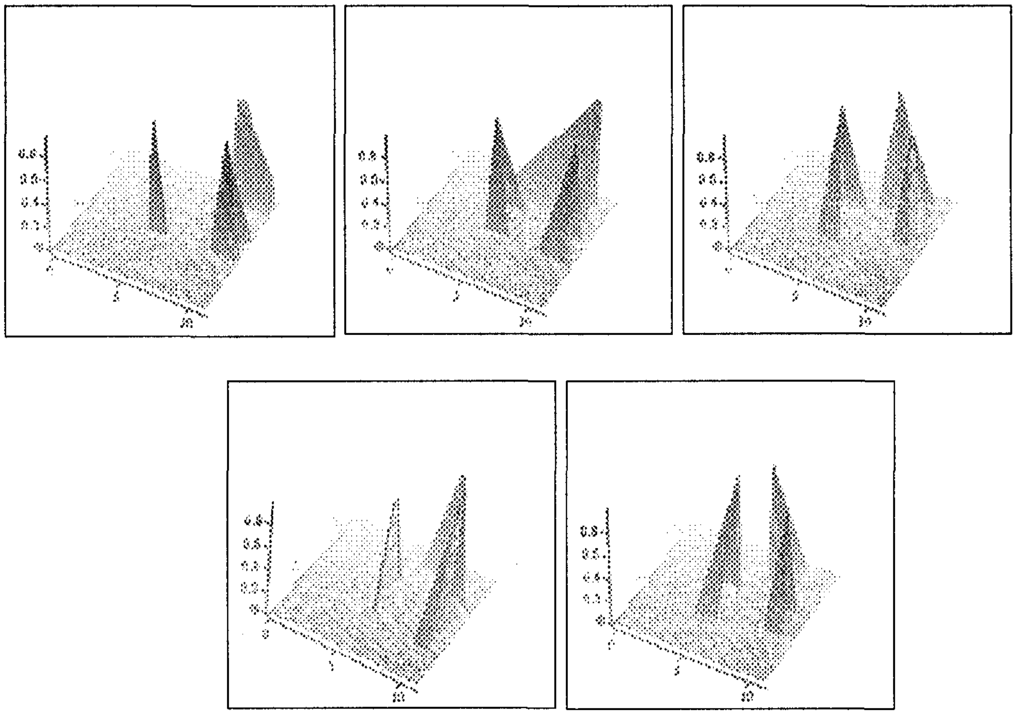
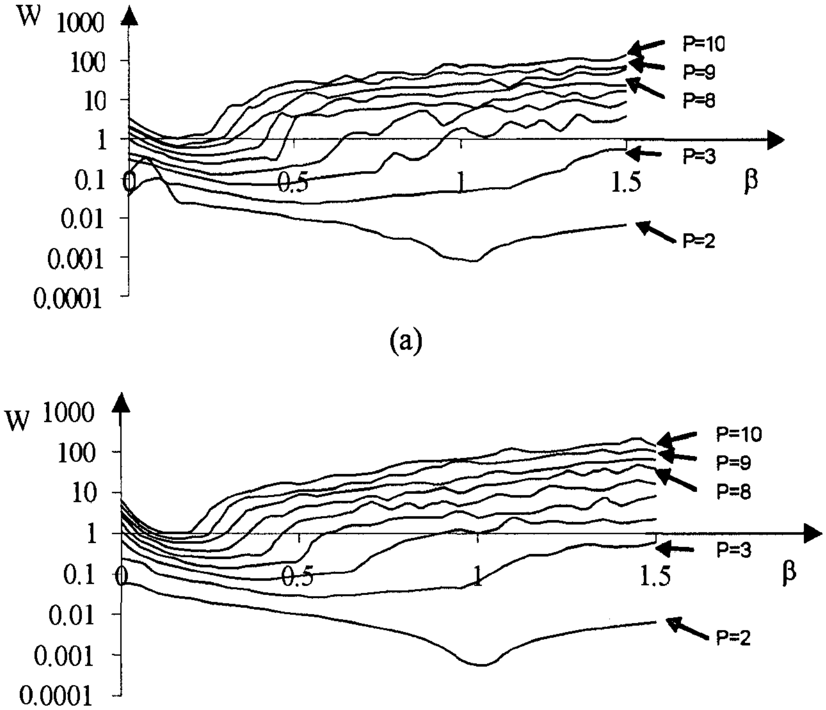


Figure 5.7 Fuzzy sets of prototypes for the datasites. The Cartesian products are constructed by taking the minimum of the membership functions formed for the two variables.

The series of plots in Figure 5.7 shows how much the granularity of the prototypes reflects the diversity of the prototypes produced locally at individual datasites. For instance, the fuzzy set of the prototype in datasite-2 (see the second graph in Figure 5.7), exhibits a substantial spread along the first variable which becomes reflective when considering the corresponding prototypes obtained for the remaining datasites.

Selected Machine Learning data. The datasets discussed in this section come from the Machine Learning Repository, see http://www.ics.uci.edu/~mlearn/MLRepository.html.

Abalone. The experiments are arranged in the same manner as the experiment with synthetic data discussed previously. Our primary interest is in the assessment of the effectiveness of the collaboration vis-à-vis the main parameters of the algorithm and the details of the setup of the environment. In particular, the results are reported for a different number of the datasites involved in the collaboration, see Figure 5.8. In general, the collaboration led to a significant improvement in the consistency results reflected in lower values of W. The effectiveness of collaboration depends on the values of β. Here a clear tendency is observed: with more datasites engaged in collaboration, the optimal intensity of collaboration becomes lower. The choice of a suitable value of β is not overly critical; we encounter regions of β over which the values of W remain quite similar. The results are reported in Table 5.3.



(a)

(b)

Figure 5.8 Values of the collaboration index reported as a function of β obtained after 20 phases of collaboration (a) c = 3, (b) c = 6.

P	2	3	4	5	6	7	8	9	10
Optimal β	1	0.55	0.40	0.30	0.25	0.20	0.15	0.15	0.15

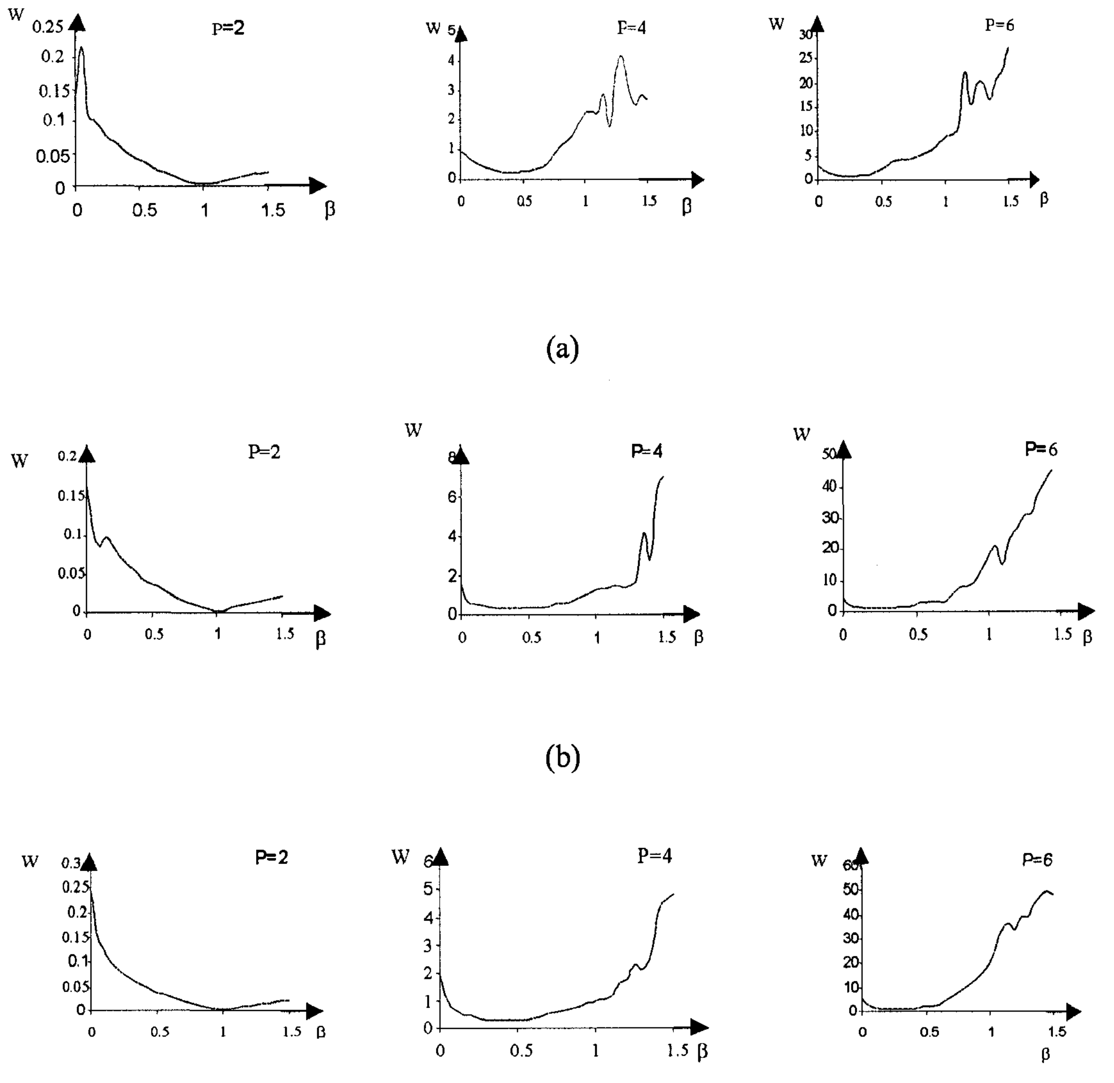
(a)

Table 5.3 Optimal values of β for c = 3 (a) and c = 6 (b); refer also to Figure 5.8.

P	2	3	4	5	6	7	8	9	10
Optimal β	1	0.55	0.35	0.30	0.25	0.20	0.20	0.15	0.15

(b)

Boston Housing The results are organized in a series of graphs, Figure 5.9, in the same way as the previous dataset. The experimental evidence suggests several conclusions. First, a higher number of datasites leads to lower values of β; this indicates that the collaboration needs to be established in a less intensive manner. This is not surprising: when a large number of parties (datasites) are involved in a collaboration, individual interactions may be weak, leading to breakdown of data exchange. This effect is quite visible in several plots in Figure 5.9: for higher values of β, the values of W increase quite rapidly for high values of P. On the other hand, in the neighborhood of the optimal value of β, slight deviations from the optimum value are not highly detrimental.



(a)

(b)

(c)

Figure 5.9 Values of the collaboration index for c = 5 (a), c = 10 (b), and c = 15 (c) obtained after 20 phases of collaboration and reported for selected values of the collaboration intensity β and selected number of datasites P.

Canada Weather Network data. Datasets for Alberta and British Columbia weather were selected to test our theories of collaborative clustering. The data is available at: http://www.climate.weatheroffice.ec.gc.ca/prods_servs/cdcd_iso_e.html.

This dataset is attractive as it captures a distributed (spatial) phenomenon; that is, data are collected at different sites. While structures could be analyzed individually (locally), development of a holistic view would be informative and could reveal the power of collaborative clustering. Ten datasites are considered for each province.

Alberta. Ten geographically distributed datasites were selected in the province of Alberta: 1. Beaver Mines, 2. Calgary INT A, 3. Kananaskis, 4. Bindloss East, 5. Big stone, 6. Alliance South, 7. Cold Lake A, 8. Athabasca-2, 9. Brule Black, and 10. Ballater. The distribution of these locations is illustrated in Figure 5.10.

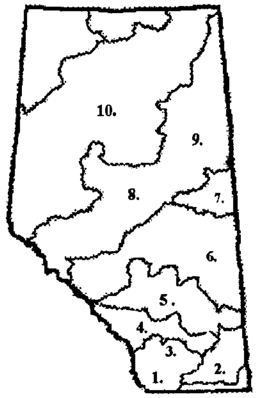


Figure 5.10 Distribution of 10 datasites (weather stations) in the province of Alberta.

These 10 datasites comprises 801 weather records collected over the winter seasons (December, January, February) of 1991–2000. Each data item is described by four features: maximum temperature, minimum temperature, average temperature, and precipitation. Experiments were run for c = 3 and c = 6 clusters. The collaboration was optimized by running the experiment for different values of β. The lowest value of W was obtained during the 11th phase of collaboration for $\beta = 0.85$. Values of the prototypes were obtained before and after collaboration. Figure 5.11 shows the reconciliation of the findings obtained locally at each datasite. We observe that the prototypes obtained after collaboration (right-hand column of plots) get closer to each other compared with prototypes obtained without collaboration (left-hand column of plots).

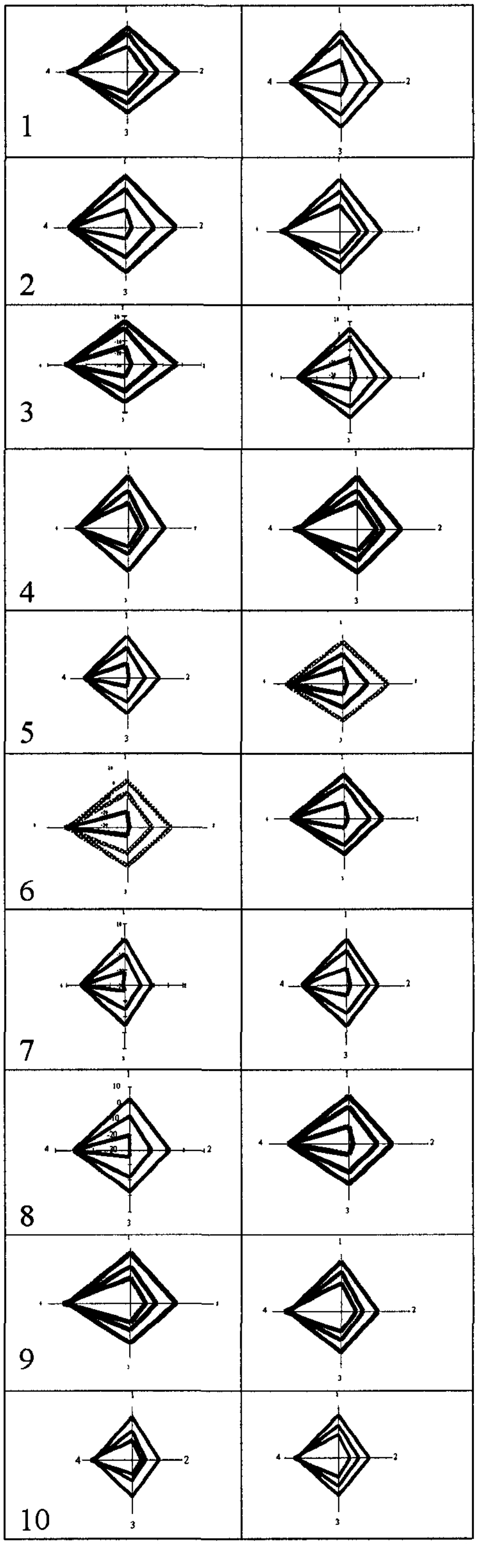


Figure 5.11 Radar plots of prototypes before collaboration (left column) and after collaboration (right column). Coordinates of the plots of the prototypes are numbered as follows: 1–maximum temperature, 2–minimum temperature, 3–average temperature, and 4–precipitation.

Granular prototypes offer another useful insight into the effects of collaboration (see Figure 5.12). We note that the fuzzy sets of prototypes exhibit higher variability with respect to precipitation while the variability with respect to average temperature remains quite limited (the support of the fuzzy set along this coordinate is very narrow).

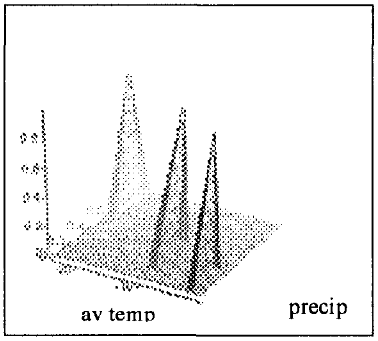
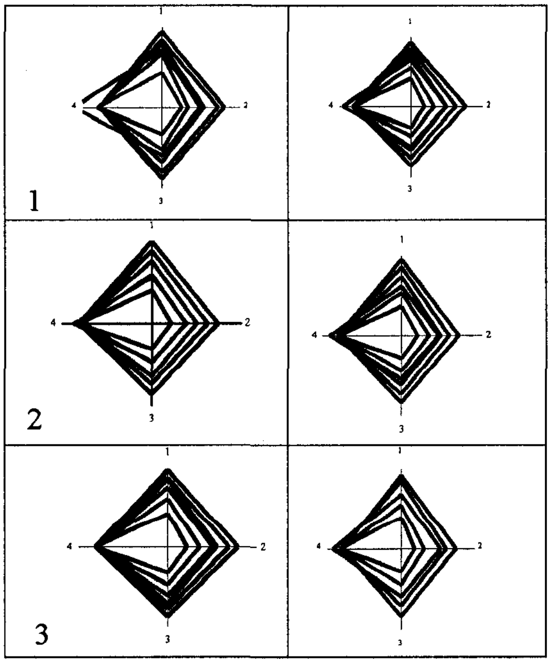


Figure 5.12 Fuzzy sets of granular prototypes constructed for datasite 6. The prototypes are reported in the space of average temperature (av. temp) and precipitation (precip).

The radar plots of the prototypes for c = 6 are shown in Figure 5.13. Here the optimal value of β is equal to 0.7 at the 20th phase of collaboration. Through the collaboration, the values of the performance index W were reduced from 11.60 (no collaboration) to 3.79.



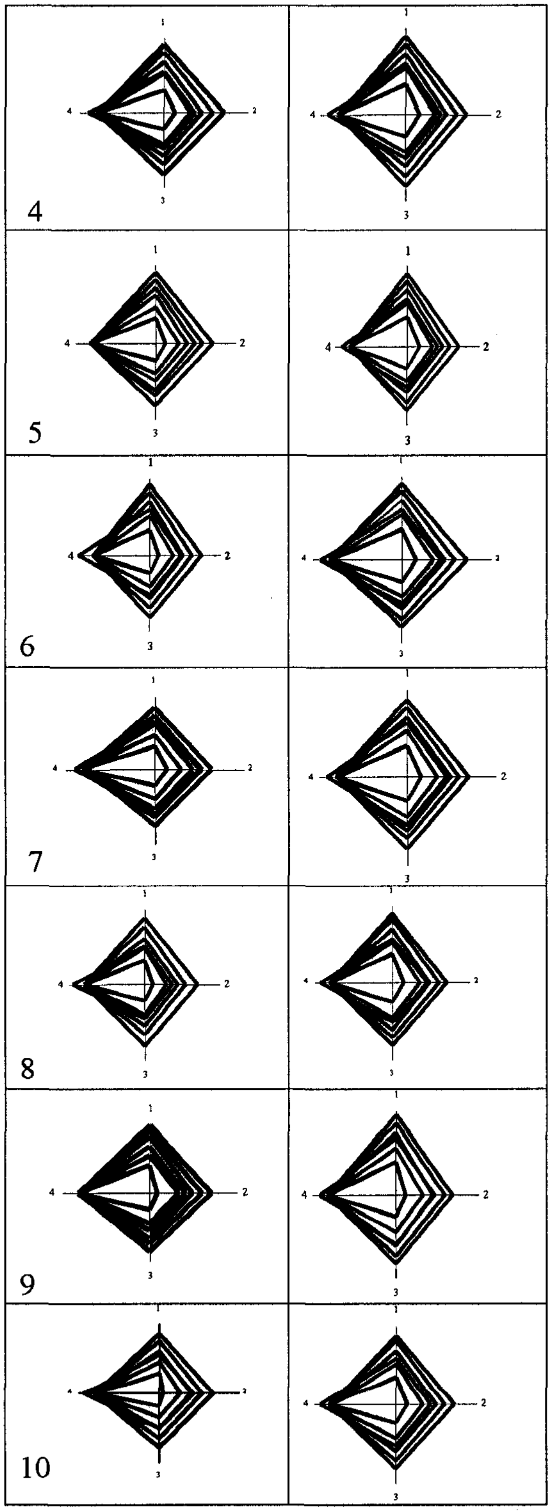
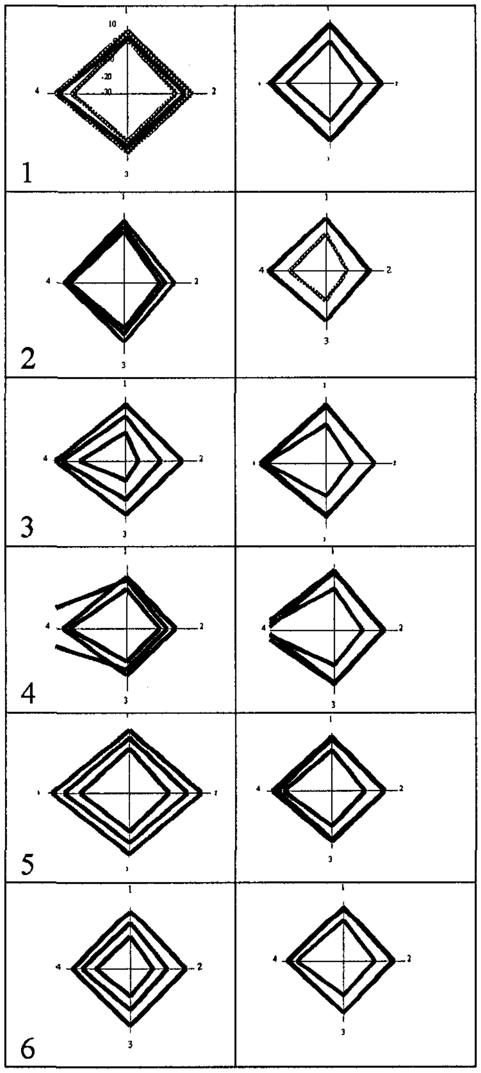


Figure 5.13 Radar plots of prototypes before collaboration (left column) and after collaboration (right column). Coordinates of the plots of the prototypes are numbered as follows: 1–maximum temperature, 2–minimum temperature, 3–average temperature, and 4–precipitation.

British Columbia Following the same scheme as discussed for the Alberta weather data, we consider 10 datasites in British Columbia: Chemainus (1), Black Creek (2), Alberni Robertson Creek (3), Boat Bluff (4), Gibsons Gower Point (5), Langara (6), Bella Coola A (7). Babine Lake Pinkut Creek (8), Hixon (9), and Penticton A (10). Radar plots for c = 3 and c = 6 are presented in Figure 5.14 and 5.15, respectively. For c = 3, the optimal value of β was 0.3 and the collaboration was realized in two phases resulting in the reduction of W from 22.34 (prior to collaboration) to 7.52 (after collaboration). For c = 6 the optimal value of β was 0.55 and the collaboration took 20 phases resulting in the reduction of the values of W from 22.82 to 7.01. For both c = 3 and c = 6 the prototypes reported at each datasite become close to each other after collaboration; note that the radar plots in the right-hand column start resembling each other. An even more profound effect is reported for c = 6; in this case the prototypes get close to each other.



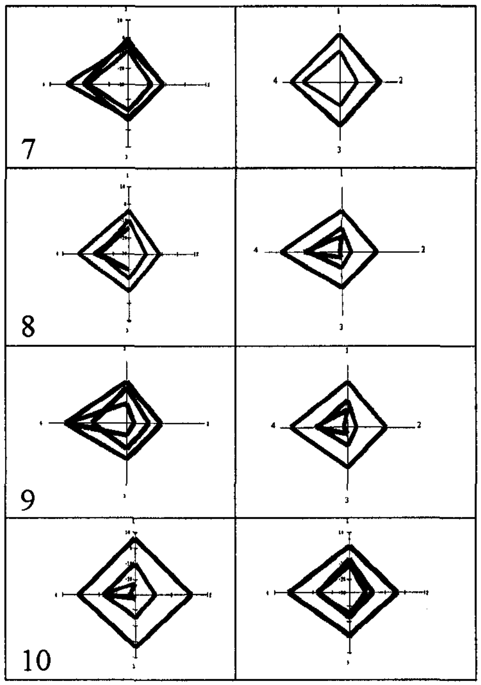
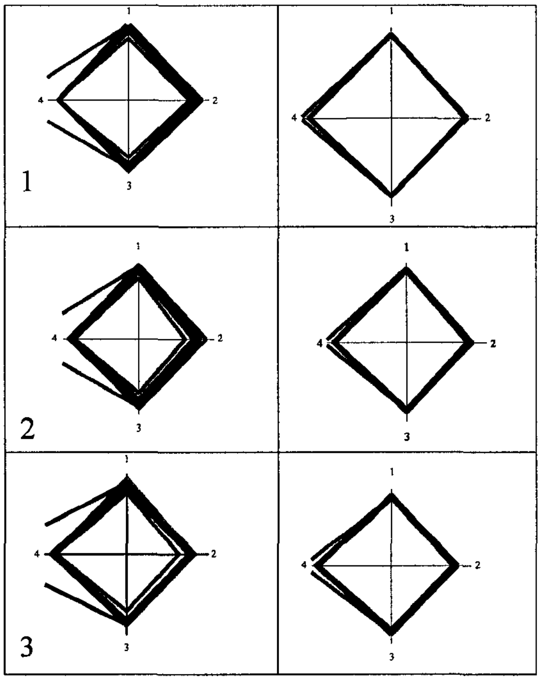


Figure 5.14 Radar plots of prototypes before collaboration (left column) and after collaboration (right column), c = 3. Coordinates of the plots of the prototypes are numbered as follows: 1–maximum temperature, 2–minimum temperature, 3–average temperature, and 4–precipitation.



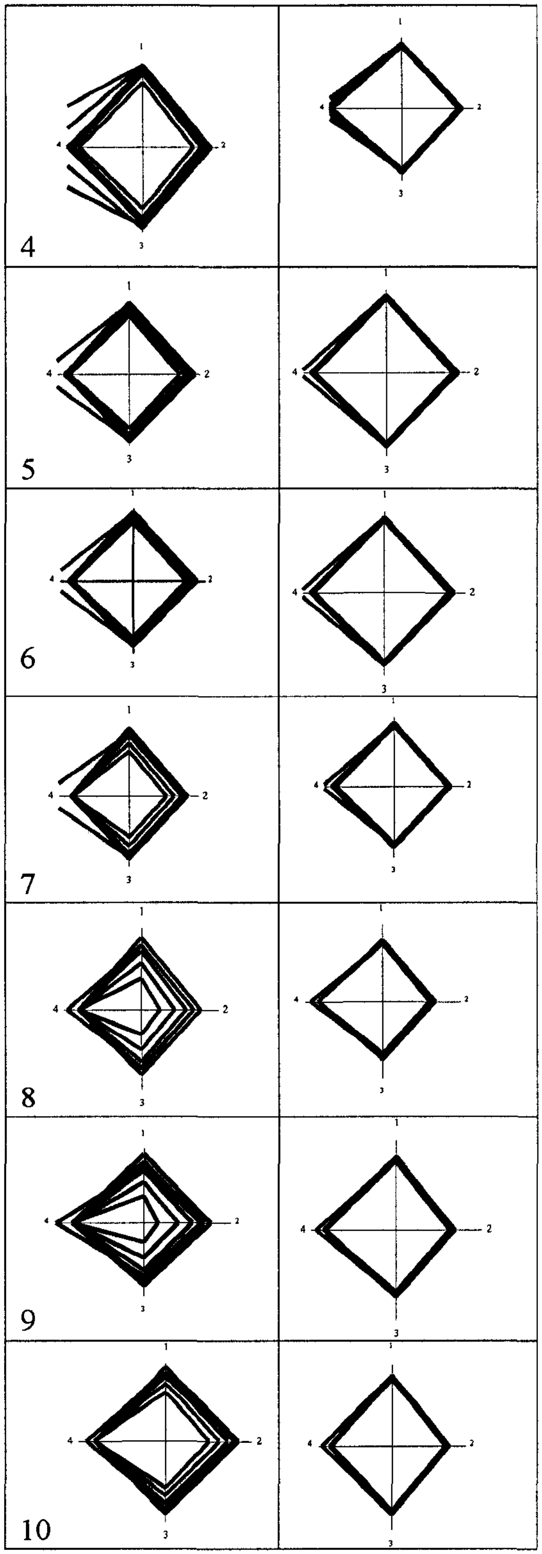


Figure 5.15 Radar plots of prototypes before collaboration (left column) and after collaboration (right column), c = 6. Coordinates of the plots of the

prototypes are numbered as follows: 1–maximum temperature, 2–minimum temperature, 3–average temperature, and 4–precipitation.

For c = 3, the plots of the fuzzy sets of prototypes at datasite 1 are shown in Figure 5.16.

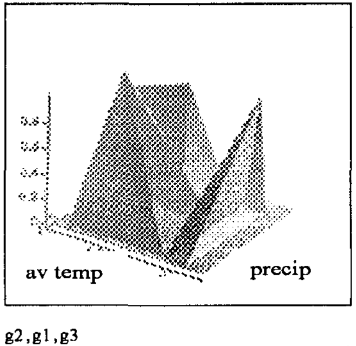


Figure 5.16 Fuzzy sets of granular prototypes constructed for datasite 1. The prototypes are reported in the space of average temperature (av. temp) and precipitation (precip).

We compare the results obtained so far with those obtained for collaborations realized in the presence of different information granularity used at the datasites.

We start with the synthetic dataset and consider a different number of clusters at each datasite, that is, c[1] = 3, c[2] = 3, c[3] = 4, c[4] = 2, and c[5] = 5. The results are reported in terms of the optimal value of β and the dynamics of collaboration quantified in terms of the values of W reported in consecutive phases of collaboration (see Figure 5.17).

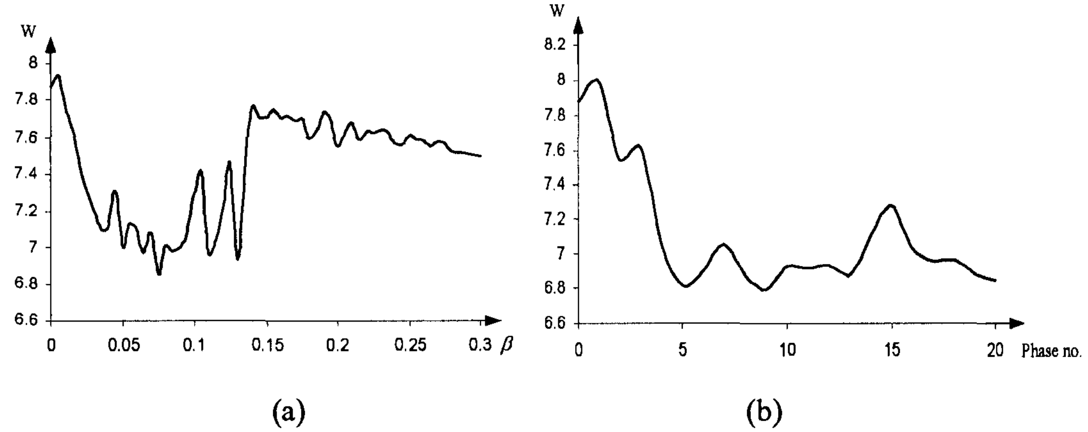
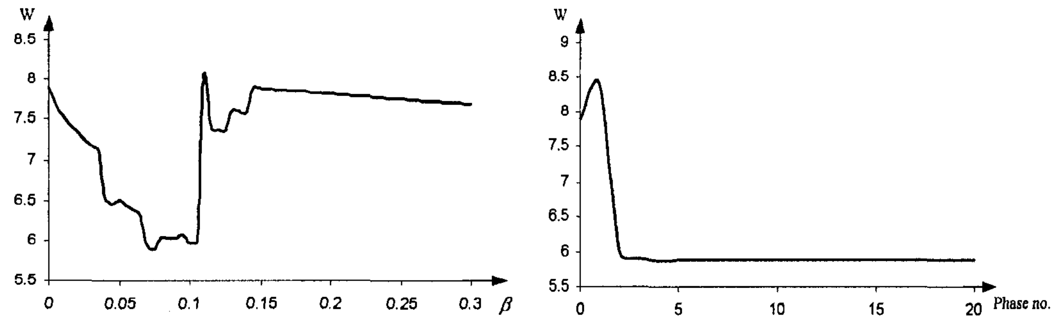


Figure 5.17 W regarded as a function of β (a), and dynamics of the collaboration process (b)

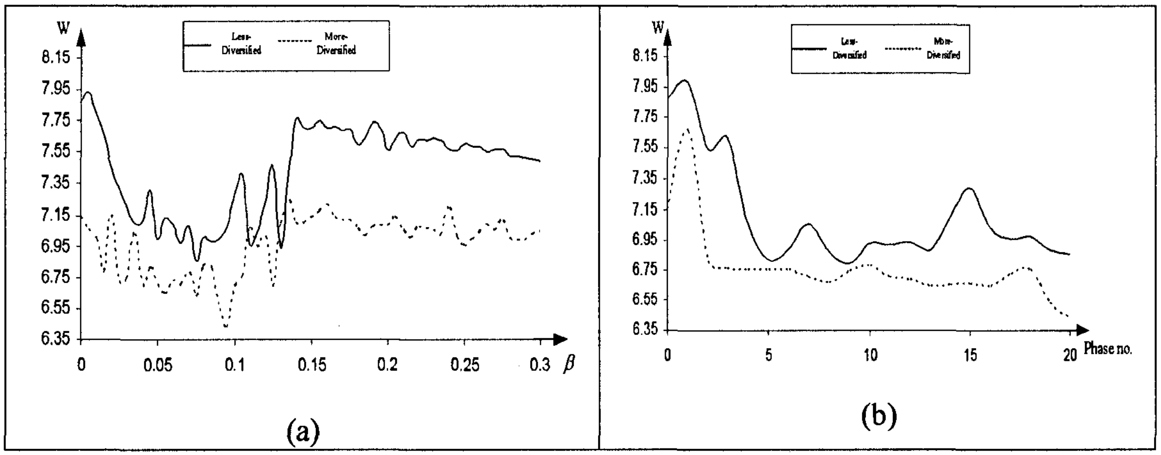
For comparison, Figure 5.18 includes the results for the case where the granularity is the same across all datasites; in this case we have c[ii] = 3. Interestingly, while the optimal value of β remains the same (β_{opt}= 0.075), the dynamics of collaboration has changes—we observe a number of ripples (oscillations) in the values of W produced in the phases which might be reflective of the heterogeneous character of information granules present at each datasite.



(a) (b)

Figure 5.18 Plots of W treated as a function of β (a), and values of W in successive phases of collaboration (b)

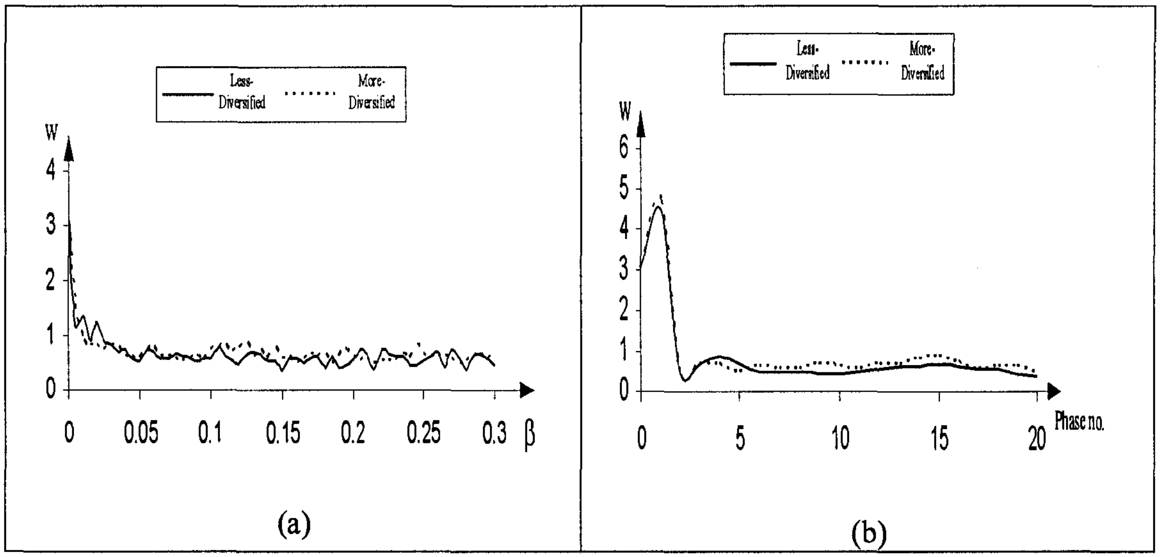
When the granularity of information becomes more diversified, the dynamics of the collaboration reflect this effect. Figure 5.19 shows that less diversified datasites come with the following numbers of clusters: c[1] = 3, c[2] = 3, c[3] = 4, c[4] = 2, c[5] = 5 and more diversified datasites have c[1] = 3, c[2] = 3, c[3] = 4, c[4] = 7, c[5] = 5.



(a) (b)

Figure 5.19 Performance of collaborative clustering expressed in terms of the values of the performance index W: (a) W regarded as a function of β, (b) Values of W reported in successive phases of collaboration.

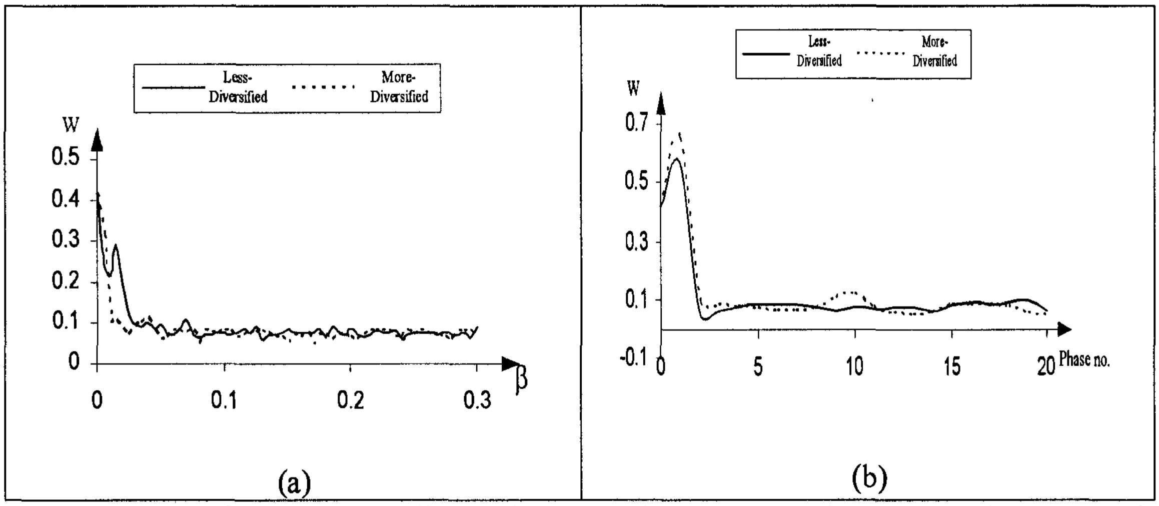
Effect of the number of information granules across datasites was tested with Boston housing (Figure 5.20) and Wine (Figure 5.21) datasets.



(a)

(b)

Figure 5.20 Performance of collaborative clustering expressed in terms of the values of the performance index W: (a) W regarded as a function of β, (b) Values of W reported in successive phases of collaboration. Clustering of higher diversity: c[1] = 3, c[2] = 5, c[3] = 7, c[4] = 10 clustering of lower diversity: c[1] = 3, c[2] =5, c[3]= 5, c[4] = 2.



(a)

(b)

Figure 5.21 Performance of collaborative clustering expressed in terms of the values of the performance index W: (a) W regarded as a function of β, and (b) Values of W reported in successive phases of collaboration. Higher diversity in the number of clusters: c[1] = 5, c[2] = 10; lower diversity of granularity is characterized by c[1] = 5, c[2] = 2.

5.10 Conclusions

We have introduced a concept of a multiple phases vertical collaborative mode of fuzzy clustering. While the FCM has been used as a generic vehicle of fuzzy clustering, the framework of collaboration presented here could be easily applicable to other objective function-based techniques of fuzzy clustering leading to the generation of prototypes and partition matrices. The crux of the collaboration concerns granular communication (exchange) of prototypes generated at different datasites and generation of induced partition matrices that are formed on a basis of data locally available at the given datasite.

The augmented objective function with a series of additional terms involving the effect of collaborative development of the clusters is used to guide the overall optimization activities. The optimal level of collaboration is calibrated by using a single numeric parameter. The collected experimental evidence demonstrates that properly selected values of β help to establish meaningful collaborative activities whose effectiveness is measured in terms of the proposed consistency measure.

We establish the framework of collaborative clustering in presence of a different number of clusters at each datasite. The underlying algorithm is augmented with a phase during which we arrive at a homogeneous form of results produced by an intermediate clustering phase where prototypes are grouped. The dynamics of the collaboration process is more complex and exhibits more oscillations in the values of the performance index W reported in consecutive phases of the collaboration.

The quality of collaboration can be concisely quantified in terms of granular prototypes that reflect the way in which collaboration led to similar results throughout the datasites. The series of experiments shows that the values of this index were significantly reduced in comparison with cases where no collaboration was established.

The concepts and algorithms presented here constitute a radical departure from the collaborative and cluster ensemble pursuits available in the literature. There are two striking differences: (a) the development of clusters is a proactive development in which we actively engage in the construction of the clusters by exchanging the findings in consecutive phases of collaboration. Other models are static and operate in a *post-mortem* manner; namely, they try to reconcile the clusters once they have been constructed, and (b) the approach provides quantification of the results of collaboration in terms of higher order granular constructs. There are some similarities in the sense that in both scenarios we are concerned with knowledge reuse and knowledge integration.

Collaborative clustering realized in the framework presented in this chapter exhibits a significant level of generality. Given this, it could be used when solving other tasks of collaboration such as, e.g., a *collaborative* development of regression models, fuzzy rule-based systems, binary classification, and neural networks.

References

1. H. Ayad, M. Kamel, Finding natural clusters using multi-cluster combiner based on shared nearest neighbors, *Proc. 4th Int. Workshop on Multiple Classifier Systems*, 2003, 166-175.

2. J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, N. York, 1981.

3. J. Costa da Silva, M. Klusch, Inference in distributed data clustering, *Engineering Applications of Artificial Intelligence*, 19, 2006, 363-369.

4. S. Dudoit, J. Fridlyand, Bagging to improve the accuracy of a clustering procedure, *Bioinformatics*, 19, 2003, 1090-1099.

5. Dimitriadou, A. Weingessed, K. Hornik, Voting-merging: An ensemble method for clustering, *In Proc. Int. Conf. on Artificial Neural Networks*, Vienna, 217-224,2001.

6. A. Jain, M. Murt, P. Flynn, Data clustering: a review, *ACM Computing Surveys*, 31, 1999, 264-323.

7. A. K. Jain, R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Englewood Cliffs, NJ, 1988.

8. A. Jain, R. Duin, J. Mao, Statistical pattern recognition: a review, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 2000, 4-37.

9. E. Johnson, H. Kargupta, Collective hierarchical clustering from distributed, heterogeneous data, *Large-Scale Parallel KDD Systems*, Lecture Notes in Computer Science, no. 1759, Springer-Verlag, London, UK, 1999, 221-244.

10. F. Hoppner et al., *Fuzzy Cluster Analysis*, J. Wiley, Chichester, England, 1999.

11. L. Hubert, P. Arabie, Comparing partitions, *Journal of Classification*, 2, 1985, 193-218.

12. Y. Leung, J. Zhang, Z. Xu, Clustering by space-space filtering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 2000, 1396-1410.

13. V. Loia, W. Pedrycz, S. Senatore, P-FCM: a proximity-based fuzzy clustering for user-centered web applications, *Int. J. of Approximate Reasoning*, 34, 2003, 121-144.

14. S. Merugu, J. Ghosh, A privacy-sensitive approach to distributed clustering, *Pattern Recognition Letters*, 26, 2005, 399-410.

15. R. Nowak, Distributed EM algorithms for density estimation and clustering in sensor networks, *IEEE Trans. on Signal Processing*, 51, 2003, 2245-2253.

16. K. L. Oehler, R. M. Gray, Combining image compression and classification using vector quantization, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17, 1995, 461–473.

17. W. Pedrycz, Collaborative fuzzy clustering, *Pattern Recognition Letters*, 23, 2002, 675-686.

18. W. Pedrycz, *Knowledge-Based Clustering: From Data to Information Granules*, J. Wiley, Hoboken, NJ, 2005.

19. W. Pedrycz, Distributed fuzzy systems modeling, *IEEE Transactions on Systems, Man, and Cybernetics*, 25, 1995, 769-780.

20. W. Pedrycz, G. Vukovich, Clustering in the framework of collaborative agents, *Proc. 2002 IEEE Int. Conference on Fuzzy Systems*, 1, 2002, 134-138.

21. W. Pedrycz, M. Reformat, Evolutionary fuzzy modeling, *IEEE Transactions on Fuzzy Systems*, 11, 2003, 652-665.

22. W. Pedrycz, Conditional fuzzy clustering in the design of radial basis function neural networks, *IEEE Transactions on Neural Networks*, 9, 1998, 601-612.

23. E. Rasmussen, *Clustering Algorithms in Information Retrieval: Data Structures and Algorithms*, Prentice-Hall, Englewood Cliffs, NJ, 1992.

24. A. Strehl, J. Ghosh, Cluster ensembles: a knowledge reuse framework for combining multiple partitions, *Journal of Machine Learning Research*, 3, 2002, 583-617.

25. A. Topchy, K. Jain, W. Punch, Clustering ensembles: models of consensus and weak partitions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 2005, 1866-1881.

26. V. S. Veryklos et al., State-of-the art in privacy preserving data mining, *SIGMOID Record*, 33, 2004, 50-57.

27. L. A. Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*, 90, 1997, 111-117.

Chapter 6

Horizontal Fuzzy Collaborative Clustering

The previous chapter was concerned with a collaborative scheme involving intelligent agents carrying out collaboration in a so-called vertical mode.

In this chapter, we focus on a broad category of problems of collaborative data analysis realized by a collection of agents having access to their individual data and exchanging findings through their collaboration activities in a so-called horizontal mode. Such problems of data analysis arise in the context of building a global view at a certain phenomenon (process) by viewing it from different perspectives (and thus engaging various collections of attributes by various agents). To effectively arrive at meaningful solutions in a vast array of problem solving, it becomes imperative to establish a sound machinery to reconcile findings which might form partial solutions to an overall problem. We develop a comprehensive optimization scheme and discuss its two-phase character in which the communication phase of the granular findings intertwines with the local optimization being realized by the agents at the level of the individual site and exploits the evidence collected from other sites. We show how the mechanism of fuzzy granulation realized in the form of a well-known fuzzy c-means (FCM) algorithm can be augmented to support collaborative activities required by the agents. We illustrate the performance of our approach by conducting experiments over synthetically generated data and several selected data sets obtained from the Machine Learning Repository.

6.1 Introduction

When traversing a path from data to knowledge, we encounter various information processing tasks in which knowledge is being effectively formed from multifaceted and multicriteria perspectives. Knowledge formation pursuits of practical relevance commonly rely on distributed data sources. On this basis we determine the underlying structure in data, construct main relationships between variables, analyze trends, etc. In this regard the role of agent systems or multiagent systems has become highly visible, cf. [2][3][4][14][16][17].

In spite of the diversity of existing methods, some important features manifest themselves throughout a number of developments. First, while a local analysis could be completed by an agent on the basis of data available there, it is desirable to share and reconcile views and findings available from other agents. Second, effective collaborations are better established by communicating findings (knowledge sharing) rather than by sharing data.

Data sharing might not be viable due to limited bandwidth, low energy (which is critical in wireless sensor networks), temporal constraints, or security issues. An alternative is to realize collaboration at the level of entities more abstract than numeric data, that is, information granules. Information granules arise through the application of a fuzzy c-means algorithm, an important abstraction process: we bring individual numeric entities under the same rubric with respect to their closeness, resemblance, spatial ties, and alike [12][18]. Interestingly, communication between humans is predominantly carried out at the level of information granules [12]. We may contemplate information granules (once they are cast in some formal structure) to serve as a generic vehicle to establish communication between individual agents. There is a great deal of literature on information granules and granular computing in general [18], and many approaches have been undertaken to address interactions between agents in problem solving. Intelligent agents and granular computing are concepts concerned with collaborative clustering [6][8] and its variant experience-based modeling. These techniques overlap with distributed modeling [9][13] and fuzzy modeling [10].

This study concentrates on a collaborative scheme involving intelligent agents carrying out collaboration in a so-called horizontal mode. The scheme is outlined in Figure 6.1.

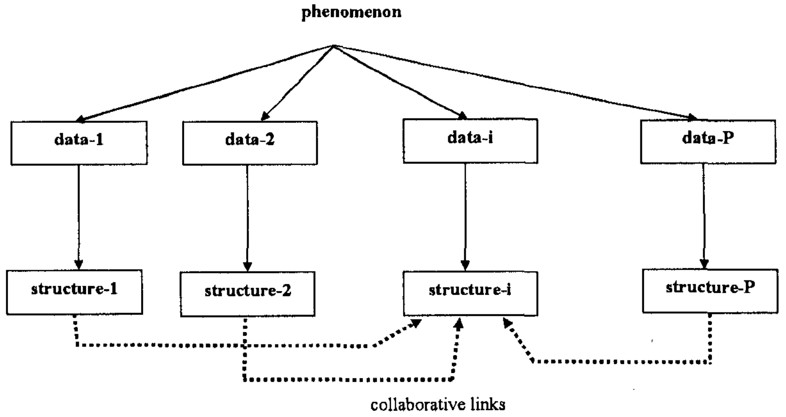


Figure 6.1 A general view of collaboration realized in the horizontal mode

In order to model complex phenomena such as economic processes, societal systems, ecosystems, or biological architectures, we need to look at them from different points of view. The *horizontal* mode of collaboration is an example of such a perspective. To develop a model of a particular phenomenon, we begin by collecting features or attributes into datasets: data-1, D[1], data-2, D[2], ..., data-P, D[P]. These sets of attributes could be common or disjoint. For instance, if our interest is in a description of the socioeconomic structure of a certain region of the world, we would collect various sets of descriptors (features) which offer different views of the country. Note that each view is established in a unique fashion by choosing some attributes. Obviously, we do not know what the most suitable descriptors could be. Bearing in mind the complexity of the phenomenon, it is very likely that various perspectives are quite legitimate and when discovering the structure in the data it would be advantageous to "compare the notes," viz., reconcile the structures being formed from different standpoints. In another example, suppose we want to model a general concept of human health, a phenomenon that is multifaceted and difficult to capture. Individuals could be characterized by the outcomes of medical tests or psychological tests or some other such identifier. The categories of tests (medical, psychological) form spaces of attributes and the entities (individuals) described by these tests constitute the corresponding datasets: data-1, data-2, ..., data-P. In contrast to the previous example, here we are not allowed to share raw data as there are strongly reinforced privacy requirements. Nevertheless, we can engage in collaboration by communicating findings in the form of information granules.

The chapter is arranged in the following manner. The formulation of the problem along with its representation as a certain optimization task is provided in section 6.2. Section 6.3 is devoted to optimizing levels of collaboration; we show that the intensity of interactions significantly affects the quality of collaborative findings. In section 6.4, we present the experimental results and provide additional observations. Conclusions are included in section 6.6.

6.2 Notations and a formulation of the problem

Before presenting a concise statement of the problem, the notation used in the derivations and methods is described.

The notation used in this chapter adheres to literature standards. We use boldface to denote vectors in n-dimensional space, for instance, $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z} \in \mathbf{R}^n$, where $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$, are vectors in space $\mathbf{R}$. The distance function used in the clustering is the standard Euclidean in which the corresponding features are normalized by including the corresponding variance, that is. $\|\mathbf{x} - \mathbf{y}\|^2 = \sum_{j=1}^{n} \frac{(x_j - y_j)^2}{\sigma_j^2}$, where σ_j^2 is the variance of the j-th attribute (feature) of the data. We consider a finite number of datasites, denoted by D[1], D[2], ..., D[P] composed of the same patterns (data) described in different feature spaces $\mathbf{F}_1$, $\mathbf{F}_2$, ..., $\mathbf{F}_P$. Their dimensionality is equal to n[1], n[2], ..., n[P], respectively. For disjoint spaces: $\mathbf{F}_i \cap \mathbf{F}_j = \varnothing$ for $i \neq j$. Overlapping spaces are not necessarily excluded from investigations. While we can reveal the structure of each datasite through fuzzy clustering, we are also interested in a collaborative discovery of structure across all data. If we cannot access certain datasites, findings at a higher conceptual level of information granules will be shared between sites engaged in the collaboration.

Information granules are constructed through fuzzy clustering, and the fuzzy c-means (FCM) algorithm [1][5][7][8]10][12] in particular. In terms of the FCM algorithm for the ii-th dataset D[ii], the resulting structural information is conveyed in the form of the partition matrix U[ii] with c[ii] clusters formed there. The standard objective function minimized by the FCM takes on the form:

$$Q[ii] = \sum_{k=1}^{N} \sum_{i=1}^{c[ii]} u_{ik}^2[ii] \|\mathbf{x}_k - \mathbf{v}_i[ii]\|^2 . \tag{1}$$

Along with the partition matrices, fuzzy clustering produces a collection of prototypes. For the ii-th datasite we have $\mathbf{v}_1[ii]$, $\mathbf{v}_2[ii]$, …, $\mathbf{v}_c[ii]$ with prototypes located in $\mathbf{F}_{ii}$.
Note that the numbers of clusters need not to be the same for all datasites. Once optimization has been completed resulting in the partition matrix and the prototypes, one notes that these two descriptors of the data structure are highly complementary: given the data, we can determine the prototypes on a basis of the partition matrix. Conversely, for the given data and the prototypes provided, we can produce a complete partition matrix.

This situation is portrayed in Figure 6.2 which illustrates that all communication and collaboration occurs at the level of information granules.

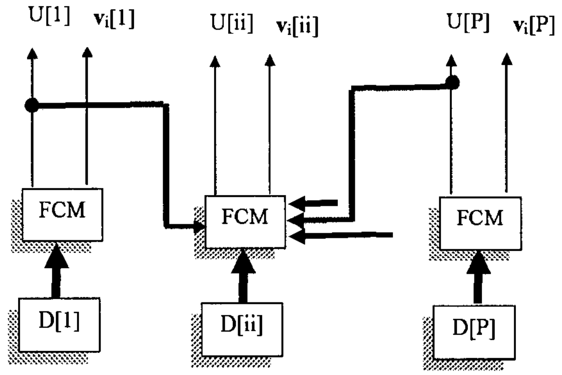


Figure 6.2 Mechanisms of collaboration realized through communication of granular findings (partition matrices).

Having established the methodology, we move on to algorithmic development of the optimization process.

6.2.1 Collaborative clustering as an optimization process

We optimize the collaborative formation of information granules by starting with an augmented objective function and deriving detailed formulas for the partition matrix and the prototypes. For now we assume that the granularity of findings at all datasites is the same, that is, c[1] = c[2] =…= c[P] = c. There are two fundamental ways to communicate findings between datasites; which one is used depends on the format of the objective function.

(a) **scheme-1.** Communication of obtained information is realized in terms of partition matrices. The objective function to be minimized at the ii-th datasite involves these matricies in the overall minimization process:

$$Q[ii] = \sum_{k=1}^{N}\sum_{i=1}^{c} u_{ik}^{2}[ii]\|\mathbf{x}_k - \mathbf{v}_i[ii]\|^2 + \beta\sum_{\substack{jj=1\\ jj\neq ii}}^{P}\sum_{k=1}^{N}\sum_{i=1}^{c}(u_{ik}[ii]-u_{ik}[jj])^2\,\|\mathbf{x}_k-\mathbf{v}_i[ii]\|^2, \qquad (2)$$

where U[jj] = [u_{ik}[jj]] is the partition matrix obtained at the jj-th datasite and made available at the ii-th datasite when searching for the structure in D[ii]. The positive coefficient β is used to establish a tradeoff between forming the structure on the basis of the data being locally available and the structure available at collaborating datasites. Optimization of the collaboration level is discussed in section 6.3.

Optimization details

Mnimization of the objective function Q[ii] is carried out with respect to the fuzzy partition U[ii] and the family of prototypes v_i[ii]. U[jj] is a partition matrix obtained for datasite D[jj]. The partition matrix satisfies the following conditions.

Given the standard identity constraint imposed on the partition matrix, viz. $\sum_{i=1}^{c} u_{ik}[ii] = 1$, in the optimization of (2) we use the Lagrange multipliers. For any data point k, k = 1, 2, ..., N, we reformulate the objective function as:

$$V[ii] = \sum_{i=1}^{c} u_{ik}^2[ii]d_{ik}^2 + \beta \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} \sum_{i=1}^{c} (u_{ik}[ii] - u_{ik}[jj])^2 d_{ik}^2 - \lambda(\sum_{i=1}^{c} u_{ik}[ii] - 1) \tag{3}$$

The necessary conditions for the minimum of V[ii] are expressed as:

$$\frac{\partial V[ii]}{\partial u_{rs}} = 0, \quad \frac{\partial V[ii]}{\partial \lambda} = 0.$$

After computing the derivative with respect to the elements of the partition matrix we obtain:

$$\frac{\partial V}{\partial u_{rs}} = 2u_{rs}[ii]d_{rs}^2 + 2\beta \sum_{\substack{jj=1 \\ jj \neq i}}^{P} (u_{rs}[ii] - u_{rs}[ii]d_{rs}^2) - \lambda = 0, \tag{4}$$

where r = 1, 2, ..., c; s = 1, 2, ..., N.

The detailed calculations are shown below:

$$2u_{rs}[ii]d_{rs}^2 + 2\beta(P-1)u_{rs}[ii]d_{rs}^2 - 2\beta d_{rs}^2 \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} u_{rs}[jj] - \lambda = 0$$

$$u_{rs}[ii]\left(2d_{rs}^2 + 2\beta(P-1)d_{rs}^2\right) = \lambda + 2\beta d_{rs}^2 \sum_{\substack{ii=1 \\ jj \neq ii}}^{P} u_{rs}[jj],$$

$$u_{rs}[ii] = \frac{\lambda + 2\beta d_{rs}^2 \sum_{\substack{jj=1 \\ jj\neq ii}}^{P} u_{rs}[jj]}{2d_{rs}^2[1+\beta(P-1)]}.$$

Finally,

$$u_{rs}[ii] = \frac{\lambda + 2\beta d_{rs}^2 \sum_{\substack{jj=1 \\ jj\neq ii}}^{p} u_{rs}[jj]}{2d_{rs}^2[1+\beta(P-1)]} = \frac{\lambda}{2d_{rs}^2[1+\beta(P-1)]} + \beta \sum_{\substack{jj=1 \\ jj\neq ii}}^{p} u_{rs}[jj] \times \frac{1}{1+\beta(P-1)}. \tag{5}$$

Given the constraint of the form $\sum_{j=1}^{c} u_{js}[ii] = 1$, we obtain:

$$\sum_{j=1}^{c} \frac{\lambda + 2\beta d_{js}^2 \sum_{\substack{jj=1 \\ jj\neq ii}}^{P} u_{js}[jj]}{2d_{js}^2[1+\beta(P-1)]} = 1,$$

$$\sum_{j=1}^{c} \frac{\lambda}{2d_{js}^2[1+\beta(P-1)]} + \sum_{j=1}^{c} \frac{2\beta d_{js}^2 \sum_{\substack{jj=1 \\ jj\neq ii}}^{P} u_{js}[jj]}{2d_{js}^2[1+\beta(P-1)]} = 1. \tag{6}$$

In the sequel we have:

$$\lambda \sum_{j=1}^{c} \frac{1}{2d_{js}^2[1+\beta(P-1)]} = 1 - \sum_{j=1}^{c} \frac{\beta \sum_{\substack{jj=1 \\ jj\neq ii}}^{P} u_{js}[jj]}{[1+\beta(P-1)]},$$

$$\lambda = \sum_{j=1}^{c} 2d_{js}^2[1+\beta(P-1)] \left(1 - \sum_{j=1}^{c} \frac{\beta \sum_{\substack{jj=1 \\ jj\neq ii}}^{P} u_{js}[jj]}{[1+\beta(P-1)]} \right),$$

$$\lambda = \frac{1 - \sum_{j=1}^{c} \frac{\beta \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} u_{js}[jj]}{[1 + \beta(P-1)]}}{\frac{1}{2\sum_{j=1}^{c} d_{js}^{2} [1 + \beta(P-1)]}}.$$

(7)

Hence,

$$\lambda = 2\frac{1 - \frac{\beta \sum_{j=1}^{c} \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} u_{js}[jj]}{[1 + \beta(P-1)]}}{\sum_{j=1}^{c} \frac{1}{d_{js}^{2}}} \langle 1 + \beta(P-1) \rangle.$$

(8)

Plugging (8) into (5) gives:

$$u_{rs}[ii] = \frac{2\frac{1 - \frac{\beta \sum_{j=1}^{c} \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} u_{js}[jj]}{[1 + \beta(P-1)]}}{\sum_{j=1}^{c} \frac{1}{d_{js}^{2}}} [1 + \beta(P-1)]}{2d_{rs}^{2}[1 + \beta(P-1)]} + \frac{2\beta d_{rs}^{2} \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} u_{rs}[jj]}{2d_{rs}^{2}[1 + \beta(P-1)]}.$$

(9)

Further simplifications lead to:

$$u_{rs}[ii] = \frac{1}{\sum_{j=1}^{c} \frac{\left\|x_{s} - v_{r}\right\|^{2}}{\left\|x_{s} - v_{j}\right\|^{2}}} \left[1 - \sum_{j=1}^{c} \frac{\beta \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} u_{js}[jj]}{[1 + \beta(P-1)]} \right] + \frac{\beta \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} u_{rs}[jj]}{[1 + \beta(P-1)]}.$$

(10)

In optimizing the objective function with regard to the prototypes, we consider the Euclidean distance (its weighted version is handled in the same manner). Given the form of the distance, the objective function is written:

$$Q[ii]=\sum_{k=1}^{N}\sum_{i=1}^{c}u_{ik}^{2}[ii]\sum_{j=1}^{n[ii]}(x_{kj}-v_{ij}[ii])^{2}+\beta\sum_{\substack{jj=1\\ jj\neq ii}}^{P}\sum_{k=1}^{N}\sum_{i=1}^{c}(u_{ik}[ii]-u_{ik}[jj])^{2}\sum_{j=1}^{n[ii]}(x_{kj}-v_{ij}[ii])^{2}\,. \tag{11}$$

The necessary condition leading to the minimization of Q[ii] comes in the form:

$$\frac{\partial Q[ii]}{\partial v_{rt}[ii]}=0\,.$$

Then we obtain:

$$\begin{aligned}\frac{\partial Q[ii]}{\partial v_{rt}[ii]}&=-2\sum_{k=1}^{N}u_{rk}^{2}[ii](x_{kt}-v_{rt}[ii])\\ &\quad-2\beta\sum_{\substack{ii=1\\ jj\neq ii}}^{P}\sum_{k=1}^{N}(u_{rk}[ii]-u_{rk}[jj])^{2}\,(x_{kt}-v_{rt}[ii])=0\end{aligned} \tag{12}$$

After further simplifications:

$$\begin{aligned}&v_{rt}[ii]\left\{\sum_{k=1}^{N}u_{rk}^{2}[ii]+\beta\sum_{\substack{jj=1\\ jj\neq ii}}^{P}\sum_{k=1}^{N}(u_{rk}[ii]-u_{rk}[jj])^{2}\right\}\\ &=\sum_{k=1}^{N}u_{rk}^{2}[ii]x_{kt}+\beta\sum_{\substack{jj=1\\ jj\neq ii}}^{P}\sum_{k=1}^{N}(u_{rk}[ii]-u_{rk}[jj])^{2}x_{kt}\end{aligned}\,.$$

Finally,

$$v_{rt}[ii]=\frac{\sum_{k=1}^{N}u_{rk}^{2}[ii]x_{kt}+\beta\sum_{\substack{jj=1\\ jj\neq ii}}^{P}\sum_{k=1}^{N}(u_{rk}[ii]-u_{rk}[jj])^{2}x_{kt}}{\sum_{k=1}^{N}u_{rk}^{2}[ii]+\beta\sum_{\substack{jj=1\\ jj\neq ii}}^{P}\sum_{k=1}^{N}(u_{rk}[ii]-u_{rk}[jj])^{2}}\,, \tag{13}$$

where r = 1, 2, …, c; t = 1, 2, …, n[ii].

(b) **scheme-2.** Communication of the structural findings is again realized in terms of the partition matrix U[jj]. However, once it becomes available at D[ii], we compute the induced prototypes, denoted here by $\mathbf{v}_i$[ii|jj], i = 1, 2, ..., c. Then the objective function to be minimized takes on the form:

$$Q[ii] = \sum_{k=1}^{N}\sum_{i=1}^{c} u_{ik}^2[ii]\|\mathbf{x}_k - \mathbf{v}_i[ii]\|^2 + \beta\sum_{\substack{jj=1\\ jj\neq ii}}^{P}\sum_{k=1}^{N}\sum_{i=1}^{c} u_{ik}[ii]^2 \parallel \mathbf{v}_i[ii] - \mathbf{v}_i[ii \mid jj] \parallel^2. \tag{14}$$

The minimization of (2) and (14) is realized with respect to the partition matrix U[ii] and the prototypes vi[ii].

Optimization Details

Given the standard identity constraint imposed on the partition matrix, the optimization of (14), we confine ourselves to the use of Lagrange multipliers. For any data point k, k =1, 2, ..., N, we expand the objective function to include the constraints:

$$V[ii] = \sum_{k=1}^{N}\sum_{i=1}^{c} u_{ik}^2[ii]\|\mathbf{x}_k - \mathbf{v}_i[ii]\|^2 + \beta\sum_{\substack{jj=1\\ jj\neq ii}}^{P}\sum_{k=1}^{N}\sum_{i=1}^{c} u_{ik}^2[ii]\|\mathbf{v}_i[ii] - \mathbf{v}_i[ii \mid jj]\|^2 - \lambda(\sum_{i=1}^{c} u_{ik}[ii] - 1). \tag{15}$$

The necessary conditions for the minimum V[ii] are expressed as:

$$\frac{\partial V[ii]}{\partial u_{rs}} = 0, \quad \frac{\partial V[ii]}{\partial \lambda} = 0.$$

After computing the derivative with respect to the elements of the partition matrix we obtain:

$$\frac{\partial V[ii]}{\partial u_{rs}} = 2u_{rs}[ii]\|\mathbf{x}_s - \mathbf{v}_r[ii]\|^2 + 2\beta\sum_{\substack{jj=1\\ jj\neq ii}}^{P}(u_{rs}^2[ii]\|\mathbf{v}_r[ii] - \mathbf{v}_r[ii \mid jj]\|^2) - \lambda = 0, \tag{16}$$

where r = 1, 2, ..., c; s = 1, 2, ..., N.

The detailed calculations are shown below:

$$u_{rs}[ii]\left(2\left\|\mathbf{x}_s - \mathbf{v}_r[ii]\right\|^2 + 2\beta\sum_{\substack{jj=1\\ jj\neq ii}}^{P}\left\|\mathbf{v}_r[ii] - \mathbf{v}_r[ii\,|\,jj]\right\|^2\right) = \lambda$$

$$u_{rs}[ii] = \frac{\lambda}{2\left\|\mathbf{x}_s - \mathbf{v}_r[ii]\right\|^2 + 2\sum_{\substack{jj=1\\ jj\neq ii}}^{P}\beta\left\|\mathbf{v}_r[ii] - \mathbf{v}_r[ii\,|\,jj]\right\|^2}.$$

(17)

Given the constraint of the form $\sum_{j=1}^{c} u_{js}[ii] = 1$, we obtain:

$$\sum_{j=1}^{c[ii]}\frac{\lambda}{2\left\|\mathbf{x}_s - \mathbf{v}_j[ii]\right\|^2 + 2\beta\sum_{\substack{jj=1\\ jj\neq ii}}^{P}\left\|\mathbf{v}_j[ii] - \mathbf{v}_j[ii\,|\,jj]\right\|^2} = 1.$$

(18)

In the sequel,

$$\lambda = \sum_{j=1}^{c} 2\left\|\mathbf{x}_s - \mathbf{v}_j[ii]\right\|^2 + \sum_{j=1}^{c} 2\beta\sum_{\substack{jj=1\\ jj\neq ii}}^{P}\left\|\mathbf{v}_j[ii] - \mathbf{v}_j[ii\,|\,jj]\right\|^2.$$

(19)

Plugging (19) into (18) gives:

$$u_{rs}[ii] = \frac{\sum_{j=1}^{c} 2\left\|\mathbf{x}_s - \mathbf{v}_j[ii]\right\|^2 + \sum_{\substack{jj=1\\ jj\neq ii}}^{P}\sum_{j=1}^{c} 2\beta\left\|\mathbf{v}_j[ii] - \mathbf{v}_j[ii\,|\,jj]\right\|^2}{2\left\|\mathbf{x}_s - \mathbf{v}_r[ii]\right\|^2 + 2\beta\sum_{\substack{jj=1\\ jj\neq ii}}^{P}\left\|\mathbf{v}_r[ii] - \mathbf{v}_r[ii\,|\,jj]\right\|^2}.$$

(20)

Further simplifications lead us to:

$$u_{rs}[ii] = \frac{1}{\sum_{j=1}^{c} \frac{\|\mathbf{x}_s - \mathbf{v}_r[ii]\|^2 + \beta \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} \|\mathbf{v}_r[ii] - \mathbf{v}_r[ii \mid jj]\|^2}{\|\mathbf{x}_s - \mathbf{v}_j[ii]\|^2 + \beta \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} \|\mathbf{v}_j[ii] - \mathbf{v}_j[ii \mid jj]\|^2}}. \tag{21}$$

In optimizing the objective function (14) with regard to the prototypes, we consider the Euclidean distance (its weighted version is handled in the same manner). Given the form of the distance, the necessary condition leading to the minimization of Q[ii] is:

$$\frac{\partial Q[ii]}{\partial v_{rt}} = 0.$$

Then we obtain:

$$\frac{\partial Q[ii]}{\partial v_{rt}} = -2\sum_{k=1}^{N} u_{rk}^2[ii](x_{kt} - v_{rt}[ii]) + 2\beta \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} \sum_{k=1}^{N} \sum_{i=1}^{c} u_{rk}^2[ii](v_{rt}[ii] - v_{rt}[ii \mid jj]) = 0. \tag{22}$$

After further simplifications:

$$\sum_{k=1}^{N} u_{rk}^2[ii] x_{kt} = \sum_{k=1}^{N} u_{rk}^2[ii] v_{rt}[ii] + \beta \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} \sum_{k=1}^{N} u_{rk}^2[ii] v_{rt}[ii] - \beta \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} \sum_{k=1}^{N} u_{rk}^2[ii] v_{rt}[ii \mid jj].$$

Finally,

$$v_{rt}[ii] = \frac{\sum_{k=1}^{N} u_{rk}^2[ii] x_{kt} + \beta \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} \sum_{k=1}^{N} u_{rk}^2[ii] v_{rt}[ii \mid jj]}{\sum_{k=1}^{N} u_{rk}^2[ii](1 + \beta(P-1))}, \tag{23}$$

where r = 1, 2, …, c; t = 1, 2, …, n[ii].

6.2.2 The general flow of collaborative processing

The algorithmic derivations presented so far are now embedded in the organization of the overall collaboration process (refer to Figure 6.3). Two underlying processes are run consecutively. Fuzzy clustering is first run independently at each datasite for a certain number iterations. The stopping criterion is the one typically used in the FCM algorithm, namely, we monitor the changes in the values of the partition matrices obtained in the consecutive iterations and terminate the process when the Tchebyshev distance between the partition matrices does not exceed a certain predefined threshold ε; say, $\max_{i,k} |u_{ik}(\text{iter}+1) - u_{ik}(\text{iter})| < \varepsilon$ with $u_{ik}(\text{iter})$ being the (i, k)th entry of the partition matrix obtained at the iteration "iter."

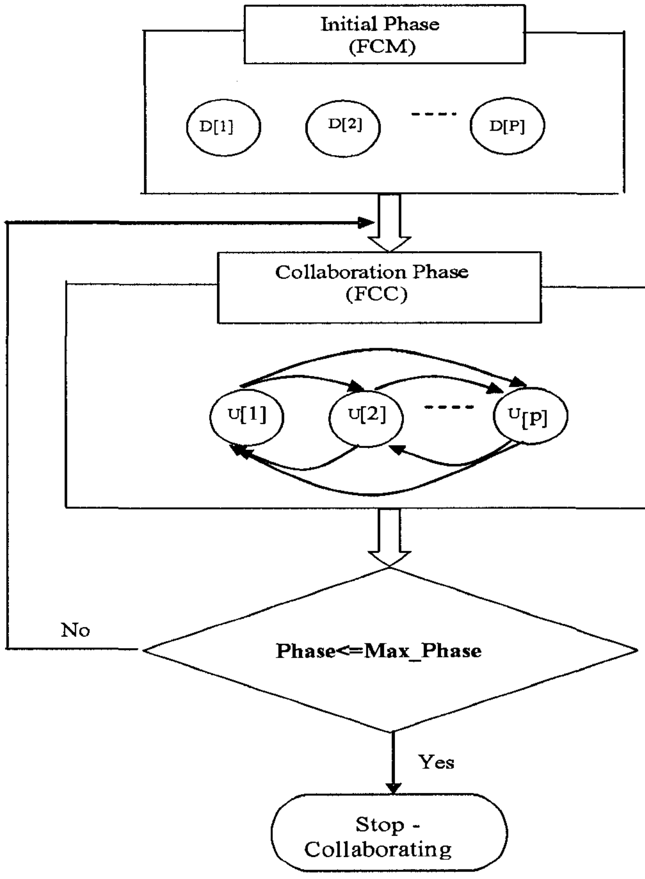


Figure 6.3 An overall process of collaborative clustering underlining two phases of clustering realized at the level of individual datasites(initial phase) and exchange of structural findings in the form of partition matrices(collaboration phase).

At this point datasites exchange findings by transferring partition matrices (illustrated in Figure 6.3). Afterward an iterative process realizes the minimization of (2) or (14). Again when convergence is reported, the results (partition matrices) are exchanged (communicated) between the datasites and the iterative computing of the partition matrices and the prototypes resumes.

6.3 Evaluation of the quality of collaboration

The quality of the results of the collaboration between datasites requires careful assessment. As there are partition matrices associated with each of the D[ii]s, one could think of treating the distances between them as a measure of the quality of the ongoing process of structural reconciliation. However, there may not be direct correspondence between the rows (respective clusters) of these matrices [6]; therefore, a direct comparison of two partition matrices may not be feasible. In a more general setting we might have different numbers of clusters at individual datasites and this could also thwart attempts to form a correspondence between partition matrices. Instead, to test the quality of our collaboration results, we test how the structure revealed at one datasite performs on the remaining ones. Consider the following expression:

$$Q[ii \mid jj] = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{m}[jj]\|\mathbf{x}_k - \mathbf{v}_i[jj]\|^2, \tag{24}$$

where $\mathbf{v}_i[jj]$ is induced by the fuzzy partition matrix U[jj] using data at D[ii]. Similarly $\mathbf{x}_k \in D[ii]$. In a nutshell, Q[ii|jj] expresses how well the structure revealed at D[jj] performs for D[ii]. By using the structure revealed at D[1], D[2], ..., D[ii-1], D[ii+1], ..., D[P] we compute the following expression:

$$W[ii] = \sum_{\substack{jj=1 \\ jj \neq ii}}^{P} Q[ii \mid jj], \tag{25}$$

which aggregates the performance of all other structures on the datasite D[ii]. Finally, we sum W[ii]s,

$$W = W[1] + W[2] + \ldots + W[P]. \tag{26}$$

This index can be regarded as an overall measure of effectiveness of collaboration. The smaller the value of W, the higher the effectiveness of collaboration. As this index is a function of β, and W = W(β), we are provided with a vehicle to optimize the strength of collaboration. Intuitively, we may observe that higher values of β imply stronger interactions between datasites. While this is helpful in reconciling the differences, too high values of β might not be suitable if there is significant structural variability between datasites leading to high values of W.

6.4 Different levels of information granularity

The collaboration procedures presented so far are quite restrictive in the sense that we have assumed that the number of clusters at each of the collaborating datasites is the same. While this is a viable alternative if all collaborating parties agree in advance on the level of granularity to be realized, in general, this assumption is quite restrictive and unrealistic. A far more flexible scenario is the one in which each party considers its own number of clusters (which could be quite legitimate considering that data structure could vary from site to site). Given this, the algorithmic settings have to be augmented. The major step would be to present information granules at each datasite at the level of granularity which has been accepted before collaboration. A viable alternative is to construct prototypes on a basis of the available partition matrices, cluster them, and use the prototypes obtained in this manner in the minimization of the associated objective functions. Proceeding with the details, let us consider the ii-th datasite. The partition matrices available at all other datasites U[1], U[2], ..., U[P] are used in the formation of the prototypes using the patterns at D[ii]. We obtain:

$\mathbf{v}_1[1], \mathbf{v}_2[1], \ldots, \mathbf{v}_{c[1]}[1]$ for the partition matrix U[1],
$\mathbf{v}_1[2], \mathbf{v}_2[2], \ldots, \mathbf{v}_{c[2]}[2]$ for the partition matrix U[2],
...

$\mathbf{v}_1[P], \mathbf{v}_2[P], \ldots, \mathbf{v}_{c[P]}[P]$ for the partition matrix U[P].

Overall, the number of induced prototypes along with the original prototypes available at D[ii] is equal to c[1] + c[2] +...+c[P]. We cluster them into c[ii] clusters so the level of granularity applied here is consistent with the number of clusters formed for D[ii]. Denote the resulting prototypes by $\mathbf{v}_1^{\sim}[ii], \mathbf{v}_2^{\sim}[ii], \ldots, \mathbf{v}_{c[ii]}^{\sim}[ii]$. These new prototypes are used in the following augmented objective function:

$$Q[ii] = \sum_{k=1}^{N} \sum_{i=1}^{c[ii]} u_{ik}^2[ii] \|\mathbf{x}_k - \mathbf{v}_i[ii]\|^2 + \beta \sum_{k=1}^{N} \sum_{i=1}^{c[ii]} u_{ik}^2[ii] \|\mathbf{v}_i[ii] - \mathbf{v}_i^{\sim}[ii]\|^2. \tag{27}$$

We follow the optimization scheme of different levels of information granularity for eq(27) as discussed in Chapter 5 ; section 5.8, where we reported the following partition matrix and prototypes:

$$u_{rs}[ii] = \frac{1}{\sum_{j=1}^{c[ii]} \frac{\|\mathbf{x}_s - \mathbf{v}_r[ii]\|^2 + \beta \|\mathbf{v}_r[ii] - \mathbf{v}_r^{\sim}[ii]\|^2}{\|\mathbf{x}_s - \mathbf{v}_j[ii]\|^2 + \beta \|\mathbf{v}_j[ii] - \mathbf{v}_j^{\sim}[ii]\|^2}}, \tag{28}$$

where r = 1, 2, ..., c; s = 1, 2, ..., N.

And

$$v_{rt}[ii] = \frac{\sum_{k=1}^{N} u_{rk}^{2}[ii]x_{kt} + \beta \sum_{k=1}^{N} u_{rk}^{2}[ii]v_{rt}^{\sim}[ii]}{\sum_{k=1}^{N} u_{rk}^{2}[ii](1+\beta)}, \qquad (29)$$

where as, r = 1, 2, …, c; t = 1, 2, …, n[ii].

The overall flow of processing can be outlined as Table 6.1 .

Table 6.1 The flow of collaborative clustering showing the main optimization phases and underlining the mechanism of communication in the form of exchange of different level of granulation at each datasite.

Given: datasites D[1], D[2], …, D[P] with different structures, viz., levels of granularity. Select a number of clusters (c[ii]) for each datasite; set up some termination criterion and establish a level of collaboration (interaction) by choosing a nonnegative value of β (it could be optimized as discussed in section 6.3).

Initial phase
Carry out clustering (FCM) for each datasite producing a collection of prototypes $\{\mathbf{v}_i[ii]\}$, i = 1, 2, …, c[ii] for each datasite.

Collaboration
Iterate {successive phases of collaboration}

Communicate the results about the structure determined at each datasite.

For each datasite (ii)
{

Collect all prototypes from other sites at datasite (ii) and run FCM on that collection of all prototypes by selecting the same number clusters at that site to generate new prototypes $\mathbf{v}^{\sim}[ii]$. Minimize (27) at each datasite by iteratively proceeding with the iterative calculations of the partition matrix ($u_{rs}[ii]$) and the prototypes (v_{rt} [ii]) using (28) and (29), respectively.

r = 1, 2, …, c[ii]; t = 1, 2, …, n[ii]; s = 1, 2, …, N

} for the datasite

until termination condition of the collaboration activities has been satisfied.

The quality of collaboration is optimized by choosing a suitable value of β (which minimizes the performance index W given by (26)).

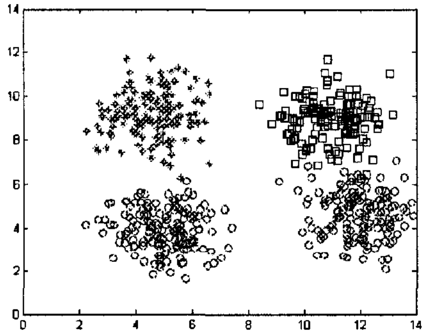
6.5 Experimental studies

In this section, we report experimental findings for synthetic and machine learning datasets (http://mlearn.ics.uci.edu/MLRepository.html).

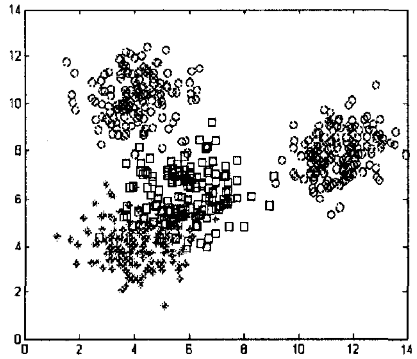
Synthetic data. For illustrative purposes we consider two-dimensional synthetic datasets with five datasites (denoted here as datasite-1, datasite-2, ..., datasite-5) as shown in Figure 6.4. Each datasite consists of 600 patterns. These data form a mixture of several Gaussian distributions with the following mean vectors:

datasite-1:	v_1=[4.5 9.2]	v_2= [11.0 9.0]	v_3 = [5.0 4]	
	v_4=[12 4.5]			
datasite-2:	v_1=[4.0 4.2]	v_2=[6.0 6]	v_3 = [4 10.2]	
	v_4=[11.5 8]			
datasite-3:	v_1=[4.5 6.0]	v_2= [7.0 9.0]	v_3 = [10.0 6.0]	
datasite-4:	v_1=[6.0 4.0]	v_2= [6.0 10.0]	v_3 = [10.0 10.0]	
datasite-5:	v_1=[4.3 10.0]	v_2= [11.0 9]	v_3 = [5 3.5]	v_4=[11.2 4.0]
	v_5=[2.0 6.5]			

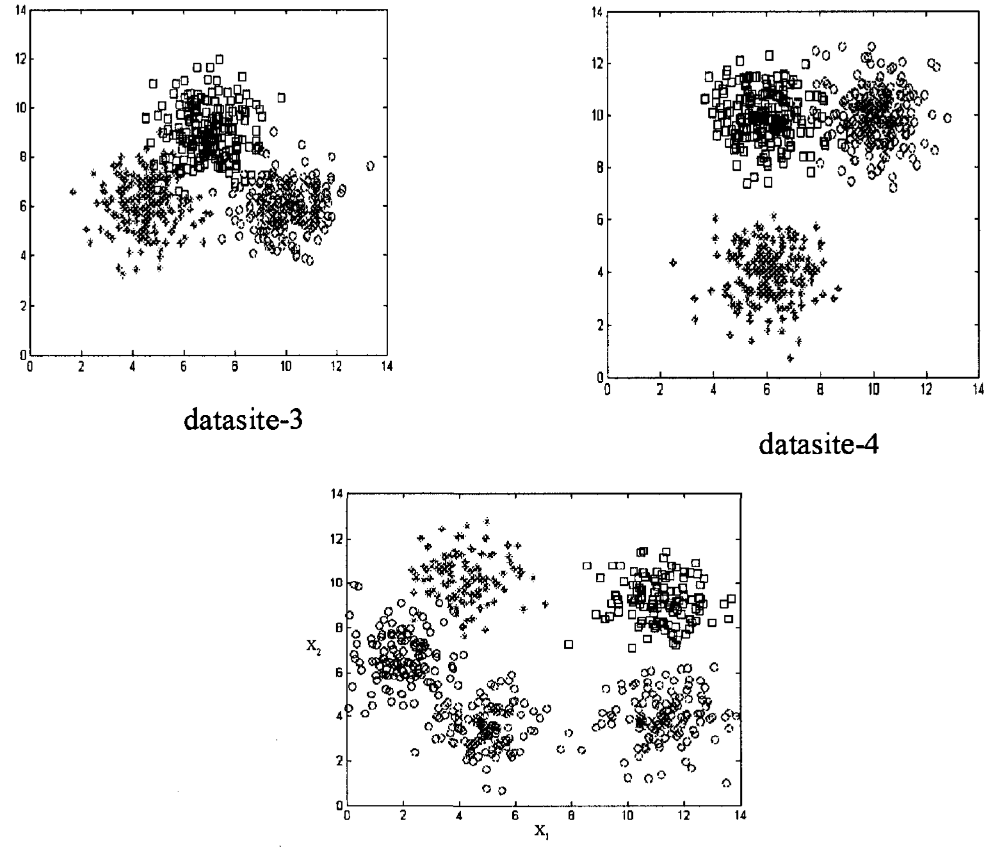
The covariance matrix of all Gaussian components is equal to $\begin{bmatrix}1.0 & 0.0\\ 0.0 & 1.0\end{bmatrix}$ and it is the same for the groups of all datasites. Different distribution of the centers of the groups results in quite different topologies of the datasites. Let us stress that at each datasite the individual data points are described by the variables specific to the given datasite as visualized in Figure 6.4. This underlines the multifaceted view of the data, a part of which is captured by each datasite.



datasite-1



datasite-2



datasite-3

datasite-4

datasite-5

Figure 6.4 Two-dimensional synthetic datasets used in the collaboration process; note that the mean vectors and covariance matrices vary significantly across datasites.

Machine learning datasets. Here we consider the Wine and Boston Housing datasites. The Wine dataset consists of 178 data items described by 13 features (attributes). For the Boston Housing dataset we have 506 13-dimensional data. We form some datasites in which there are groups of the original features which allow us to assume a certain point of view of the data. For the Wine data:

datasite-1: data are described from the standpoint of strong/acidic compounds: alcohol, malic acid, ash, and proline.

datasite-2: data are concerned with healthy compounds: magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins.

datasite-3: the attributes used here pertain to color intensity, hue, OD280/OD315 of diluted wines, proline, and proanthocyanins.

For the Boston Housing data we consider 3 datasites formed by the following attributes:

datasite-1: It is concerned with a general characterization of surroundings and uses the following attributes:
1. per capita crime rate by town
2. proportion of residential land zoned for lots over 25,000 sq.ft.
3. proportion of nonretail business acres per town.
4. concentration of nitric oxides (parts per 10 million)
5. median value of owner-occupied homes in $1000s

datasite-2: Here the focus is on convenience aspects related to the following attributes:
1. average number of rooms per dwelling
2. proportion of owner-occupied units built prior to 1940
3. weighted distances to five Boston employment centers
4. index of accessibility to radial highways
5. median value of owner-occupied homes in $1000s

datasite-3: The attributes available at this datasite deal with economic aspects:
1. full-value property-tax rate per $10,000
2. pupil-teacher ratio by town
3. 1000 (Bk - 0.63)^2 where Bk is the proportion of blacks by town
4. percent lower status of the population
5. median value of owner-occupied homes in $1000s

The collaboration was completed for a suite of parameters. We varied the granularity of information granules by changing the number of clusters from 2 to 7. There were 20 phases of collaboration (exchange of findings), while in-between these exchanges the clustering algorithm was run for 80 iterations. Two aspects of the collaboration are of particular interest: the optimal values of β and the values of W in successive phases of collaboration which reflect the dynamics of the collaboration. The pertinent experimental findings are plotted in Figure 6.5. In this case we have implemented scheme 1 of collaboration as described by (2).

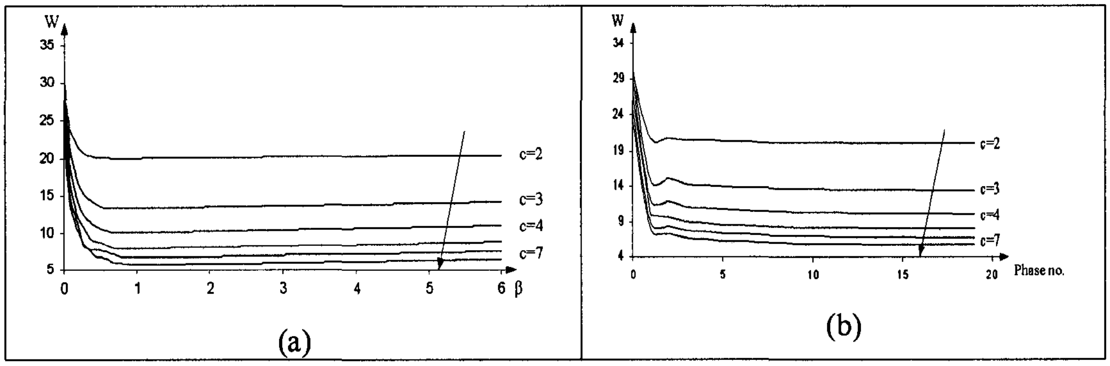


Figure 6.5 Plot of W treated as a function of β for different values of c (a), and the dynamics of W reported over successive phases of collaboration—these results are reported for optimal values of β (b).

We note that when dealing with different levels of information granularity, the optimal values of β are around 1. Furthermore, the values of W do not change substantially with an increase in the value of β. In all cases the collaboration was beneficial as the values of W for β = 0, W(0), were higher than those reported for nonzero values of β. The dynamics of the collaboration process are depicted in Figure 6.5 (b). The figure shows that for the optimal value of β the values of W were reduced monotonically with the most substantial drop reported in the first few phases of the collaboration. Slight oscillations in the value of W are noticeable at the beginning of the collaboration. The findings are summarized in Table 6.2.

Table 6.2 Summary of collaboration quantified in terms of the performance index W.

c	2	3	4	5	6	7
β_{opt}	0.8	0.8	0.9	0.85	1.05	1.05
W (0)	29.267	29.996	28.449	26.586	24.583	22.950
W (20)	19.992	13.372	10.090	8.038	6.748	5.783

For the Wine data we report the values of W treated as a function of β and show the dynamics of W in successive phases of collaboration. As reported for the Boston Housing analysis, W does not seem to be very dependent on β; however, the collaboration is beneficial (Table 6.3). The dynamics of the Wine dataset collaboration experiment are similar to those of the Boston Housing dataset.

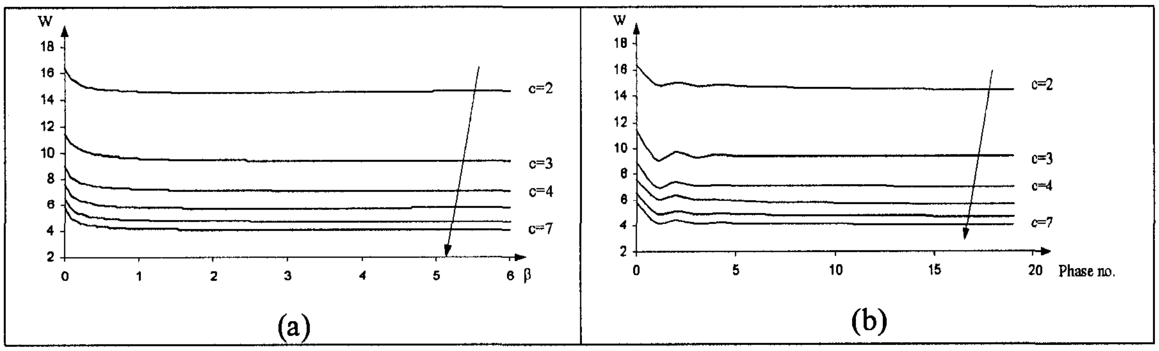


Figure 6.6 Plot of W treated as a function of β for different values of c (a), and the dynamics of W reported over successive phases of collaboration—the results are reported for optimal values of β (b).

Table 6.3 Summary of collaboration quantified in terms of the performance index W.

c	2	3	4	5	6	7
β_{opt}	2.2	6	4.45	2.85	4.1	3.65
W (0)	16.370	11.422	8.985	7.632	6.550	5.834
W (20)	14.465	9.370	7.081	5.705	4.760	4.091

It is interesting to see how much the collaboration became reflected in the results of clustering itself, such as prototypes. To visualize this effect, Figure 6.7 provides a radar plot of the prototypes before and after collaboration.

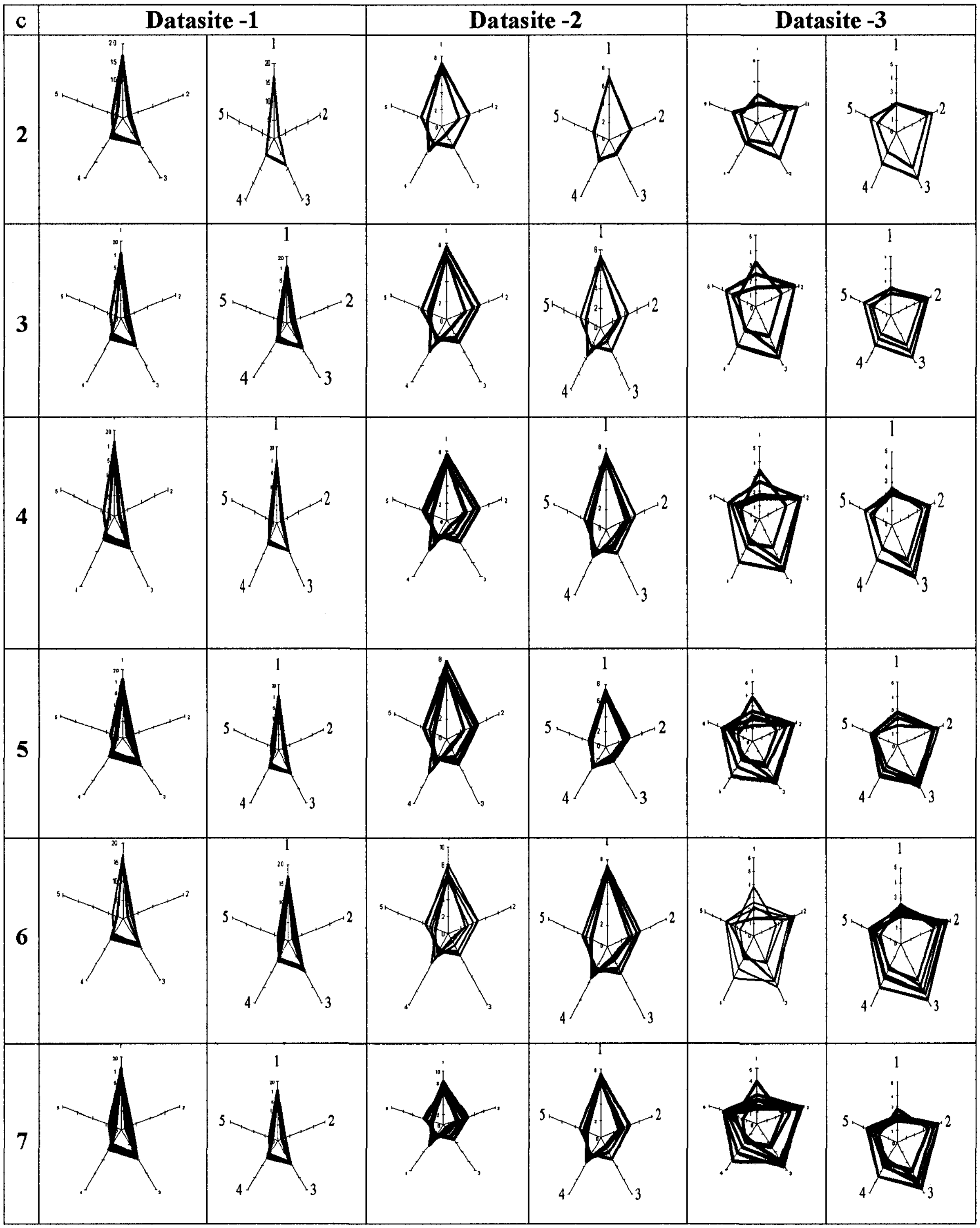


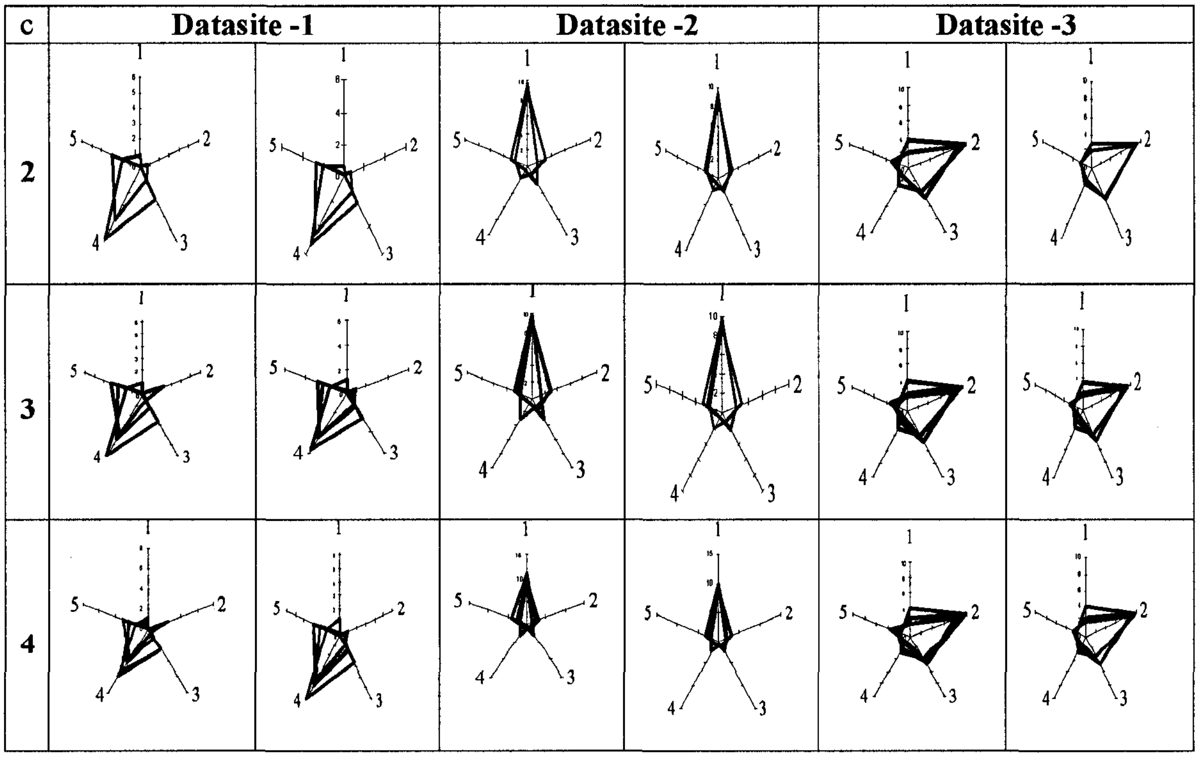
Figure 6.7 Radar plots of prototypes obtained for datasite-1, datasite-2, and datasite-3. Consecutive columns refer to the results before and after collaboration,Wine

dataset. The results for different levels of granularity c = 2, 3, ..., 7, are shown in successive rows.

The qualitative aspects of the Boston Housing data collaboration are similar to the results for the Wine data. However, the optimal values of β are higher and located in the range of 3–12 for Boston Housing. The dynamics of collaboration exhibits a quick and smooth convergence. In all cases, we note that there is a reduction of the performance index W which points to a positive effect of collaboration established by the sites. In all cases, the reduction in the values of W (comparing W(20) with W(0)) is in the range of 8% to 34%. The resulting radar plots are shown in Figure 6.8.

Table 6.4 Summary of collaboration quantified in terms of the performance index W.

c	2	3	4	5	6	7
β_{opt}	12	9	4.5	3.4	2.95	4.35
W (0)	14.964	10.767	8.625	7.576	6.606	5.955
W (20)	13.733	8.969	6.783	5.483	4.584	3.915



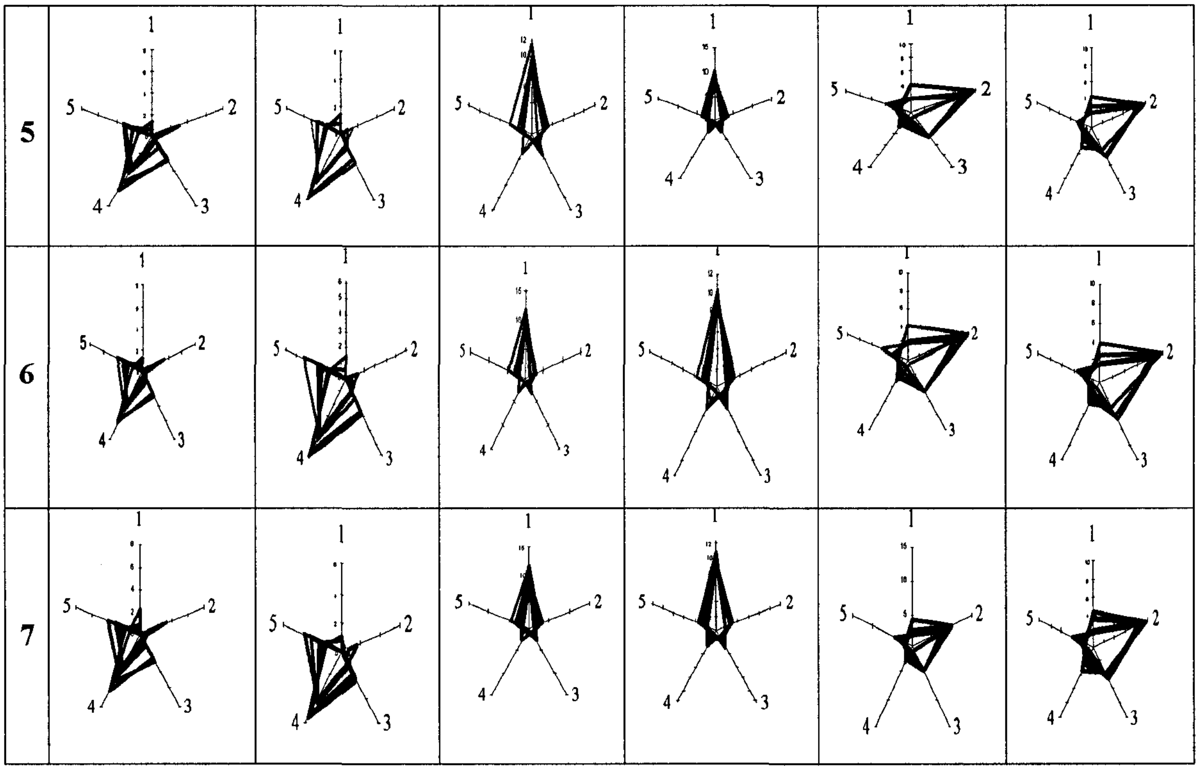


Figure 6.8 Radar plots of prototypes obtained for datasite-1, datasite-2, and datasite-3. Consecutive columns refer to the results before and after collaboration, Boston Housing dataset. Results for different levels of granularity c = 2, 3, ..., 7, are shown in successive rows.

We proceed now with the scheme 2 of collaboration. Starting with the synthetic data, there is a substantial reduction in the value of the performance index W as reported in Table 6.5. The dynamics of collaboration are the same as reported in scheme 1. The optimal value of β is 0.4 and the quality of collaboration is quite insensitive to β as visualized in Figure 6.9 (a).

Table 6.5 Summary of collaboration reported in terms of the performance index W and optimal values of β.

c	2	3	4	5	6	7
β_{opt}	0.4	0.4	0.4	0.4	0.4	0.4
W (0)	29.26	30.00	28.45	26.62	24.58	22.93
W (20)	19.97	13.31	9.98	7.99	6.66	5.70

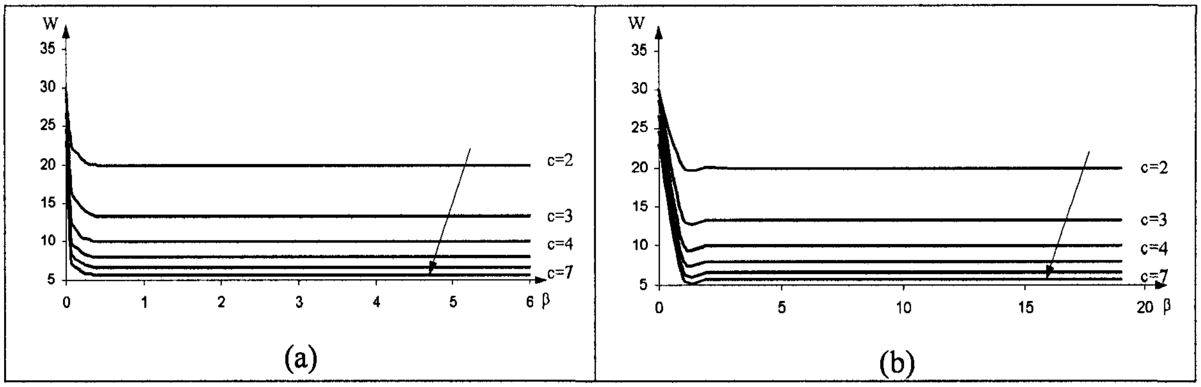
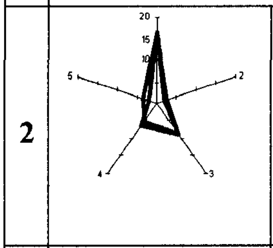
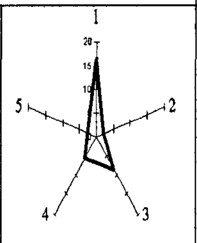
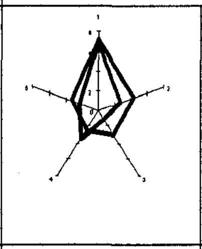
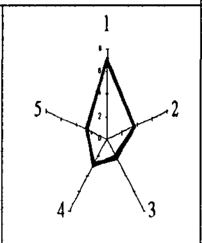
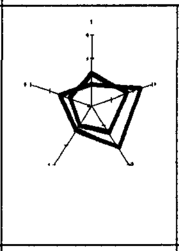
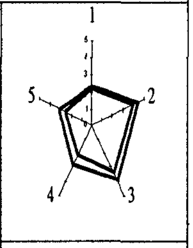
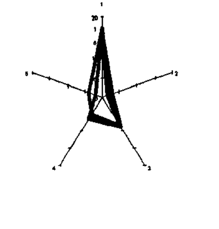
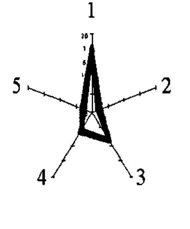
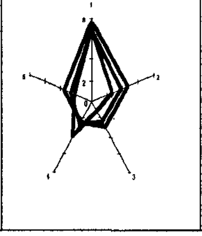
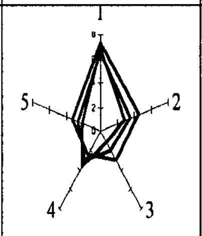
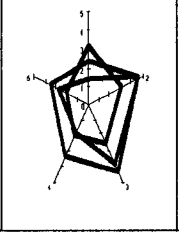
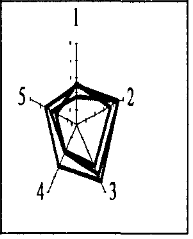


Figure 6.9 Plots of W treated as a function of β for different values of c (a), and the dynamics of W reported over successive phases of collaboration—the results reported are for optimal values of β (b).

Results of the Wine data analysis are presented in the same way as the results of the Boston Housing data analysis. We report the values of W treated as a function of β and show the dynamics of W in successive phases of collaboration. The main trend for the Wine data is similar to that of the Boston Housing data; i.e., W does not seem to be very dependent on β. The collaboration effect is beneficial and its effect is quantified in Table 6.6.

Table 6.6 Summary of collaboration quantified in terms of the performance index W and optimal values of β.

c	2	3	4	5	6	7
β_{opt}	0.85	2.7	2.3	3.85	0.05	0.5
W (0)	16.37	11.42	9.02	7.63	6.57	5.85
W (20)	14.92	10.39	7.72	6.50	5.45	4.50

c	Datasite -1		Datasite -2		Datasite -3	
2						
3						

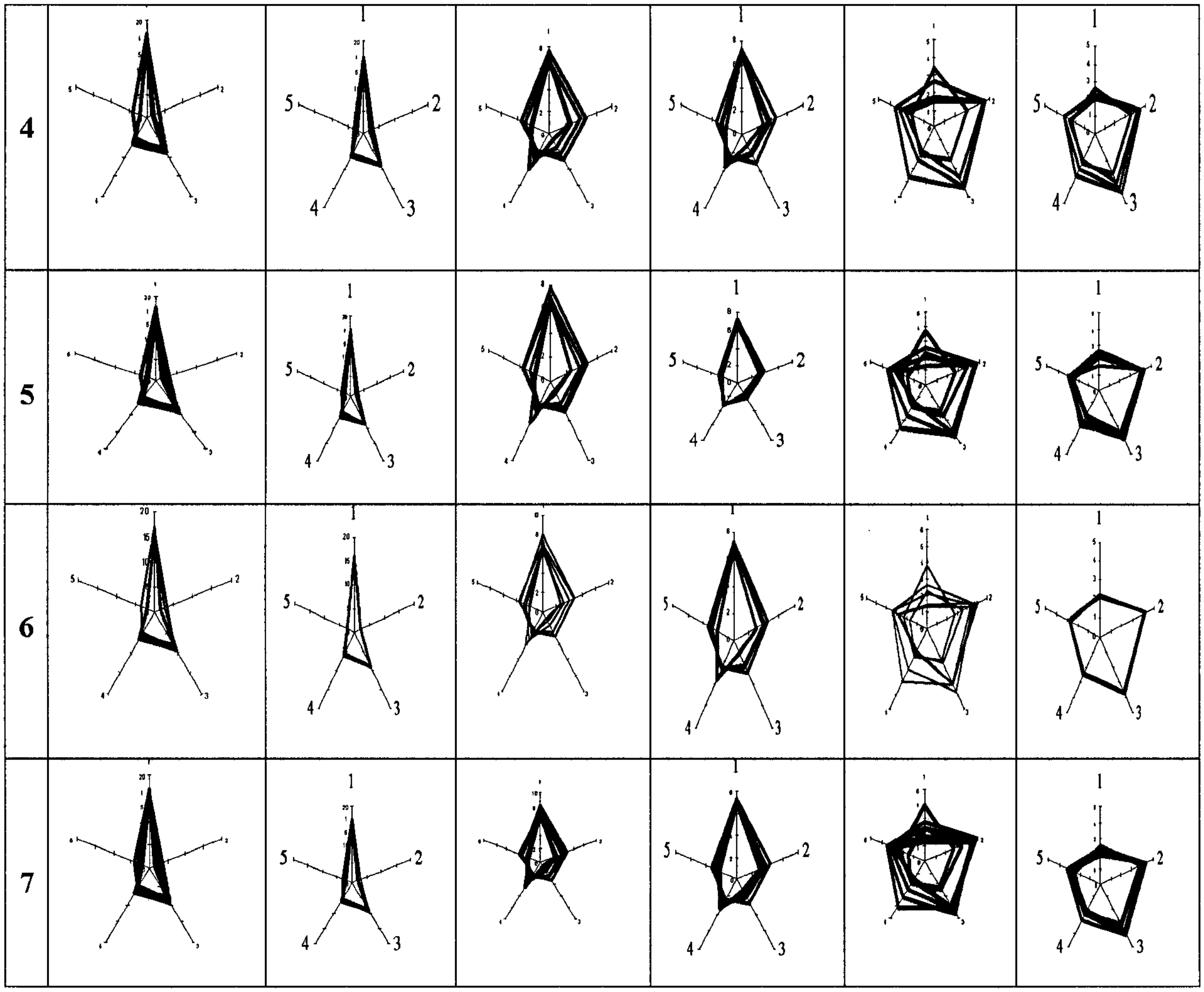


Figure 6.10 Radar plots of prototypes obtained for datasite-1, datasite-2, and datasite-3. Consecutive columns refer to the results before and after collaboration. The results for different levels of granularity c = 2, 3, ...,7, are shown in successive rows.

For the Boston Housing data, the qualitative aspect of the collaboration is the same as for the Wine data. However, the optimal values of β for Boston Housing data are higher, located in the range of 0–6.

Table 6.7 Summary of collaboration quantified in terms of the performance index W.

c	2	3	4	5	6	7
β_{opt}	0	3.85	6	5	0.55	2.9
W (0)	14.96	10.77	8.63	7.58	6.62	6.01
W (20)	14.96	10.67	7.66	6.05	5.07	4.34

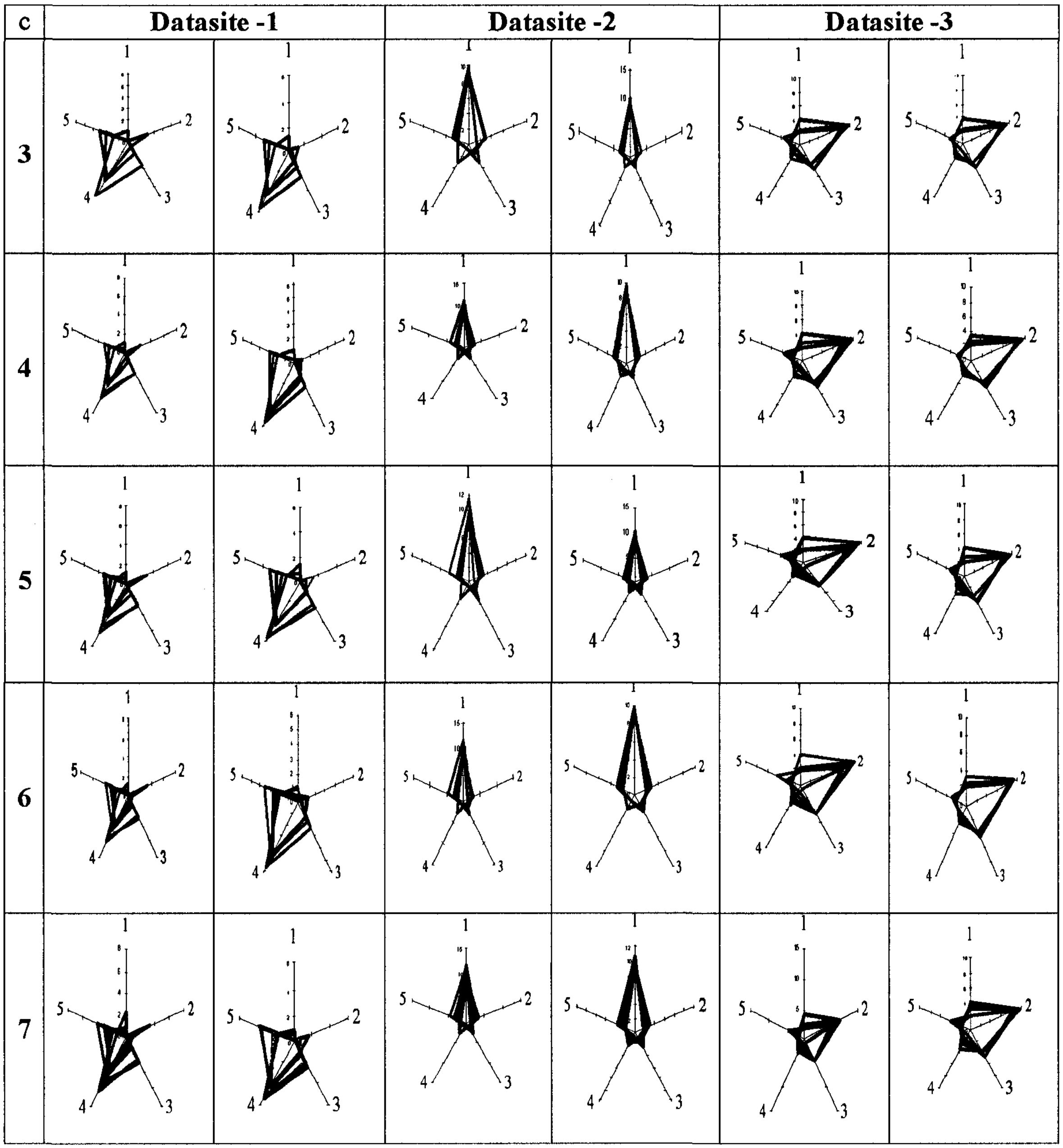


Figure 6.11 Radar plots of prototypes obtained for datasite-1, datasite-2, and datasite-3. Consecutive columns refer to the results before and after collaboration. The results for different levels of granularity c = 2, 3, ..., 7 are shown in successive rows.

For the synthetic data (Figure 6.4), where we consider a scenario in which the level of granularity varies between datasites. More specifically, we consider that c[1] = 4, c[2] = 3, c[3] = 3, c[4] = 3, c[5] = 5. Variation in levels of granularity produces qualitative changes in the nature of W. We note some oscillations in the values of W with respect to β. Furthermore, the optimal value of β is substantially lower ($\beta = 0.06$).

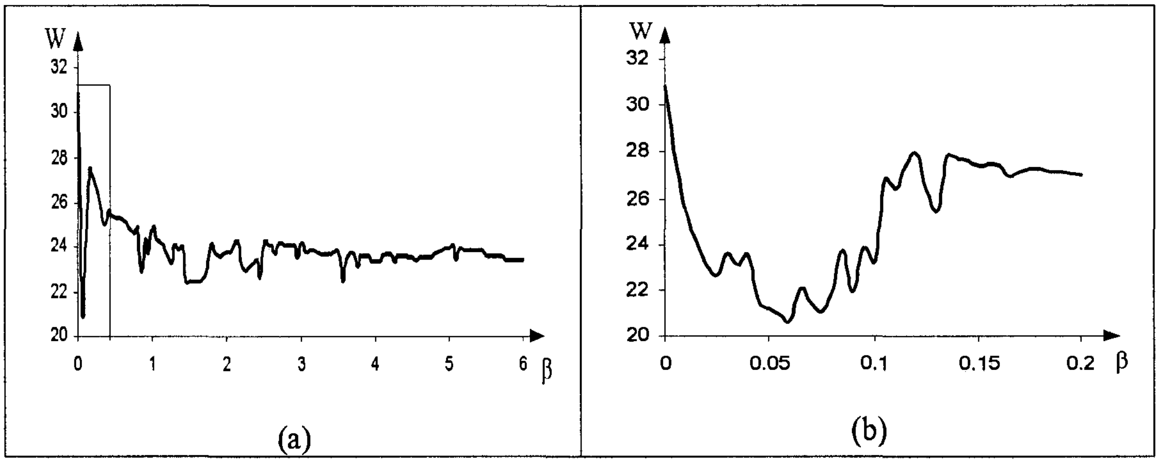


Figure 6.12 Plot of W treated as a function of β for different values of c: c[1] = 4, c[2] = 3, c[3] = 3, c[4] = 3, c[5] = 5 (a), and the more detailed plot of W for lower range values of β (b).

The distribution of prototypes is illustrated in Figure 6.13. Given the low level of optimal collaboration, we note that the most significant changes in the location of the prototypes are visible in datasite-5 where one prototype is significantly moved around to become consistent with the structure revealed from the perspective of other datasites.

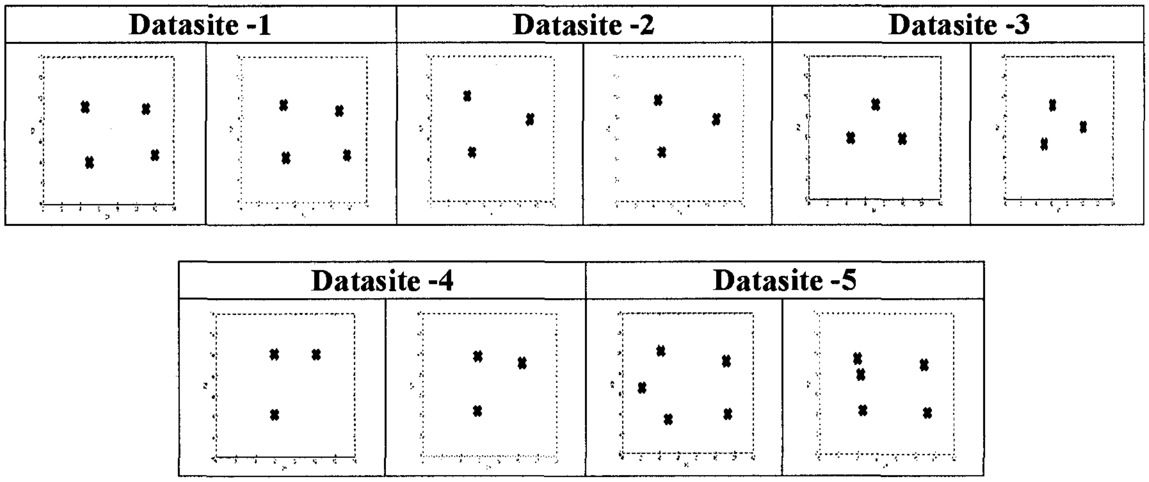


Figure 6.13 Distribution of prototypes in datasites before and after collaboration; β = 0.06.

6.6 Conclusions

Building a global and coherent view of complex systems (phenomena) has always been an ongoing challenge. This study offers an approach to this problem by presenting a way of reconciling local findings through collaborative clustering in which information granules are exchanged and actively engaged in further structuralization of data by means of fuzzy clustering. While the FCM algorithm, commonly used in information granulation, has been used to demonstrate detailed algorithmic aspects, the proposed framework can be adopted to deal with any objective function-based fuzzy clustering.

In all cases tested, experiments revealed that collaboration leads to a higher consistency of results (quantified in terms of the performance index), however, there are differences in terms of the intensity of collaboration. Interestingly, in a number of cases the choice of the intensity parameter β does not seem to be overly critical and the performance of the collaborative environment is retained over a relatively large range of values of β.

Two open design aspects deserve further attention. One is concerned with more levels of collaboration—this reflects the willingness of agents to collaborate (e.g., by accepting findings coming from others and taking them into consideration when running clustering for the agent's local data). This means that instead of a single scalar quantity β we may envision individual coefficient β_{ij} for each pair of agents. Note that no symmetry is required so one might have different values of β_{ij} and β_{ji}. The optimization of these coefficients needs further investigation. Coefficients could also be made time dependent allowing a variable collaboration in consecutive phases. The other issue is related to the aggregation of findings at different datasites when the collaboration has been completed. This may result in higher order granular constructs such as type-2 fuzzy sets. Fuzzy partition matrices are comprised of fuzzy set entries rather than the single numeric membership grades employed in this study.

References

1. J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, N. York, 1981.

2. P. K. Biswas, H. Qi, Y. Xu, Mobile-agent-based collaborative sensor fusion, *Information Fusion*, 2007, to appear.

3. M. Chau, D. Zeng, H. Chen, M. Huang, D. Hendriawan, Design and evaluation of a multi-agent collaborative Web mining system, *Decision Support Systems*, 35, 1, 2003, 167- 183.

4. J. Costa da Silva, M. Klusch, Inference in distributed data clustering, *Engineering Applications of Artificial Intelligence*, 19, 2006, 363-369.

5. A. Jain, M. Murt, P. Flynn, Data clustering: a review, *ACM Computing Surveys*, 31, 1999, 264-323.

6. E. Johnson, H. Kargupta, Collective hierarchical clustering from distributed, heterogeneous data, *Large-Scale Parallel KDD Systems*, Lecture Notes in Computer Science, no. 1759, Springer-Verlag, London, UK, 1999, 221-244.

7. F. Hoppner et al., *Fuzzy Cluster Analysis*, J. Wiley, Chichester, England, 1999.

8. V. Loia, W. Pedrycz, S. Senatore, P-FCM: a proximity-based fuzzy clustering for user-centered web applications, *Int. J. of Approximate Reasoning*, 34, 2003, 121-144.

9. S. Merugu, J. Ghosh, A privacy-sensitive approach to distributed clustering, *Pattern Recognition Letters*, 26, 2005, 399-410.

10. S. Mitra, Y. Hayashi, Neuro-fuzzy rule generation: survey in soft computing framework, *IEEE Transaction on Neural Networks*, 11, 3, 2000, 748-768.

11. W. Pedrycz, Collaborative fuzzy clustering, *Pattern Recognition Letters*, 23, 2002, 675-686.

12. W. Pedrycz, *Knowledge-Based Clustering: From Data to Information Granules*, J. Wiley, Hoboken, NJ, 2005.

13. W. Pedrycz, Distributed fuzzy systems modeling, *IEEE Transactions on Systems, Man, and Cybernetics*, 25, 1995, 769-780.

14. J. L. Posadas, J. L. Poza, J. E. Simó, G. Benet, F. Blanes, Agent-based distributed architecture for mobile robot control, *Engineering Applications of Artificial Intelligence*,2007, to appear.

15. A. Topchy, K. Jain, W. Punch, Clustering ensembles: models of consensus and weak partitions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 2005, 1866-1881.

16. J. Tweedale, N. Ichalkaranje, C. Sioutis, B. Jarvis, A. Consoli, G. Phillips-Wren, Innovations in multi-agent systems, *Journal of Network and Computer Applications*, 30, 3, 2007, 1089-1115.

17. T. W. Wang, S. K. Tadisina, Simulating Internet-based collaboration: A cost-benefit case study using a multi-agent model, *Decision Support Systems*, 43, 2, 2007, 645-662.

18. L. A. Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*, 90, 1997, 111-117.

Chapter 7

Experience-Consistent Modeling: Regression and Classification Problems

This chapter develops an extension of two traditional statistical method of modeling and pattern classification. With regard to the usage of data, we encounter several fundamental problems that require careful attention. The previously available datasets could be substantially larger, meaning that relying on the models formed in the past could be beneficial for the development of the current model.

The experimental studies presented include some synthetic data, selected datasets from the Machine Learning Repository and census housing data from the Statistical Library.

7.1 Introduction

System modeling is inherently associated with an intensive and prudent usage of experimental data. In spite of the evident diversity of available models, all of them exploit the existing data in order to establish a structure of the respective model and estimate its parameters. This general observation applies equally well to regression models [5][6][8][10][14][15], neural networks, cognitive maps, rule-based systems, fuzzy systems [2][9][12][13][17] and alike. Generalization capabilities of the models rely in a direct manner on the characteristics of data (in particular their representative capabilities with respect to the problem at hand) and the nature of the model itself. The characteristics of data deserve particular attention in case we encounter small data sets that could be also biased by some noise. The models developed on a basis of the limited and noisy dataset typically exhibit low prediction capabilities. A certain alleviation of the problem of this nature could be realized by contemplating reliance on other sources of knowledge about the system to be modeled, especially where they might have been acquired in the past. They are not necessarily data themselves (whose accessibility could be limited to various reasons) but could be available in the format of the parameters of the models. In the anticipated modeling scenario, it becomes advantageous not only to consider currently available data but also to actively exploit previously obtained findings. Such observations bring us to the following formulation of the problem:

> Given some experimental data, construct a model which is *consistent* with the findings (models) produced for some previously available data. Owing to the existing requirements, such as privacy or security of data, as well as some other technical limitations in the construction of the model an access to these previous data is not available. However, we can take advantage of the knowledge of the parameters of the existing models.

Considering the need to achieve a certain desired consistency of the proposed model with the previous findings, we refer to the development of such models as *experience-based* or *experience-consistent* modeling. Depending on the nature of the model, we distinguish between regression models (in which the output variable is continuous) and classification problems (classifiers), where we encounter a small number of discrete class labels [3][4][11][16][18][17].

In the experience-consistent models we may encounter a number of essential constraints that imply a way in which the underlying processing can be realized. For instance, it is common that the currently available data is quite limited in terms of its size (which implies limited evidence of the dataset), while the previously available datasets could be substantially larger, meaning that relying on the models formed in the past could be beneficial for the development of the current model.

In this chapter, we are concerned with system modeling that involves data and reconciles the developed model with some previously acquired domain knowledge being captured in the format of already constructed models that were based on auxiliary datasets. To emphasize the nature of modeling guided by the reconciliation mechanisms, we refer to this mode of identification as experience –consistent modeling. The chapter presents the conceptual and algorithmic framework by considering regression models. By forming a certain extended form of the performance index, we can show that the domain knowledge captured by regression models can play a similar role as a regularization component used in identification problems. It will be shown that a level of achieved consistency can be quantified through fuzzy sets (fuzzy numbers) of the parameters of the model, subsequently leading to the concept of fuzzy linear regression. Experimental results involve both synthetic low-dimensional data and selected data from the Machine Learning Repository. Furthermore, the data used in the experiments give rise to regression models as well classification problems.

7.2 Problem statement

We consider a regression model formed on a basis of some data D and influenced by the models formed with the use of other data D_1, D_2, …, and D_P. To realize a mechanism of experience consistency, we introduce several pertinent performance indexes that help quantify interaction between the models.

The optimization process starts from the optimal estimation of the parameters of the linear regression model for D, which assumes the form $\mathbf{a}^T\mathbf{x}$ +a_0. As usual, the optimal parameters of the model minimize the standard sum of squared errors

$$Q = \frac{1}{N}\sum_{\substack{x_k \in D \\ y_k \in D}} (\mathbf{a}^T \mathbf{x}_k + a_0 - y_k)^2 \tag{1}$$

With each datasite (dataset) D_1, D_2,....D_p we associate its local linear regression model described by the vector of parameters **a**[ii] and a_0[ii], ii=1,2,...P. The indexes in squared brackets point at the specific dataset. The crux of the consistency-driven modeling is to form the regression model on a basis of D while making the model perform consistently close to the individual models formed for the respective D_is. Using the following performance index, we strike a balance between the model formed exclusively on a basis of data D and the consistency of the model with the results produced by the models formed on a basis of some other datasites D_is

$$V = \sum_{\substack{x_k \in D \\ y_k \in D}} (a^T x_k + a_0 - y_k)^2 + \alpha \sum_{j=1}^{P} \sum_{\substack{x_k \in D \\ y_k \in D}} (a^T x_k + a_0 - a^T[j] x_k - a_0[j])^2 \tag{2}$$

The minimization of V for some predefined value of α leads to the optimal vector of parameter relies on the constraints of consistency(detail derivation is in Appendix-A) denote the vector of these parameters by $\mathbf{a}_{opt}$. An overall balance captured by (2) is achieved for a certain value of α. Here a certain tendency becomes clearly visible: higher values of α stress higher relevance of other models and their more profound impact on the constructed model. The essence of the minimized performance index is schematically illustrated in Figure 7.1 (a). Observe that the assessment of consistency is realized by making use of the dataset D. First, the model is constructed on the basis of D. Second, the consistency is expressed on a basis of differences between the constructed model and those models coming from D_is where the differences are assessed with the use of data D. There is yet another interesting view of the format of the minimized performance index. The second component in V plays a role that is similar to a regularization term being typically used in estimation problems. However, its origin here has a substantially different format from the one encountered in the literature. In other words, we consider other data (and models) rather than focusing on the complexity of the model expressed in terms of its parameters.

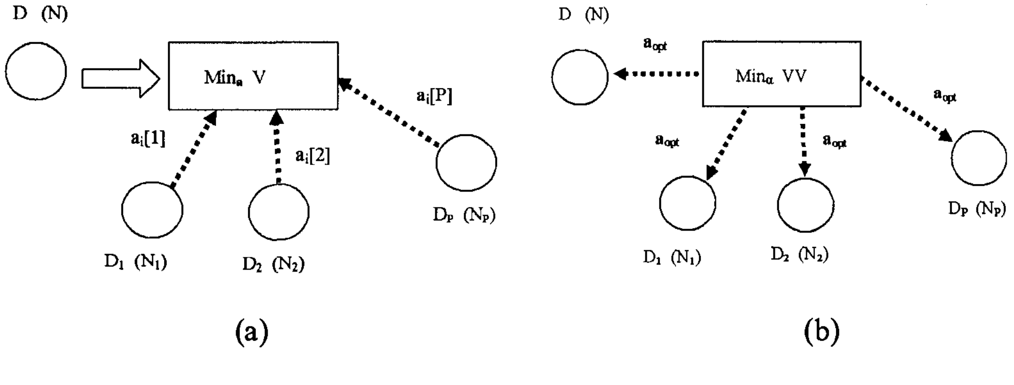


Figure 7.1 Minimization of the performance index V – a schematic view of the construction of the model (a) and a way of the maximization of consistency of the model across all data sets (b).

While the semantics of the above performance index (2) is straightforward, a choice of the value of α requires some attention. To optimize the level of contribution coming from the datasets, we may adhere to the following evaluation process that consists of two fundamental components. The quality of the optimal model is evaluated with respect to data D. The same optimized model is transferred to each D_i (viz. the parameters of the model are made available at D_i) and its quality is evaluated there. We combine the results (viz. the corresponding squared errors) by adding their normalized values. Given these observations, the index quantifying a global behaviour of the optimal model arises in the following form

$$VV = \frac{1}{N} \sum_{\substack{x_K \in D \\ y_k \in D}} (\mathbf{a}_{opt}^T \mathbf{x}_k + \mathbf{a}_{0opt} - y_k)^2 + \sum_{j=1}^{P} \frac{1}{N_j} \sum_{\substack{x_k \in D_j \\ y_k \in D_j}} (\mathbf{a}_{opt}^T \mathbf{x}_k + \mathbf{a}_{0opt} - y_k)^2 \tag{3}$$

Apparently the expression of VV is a function of α and the optimized level of consistency is such for which VV attains its minimal value, namely
α_{opt} = arg Min VV(α).

The optimization scheme (2) along with its evaluation mechanisms governed by (3) can be generalized by admitting the various levels of impact that each data D_i could have in the process of achieving consistency. We introduce some positive weights w_i, i=1, 3, ...p which are used in the performance index

$$V = \sum_{\substack{x_k \in D \\ y_k \in D}} (\mathbf{a}^T \mathbf{x}_k + \mathbf{a}_0 - y_k)^2 + \alpha \sum_{j=1}^{P} w_j \sum_{x_k \in D} (\mathbf{a}^T \mathbf{x}_k + \mathbf{a}_0 - \mathbf{a}^T[j]\mathbf{x}_k - \mathbf{a}_0[j])^2 \tag{4}$$

Lower values of w_i indicate lower influence of the model formed on a basis of data D_i when constructing the model for data D. The role of such weights is particularly apparent when dealing with data D_i, which are in some temporal or spatial relationships with respect to D. In these circumstances, the values of the weights are reflective of how far (in terms of time or distance) the sources of the individual data are from D. For instance, if D_j denotes a collection of data gathered some time ago in comparison to the currently collected data D_i, then it is intuitively clear that the value of weight w_j should be lower than w_i.

As an auxiliary performance index that expresses a quality of the model for which (2) has been minimized with α being selected with regard to (3), we consider the following expression

$$Q^{\sim} = \frac{1}{N} \sum_{\substack{x_k \in D \\ y_k \in D}} (\mathbf{a}_{opt}^T \mathbf{x}_k + \mathbf{a}_{0opt} - y_k)^2 \tag{5}$$

The values of $Q^{\sim}$ considered vis-à-vis the results expressed by (1) are helpful in assessing the extent the regression model optimized with regard to data D while achieving consistency with D_1, D_2, ..., D_p deteriorates when applied to D over the optimal regression model being optimized exclusively on a basis of D.

The same way of incorporating experiential consistency is realized in case of linear classifiers [7]. Considering a two-class problem $\{\omega_1, \omega_2\}$, we use the following labeling scheme of the form: if x_k belongs to class ω_1, the output y_k is taken as +1. Otherwise (for class ω_2) we assign value y_k equal to -1. Given this dataset, we run a standard regression procedure. The pattern $\mathbf{x}_k$ is said to be correctly classified when one of the following conditions is satisfied

$$\mathbf{a}^T\mathbf{x}_k + a_0 \geq 0 \text{ and } \mathbf{x}_k \text{ belongs to class } \omega_1, \text{ viz. } y_k = +1$$
$$\mathbf{a}^T\mathbf{x}_k + a_0 < 0 \text{ and } \mathbf{x}_k \text{ belongs to class } \omega_2, \text{ namely } y_k = -1 \qquad (6)$$

Otherwise a classification error occurs. The classification error rate (classification error) is taken as a ratio of the number of misclassified patterns (n) and all patterns (N), that is e = n/N. An error (expressed in %) produced for all datasites (D, and D_i) is computed in the form

$$E = \left(\frac{n[1] + n[2] + ... + n[P] + n}{N[1] + N[2] + ... + N[P] + N}\right) * 100 \qquad (7)$$

The above formula is a certain analog of the expression provided by (3).

7.3 Models through characteristics of granular parameters

Once the mechanism of experience consistency has been invoked, the determined parameters of the optimized model can be further evaluated. In particular, we are interested in quantifying the nature of the parameters of the model so that they reflect upon their reliance on several models formed for the previous data. The essence of the problem is in a representation of a collection of numeric values of the same parameter of the models in a form of some aggregate. A certain alternative is to form a fuzzy set whose membership function captures individual numeric values. The idea introduced in [7] follows this observation. Let us consider a finite number of numeric values $\{z_1, z_2, ..., z_P\}$.

We intend to span a unimodal fuzzy set A[13] over data z_i in such a way that it represents these data to the highest possible extent. The form of the membership is also defined in advance. For instance, we could consider triangular membership functions, such as Gaussian, parabolic, etc. Furthermore we consider that a modal value of A, say "u", is given. Let us look at the values of z_i that are lower than u, $z_i < u$. We use them to estimate the parameters of the left-hand side of the membership function. The determination of the right-hand side of the membership function is realized in an analogous manner. The method for computation of a membership function of fuzzy set of type-2 is covered in in Chapter 4 (Section 4.2.1).

The procedure is applied to the formation of fuzzy sets of the parameters of the regression model in D. As an example, consider the i-th parameter of the regression model at D, say

a_i. The corresponding values of the same parameter of the same coefficient of the regression model available at D_1, D_2, …, D_P are $a_i[1]$, $a_i[2]$, …, $a_i[P]$. The resulting fuzzy set A_i has ai as its modal value and its parameters are determined on the basis of the numeric evidence available in this manner, viz. $z_1 = a_i[1]$, $z_2=a_i[2]$,…, $z_P=a_i[P]$. A particular form of the fuzzy set could come with its triangular membership function. This fuzzy set is fully characterized by its three parameters, such as u_i, a_i, b_i, with a_i and b_i denoting its lower and upper bound. We also use the concise notation $A_i(a_i, u_i, b_i)$ to describe this fuzzy set. In addition to their evident simplicity and good interpretability, fuzzy numbers of this nature are easy to compute with, especially when dealing with linear models. The extension of the linear model $y = a_0 + a_1x$ to its fuzzy counterpart reads as $Y = A_0 \oplus A_1 \otimes x$ where $A_0= <c, n, d>$ and $A_1 =<a, m, b>$ are abbreviated descriptors of triangular fuzzy numbers of the parameters. The symbols $\oplus$ and $\otimes$ are used to underline the nonnumeric nature of the arguments being used in the model over which the multiplication and addition are carried out. The resulting output Y of this regression model is again a triangular fuzzy number $Y = <w, y, z>$ where their parameters are computed as follows

Modal value $n + mx$
Lower bound $c + ax$ if $x>0$ and $d + bx$ otherwise
Upper bound $d+ bx$ if $x> 0$ and $c+ax$ otherwise

The spread of the output Y reflectes of the consistency of the sources of data and knowledge being used in the formation of the regression model.

7.4 Experimental studies

In the ensuing series of experiments, we consider a suite of synthetic and real-world data that involve both regression and classification nature of problems. The characteristics of data used along with their origin are summarized in Table 7.1.

Table 7.1 A suite of experimental datasets (used in the design of regression models)

No.	Dataset	Source	Number of data (N)	Number of features (attributes) (n)
1	Machine-CPU	UCI- Web*	209	7
2	Auto-mpg	UCI	398	8
3	Breast Cancer	UCI	194	33
4	Auto Car Price	UCI	159	17
5	California House	StatLib Repository – Web&	20,640	9
6	Friedman Synthetic	[1]	40,768	11
7	Abalone	UCI	4177	9
8	Boston Housing	UCI	506	14

Note: Web*: http://www.ics.uci.edu/~mlearn/MLRepository.html.
Web&: http://lib.stat.cmu.edu/

We start with several two-dimensional synthetic datasets that are helpful in illustrating the very nature of the problem and quantify the effect of forming consistency under various circumstances.

Synthetic data-1 We consider a two-dimensional synthetic data shown in Figure 7.2. The 300 data points are linearly distributed with a strong linear tendency between "x" and "y". In the series of completed experiments, we distribute the data among "P" subsets of equal size. The value of "P" varied in-between 1 and 9. The obtained results are reported in Table 7.2 in which we summarize the values of all performance indexes discussed in Section 7.2.

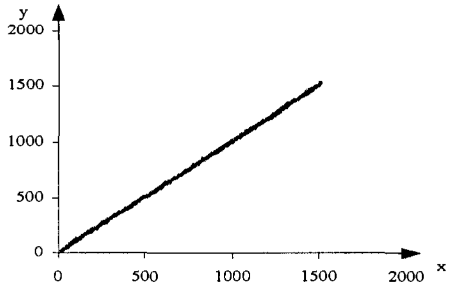


Figure 7.2 Plot of two-dimensional x-y synthetic data

Table 7.2 Values of performance index obtained for optimal values of α; αopt are also reported here.

P	1	2	3	4	5	6	7	8	9
Q	37.24	33.73	33.06	39.71	35.85	33.11	26.30	33.10	37.10
α_{opt}	0.89	1.18	0.73	1.66	0.83	2.34	1.45	1.59	0.72
VV (α=0)	72.04	108.91	151.22	182.26	221.98	251.42	306.40	332.03	377.97
VV(α_{opt})	71.20	106.81	142.46	178.08	213.60	249.82	284.25	320.05	356
$Q^{\sim}$	37.69	34.35	35.36	40.51	37.11	33.31	29.35	34.38	39.84

The results are very similar for different splits. The model constructed on a basis of data D exhibits a very similar performance to the individual models formed on a basis of D$_i$s. This result is not surprising at all given the nature of the data, which in essence leads to very homogeneous subsets of data generated in this manner. In essence, we encountered some very minor differences when sampling the data shown in Figure 7.2. Subsequently the values of Q and $Q^{\sim}$ (which pertain to the optimal value of α) become very similar. There is no noticeable differences in the values of VV obtained for α =0 and α_{opt}.

Quantification effect of collaboration:

The effect of consistency enhancement is quantified in terms of fuzzy sets of parameters of the regression model. Following the construction discussed in Section 7.3, we arrive at fuzzy numbers describing granular parameters of the linear model. The obtained results for selected values of P are shown in Figure 7.3 and 7.4. It is noticeable that the spreads of the fuzzy numbers obtained after experience-consistent adjustment become more confined (viz. the support of the corresponding fuzzy numbers gets smaller). There is also some shift in the location of the support of the resulting fuzzy numbers.

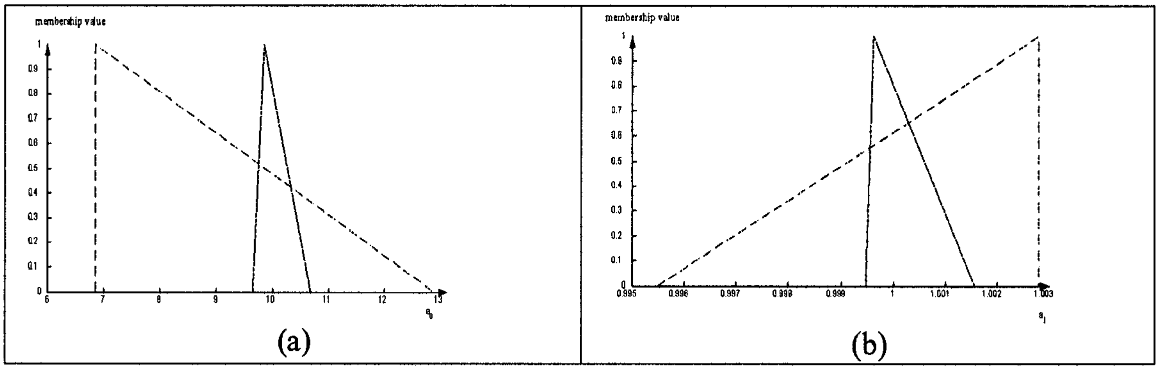


Figure 7.3 Membership functions of fuzzy sets of parameters of the regression model (a) fuzzy set a_0 and (b) Fuzzy set a_1 with P = 9. Dotted lines denote membership functions obtained before experience consistent development of the model while the solid lines deal with the model's parameters that has been developed when invoking the consistency mechanism.

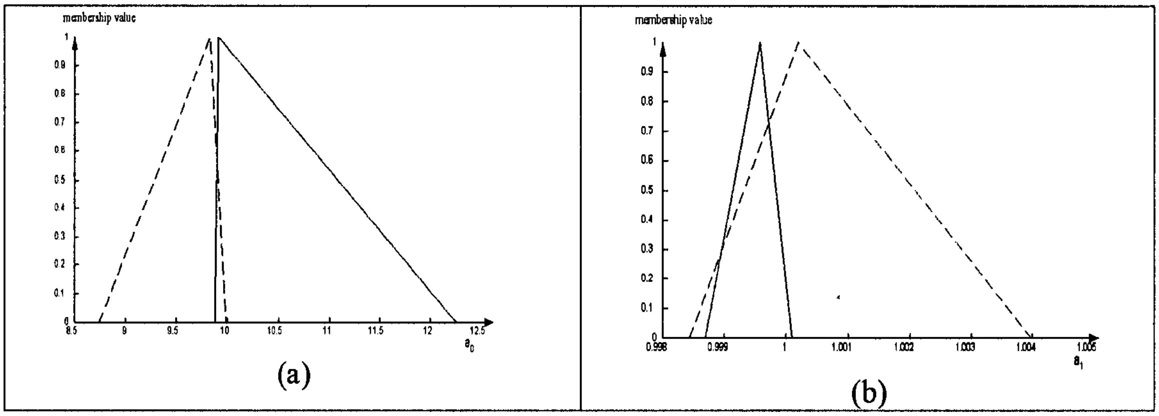
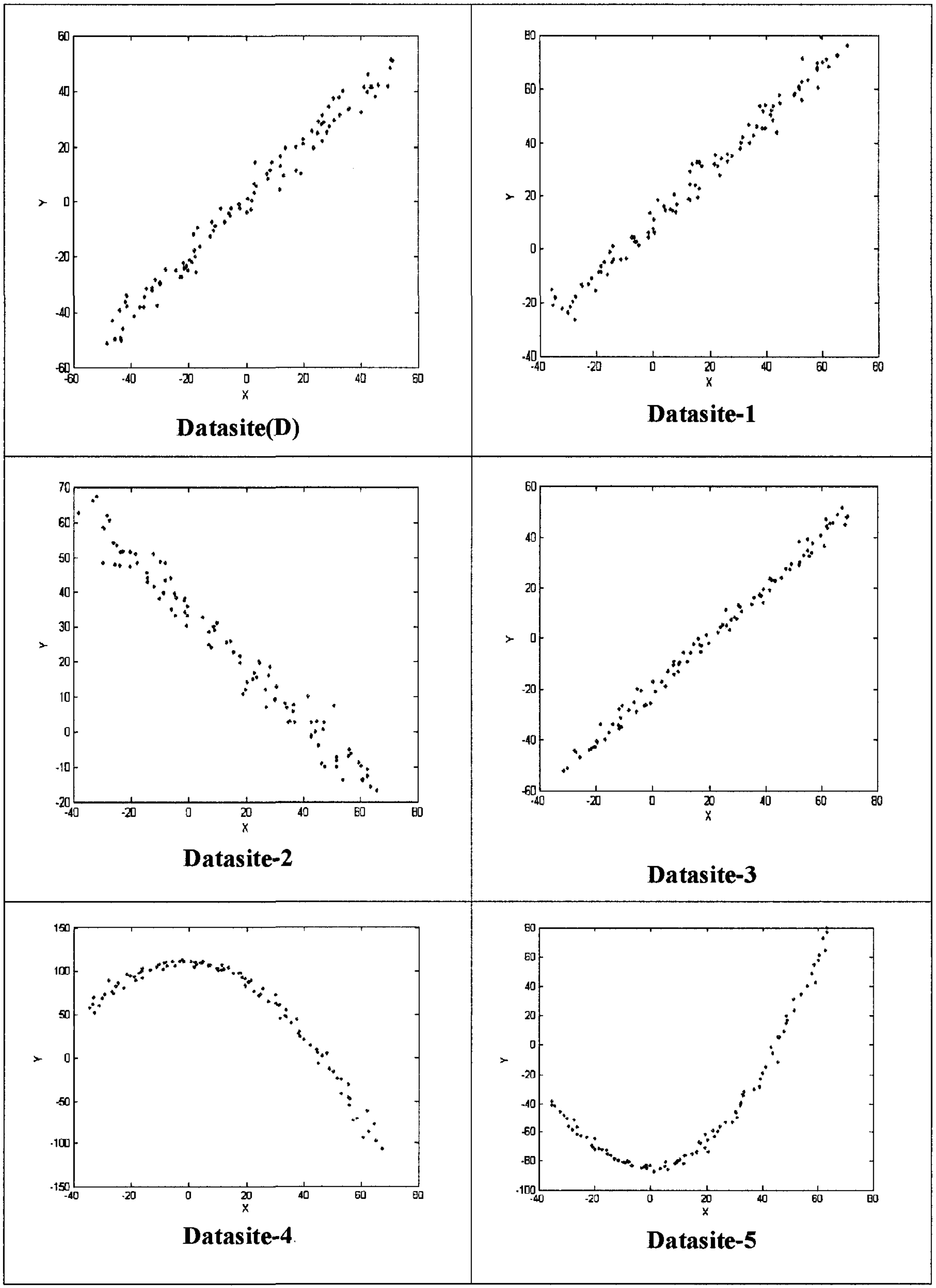


Figure 7.4 . Membership functions of fuzzy sets of parameters of the regression model (a) fuzzy set a_0 and (b) Fuzzy set a_1 with P = 6. Dotted lines denote membership functions obtained before experience consistent development of the model while the solid lines deal with the model's parameters that here been developed when invoking the consistency mechanism.

Synthetic data-2. Here we consider D, D_1, D_2,..., D_5 as shown in Figure 7.5. These datasets (each of them consists of 100 elements) exhibit some diversity as to the corresponding relationships between input and output variables. Furthermore each of them is contaminated by some noise.



Datasite(D)

Datasite-1

Datasite-2

Datasite-3

Datasite-4

Datasite-5

Figure 7.5 Two-dimensional synthetic datasets: Datasite, Datasite-1, …, Datasite-5

Table 7.3 Values of performance indexes for selected valued of P.

P	1	2	3	4	5
Q	16.62	16.62	16.62	16.62	16.62
α_{opt}	0.98	0.67	1.08	0.79	0.95
VV (α=0)	125.71	2,878	3,280	11,895	16,042
VV(α_{opt})	78.98	2,024	2,558	8,516	13,554
$Q^{\sim}$	40.05	383.93	216.84	945.70	447.27

As revealed in Table 7.3, the higher number of datasets lead to the growing role of the consistency based modeling that becomes reflected in higher values of differences between VV obtained for α equal to zero and its optimized value (α_{opt}). There is no strong relationship between an optimal value of α and P; it has been found, however, that these optimal values are located around 1.

Real-world data. The series of experiments was carried out by making use of the Machine Learning repository datasets as well as others included in Table 7.1. We considered a series of splits (p). For any scenario, we repeated an experiment 10 times to achieve higher confidence in the obtained results. For example, for a certain number of datasets (p), the split was repeated 10 times and the results are reported in terms of their mean values and standard deviations. The results are reported in a tabular format, Table 7.4.

In spite of some differences, several general observations can be made. The experience – consistent model contributes to the lower values of the performance index (VV) and the differences between its values for α =0.0 and α_{opt} are higher when the number of data sets is higher, demonstrating that experience consistency contributes to the models of enhanced quality. The optimal values of α depend on data and the number of datasets. However, a range of their values lies between 1 and 5.

Table 7.4 Results of experience-based modeling for selected Machine Learning datasets: (a) Abalone, (b) Boston Housing, (c) Auto-mpg, (d) California Housing, (e) Fried Synthetic and (f) Machine-CPU activity.

P	1	2	3	4	5	6	7	8	9
Q	4.80 ±0.15	4.88 ±0.21	4.93 ±0.22	4.78 ±0.32	4.74 ±0.39	4.77 ±0.38	4.82 ±0.48	4.89 ±0.27	4.73 ±0.24
α_{opt}	1.29 ±0.84	1.29 ±0.84	2.52 ±3.13	4.43 ±4.81	3.27 ±4.19	3.52 ±4.50	2.68 ±4.03	2.05 ±3.23	3.18 ±4.71
VV (α=0)	9.75 ±0.06	9.75 ±0.06	19.76 ±0.30	24.74 ±0.40	30.01 ±0.57	34.78 ±0.56	39.97 ±0.76	44.69 ±0.39	50.19 ±1.44
VV (α_{opt})	9.66 ±0.002	9.66 ±0.002	19.38 ±0.04	24.26 ±0.05	29.16 ±0.09	34.00 ±0.06	38.89 ±0.06	43.81 ±0.09	48.69 ±0.12
$Q^{\sim}$	4.84 ±0.15	4.84 ±0.15	5.03 ±0.21	4.86 ±0.30	4.90 ±0.45	4.91 ±0.38	4.98 ±0.54	4.98 ±0.24	4.87 ±0.27

(a)

P	1	2	3	4	5	6	7	8	9
Q	22.23 ±2.69	18.72 ±3.83	17.98 ±4.74	18.80 ±3.10	17.74 ±5.68	17.60 ±6.04	12.75 ±5.46	13.71 ±5.23	16.09 ±8.98
α_{opt}	0.918 ±0.12	1.008 ±0.30	1.403 ±0.68	0.90 ±0.44	0.967 ±0.62	1.81 ±1.81	1.94 ±2.88	1.94 ±2.25	1.745 ±2.97
VV (α=0)	45.74 ±0.66	71.02 ±3.51	102.75 ±11.59	126.32 ±12.73	164.39 ±10.18	215.07 ±72.98	233.55 ±26.79	335.80 116.70	366.94 ±149.29
VV (α_{opt})	43.85 ±0.03	65.60 ±1.45	87.91 ±0.63	110.27 ±1.10	133.15 ±1.30	155.64 ±3.36	180.47 ±3.91	202.05 ±2.17	229.75 ±10.39
$Q^{\sim}$	23.22 ±2.79	20.28 ±3.82	20.66 ±3.81	21.49 ±4.18	22.56 ±6.96	22.23 ±6.16	17.73 ±5.66	20.50 ±4.73	23.85 ±10.04

(b)

P	1	2	3	4	5	6	7	8	9
Q	10.68 ±1.07	10.74 ±0.87	10.28 ±1.94	11.31 ±1.88	7.99 ±1.29	8.58 ±1.35	8.30 ±2.19	9.61 ±2.37	9.98 ±3.20
α_{opt}	1.00 ±0.14	0.96 ±0.19	1.17 ±0.29	0.95 ±0.38	1.12 ±1.18	1.06 ±0.24	2.20 ±2.99	2.01 ±2.89	1.82 ±2.90
VV (α=0)	22.34 ±0.30	34.54 ±0.96	48.34 ±1.53	60.53 ±2.88	71.65 ±3.28	91.71 ±9.61	105.24 ±11.12	119.71 ±7.86	139.13 ±23.58
VV (α_{opt})	21.84 ±0.01	32.71 ±0.32	43.78 ±0.16	54.64 ±0.64	65.44 ±0.82	76.01 ±0.99	87.56 ±0.91	98.75 ±0.51	109.24 ±1.80
$Q^{\sim}$	10.93 ±1.12	11.33 ±0.94	11.31 ±1.88	10.92 ±1.47	8.94 ±1.32	10.44 ±0.91	10.06 ±2.74	11.43 ±2.39	12.50 ±4.44

(c)

P	1	2	3	4	5	6	7	8	9
Q	4.84e+9 ±8.76e+7	4.75e+9 ±1.49e+8	4.80e+9 ±1.22e+8	4.76e+9 ±1.43e+8	4.73e+9 1.36e+8	4.86e+9 ±1.65e+8	4.87e+9 ±2.16e+8	4.69e+9 ±2.59e+8	4.76e+9 ±7.02e+7
α_{opt}	1.062 ±0.20	1.22 ±0.44	2.53 ±2.81	1.45 ±1.31	3.81 4.40	4.41 ±3.58	4.45 ±4.78	3.21 ±4.00	6.74 ±4.40
VV (α=0)	9.68e+9 ±9.03e+6	1.45e+10 ±1.11e+7	1.94e+10 ±5.45e+7	2.43e+10 ±7.12e+7	2.93e+10 ±2.60e+8	3.41e+10 ±1.34e+8	3.90e+10 ±2.42e+8	4.41e+10 ±3.93e+8	4.87e+10 ±2.59e+8
VV (α_{opt})	9.66e+9 ±69.7	1.45e+10 ±164128	1.93e+10 ±1.21e+6	2.42e+10 ±3.11e+6	2.90e+10 ±2.07e+6	3.38e+10 ±5.58e+6	3.87e+10 ±5.36e+6	4.35e+10 ±1.07e+7	4.83e+10 ±8.29e+6
$Q^{\sim}$	4.85e+9 ±8.73e+7	4.76e+9 ±1.51e+8	4.82e+9 ±1.27e+8	4.79e+9 ±1.42e+8	4.77e+9 ±1.54e+8	4.89e+9 ±1.71e+8	4.91e+9 ±2.30e+8	4.77e+9 ±2.78e+8	4.80e+9 ±8.34e+7

(d)

P	1	2	3	4	5	6	7	8	9
Q	6.90 ±0.05	6.88 ±0.08	6.84 ±0.12	6.86 ±0.13	6.94 ±0.123	6.88 ±0.203	6.83 ±0.131	6.83 ±0.129	6.901 ±0.194
α_{opt}	1.002 ±0.01	1 ±0.02	0.99 ±0.019	1.01 ±0.02	1.001 ±0.031	0.993 ±0.04	0.984 ±0.03	1 ±0.05	1.001 ±0.067
VV (α=0)	13.84 ±0.003	20.77 ±0.006	27.70 ±0.005	34.64 ±0.01	41.57 ±0.016	48.54 ±0.036	55.47 ±0.038	62.39 ±0.040	69.37 ±0.055
VV (α_{opt})	13.84 ±5.02e-7	20.76 ±0.0002	27.679 ±1.83e-6	34.60 ±0.001	41.52 ±0.003	48.44 ±9.11e-6	55.36 ±1.02e-5	62.28 ±0.003	69.2 ±0.006

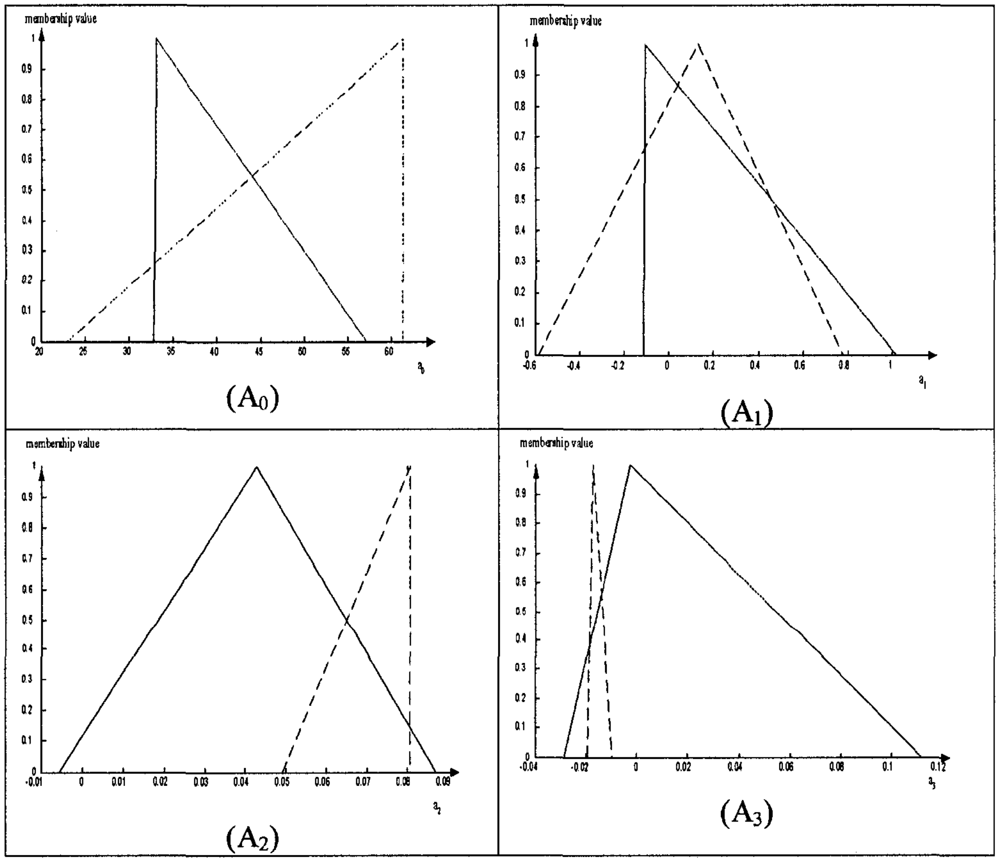
$Q^{\sim}$	6.90 ±0.05	6.89 ±0.08	6.84675 ±0.115	6.87 ±0.13	6.95 ±0.124	6.89 ±0.201	6.84 ±0.131	6.84 ±0.130	6.92 ±0.195

(e)

P	1	2	3	4	5	6
Q	1772.84 ±733.67	1533.12 ±1019.79	1231.58 ±645.81	2595.66 ±1785.79	856.969 ±403.87	1164.67 ±1318.84
α_{opt}	2.23 ±2.59	4.008 ±4.09	6.22 ±4.69	1.97 ±3.16	5.55 ±4.81	6.43 ±4.48
VV (α=0)	17220.8 ±3576.26	24237.3 ±5403.05	31183.3 ±7643.06	42424.4 ±15353.5	45970.9 ±17708.6	72711.6 ±30750.9
VV (α_{opt})	11199 ±437.90	15857.4 ±1389.58	19775.5 ±1320.46	25034.9 ±2827.43	29331.5 ±4881.99	32931.3 ±1614.58
$Q^{\sim}$	3229.45 ±1096.85	3050.02 ±1734.26	2784.65 ±1159.20	5166.55 ±3292	2844.94 ±3171.75	3941.63 ±3771.08

(f)

The fuzzy numbers of the parameters of the regression models are included in Figures 7.6- 7.7. It is noticeable how the parameters of the models are affected by the experience consistency component.



(A_0)

(A_1)

(A_2)

(A_3)

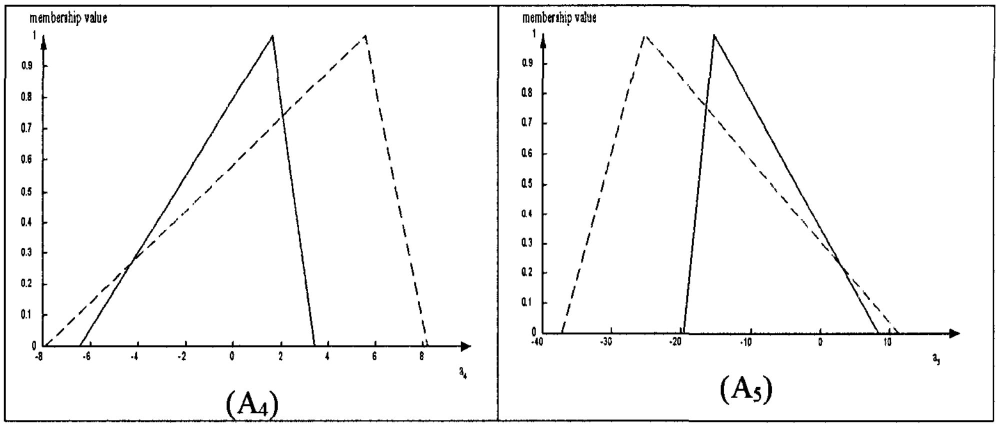
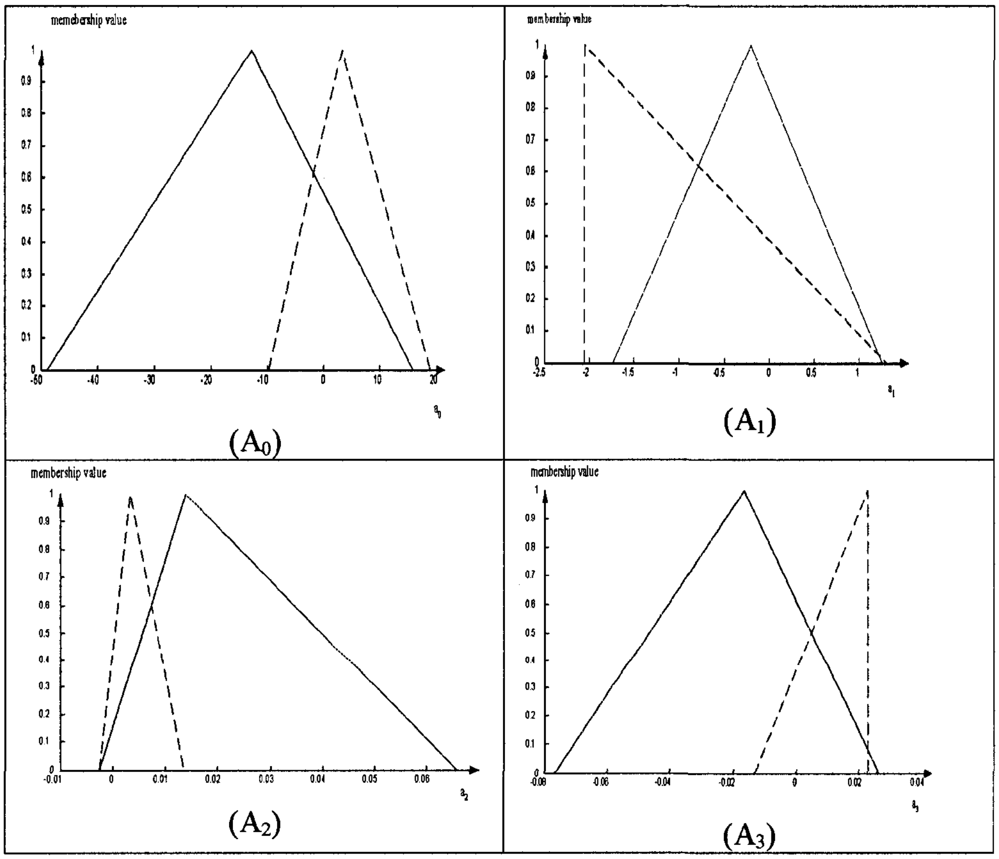


Figure 7.6 Fuzzy numbers of the regression model (a_0, a_1, …, a_5) for the Boston housing with P=6. Dotted lines denote membership functions obtained before experience consistent development of the model, while the solid lines deal with the model's parameters that have been developed when invoking the consistency mechanism.



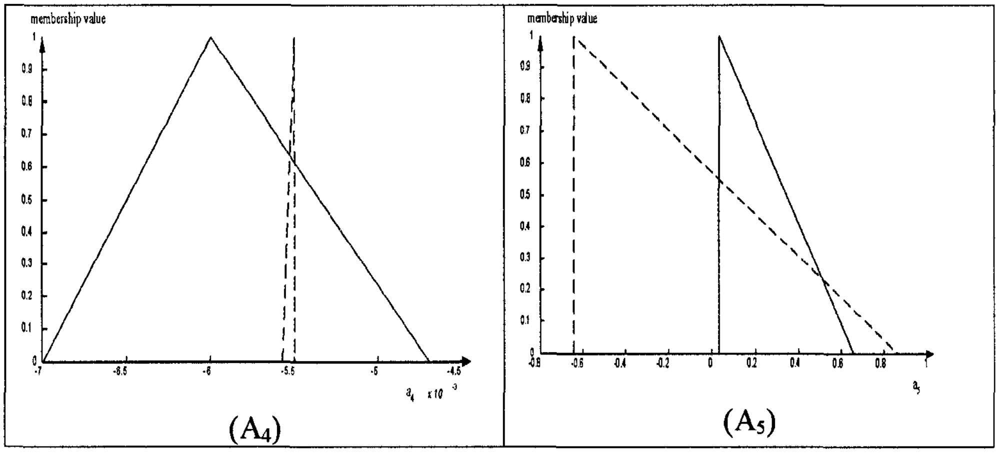


Figure 7.7 Fuzzy numbers of the regression model (a_0, a_1, …, a_5) for the Auto-mpg with P=6. Dotted lines denote membership functions obtained before experience consistent development of the model, while the solid lines deal with the model's parameters that have been developed when invoking the consistency mechanism.

7.5 Two-Class classification problem

Moving on with the classification problems, we consider a collection of datasets shown in Table 7.5. Furthermore we experiment with a two-dimensional Gaussian dataset illustrated in Figure 7.8. In all cases, the development of the classifiers follows the same scheme as presented in Section 7.2.

Table 7.5 Classification data sets overview.

S.N	Dataset	Source	Observations(N)	Features(n)
1	Breast Cancer	UCI	569	31
2	Ionosphere	UCI	351	34
3	Contraception Choice	UCI	1473	10
4	Liver Disorder	UCI	345	7
5	Ringnorm	Web*	7400	21
6	Twonorm	Web&	7400	21

Web*: http://www.cs.toronto.edu/~delve/data/ringnorm/desc.html
Web&: http://www.cs.toronto.edu/~delve/data/twonorm/desc.html

The synthetic data are Gaussian with the following characteristics $\mathbf{m}_1$=[4.5 5.0] $\mathbf{m}_2$ = [7.5 6.5] and the covariance matrices equal to $\Sigma_1=\begin{bmatrix}1.0 & 0.0\\0.0 & 1.0\end{bmatrix}$ and $\Sigma_2=\begin{bmatrix}1.0 & 0.0\\0.0 & 1.0\end{bmatrix}$.

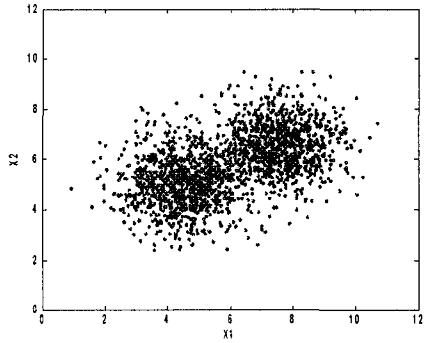


Figure 7.8 Two class normally distributed classification data.

The obtained classification errors (along with their standard deviations) are included in Figure 7.9. The dotted line shows the error for the classifier that was built when using only data in D. The solid line shows the classification error for the experience –consistent classifier. It is apparent that for higher values of "P" the experience consistency plays an important role.

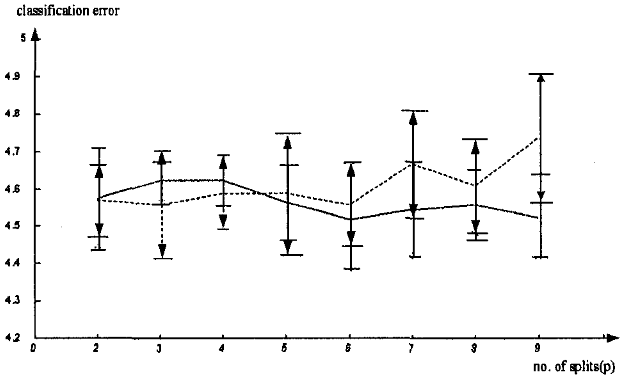


Figure 7.9 Classification error versus the number of splits of data (P):The dotted line - experience consistency- was not involved (α=0), while the solid line- optimal experience consistency- was established (α_{opt}).

The results for Machine Learning datasets are shown in Figure 7.10. The results are reported in the same manner as for the synthetic data. In all cases, we acknowledge an essential role of experience consistency, which becomes even more profound in the case of the higher values of "P".

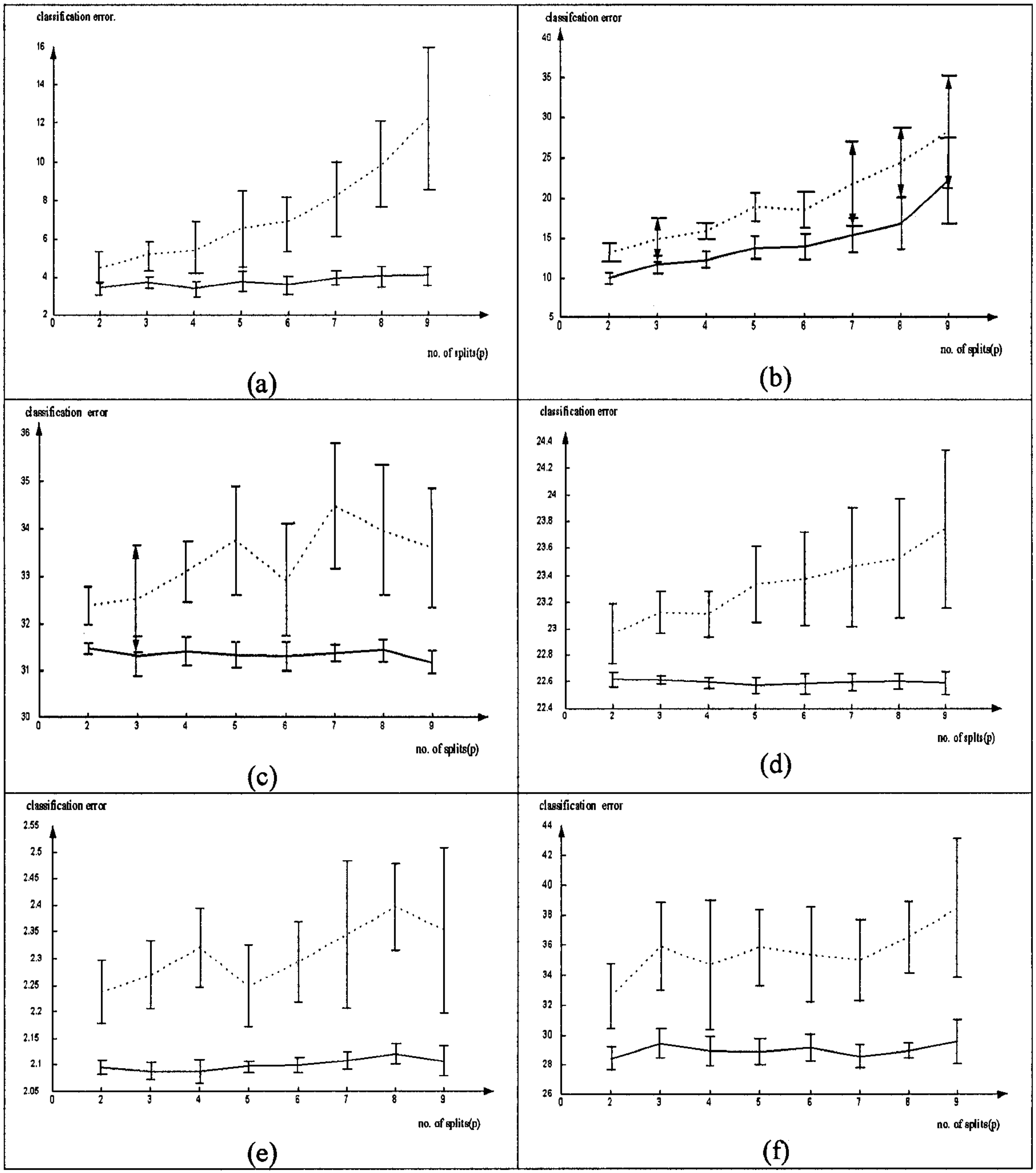


Figure 7.10 Classification results obtained for several Machine Learning dataset: (a) Breast Cancer, (b) Ionosphere, (c) Contraception, (d) Ringnorm, (e) Twonorm, and (f) Liver Disorder. The dotted line – experience consistency- was not involved (α=0). The solid line- optimal experience consistency- was established (α_{opt}).

7.6 Conclusions

In this chapter, we have discussed an approach of system identification realized in a collaborative framework of data (experimental evidence) and past experience (knowledge evidence). We demonstrated how to reconcile these two essential sources of guidance in the form of a single regression model. In particular, one could note that the knowledge-based component (previously constructed models) can serve as a certain form of the regularization mechanism encountered in various modeling platforms. A level of achieved consistency is modeled in terms of fuzzy sets and quantified by fuzzy numbers of their parameters, giving rise to so-called fuzzy linear regression models. The introduced optimization procedure helps us strike a sound balance between the data-driven and knowledge-driven evidence.

The proposed concept has broader ramifications than discussed in this chapter. It does not relate to a specific category of models. While the regression models have been used to focus our discussion on a relatively simple identification scheme, one can consider any other architectures such as neural networks or fuzzy rule-based systems. Furthermore, one could envision a variety of models coming from different data sites (say, neural networks, fuzzy models, nonlinear polynomials) that could be used effectively in the build up of the high level of consistency. More flexibility could be achieved by differentiating the contribution of the individual models in the development of the knowledge- and data-navigated model.

References

1. L. Breiman, Bagging predictors, *Machine Learning* , 24(3), 1996,123-140.

2. S. G. Cao, N. W. Rees, G. Feng, Analysis and design for a class of complex control systems, Part I and Part II, *Automatica*, 33, 6, 1997, 1017-1028 and 33, 6, 1997, 1029-1039.

3. C. Chen, *Statistical Pattern Recognition*, Spartan Books, 1973.

4. R. O. Duda, P. E. Hart, D. G. Stroke, *Pattern Classification*, John Wiley, 2001.

5. K. H. Fasol, H. P. Jörgl, Principles of model building and identification, *Automatica*, 16, 5, 1980, 505-518.

6. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, 1972

7. A. Jain, R. Duin, J. Mao, Statistical pattern recognition: a review, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 2000, 4-37.

8. J. Hromkovic, *Algorithmics for Hard Problems*, Springer, 2001.

9. M. Kumar, N. Stoll, R. Stoll, An energy-gain bounding approach to robust fuzzy identification, *Automatica*, 42, 5, 2006, 711-721.

10. J. P. Michael, *Classical Optimization: Foundations and Extensions*, North-Holland Publishing Company., 1976.

11. W. Pedrycz, G. Vukovich, On elicitation of membership functions, *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, 32, 2002, 761-767.

12. W. Pedrycz, *Knowledge-Based Clustering: From Data to Information Granules*, J. Wiley, Hoboken, NJ, 2005.

13. W. Pedrycz, F. Gomide, *Fuzzy Systems Engineering*, J. Wiley, Hoboken, NJ, 2007.

14. F. L. Ramsey, D. W. Schafer, *The Statistical Sleuth: A course in Methods of Data Analysis* , Duxbury, 2002.

15. H. W. Sorenson, *Parameter Estimation : Principles and Problems*, Mrcel Dekker, 1980.

16. W. J. Thompson, *Computing For Scientists and Engineers*, John wiley, 1992.

17. L. A. Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*, 90, 1997, 111-117.

18. A. Zaknich, *Principles of Adaptive Filters and Self-learning Systems*, Springer, 2005.

Chapter 8

Experience-Consistent Fuzzy Rule-Based System Modeling

In this chapter, we develop an approach to fuzzy rule-based model identification realized in a collaborative framework of data (experiential evidence) and past experience (knowledge evidence). We demonstrated how to reconcile these two essential sources of guidance in the form of local regression models. We further elaborate as to how the collaboration mechanism gives rise to higher order granular constructs such as type-2 fuzzy sets in distributed fuzzy modeling.

The experimental studies presented include some synthetic data and selected datasets coming from the Machine Learning Repository.

8.1 Introduction

We introduce an experience-consistent development of fuzzy rule-based systems, which expand the notions and applicability of fuzzy sets introduced previously. The design of such fuzzy models involves some locally available data and reconciles the constructed model with some previously acquired domain knowledge. This type of domain knowledge is captured in the format of some other rule-based models constructed on a basis of some auxiliary data sets. By forming a certain extended form of the optimized performance index, we show that the domain knowledge captured by the individual rule-based models plays a similar role as a regularization component typically encountered in identification problems. We will show that a level of achieved experience-driven consistency can be quantified through fuzzy sets of the parameters of the local models standing in the conclusion parts of the rules. Experimental study involves both synthetic low-dimensional data and selected datasets from Machine Learning Repository.

8.2 Notations and a formulation of the problem

Fuzzy rule-based models [9][13] and fuzzy modeling play a predominant role in system modeling realized in the context of fuzzy sets. The most recent studies, cf. [1][3][4][6][7][8][10][14], are a genuine testimony to the wealth of approaches and new computational pursuits in this area. As usual in system modeling [2][5], including the development of fuzzy rule-based systems, we rely on an intensive and prudent usage of experimental data. We exploit the existing data in order to establish a structure of the respective model and estimate its parameters. With regard to the character of the usage of data, we encounter several fundamental problems that require careful attention.

Generalization capabilities of the models rely in a direct manner on the characteristics of data (in particular their representative capabilities with respect to the problem at hand) and the nature of the model itself. The characteristics of data deserve particular attention in case we encounter small datasets that could be also biased by some noise. The models developed on a basis of the limited and noisy dataset typically exhibit low prediction capabilities.

When dealing with experience-consistent models, we may encounter a number of essential constraints that imply a way in which the underlying processing can be realized. For instance, it is common that the currently available data is quite limited in terms of its size (which implies limited evidence of the data set, while the previously available datasets could be substantially larger, meaning that relying on the models formed in the past could be beneficial for the development of the current model. There is also another reason in which the experience –driven component plays a pivotal role. The dataset D could be quite small and affected by a high level of noise – in this case it becomes highly legitimate to seriously consider any additional experimental evidence available around.

In the realization of the consistent-oriented modeling, we consider the following scenario. Given is a dataset D, using which we intend to construct a fuzzy rule-based model. There is a collection of datasets D_1, D_2, …, D_P. For each of them, an individual fuzzy model is developed. Those local models are available when seeking consistency with the fuzzy models formed for D_{ii}, ii=1, 2, …, P. At the same time, it is worth stressing that the datasets themselves are not available to any processing and modeling realized at the level of D.

The underlying architectural details of the rule-based model considered in this study are as follows. For each datasite D and D_{ii}, we consider the rules with local regression models assuming the form

Data D

$$\text{-if } \mathbf{x} \text{ is } B_i \text{ then } y = \mathbf{a}_i^T\mathbf{x} \tag{1}$$

where $\mathbf{x} \in \mathbf{R}^{n+1}$ and B_i are fuzzy sets defined in the n-dimensional input space, i=1, 2,…, c. The local regression model standing in the i-th rule is a linear regression function described by a certain vector of parameters $\mathbf{a}_i$. More specifically, the n-dimensional vector of the original input variables is augmented by a constant input so we have $\mathbf{x} =[x_1\ x_2\ \dots\ x_n\ 1]^T$ and $\mathbf{a} =.[a_1\ a_2\ \dots\ a_n\ a_0]^T$ where a_0 stands for a bias term that translates the original hyperplane.

The same number of rules (c) is encountered at all other datasites, D_1, D_2, …, D_P. The format of the rules is the same as for D, that is, for the ii-th datasite D_{ii} we have

$$\text{-if } \mathbf{x} \text{ is } B_i[ii] \text{ then } y = \mathbf{a}_i[ii]^T\mathbf{x} \tag{2}$$

As before, the fuzzy sets in the condition part of the i-th rule are denoted by $B_i[ii]$, while the parameters of the local model are denoted by $\mathbf{a}_i[ii]$. The index in the square brackets refers to the specific datasite, that is, D_{ii} for $\mathbf{a}_i[ii]$.

The format of the data at D, comes in the form of input – output pairs $(\mathbf{x}_k, y_k)$, k=1, 2,..., N which are used to carry out learning in a supervised mode. The previously collected datasets denoted by D_1, D_2, ..., D_P consist of N_1, N_2, and N_P data points. We assume that due to some technical and non-technical reasons, the data available at D_j cannot be shared with D. However, the communication between the datasites can be realized at the higher conceptual level such as those involved in the parameters of the fuzzy models.

8.3 Development of the rule-based model

Alluding to the formulation of the problem, we consider a rule-based model constructed on a basis of data D where in the construction of the model we are influenced by the models formed with the use of D_1, D_2, ..., and D_P. To realize a mechanism of experience consistency, we introduce several pertinent performance indexes that are crucial in the quantification of this mechanism.

Given the architecture of the rule-based system, it is well known that we encounter here two fundamental design phases, that is (a) a formation of the fuzzy sets appearing in the conditions of the rules and (b) the estimation of the corresponding conclusion parts. There are numerous ways of carrying out this construction. Typically, when it comes to the condition parts of the rules, the essence of the design is to granulate data by forming a collection of fuzzy sets. The common technique relates to fuzzy clustering when the condition part of the rule involves a fuzzy set defined in $\mathbf{R}^n$ or a Cartesian product of fuzzy sets defined in $\mathbf{R}$. The conclusion part where we encounter local regression models is formed by estimating the parameters $\mathbf{a}_i$. Such an estimation process is standard to a high extent as it is nothing but a global minimization of the well-known squared error criterion.

The organization of the consistency–driven optimization relies on the reconciliation of the conclusion parts of the rules. We assume that the condition parts, viz. fuzzy sets, are developed independently from each other. In other words, we cluster data in the input space of D, D_1, ... , D_P assuming the same number of clusters (c) results in the same collection of rules. Then the mechanism of experience consistency is realized for the conclusions of the rules. Given the independence of the construction process of the clusters at the individual sites, before moving on with the quantification of the obtained consistency of the conclusion parts of the rules, it becomes necessary to align the information granules obtained at D and the individual datasites D_i.

8.3.1 Construction of information granules of conditions of the rules

Information granules in the input space can be developed in many different ways, cf. [3][6][11][12[17]. We are of the opinion that they need to directly reflect of the nature of data available, which makes fuzzy clustering an intuitively appealing alternative. More specifically, an FCM algorithm [12] comes as a suitable algorithmic vehicle. For the given number of clusters (c), we minimize a standard objective function and as a result obtain a collection of prototypes and a partition matrix. In the ensuing communication schemes of consistency development, we will be relying on the exchange of the prototypes.

8.3.2 Consistency-based optimization of local regression models

To make the ensuing formulas concise, we use a shorthand notation FM, FM[1], FM[2], …, FM[P] to denote rule-based models pertaining to data D, D[1],… etc.

As usual the optimal parameters of the local models occurring in the conclusions of the rules are chosen in such a way so that they minimize the sum of squared errors

$$Q = \frac{1}{N} \sum_{\substack{x_k \in D \\ y_k \in D}} (\mathrm{FM}(\mathbf{x}_k) - y_k)^2 \tag{3}$$

For given fuzzy sets of conditions, the determination of the parameters of the linear models is standard and well documented in the literature, cf. [13][15[16]. When we consider the form of the rule-based system, the output of the fuzzy model is determined as a weighted combination of the local models with the weights being the levels of activation of the individual rules. More specifically, we have

$$\hat{y}_k = \sum_{i=1}^{c} u_i(\mathbf{x}_k) \mathbf{a}_i^T \mathbf{x}_k \tag{4}$$

where $u_{ik}= u_i(\mathbf{x}_k)$ is a membership degree of the k-th data $\mathbf{x}_k$ to the i-th cluster being computed on a basis of the already determined prototypes in the input space. In a nutshell, (4) comes as a convex combination of the local models that aggregates the local models by taking advantage of the weight factors, expressing a contribution of each model based upon the activation reported in the input space. The detailed computations of the optimal parameters of the local models standing in the conclusion part of the rules are covered in the next section.

Optimization Details of the local models standing in the rules-based model

In the development of the Takagi-Sugeno rule-based model, we consider some finite data set of the form of input – output pairs, $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$,…., $(\mathbf{x}_N, y_N)$ where $\mathbf{x}_k \in \mathbf{R}^{n+1}$ and $y \in \mathbf{R}$.

Assuming a given number of rules (c) (which is the number of clusters), we apply the FCM for the input data $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$. The clustering method returns a collection of prototypes $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c$ and a partition matrix U whose rows $U_1, U_2, \ldots, U_c$ can be regarded as fuzzy sets forming a condition part of the rules. Thus the i-th rule reads as

$$\text{-if } \mathbf{x} \text{ is } U_i \text{ then } y_i = \mathbf{a}^T{}_i\mathbf{x}, \ i=1,2,\ldots,c, \tag{5}$$

The estimation of the parameters $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_c$ follows a standard least-square error minimization problem in which the performance index comes in the form

$$Q = \sum_{k=1}^{N} (y_k - \hat{y}_k)^2 . \tag{6}$$

The output of the model comes as a weighted sum of each of the local models

$$\hat{y}_k = \sum_{i=1}^{c} u_i(\mathbf{x}_k)\mathbf{a}_i^T \mathbf{x}_k , \tag{7}$$

where $u_{ik} = u_i(\mathbf{x}_k)$ is a membership degree of the k-th data $\mathbf{x}_k$ to the i-th cluster being computed on a basis of the already determined prototypes. More specifically, we compute this membership degree to be

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\| \mathbf{x}_k - \mathbf{v}_i \|}{\| \mathbf{x}_k - \mathbf{v}_j \|} \right)^{2/m-1}}, \tag{8}$$

To arrive at the optimal parameters of the local linear models, we introduce some notation which makes the formulation of the problem more concise and readable. First, each input data is weighted by the membership degree of the individual cluster thus yielding an (n+1) vector $\hat{\mathbf{x}}_{ik} = u_{ik}\mathbf{x}_k$. The above fuzzy model (7) can be written in a matrix format as

$$\hat{y}_k = [\hat{\mathbf{x}}_{1k}^T \ \hat{\mathbf{x}}_{2k}^T \ldots \hat{\mathbf{x}}_{ck}^T] \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \\ \mathbf{a}_c \end{bmatrix}, \tag{9}$$

where all parameters of the models are collected in the form of a single c(n+1)-dimensional vector such that

$$\mathbf{a} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \\ \mathbf{a}_c \end{bmatrix}.$$

(10)

The input data transformed through the fuzzy sets (clusters) are organized in a single matrix $\hat{X}$ with N rows and c(n+1) columns.

Furthermore, collect all outputs in a single vector **y** with the entries

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \\ y_N \end{bmatrix}.$$

(11)

The minimization problem reads as

$$\mathrm{Min}_{\mathbf{a}}\, Q,$$

(12)

where

$$Q = \|\mathbf{y}- \hat{X}\mathbf{a}\|^2 = (\mathbf{y}- \hat{X}\mathbf{a})^T(\mathbf{y}- \hat{X}\mathbf{a}).$$

(13)

The global solution is straightforward and could be easily encountered in many standard references. It reads as

$$\mathbf{a}_{\mathrm{opt}} = (\hat{X}^T\hat{X})^{-1}\hat{X}^T y = \hat{X}^{\#}\mathbf{y},$$

(14)

where $\hat{X}^{\#}$ is a pseudoinverse of $\hat{X}$.

The essence of the consistency-driven modeling is to form local regression models occurring in the conclusions of the rules on a basis of data D, while at the same time making the model perform in a consistent manner (viz. close enough) to the rule-based model formed for the respective D$_i$s. The following performance index strikes a sound balance between the model formed exclusively on a basis of data D and the consistency of the model with the results produced by the models formed on a basis of some other datasites D$_i$s, that is FM[j]($\mathbf{x}_k$).

$$V = \sum_{\substack{x_k \in D \\ y_k \in D}} (FM(\mathbf{x}_k) - y_k)^2 + \alpha \sum_{j=1}^{P} \sum_{\substack{x_k \in D \\ y_k \in D}} (FM(\mathbf{x}_k) - FM[j](\mathbf{x}_k))^2.$$

(15)

The calculations of FM[j]($\mathbf{x}_k$) for some $\mathbf{x}_k$ in D require some words of explanation. The model is communicated to D by transferring the prototypes of the clusters (fuzzy sets) and the coefficients of the linear models standing in the conclusions of the rules refer to Figure 8.1.

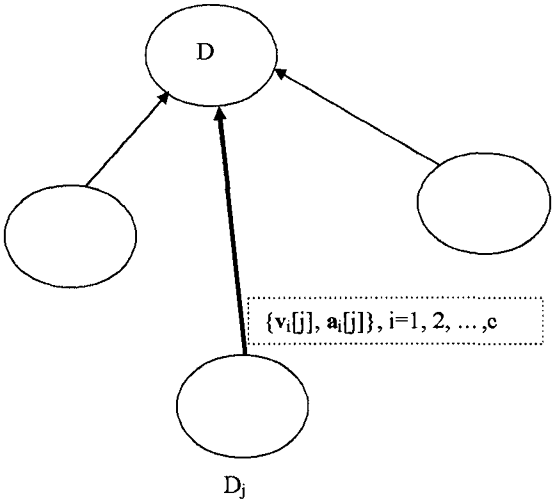


Figure 8.1 Communication between D and D_j realized by transferring parameters of the rule-based model available at individual data sites D_j

When used at D, the prototypes $\mathbf{v}_i[j]$, i=1, 2,...,c give rise to an induced partition matrix in which the k-th column (for data $\mathbf{x}_k$) assumes the following membership values $w_i(\mathbf{x}_k)$ computed in the standard manner as being encountered when running the FCM algorithm, that is,

$$w_i(\mathbf{x}_k)[j] = \frac{1}{\sum_{l=1}^{c}\left(\frac{\mathbf{x}_k - \mathbf{v}_i[j]}{\mathbf{x}_k - \mathbf{v}_l[j]}\right)^{1/m-2}} . \tag{16}$$

The transferred parameters of the local models obtained at the j-th datasite produce the output of the model FM[j]($\mathbf{x}_k$) obtained at D as a weighted sum of the form

$$\mathrm{FM}[j](\mathbf{x}_k) = \sum_{i=1}^{c} w_i(\mathbf{x}_k)[j] a_i^T(j)\mathbf{x}_k , \tag{17}$$

where $\mathbf{x}_k \in D$.

The minimization of the performance index V for some predefined value of α leads to the optimal vectors of the parameters of the linear models $\mathbf{a}_i$(opt), i=1, 2,..., c which reflects the process of satisfying the consistency constraints. The following section shows the details of optimization.

Optimization details

Considering the form of the performance index, we rewrite it as follows

$$Q = ||\mathbf{y} - \hat{X}\mathbf{a}||^2 + \alpha\ ||\mathbf{y}_1 - \hat{X}\mathbf{a}||^2 + \alpha\ ||\mathbf{y}_2 - \hat{X}\mathbf{a}||^2 + \ldots. + \alpha\ ||\mathbf{y}_P - \hat{X}\mathbf{a}||^{2.} \tag{18}$$

In the above expression, we have used the same notation as already introduced in optimization section of local models. Furthermore, the vectors $\mathbf{y}_1$, $\mathbf{y}_2$, …, $\mathbf{y}_P$ denote the outputs produced by the consecutive fuzzy models developed for D_1, D_2, .., and D_P. Taking the gradient of Q with respect to **a**, we obtain

$$\begin{aligned}\frac{dQ}{d\mathbf{a}} &= -\hat{X}^T\mathbf{y} - \hat{X}^T\mathbf{y} + 2(\hat{X}^T\hat{X})\mathbf{a} - \alpha(\hat{X}^T\mathbf{y}_1 + \hat{X}^T\mathbf{y}_1) + 2\alpha(\hat{X}^T\hat{X})\mathbf{a} - \alpha(\hat{X}^T\mathbf{y}_2 + \hat{X}^T\mathbf{y}_2) + \\ &2\alpha(\hat{X}^T\hat{X})\mathbf{a}\ldots. - \alpha(\hat{X}^T\mathbf{y}_P + \hat{X}^T\mathbf{y}_P) + 2\alpha(\hat{X}^T\hat{X})\mathbf{a}.\end{aligned} \tag{19}$$

By setting $\frac{dQ}{d\mathbf{a}} = \mathbf{0}$, we arrive at the solution, giving a global solution to (18)

$$\begin{aligned}\mathbf{a}_{opt} &= \frac{1}{\alpha P + 1}(\hat{X}^T\hat{X})^{-1}\hat{X}^T(\mathbf{y} + \alpha\mathbf{y}_1 + \alpha\mathbf{y}_2 + \ldots. + \alpha\mathbf{y}_P) = \\ &= \frac{1}{\alpha P + 1}\hat{X}^{\#}(\mathbf{y} + \alpha\mathbf{y}_1 + \alpha\mathbf{y}_2 + \ldots. + \alpha\mathbf{y}_P),\end{aligned} \tag{20}$$

where $\mathbf{y}_i$ is a vector of the outputs of the i-th fuzzy model (formed on a basis of D_i) where the corresponding coordinate of this vector produced the output obtained for the corresponding input, that is,

$$\mathbf{y}_i = \begin{bmatrix} FM[i](\mathbf{x}_1) \\ FM[i](\mathbf{x}_2) \\ \\ FM[i](\mathbf{x}_N) \end{bmatrix},$$

and where $\hat{X}^{\#}$ is a pseudoinverse of the data matrix.

An overall balance captured by (15) is achieved for a certain value of α. An evident tendency of increased impact becomes clearly visible: higher values of α stress higher relevance of other models and their more profound impact on the constructed model. First, the model is constructed on the basis of D. Second, the consistency is expressed on a basis of differences between the constructed model and those models coming from D_is where the differences are assessed with the use of data D. There is another interesting view of the format of this performance index under minimization. The second component in V plays a role that is similar to a *regularization* term typically used in estimation problems. However, its origin here has a substantially different format from the one encountered in the literature. Here, we consider other data (and models) rather than focusing on the complexity of the model expressed in terms of its parameters to evaluate the performance of the model.

While the semantics of the above performance index (15) is straightforward, a choice of the value of α requires some attention. To optimize the level of contribution coming from the datasets, we may adhere to the following evaluation process, which invokes two fundamental components. As usual, the quality of the optimal model is evaluated with respect to data D. The same optimized model (viz. its prototypes and the parameters of the local regression models) is made available at D_i and the quality of the model is evaluated there with the use of the local data present there. We combine the results (viz. the corresponding squared errors) by adding their normalized values. Given these motivating notes, an index quantifying a global behaviour of the optimal model arises in the following form

$$VV = \frac{1}{N}\sum_{\substack{X_K \in D \\ y_k \in D}}(FM(\mathbf{x}_k) - y_k)^2 + \sum_{j=1}^{P}\frac{1}{N_j}\sum_{\substack{x_k \in D_j \\ y_k \in D_j}}(FM(\mathbf{x}_k) - y_k)^2 .$$

(21)

A schematic view of computing and communication of findings realized with the aid of (21) is illustrated in Figure 8.2.

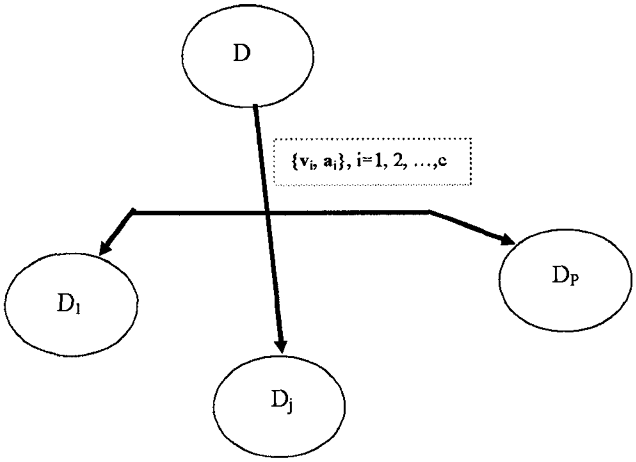


Figure 8.2 A quantification of the global behaviour of the consistency – based fuzzy model

Note that when the fuzzy model FM(.) is transferred, as before to D_j, we communicate, the prototypes obtained at D and the coefficients of the local linear models of the conclusion part of the rules. Likewise, as shown in (8.2), the output of the fuzzy model obtained for $\mathbf{x}_k \in D_j$ involves the induced value of membership degree $w_j(\mathbf{x}_k)$ and an aggregation of the local regression models.

Apparently the expression of VV is a function of α and the optimized level of consistency is such for which VV attains its minimal value, namely

$$\alpha_{opt} = \arg \mathrm{Min}\ \mathrm{VV}(\alpha). \tag{22}$$

The optimization scheme (15) along with its evaluation mechanisms governed by (21) can be generalized by admitting the various levels of impact that each data D_i might have in the process of achieving consistency. To do so, we introduce some positive weights, w_i, $i=1, 3, \ldots p$, which are afterwards used in the performance index

$$V = \sum_{\substack{x_k \in D \\ y_k \in D}} (FM(\mathbf{x}_k) - y_k)^2 + \alpha \sum_{j=1}^{P} w_j \sum_{\substack{x_k \in D \\ y_k \in D}} (FM(\mathbf{x}_k) - y_k)^2. \tag{23}$$

Lower values of w_i indicate lower influence of the model formed on a basis of data D_i when constructing the model for data D. The role of such weights is particularly apparent when dealing with data D_i , which are in some temporal or spatial relationships with respect to D. In these circumstances, the values of the weights reflect how far (in terms of time or distance) the sources of the individual data are from D. For instance, if D_j denotes a collection of data gathered some time ago in comparison to the currently collected data D_i, then it is intuitively clear that the weight w_j is lower than w_i.

For an auxiliary performance index that expresses a quality of the model for which (15) has been minimized with α being selected with regard to (22), we consider the following expression

$$Q^{\sim} = \frac{1}{N} \sum_{\substack{x_k \in D \\ y_k \in D}} (FM(\mathbf{x}_k) - y_k)^2. \tag{24}$$

The values of $Q^{\sim}$ considered vis-à-vis the results expressed by (24) are helpful in assessing the extent that the fuzzy model optimized with regard to data D while achieving consistency with D_1, D_2, ..., D_p will deteriorates when applied to D over the optimal model being optimized exclusively on a basis of D.

In what follows, we also introduce a computationally effective measure articulating a level of experience consistency obtained for D in the form of *granular* characterization of the parameters of local regression models. Before proceeding with the details, we elaborate a method in which individual rules existing in the models formed for D and the datasites D_1, D_2, ..., D_p are "synchronized" (aligned).

8.3.3 Alignment of information granules

The rules forming each fuzzy model have been formed independently at each datasite. If we intend to evaluate a level of consistency of the rules at D vis-à-vis the modeling evidence available at D_j, some alignment of the rules become essential. Such an alignment concerns a way of lining up the prototypes forming the condition part of the rules. We consider the models obtained at D and D_j, j=1, 2, …, P with their prototypes $\mathbf{v}_1$, $\mathbf{v}_2$, …, $\mathbf{v}_c$ and $\mathbf{v}_1[j]$, $\mathbf{v}_2[j]$,…, $\mathbf{v}_c[j]$. We say that the rule "i" at D and the rule "*l*" at D_j are aligned if the prototypes $\mathbf{v}_k$ and $\mathbf{v}_l[j]$ are the closest within the collections of the prototypes produced for D and D_j. The alignment process is realized by successively finding the pairs of the prototypes being characterized by the lowest mutual distance. Overall, the alignment process can be described in the following manner:

Form two sets of integers (indexes) **I** and **J**, where **I** = **J** = {1, 2, …,c}. Start with an empty list of alignments, L= ∅.

```
Repeat
    Find a pair of indexes $i_0$ and $j_0$ for which the distance attains minimum
        $(i_0, j_0) = \arg\min_{i,l} \|\mathbf{v}_i - \mathbf{v}_l(j)\|$
    The pair $(i_0, j_0)$ is added to the list of alignments, $L = L \cup (i_0, j_0)$
    Reduce the set of indexes I and J by removing the elements that were placed on
    the list of alignments, $\mathbf{I} = \mathbf{I} \setminus \{i_0\}$ and $\mathbf{J} = \mathbf{J} \setminus \{j_0\}$
until $I = \emptyset$
```

Once the above loop has been completed, we end up with the list of alignment of the prototypes in the form of pairs (i_1, j_1), (i_2, j_2),…, (i_c, j_c).

8.4 Granular parameters as a characterization of experience-consistent models

Once the mechanism of experience consistency has been completed and the local models have been aligned (following the scheme provided in the previous section), we can now look at the characterization of the set of the related parameters of the local regression models. In essence, through the alignment of the prototypes at D and D_j, we obtain the corresponding vectors of the parameters of the regression models of the conclusion parts. Denote these vectors corresponding to a certain rule by **a**, $\mathbf{a}_i$, $\mathbf{a}_k$, …, and $\mathbf{a}_l$ altogether arriving at D and Dj datasites. If we now consider the j-th coordinate of all of them, we obtain the numeric values a_j, a_{ij}, …, a_{lj}. The essence of their aggregation concerns their global representation completed in the form of a single fuzzy set. The datasites' modal value is just a_j ,while the membership function is reflective of the numeric values of the corresponding parameters of the local models. The idea introduced in [11] follows this observation. Let us consider a finite number of numeric values z_i={a_j, a_{ij}, …, a_{lj}}. We intend to span a unimodal fuzzy set A over data a_j in such a way that it represents all collaborative datasites to the highest possible extent. The form of the membership is also defined in advance. For instance, we could consider a certain type of membership functions, say triangular, Gaussian, parabolic, etc. Furthermore, we consider that a modal value of A_j, that is a_j , is given. Let us look at the values of a_{ji} that are lower than a_j, $a_{ji} < a_j$. Denote this set by Ω_-, $z_{i-} = \{a_{ji} \mid a_{ji} < a_j\}$.We use them to estimate the parameters of the left-hand side of the membership function. The determination of the right-hand side of the membership function is realized in an analogous manner by considering the set z_{i+} where $z_{i+} = \{a_{ji} \mid a_{ji} > a_j\}$. The method for computation of a membership function of fuzzy set of type-2 is followed in the same way as discussed in Chapter 4(Section 4.2.1).

The result of the aggregation becomes a triangular fuzzy number of the j-th parameter of the local regression model. Denote it by $A_j = (a_{j-}, a_j, a_{j+})$ with the three parameters denoting the lower, modal, and upper bound of the fuzzy number. Applying the same procedure to all remaining parameters of the vector **a**, we produce the corresponding fuzzy numbers A_1, A_2, …, A_{j-1}, A_{j+1}, …, A_n, and A_0. Given them, the rule in D reflects the nature of the incorporated evidence offered by the remaining models D_1, D_2, etc. If there is a fairly high level of consistency, this effect is manifested through a fairly "concentrated" fuzzy number. Increasing inconsistency results in a broader, less specific fuzzy number of the parameters. In summary, a certain fuzzy rule assumes the following format

$$\text{If x is B, then } Y = A_0 \oplus A_1 \otimes x_1 \oplus A_2 \otimes x_2 \oplus \ldots \oplus A_n \otimes x_n \tag{25}$$

The symbols ⊕ and ⊗ being used above underline the nonnumeric nature of the arguments standing in the model over which the multiplication and addition are carried out. For given numeric inputs $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$, the resulting output Y of this local regression model is again a triangular fuzzy number $Y = <w, y, z>$ where their parameters are computed as follows

Modal value $y = a_0 + a_1x_1 + a_2x_2 + \ldots + a_nx_n$

Lower bound $w = a_0 + \min(a_{1-}x_1, a_{1+}x_1) + \min(a_{2-}x_2, a_{2+}x_2) + \ldots + \min(a_{n-}x_n, a_{n+}x_n)$

Upper bound $z = a_0 + \max(a_{1-}x_1, a_{1+}x_1) + \max(a_{2-}x_2, a_{2+}x_2) + \ldots + \max(a_{n-}x_n, a_{n+}x_n)$

The above process represents the formation of the fuzzy numbers of the local regression model of the rule is repeated for all rules. At the end we arrive at the rules of the form

$$\begin{gathered}\text{If x is } B_1 \text{ then } Y = A_{10} \oplus A_{11} \otimes x_1 \oplus A_{12} \otimes x_2 \oplus \ldots \oplus A_{1n} \otimes x_n \\ \text{If x is } B_2 \text{ then } Y = A_{20} \oplus A_{21} \otimes x_1 \oplus A_{22} \otimes x_2 \oplus \ldots \oplus A_{2n} \otimes x_n \\ \ldots. \\ \text{If x is } B_c \text{ then } Y = A_{c0} \oplus A_{c1} \otimes x_1 \oplus A_{c2} \otimes x_2 \oplus \ldots \oplus A_{cn} \otimes x_n\end{gathered} \tag{26}$$

Given this structure, the input vector **x** implies the output fuzzy set with the following membership function

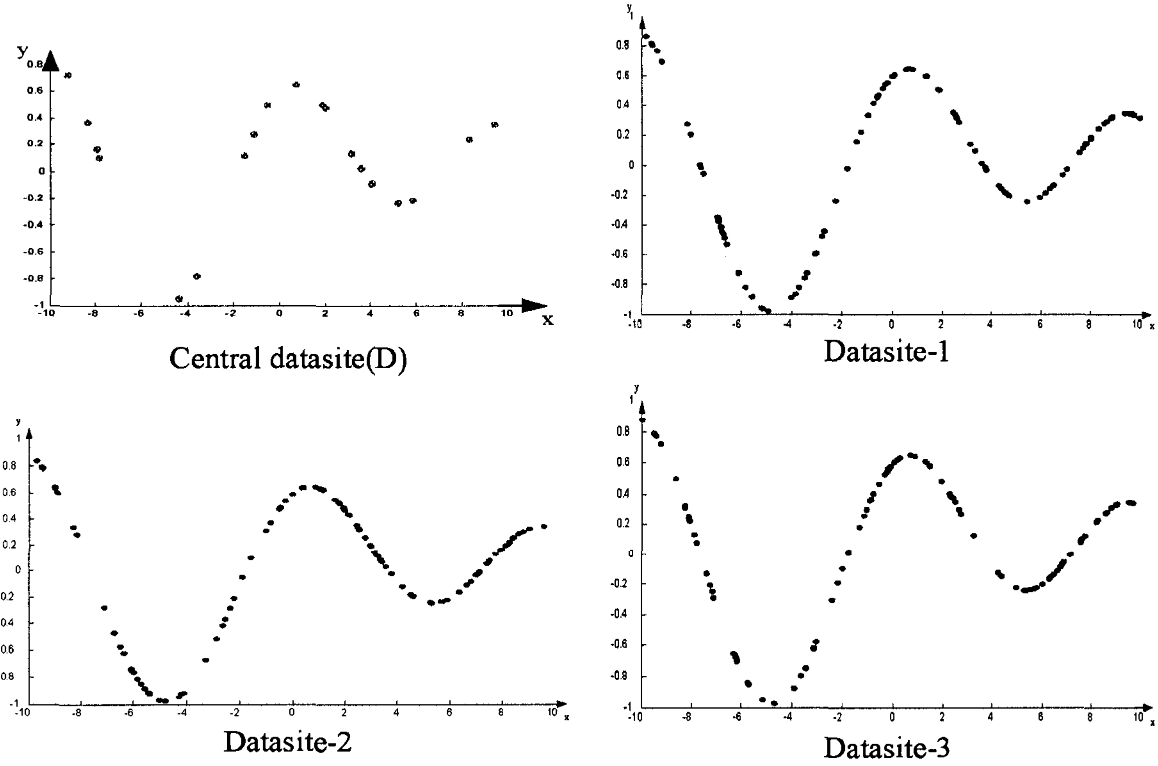
$$Y=\sum_{i=1}^{c} w_i(\mathbf{x}\ )\otimes[A_{i0}\oplus(A_{i1}\otimes x_1)\oplus(A_{i2}\otimes x_2)\oplus\ldots\oplus(A_{in}\otimes x_n)] \tag{27}$$

Owing to the fact of having fuzzy sets of the parameters of the regression model in the conclusion part of the rules, Y becomes a fuzzy number rather than a single numeric value.

8.5 Experimental studies

We start with several one-dimensional synthetic data sets that are helpful in illustrating the very nature of the problem and quantify the effect of building consistency under various circumstances.

Synthetic data In this series of experiments, we consider a collection of synthetic data D, D_j, j=1, 2, …, 5 The plots of data D, D_1- D_5 are illustrated in Figure 8.3. Each of them consists of 100 data points. In contrast, D is far smaller as it consists of 20 data. Furthermore, it is also affected by Gaussian noise of zero mean value and some standard deviation. Some examples of the dataset D are presented in Figure 8.4. For each dataset we consider c = 4 rules, which reflect the structure of data. We may anticipate that this number of rules could be capable of modeling the data to a high extent.

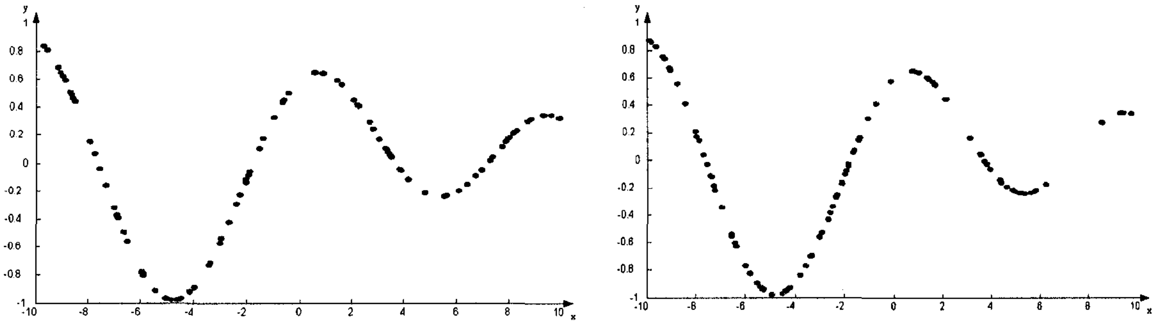


Central datasite(D)

Datasite-1

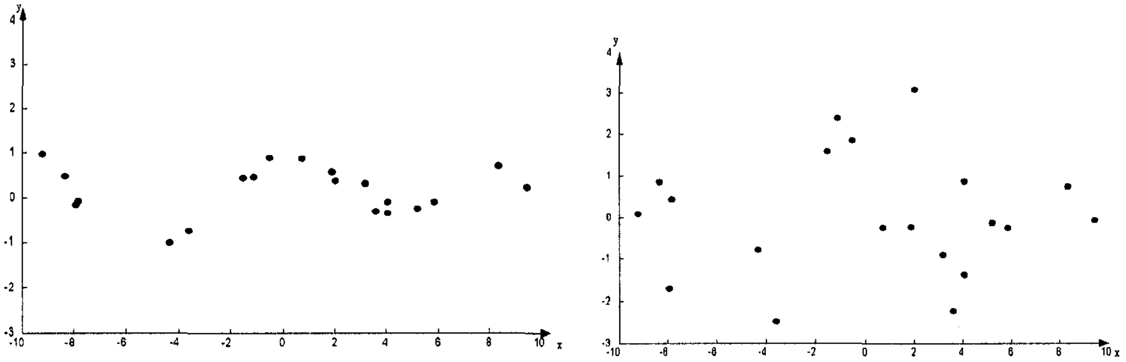
Datasite-2

Datasite-3



Datasite-4 Datasite-5

Figure 8.3 Data sets used in the experiment.



(a) (b)

Figure 8.4 Datasite (D) affected by two levels of noise: $\sigma = 0.2$ (a) and $\sigma=1.5$ (b)

Since we repeated the experiments for each noise level 10 times, this organization of the experiments helps to reveal any meaningful tendency that might exist. The obtained results are summarized in Table 8.1 where we report the values of VV for $\alpha = 0$ and α_{opt}. The results are provided in terms of the mean value and the standard deviation. Given the results, the tendency becomes quite straightforward as to the level of noise and the optimal value of α: higher noise level comes with higher values of α. It is not surprising that as the quality of data deteriorates (higher level of noise), we anticipate a somewhat stronger impact from other data sites to compensate for the lower quality of the data.

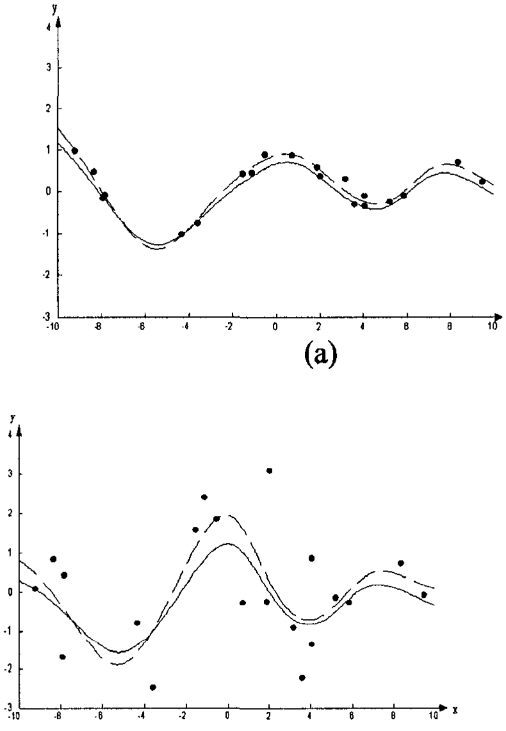
Table 8.1 Performance index VV (mean and standard deviation) for $\alpha = 0$ and the optimal values of α.

	σ=0.0	σ=0.1	σ=0.2	σ=0.3	σ=0.4	σ=0.5
VV(α =0)	0.003	0.04 ± 0.02	0.20 ± 0.13	0.33 ± 0.17	0.62 ± 0.33	0.77 ± 0.27
VV(α opt)	0.003	0.04 ± 0.02	0.17 ± 0.08	0.29 ± 0.17	0.52 ± 0.25	0.67 ± 0.18
α opt	0.001	0.001 ± 0.002	0.011 ± 0.012	0.015 ± 0.009	0.023 ± 0.016	0.021 ± 0.019

	σ =0.6	σ =0.7	σ =0.8	σ =0.9	σ =1.0
VV(α =0)	1.48 ± 0.61	2.24 ± 0.99	2.48 ± 1.78	2.68 ± 1.07	2.30 ± 1.41
VV(α opt)	1.09 ± 0.37	1.56 ± 0.68	1.70 ± 1.03	2.04 ± 0.96	1.95 ± 1.11
α opt	0.05 ± 0.03	0.07 ± 0.03	0.07 ± 0.05	0.07 ± 0.03	0.06 ± 0.05

	σ =1.1	σ =1.2	σ =1.3	σ =1.4	σ =1.5
VV(α =0)	3.71 ± 1.35	4.63 ± 2.28	4.79 ± 2.11	6.39 ± 2.59	9.77 ± 5.27
VV(α opt)	2.8616 ± 0.95	3.05 ± 0.78	3.34 ± 1.31	3.85 ± 1.26	5.01 ± 1.54
α opt	0.09 ± 0.05	0.10 ± 0.07	0.13 ± 0.09	0.19 ± 0.11	0.36 ± 0.44

The plot of the fuzzy model for α = 0 for the two levels of noise presented in Figures 8.4 is incorporated in Figure 8.5 that also illustrate the model in case of the optimal value of α.



(a)

(b)

Figure 8.5 Plot of the fuzzy model for D: for α =0 (dotted lines) and α_{opt} (solid lines) for two levels of noise, σ = 0.2 (a) and σ=1.5 (b).

By incorporating the knowledge about parameters of the models of D_is we end up with triangular fuzzy numbers of the parameters, shown in Figure 8.6.

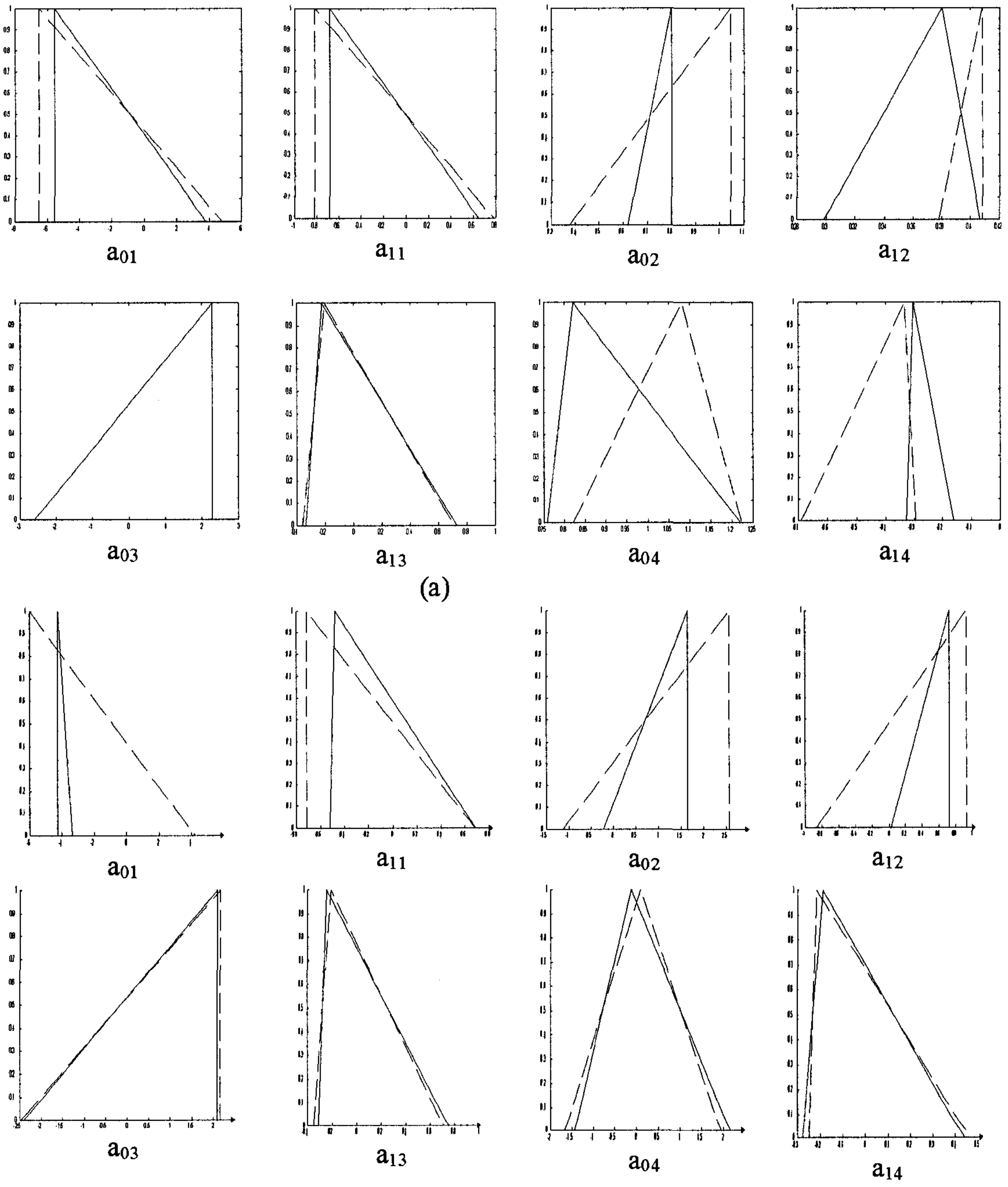
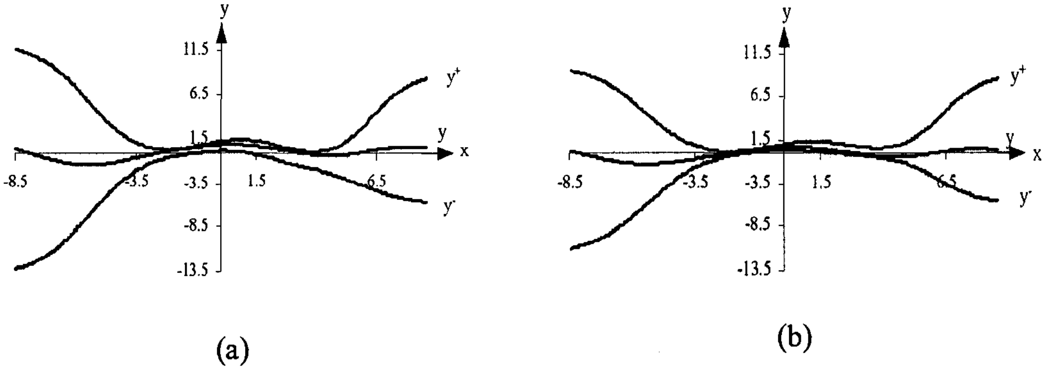


Figure 8.6 Triangular fuzzy numbers of the parameters of the rule-based model for σ = 0.2 (a) and σ=1.5 (b). Dotted lines represent α =0, while the solid lines are obtained for α_{opt}.

The use of granular (fuzzy) parameters in the model results in the granular form of the output (expressed in terms of triangular fuzzy numbers). The bounds (lower and upper bound, respectively) are illustrated in Figure 8.7. Notably, the bounds are quite tight over almost the entire range of the input variable. The substantial spread of the range occurs for the values outside the lowest and highest prototypes formed in the input space.



(a) (b)

Figure 8.7 The bounds of the output triangular fuzzy set (also shown is its modal value) for the model before (a) and after the completion of experience-consistent optimization.

Machine Learning data In the ensuing series of experiments, we consider real-world data available at StatLib and Machine Learning repositories. The main characteristics of data used here along with their origin are summarized in Table 8.2.

Table 8.2 A suite of experimental data.

No.	Dataset	Source	Number of data (N)	Number of features (n)
1	Abalone	UCI- Web*	4,177	8
2	California House	StatLib Repository – Web$^{\&}$	20,640	8
3	Auto-mpg	UCI- Web*	398	7
4	Boston Housing	UCI- Web*	506	13

Note: Web*: http://www.ics.uci.edu/~mlearn/MLRepository.html.
Web$^{\&}$: http://lib.stat.cmu.edu/

In the experiments we use 5% of all the data to form D, while the rest of the data set is split into equal parts whose size depends upon the assumed value of P forming in this way D_1, D_2, …, and D_P. We also experiment with a collection of datasites (P) contributing to the experience-based modeling. The number of clusters (c) was varied in-between 2 and 6. For each dataset from the above list, we report in a tabular fashion the values of VV for α =0 and α_{opt}. The optimization with respect to α is carried out by running the experiment for different values of this parameters starting with α =0.0 and successively increasing its value up to the point we have reached the minimal value of the index.

	c=2	c=3	c=4	c=5	c=6
VV(α=0)	9.43 ± 0.85	9.22 ± 0.85	9.19 ± 0.87	9.52 ± 0.97	9.76 ± 0.95
VV(αopt)	9.16 ± 0.78	8.96 ± 0.72	8.83 ± 0.62	9.22 ± 0.86	9.39 ± 0.85
αopt	0.816 ± 0.788	0.289 ± 0.203	0.398 ± 0.695	0.206 ± 0.103	0.183 ± 0.084
κ	0.97	0.97	0.96	0.97	0.96

P=1

	c=2	c=3	c=4	c=5	c=6
VV(α=0)	14.83 ± 1.04	14.78 ± 1.18	14.95 ± 1.41	15.84 ± 1.67	16.47 ± 1.54
VV(αopt)	14.42 ± 0.91	13.95 ± 0.83	14.11 ± 0.69	14.39 ± 0.97	15.29 ± 1.45
αopt	0.187 ± 0.111	0.329 ± 0.137	0.183 ± 0.156	0.327 ± 0.259	0.168 ± 0.074
κ	0.97	0.94	0.94	0.91	0.93

P=2

	c=2	c=3	c=4	c=5	c=6
VV(α=0)	20.22 ± 1.24	20.34 ± 1.53027	20.70 ± 1.99	22.16 ± 2.41	23.17 ± 2.16
VV(αopt)	19.23 ± 1.08	18.69 ± 0.92	18.79 ± 1.10	19.74 ± 1.28	21.23 ± 2.02
αopt	0.458 ± 0.433	0.380 ± 0.261	0.302 ± 0.227	0.252 ± 0.166	0.150 ± 0.058
κ	0.95	0.92	0.91	0.89	0.92

P=3

	c=2	c=3	c=4	c=5	c=6
VV(α=0)	25.62 ± 1.46	25.90 ± 1.92	26.45 ± 2.60	28.48 ± 3.19	29.87 ± 2.81
VV(αopt)	24.28 ± 1.15	23.75 ± 1.11	23.78 ± 0.81	24.71 ± 1.77	26.28 ± 2.20
αopt	0.380 ± 0.431	0.303 ± 0.223	0.257 ± 0.227	0.259 ± 0.094	0.153 ± 0.070
κ	0.95	0.92	0.90	0.87	0.88

P=4

	c=2	c=3	c=4	c=5	c=6
VV(α=0)	31.01 ± 1.66	31.46 ± 2.30	32.20 ± 3.21	34.79 ± 3.95	36.55 ± 3.45
VV(αopt)	29.8249 ± 1.41	28.8912 ± 1.18	28.9101 ± 1.41	30.4058 ± 2.02	31.91 ± 2.09
αopt	0.178 ± 0.303	0.186 ± 0.080	0.180 ± 0.058	0.173 ± 0.076	0.141 ± 0.062
κ	0.96	0.92	0.90	0.87	0.87

P=5

	c=2	c=3	c=4	c=5	c=6
VV(α =0)	57.95 ± 2.78	59.24 ± 4.28	60.94 ± 6.27	66.35 ± 7.85	69.99 ± 6.70
VV(α opt)	55.06 ± 1.74	53.34 ± 1.77	53.06 ± 2.22	56.73 ± 3.56	60.75 ± 5.25
α opt	0.067 ± 0.038	0.122 ± 0.073	0.160 ± 0.150	0.131 ± 0.105	0.066 ± 0.024
κ	0.95	0.90	0.87	0.86	0.87

P=10

Abalone

	c=2	c=3	c=4	c=5	c=6
VV(α=0)	9.43×10^9 ± 1.6×10^8	9.07×10^9 ± 1.86×10^8	8.99×10^9 ± 2.01×10^8	8.89×10^9 ± 2.12×10^8	8.83×10^9 ± 2.11×10^8
VV(αopt)	9.40×10^9 ± 1.64×10^8	9.04×10^9 ± 1.86×10^8	8.98×10^9 ± 2.00×10^8	8.86×10^9 ± 2.23×10^8	8.81×10^9 ± 2.14×10^8
αopt	0.366 ± 0.470	0.289 ± 0.468	0.071 ± 0.038	0.090 ± 0.141	0.071 ± 0.027
κ	1.00	1.00	1.00	1.00	1.00

P=1

	c=2	c=3	c=4	c=5	c=6
VV(α=0)	1.43×10^{10} ± 1.76×10^8	1.38×10^{10} ± 1.80×10^8	1.38×10^{10} ± 2.05×10^8	1.36×10^{10} ± 2.35×10^8	1.36×10^{10} ± 2.49×10^8
VV(αopt)	1.42×10^{10} ± 1.47×10^8	1.38×10^{10} ± 1.74×10^8	1.37×10^{10} ± 2.03×10^8	1.36×10^{10} ± 2.37×10^8	1.35×10^{10} ± 2.24×10^8
αopt	0.437 ± 0.438	0.128 ± 0.126	0.055 ± 0.031	0.062 ± 0.033	0.070 ± 0.030
κ	0.99	1.00	0.99	1.00	0.99

P=2

	c=2	c=3	c=4	c=5	c=6
VV(α=0)	1.92×10^{10} ± 1.98×10^8	1.86×10^{10} ± 2.08×10^8	1.85×10^{10} ± 2.43×10^8	1.84×10^{10} ± 2.93×10^8	1.84×10^{10} ± 3.36×10^8
VV(αopt)	1.90×10^{10} ± 1.22×10^8	1.85×10^{10} ± 1.73×10^8	1.84×10^{10} ± 2.24×10^8	1.85×10^{10} ± 2.43×10^8	1.82×10^{10} ± 2.73×10^8
αopt	0.466 ± 0.493	0.082 ± 0.054	0.045 ± 0.025	0.039 ± 0.020	0.053 ± 0.020
κ	0.99	0.99	0.99	1.01	0.99

P=3

	c=2	c=3	c=4	c=5	c=6
VV(α=0)	2.41×10^{10} $\pm 2.34\times10^{8}$	2.34×10^{10} $\pm 2.57\times10^{8}$	2.33×10^{10} $\pm 3.004\times10^{8}$	2.31×10^{10} $\pm 3.68\times10^{8}$	2.3×10^{10} $\pm 4.47\times10^{8}$
VV(αopt)	2.38×10^{10} $\pm 1.66\times10^{8}$	2.32×10^{10} $\pm 1.74\times10^{8}$	2.31×10^{10} $\pm 2.41\times10^{8}$	2.30×10^{10} $\pm 3.15\times10^{8}$	2.30×10^{10} $\pm 4.09\times10^{8}$
αopt	0.385 ± 0.553	0.075 ± 0.046	0.066 ± 0.042	0.036 ± 0.017	0.036 ± 0.012
κ	0.99	0.99	0.99	1.00	1.00

P=4

	c=2	c=3	c=4	c=5	c=6
VV(α=0)	$2.90\ \times10^{10}$ $\pm 2.79\times10^{8}$	2.81×10^{10} $\pm 3.20\times10^{8}$	2.80×10^{10} $\pm 3.72\times10^{8}$	2.79×10^{10} $\pm 4.54\times10^{8}$	2.79×10^{10} $\pm 5.66\times10^{8}$
VV(αopt)	2.87×10^{10} $\pm 1.45\times10^{8}$	2.79×10^{10} $\pm 2.76\times10^{8}$	2.78×10^{10} $\pm 3.16\times10^{8}$	2.77×10^{10} $\pm 3.83\times10^{8}$	2.76×10^{10} $\pm 4.04\times10^{8}$
αopt	0.200 ± 0.198	0.055 ± 0.027	0.051 ± 0.042	0.031 ± 0.014	0.046 ± 0.017
κ	0.99	0.99	0.99	0.99	0.99

P=5

	c=2	c=3	c=4	c=5	c=6
VV(α=0)	5.34×10^{10} $\pm 5.59\times10^{8}$	5.20×10^{10} $\pm 6.92\times10^{8}$	5.19×10^{10} $\pm 7.80\times10^{8}$	5.17×10^{10} $\pm 9.31\times10^{8}$	5.17×10^{10} $\pm 1.20\times10^{8}$
VV(αopt)	5.28×10^{10} $\pm 2.39\times10^{8}$	5.15×10^{10} $\pm 4.59\times10^{8}$	5.14×10^{10} $\pm 5.29\times10^{8}$	5.12×10^{10} $\pm 6.97\times10^{8}$	5.11×10^{10} $\pm 8.94\times10^{8}$
αopt	0.105 ± 0.096	0.038 ± 0.021	0.026 ± 0.019	0.020 ± 0.011	0.024 ± 0.012
κ	0.99	0.99	0.99	0.99	0.99

P=10

California Housing

	c=2	c=3	c=4	c=5	c=6
VV(α=0)	0.87×10^{2} $\pm 0.67\times10^{2}$	5.63×10^{4} $\pm 6.17\times10^{4}$	1.23×10^{7} $\pm 2.89\times10^{7}$	2.17×10^{9} $\pm 6.23\times10^{9}$	5.87×10^{12} $\pm 1.86\times10^{13}$
VV(αopt)	0.25×10^{2} $\pm 0.15\times10^{2}$	1.93×10^{4} $\pm 3.13\times10^{4}$	8.22×10^{6} $\pm 2.08\times10^{7}$	1.86×10^{9} $\pm 5.29\times10^{9}$	1.88×10^{8} $\pm 4.49\times10^{8}$
αopt	7.079 ± 6.422	7.175 ± 4.618	1.810 ± 4.624	4.474 ± 6.461	6.344 ± 6.690
κ	0.287	0.343	0.668	0.857	0.000

P=1

	c=2	c=3	c=4	c=5	c=6
VV(α=0)	1.73×10^{2} $\pm 1.34\times10^{2}$	9.16×10^{4} $\pm 1.03\times10^{5}$	2.36×10^{7} $\pm 5.71\times10^{7}$	3.57×10^{9} $\pm 1.00\times10^{10}$	1.18×10^{13} $\pm 3.73\times10^{13}$
VV(αopt)	0.53×10^{2} $\pm 0.25*10^{2}$	3.62×10^{4} $\pm 5.82*10^{4}$	2.08×10^{7} $\pm 5.65*10^{7}$	3.55×10^{9} $\pm 1.00*10^{10}$	1.18×10^{13} $\pm 3.73*10^{13}$
αopt	3.249 ± 4.660	5.552 ± 4.093	3.066 ± 5.383	6.216 ± 6.897	2.150 ± 4.141
κ	0.306	0.395	0.881	0.994	1.000

P=2

	c=2	c=3	c=4	c=5	c=6
VV(α =0)	2.59×10^{2} $\pm 2.01\times10^{2}$	1.27×10^{5} $\pm 1.44\times10^{5}$	3.51×10^{7} $\pm 8.60\times10^{7}$	4.94×10^{9} $\pm 1.38\times10^{10}$	1.78×10^{13} $\pm 5.63\times10^{13}$
VV(α opt)	0.75×10^{2} $\pm 0.25\times10^{2}$	4.91×10^{4} $\pm 7.53\times10^{4}$	2.83×10^{7} $\pm 8.49\times10^{7}$	4.6×10^{9} $\pm 1.28\times10^{10}$	2.32×10^{9} $\pm 7.07\times10^{9}$
α opt	2.572 ± 4.521	4.780 ± 5.058	1.856 ± 3.075	3.926 ± 6.315	1.924 ± 4.507
κ	0.290	0.387	0.806	0.931	0.000

P=3

	c=2	c=3	c=4	c=5	c=6
VV(α =0)	3.44×10^{2} $\pm 2.69\times10^{2}$	1.62×10^{5} $\pm 1.86\times10^{5}$	4.63×10^{7} $\pm 1.1410^{8}$	6.4410^{9} $\pm 1.7910^{10}$	2.3510^{13} $\pm 7.4410^{13}$
VV(α opt)	1.07×10^{2} ± 41.15	6.50×10^{4} $\pm 1.0310^{5}$	3.66×10^{7} $\pm 1.0610^{8}$	6.19×10^{9} $\pm 1.7210^{10}$	6.19×10^{9} $\pm 1.7410^{10}$
α opt	2.158 ± 4.536	8.495 ± 4.936	5.684 ± 6.176	1.44 ± 4.503	1.524 ± 4.616
κ	0.311	0.401	0.790	0.961	0.000

P=4

	c=2	c=3	c=4	c=5	c=6
VV(α =0)	4.31×10^{2} $\pm 3.35\times10^{2}$	1.97×10^{5} $\pm 2.27\times10^{5}$	5.66×10^{7} $\pm 1.40\times10^{8}$	7.88×10^{9} $\pm 2.18\times10^{10}$	2.88×10^{13} $\pm 9.12\times10^{13}$
VV(α opt)	1.00×10^{2} ± 32.2711	9.15×10^{4} $\pm 1.45\times10^{5}$	3.53×10^{7} $\pm 9.47\times10^{7}$	5.10×10^{9} $\pm 1.32\times10^{10}$	2.88×10^{13} $\pm 9.12\times10^{13}$
α opt	2.241 ± 4.510	4.501 ± 5.977	5.364 ± 5.375	2.257 ± 4.292	0.168 ± 0.422
κ	0.232	0.464	0.624	0.647	1.000

P=5

	c=2	c=3	c=4	c=5	c=6
VV(α =0)	8.60×10^{2} $\pm 6.77\times10^{2}$	3.76×10^{5} $\pm 4.41\times10^{5}$	1.14×10^{8} $\pm 2.85\times10^{8}$	1.57×10^{10} $\pm 4.37\times10^{10}$	5.45×10^{13} $\pm 1.72\times10^{14}$
VV(α opt)	2.08×10^{2} ± 93.50	1.822×10^{5} $\pm 3.2\times10^{5}$	9.01×10^{7} $\pm 2.25\times10^{8}$	1.56×10^{10} $\pm 4.37\times10^{10}$	1.70×10^{12} $\pm 5.37\times10^{12}$
α opt	2.332 ± 4.600	7.339 ± 6.768	1.094 ± 2.297	0.006 ± 0.008	0.0005 ± 0.001
κ	0.242	0.485	0.790	0.994	0.031

P=10

Auto-mpg

	c=2	c=3	c=4	c=5	c=6
VV(α =0)	5.10×10^{5} $\pm 1.45\times10^{6}$	5.20×10^{6} $\pm 9.11\times10^{6}$	2.61×10^{8} $\pm 7.49\times10^{8}$	1.03×10^{9} $\pm 2.16\times10^{9}$	1.63×10^{10} $\pm 4.83\times10^{10}$
VV(α opt)	6.52×10^{3} $\pm 1.51\times10^{4}$	3.36×10^{6} $\pm 6.01\times10^{6}$	2.47×10^{8} $\pm 7.35\times10^{8}$	8.69×10^{8} $\pm 1.83\times10^{9}$	3.66×10^{8} $\pm 6.27\times10^{8}$
α opt	10.737 ± 4.867	7.024 ± 7.402	5.426 ± 6.264	2.532 ± 4.929	1.796 ± 2.954
κ	0.013	0.646	0.946	0.844	0.022

P=1

	c=2	c=3	c=4	c=5	c=6
VV(α =0)	1.01×10^{6} ± 2.90×10^{6}	9.28×10^{6} ± 1.63×10^{7}	5.13×10^{8} ± 1.49×10^{9}	2.03×10^{9} ± 4.26×10^{9}	3.22×10^{10} ± 9.54×10^{10}
VV(α opt)	2.95×10^{3} ± 3.69×10^{3}	7.15×10^{6} ± 1.52×10^{7}	5.03×10^{8} ± 1.49×10^{9}	2.02×10^{9} ± 4.26×10^{9}	8.3410^{8} ± 1.83×10^{9}
α opt	10.585 ± 4. 684	4.462 ± 6.962	1.416 ± 4.460	0.021 ± 0.038	3.943 ± 5.913
κ	0.003	0.770	0.981	0.995	0.026

P=2

	c=2	c=3	c=4	c=5	c=6
VV(α =0)	1.53×10^{6} ± 4.40×10^{6}	1.39×10^{7} ± 2.47×10^{7}	7.67×10^{8} ± 2.23×10^{9}	2.90×10^{9} ± 6.08×10^{9}	4.79×10^{10} ± 1.42×10^{11}
VV(α opt)	5.00×10^{3} ± 5.77×10^{3}	1.10×10^{7} ± 2.24×10^{7}	7.41×10^{8} ± 2.24×10^{9}	3.28×10^{8} ± 8.02×10^{8}	4.78×10^{10} ± 1.42×10^{11}
α opt	10.253 ± 4.936	1.098 ± 2.918	3.144 ± 4.552	2.712 ± 5.351	0.002 ± 0.004
κ	0.003	0.791	0.966	0.113	0.998

P=3

	c=2	c=3	c=4	c=5	c=6
VV(α =0)	2.00×10^{6} ± 5.74×10^{6}	1.46×10^{7} ± 2.32×10^{7}	9.97×10^{8} ± 2.91×10^{9}	3.77×10^{9} ± 7.89×10^{9}	6.39×10^{10} ± 1.90×10^{11}
VV(α opt)	5.96×10^{3} ± 7.82×10^{3}	1.10×10^{7} ± 2.01×10^{7}	9.77×10^{8} ± 2.92×10^{9}	2.31×10^{9} 6.04×10^{9}	6.19×10^{10} ± 1.91×10^{11}
α opt	8.568 ± 5.395	2.516 ± 3.592	0.200 ± 0.604	1.468 ± 4.559	1.383 ± 4.320
κ	0.003	0.753	0.980	0.613	0.969

P=4

	c=2	c=3	c=4	c=5	c=6
VV(α =0)	2.33×10^{6} ± 6.65×10^{6}	1.89×10^{7} ± 3.10×10^{7}	1.22×10^{9} ± 3.58×10^{9}	4.67×10^{9} ± 9.86×10^{9}	8.18×10^{10} ± 2.44×10^{11}
VV(α opt)	7.31×10^{3} ± 8.70×10^{3}	1.52×10^{7} ± 2.79×10^{7}	1.21×10^{9} ± 3.58×10^{9}	4.58×10^{9} ± 9.72×10^{9}	7.44×10^{10} ± 2.21×10^{11}
α opt	9.278 ± 4.267	0.433 ± 1.251	0.796 ± 2.502	0.443 ± 1.370	0.698 ± 2.122
κ	0.003	0.804	0.992	0.981	0.910

P=5

	c=2	c=3	c=4	c=5	c=6
VV(α =0)	4.44×10^{6} ± 1.28×10^{7}	3.37×10^{7} ± 5.35×10^{7}	2.59×10^{9} ± 7.67×10^{9}	7.21×10^{9} ± 1.51×10^{10}	2.07×10^{11} ± 6.23×10^{11}
VV(α opt)	1.87×10^{4} ± 1.98×10^{4}	2.56×10^{7} ± 4.66×10^{7}	2.49×10^{9} ± 7.71×10^{9}	6.75×10^{9} ± 1.42×10^{10}	2.07×10^{11} ± 6.23×10^{11}
α opt	5.793 ± 5.509	1.951 ± 4.00	0.0009 ± 0.001	0.0002 ± 0.0004	0 0
κ	0.004	0.760	0.961	0.936	1.000

P=10

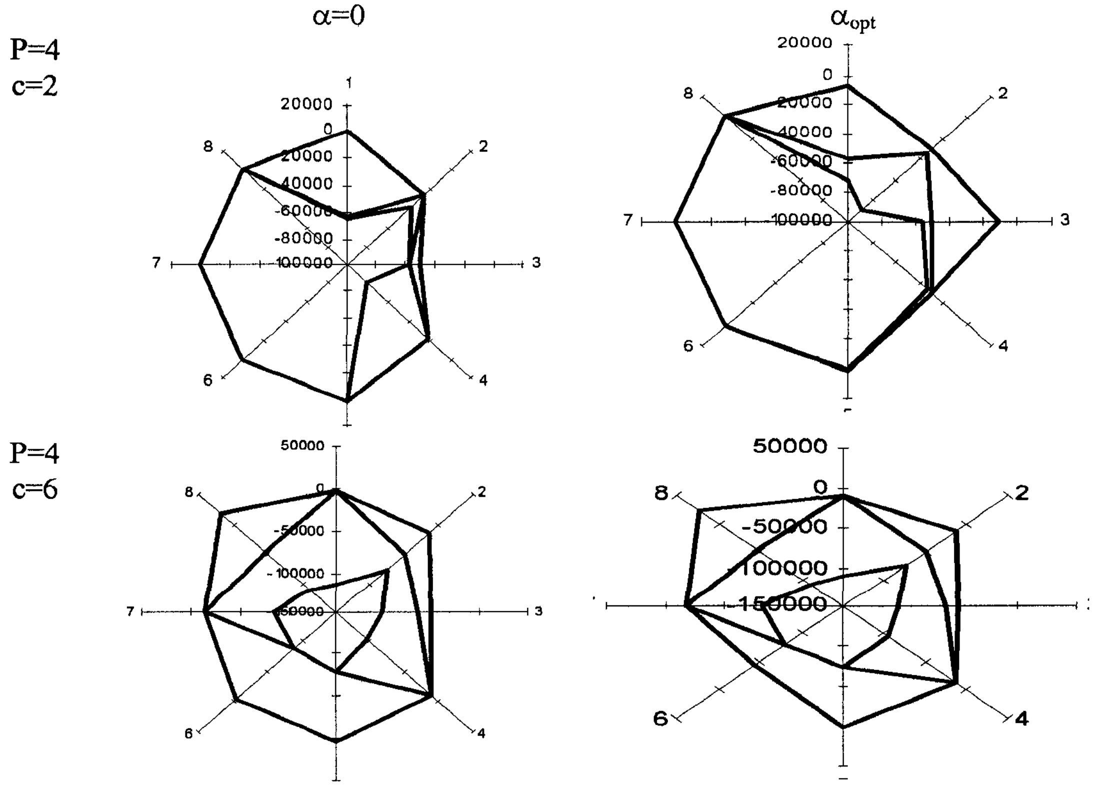
Boston Housing

Interestingly, in all cases we observe a substantial reduction of the values of VV(α_{opt}) over the values VV(α=0). A ratio $\kappa = \frac{VV(\alpha_{opt})}{VV(\alpha = 0)}$ can serve as a suitable indicator that quantifies the achieved improvement resulting from the mechanism of incorporating experience consistency. Overall, for all the data presented so far, the values of κ range in-between 0.0029 and 0.9978 with an average value assuming 0.780525. More specifically, for each data the results are as follows:

Abalone: κ ranges in-between 0.855 and 0.973 with the average equal to 0.924
California housing: κ is in the range of 0.9853and 0.997 where the average is 0.993
Auto-mpg: κ is in 0.2317and 0.994 and the average is equal to 0.586
Boston housing: κ is in-between 0.0029 and 0.998, average is 0.619

On average the most improvement is noted for the Auto-mpg data set. Similar average improvement is reported for the Boston housing data, while less reduction of κ has occurred for the Abalone and California housing.

The plots shown in Figure 8.8 illustrate radar plots of the parameters of linear model occurring in the first rule of the rule-based system. Notably, a number of these parameters have changed their values as a result of the experience-oriented modeling.



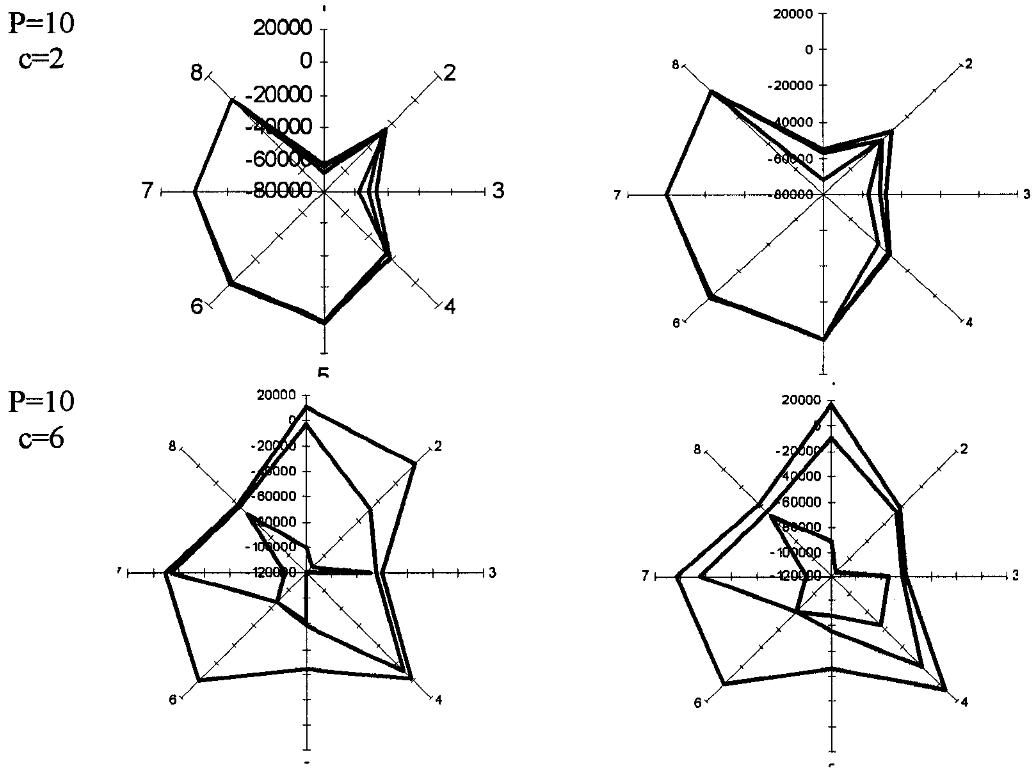
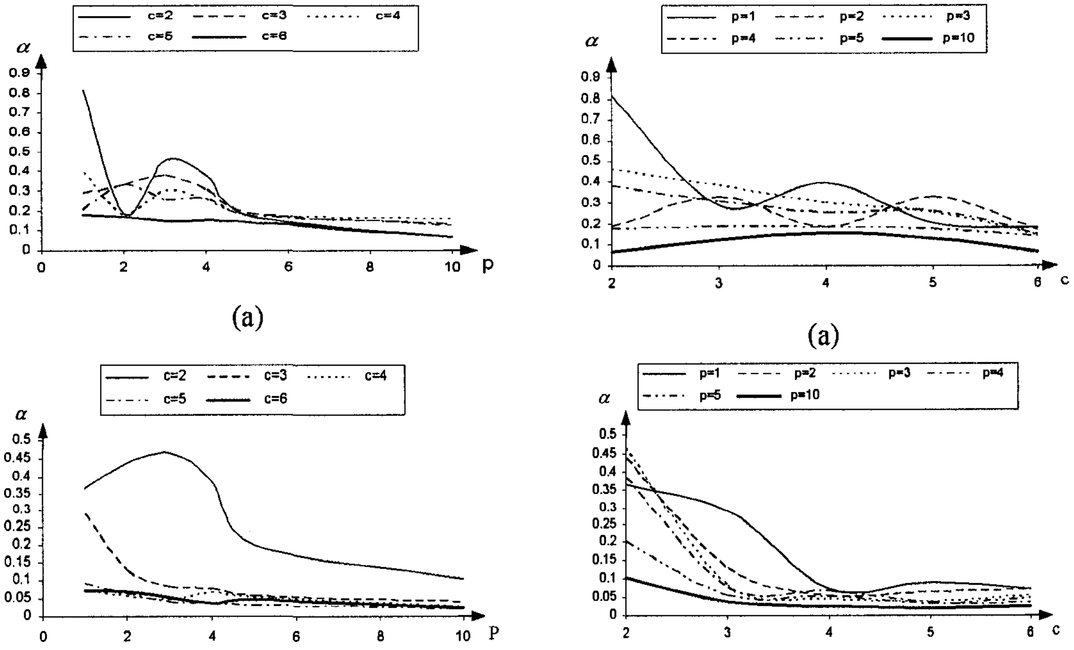


Figure 8.8 Radar-Plot for California Housing dataset: the parameters of the first rule (local model of its conclusion part).

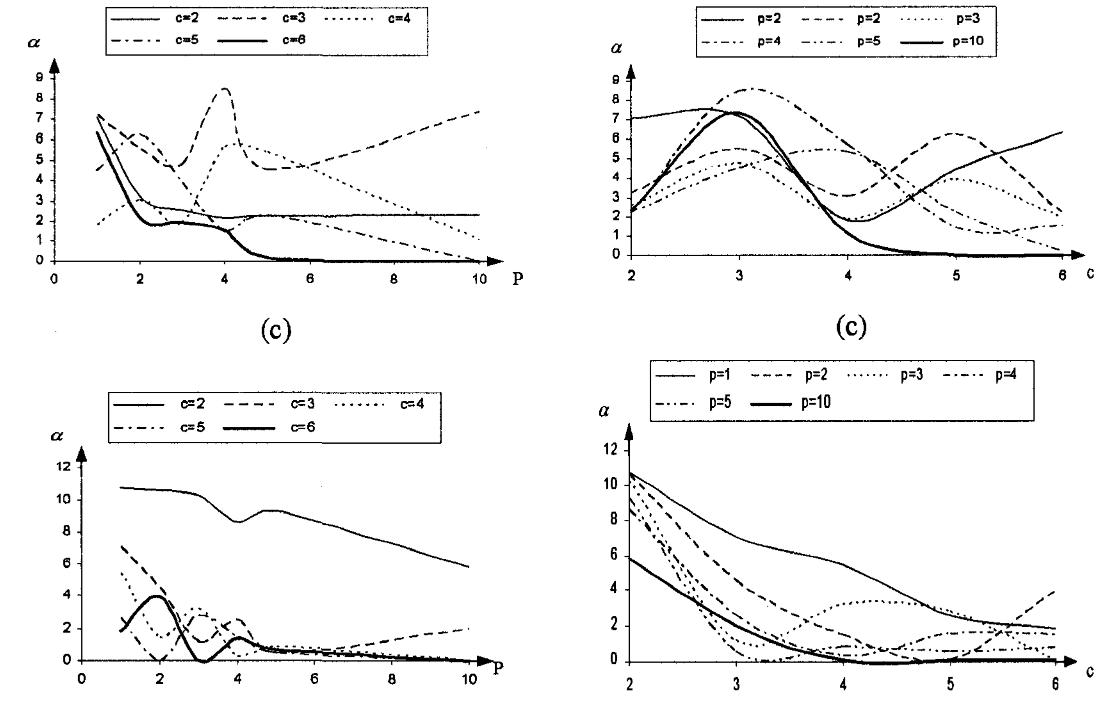


(a)

(a)

(b)

(b)



(c) (c)

(d) (d)

Figure 8.9 Optimal values of alpha versus P indexed by c (Column1), Optimal values of alpha versus c indexed by P (b) for datasets : Abalone(a), California Housing (b), Auto-mpg(c), and Boston Housing(d).

As illustrated in a series of graphs, Figure 8.9, some general tendency could be observed for a number of datasets. First, for a given number of models being available (P), the level of optimal level of acceptance of guidance from these models tends to assume lower values (a decreasing tendency reported for the optimal values of α). There are some occasional departures from this tendency, particularly for the lower values of "c". For a fixed number of rules (c), we note that while increasing the number of datasites, the optimal value of α becomes lower. These observed trends appeal to our intuition: if we encounter a larger number of models or the models are expressed by means of a larger number of rules, the level of reliance coming from them tends to become reduced. This could be partially caused by the fact that there is a higher level of diversity between the models, so we reduce their contributing role in the development of the fuzzy model and focus to a higher extent on the data D.

8.6 Conclusions

In this chapter, we have developed an approach to fuzzy rule-based model identification realized in a collaborative framework of data (experiential evidence) and past experience (knowledge evidence). We demonstrated how to reconcile these two essential sources of guidance in the form of local regression models. In particular, one could note that the knowledge-based component (previously constructed models) can serve as a certain form of the regularization mechanism encountered frequently in various modeling platforms. A level of achieved consistency is expressed in terms of fuzzy sets of the regression parameters of the local models occurring in the conclusions of the rules. In other words, the form of the parameters gives rise to so-called fuzzy linear regression models. The optimization procedure applied there helps us strike a sound balance between the data-driven and knowledge-driven evidence. It is also important to note that the granularity of the fuzzy numbers of the parameters of the local regression models becomes helpful in the quantification of the reconciled differences between the models.

The proposed approach to the development of granular rule-based architectures could be refined by allowing for a different treatment of individual rule-based systems depending upon their temporal or spatial relationships with the original data set D. Furthermore, we could consider the use of the developed scheme in case of other fuzzy models or neurofuzzy architectures.

The experience-based modeling could be realized when dealing with a variety of models available at each datasite D_j. While the optimization is carried out in the same manner as before and the performance of the model can be evaluated as given in (21), the heterogeneity of the involved models prevents us from forming the fuzzy numbers of the parameters of the overall model.

References

1. R. J. G. B. Campello, W. Caradori do Amaral, Hierarchical fuzzy relational models: linguistic interpretation and universal approximation, IEEE Trans. on Fuzzy Systems, 14, 3, 2006, 446- 453.

2. S. G. Cao, N. W. Rees, G. Feng, Analysis and design for a class of complex control systems, Part I and Part II, Automatica, 33, 6, 1997, 1017-1028 and 33, 6, 1997, 1029-1039.

3. J. Casillas et al. (eds.), Interpretability Issues in Fuzzy Modeling, Springer Verlag, Berlin, 2003.

4. O. Cordón, F. Herrera, I. Zwir, A hierarchical knowledge-based environment for linguistic modeling: models and iterative methodology, Fuzzy Sets and Systems, 138, 2, 2003, 307-341.

5. K. H. Fasol, H. P. Jörgl, Principles of model building and identification, Automatica, 16, 5, 1980, 505-518.

6. A. Gaweda, J. Zurada, Data-driven linguistic modeling using relational fuzzy rules, IEEE Trans. Fuzzy Systems, 11, 1, 2003, 121-134.

7. C. Janikow, Fuzzy decision trees: issues and methods, IEEE Trans. Systems Man, Cybernetics—Part B, 28, 1, 1998, 1–14.

8. C. Mencar, G. Castellano, A. Fanelli, Interface optimality in fuzzy inference systems, Int. Journal of Approximate Reasoning, 41, 2, 2006, 128-145.

9. S. Mitra, Y. Hayashi, Neuro-fuzzy rule generation: survey in soft computing framework, IEEE Transaction on Neural Networks, 11, 3, 2000, 748-768.

10. C. Molina, L. Rodríguez-Ariza, D. Sánchez, M. Amparo Vila, A new fuzzy multidimensional model, IEEE Trans. on Fuzzy Systems, 14, 6, 2006, 897-912.

11. W. Pedrycz, G. Vukovich, On elicitation of membership functions, IEEE Trans. on Systems, Man, and Cybernetics, Part B, 32, 2002, 761-767.

12. W. Pedrycz, Knowledge-Based Clustering: From Data to Information Granules, J. Wiley, Hoboken, NJ, 2005.

13. W. Pedrycz, F. Gomide, Fuzzy Systems Engineering, J. Wiley, Hoboken, NJ, 2007.

14. G. Serra, C. Bottura, An IV-QR algorithm for neuro-fuzzy multivariable online identification, IEEE Trans. on Fuzzy Systems, 15, 2, 2007, 200-210.

15. T. Takagi, M. Sugeno, Fuzzy identification of systems and its application to modelling and control, IEEE Trans. on Systems, Man, and Cybernetics, 15, 1985, 116-132.

16. J. Yen, L. Wang, C. Gillespie, Improving the interpretability of TSK fuzzy models by combining global learning and local learning, IEEE Trans. Fuzzy Systems, 6, 4, 1998, 530-537.

17. L. A. Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, Fuzzy Sets and Systems, 90, 1997, 111-117.

Chapter 9

Experience-Consistent Modeling Using Radial Basis Function Neural Networks

We discuss a new approach to system modeling that utilizes a collaborative framework of data-driven experience and knowledge-driven experience. Data-driven knowledge is obtained by building a model from the currently available data, while knowledge-driven experience is based on parameters of models built in the past. We construct and present a conceptual and algorithmic framework to reconcile these two essential sources of knowledge by considering gradient-based neural network models. The models use radial basis function networks (RBFN). We define appropriate performance indexes which demonstrate the effect of collaboration between the models. Experimental results are obtained for several low-dimensional synthetic datasets and for datasets from the Machine Learning Repository.

9.1 Introduction

Successful model building depends on the quality of experimental data. Knowledge about models is encoded in the architecture and parameters of the models. For example, in neural networks, regression analysis, and rule-based systems such as decision trees, knowledge is encoded in the respective weights, regression coefficients, and simple, crisp rules, respectively. In fuzzy systems [5][8][9][10][11][14][15], knowledge is represented by fuzzy sets and their membership values and fuzzy rules. If a dataset used to build a model contains a relatively small number of patterns and/or contains a high amount of noise, one can expect that the usefulness of the model and its prediction capability will be poor. In such a scenario, a reliance on models developed in the past might be quite useful in enhancing the predictive quality of the currently built model. Specifically, in constructing a new model, one solution might be to utilize not necessarily the past data themselves (which may no longer be available), but rather the parameters of the already existing models.

Business, government, and scientific organizations distribute their massive quantities of data in physically separated data warehouses located far from each other. Due to security, privacy, the sensitive nature [1][6][12] of the data, or for technical reasons, access to the data warehouses may be limited or even prohibited. Furthermore, even if access to the old data is available, transmitting very large volumes of data may not be feasible due to transmission problems or high traffic on computer networks.

There are other reasons for using existing models rather than past data to build a new model. The datasets used to build previous models may be larger, less noisy, or more reliable (if the data collection was performed in a meticulous manner). Moreover, existing models built on recent data could have a large impact on models currently being created.

A new model may not work well for a number of reasons. Some variables might not be included in a dataset used for model building because of their high complexity, cost of measurement, or simply because the modeller does not understand the importance of some features and their influence on the model. These phenomena can be overcome to some extent by using the parameters of an existing model as a guide in new model building.

For these reasons a reliance on previously constructed models when creating a current model makes a great deal of sense. Here we consider regression models, i.e., models which deal with prediction of the continuous value of the output variable. We implement a prediction learning function [2][4][7][13], a radial basis function network (RFBN) employing a single linear neuron, which essentially works as a summation node in an output layer. We use RBFN as activation functions for all neurons in a hidden layer and employ the FCM algorithm to determine parameters of the RBFN.

This chapter is organized as follows. The notation used and the mathematical formulation of the problem are presented in section 9.2. In section 9.3 we show and discuss experimental studies. Section 9.4 offers conclusions and suggests ways to extend the work done in this work. An outline of the algorithm used, mathematical derivations, an optimization scheme for neural network models, and the steps followed in experiments, are included in the Appendices.

9.2 Notation and a formulation of the problem

We continue to use the standard notation of previous chapters. Input patterns are represented as n-dimensional vectors in R^n. C is the number of clusters, **v** represents prototype vectors, and **U** or **R** denotes a partition matrix.

Assume that we wish to build a new model using a given dataset (site) D that contains N patterns and n dimensions. The dataset D consists of the following pairs $(\mathbf{x}_k, y_k)$, k = 1, 2, ..., N. Further assume that we have P auxiliary datasets (sites) denoted as D[1], D[2], ..., D[P] collected in the past. Each of these P datasets (sites) consists of N_1, N_2, ..., N_P patterns, respectively. All patterns on site D and auxiliary sites D[1] through D[P] are defined in the same feature space **X**. Suppose that in the past, we used each of these P datasets to construct P independent models. Though the datasets used to construct the P models may no longer be available, we can utilize knowledge encoded in the parameters of these models to influence building the new model utilizing the currently available data on site D.

On each of the auxiliary sites D[1] through D[P] and on site D we partition the data subsets into C fuzzy clusters using the FCM method. The clustering process performed for each site yields a collection of prototypes $\mathbf{v}_i[ii]$(cluster centers).Next we compute the partition matrix R[ii] using radial basis approach, as mentioned in the Chapter 4 section 4.7.3. The subscript ii = 1, 2, ..., P individually identifies each site; i = 1, 2, ..., c denotes a cluster number; and N_1, N_2, ..., N_P indicates the number of patterns available on the ii-site, ii = 1, 2, ..., P.

To accomplish an exchange mechanism of local constructs (parameters) between the main site D and auxiliary sites D[1] through D[P], we define two performance indexes V and G to assist in calculating the level of communication between the sites (models).

$$V = \sum_{\substack{k=1 \\ x_k, \mathrm{target}_k \in D}}^{N} (\sum_{i=1}^{C} w_i R_{ki}(x_k) - \mathrm{target}_k)^2 + \alpha \sum_{ii=1}^{P} \sum_{\substack{k=1 \\ x_k \in D}}^{N} (\sum_{i=1}^{C} w_i R_{ki}(x_k) - \sum_{i=1}^{C} w_i[ii] R_{ki}(x_k))^2 \cdot \tag{1}$$

More specifically, indices V and G assist in measuring the level of interaction between models created exclusively on the basis of dataset D and auxiliary models built on datasets D[1] through D[P]. Figures 9.1(a) and 9.1(b) show the meaning and effect of indexes V and G which minimize the standard sum of squared errors and the normalized (by N) standard sum of squared errors, respectively, between predicted and actual values. The semantics for both indices are as follows.

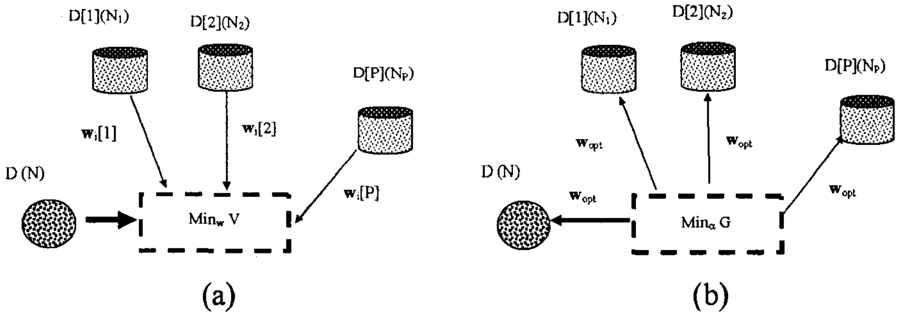


Figure 9.1 Minimization of the performance indexes V and G—a schematic view.

First, we introduce performance index V to achieve a balance (stability) and consistency between the model created only on the basis of site D (the left part of equation 1) and the results produced formerly by the models on sites D[ii], ii = 1, 2, ..., P (the right part of equation 1 and further details regarding optimal estimation of weights w_i and w_i[ii] refer Appendices A, B and C). The value of index V is based on the optimal estimation of weights w_i and w_i[ii] of the neural network models computed for datasets D and D[1] through D[P], respectively. We wish to minimize index V for some value of α, changing from 0 to 2 with step 0.001, to obtain the optimal set of parameters that relies on the constraints of consistency. The set of these parameters is denoted by w_i(opt). The overall balance embedded in index V is obtained for a certain value of α. Higher values of α imply a higher influence of other models on the currently constructed model. The nature of index V is presented in Figure 9.1(a) and formula (1). One sees that the computation of V is performed on the basis of dataset D only as well as the optimal sets of weights w_i and w_i[ii] of the models found earlier are computed using datasets D and D[1] through D[P], respectively. We note that the second component of (1) could be viewed as a regularization term.

While the semantics of V (which evaluates the quality of the optimal model based on dataset D) should be quite clear by now, the second performance index G requires some attention. As shown in Figure 9.1(b), the optimal sets of weights $\mathbf{w}_{\mathrm{opt}}$, which minimized V for some value of α, are now transferred to sites D[1] through D[P] and their quality is evaluated there by the normalized sum of squared errors. One can say that index G expressed by (2) measures the global performance of the optimal model as its computation is based on optimal vector $\mathbf{w}_{\mathrm{opt}}$, common to all sites, as well as dataset D and datasets D[1] through D[P].

$$G=\frac{1}{N}\sum_{x_k,\mathrm{target}_k\in D}\left(\sum_{i=1}^{C}w_i(\mathrm{opt})R_{ik}(x_k)-\mathrm{target}_k\right)^2+\frac{1}{N_1}\sum_{x_k,\mathrm{target}_k\in D[1]}\left(\left(\sum_{i=1}^{C}w_i(\mathrm{opt})R_{ik}(x_k[1])\right)-\mathrm{target}_k\right)^2$$
$$+\ldots\ldots+\frac{1}{N_P}\sum_{x_k,\mathrm{target}_k\in D[P]}\left(\left(\sum_{i=1}^{C}w_i(\mathrm{opt})R_{ik}(x_k[P])\right)-\mathrm{target}_k\right)^2. \tag{2}$$

Apparently the expression of G is a function of α and the optimized level of consistency is that for which G attains its minimal value, namely $\alpha_{\mathrm{opt}} = \arg\operatorname{Min} G(\alpha)$. A schematic view of computing realized with the aid of (2) is presented in Figure 9.1(b).

More detailed derivations for computing performance indexes V and G as well as the computational steps and optimization schemes for neural network models are presented in Appendices B through D.

9.3 Experimental studies

In order to demonstrate the effectiveness of RBF neural network experience-consistent modeling, we experiment with synthetic datasets and the Machine Learning Repository at the University of California, Irvine (UCI). The basic properties of the datasets used are summarized in Table 9.1.

Table 9.1 Experimental data used in constructing RBFN models.

No.	Name of dataset	Number of data (N)	Number of input and output attributes (n)
1	Synthetic-1	1000	2
2	Synthetic-2	1000	2
3	Auto-mpg	398	8
4	Boston Housing	506	14
5	Abalone	4177	9

Source(No. 3,4 and 5): http://www.ics.uci.edu~mlearn/MLRepository.html

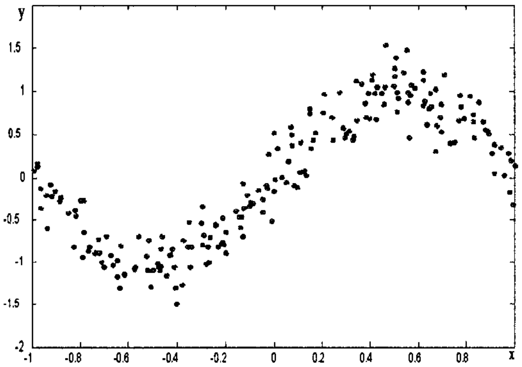
For the synthetic 2-dimensional datasets, the number of clusters c and the number of auxiliary sites P were set to 3 and 4, respectively. The 1000 patterns generated for two synthetic datasets were randomly divided into 5 subsets, D and D[1] through D[4], each containing 200 patterns. Datasets obtained from the Machine Learning Repository, have more dimensions; P varies from 1 to 5, and c changes from 2 to 5. The total number of patterns available for datasets 3, 4, and 5 (Table 9.1) were randomly and evenly distributed among the D site and P auxiliary sites so that each of the sites contains an equal number of patterns. For example, an Auto-mpg dataset contains 398 patterns. Dividing the dataset into 4 sites (D and D[1] through D[3]) yields 99 patterns on each site.

The different learning rates were investigated in the Machine Learning datasets. Generally lower learning rates tend to give better results but with higher learning rates the network oscillates. In our collaborative modeling approach, when the learning rate is greater than 0.005 in synthetic, Auto-mpg, and Boston Housing datasets, and in case of the Abalone dataset greater than 0.0005, the collaborative model tends to oscillate. To better interpret the simulation results, we define a coefficient called Ratio. As Ratio=$(G(\alpha=0)-G(\alpha_{opt}))/G(\alpha=0)$, it reflects the reduction of $G(\alpha_{opt})$ obtained for optimal α in comparison to the value of G calculated for $\alpha=0$.

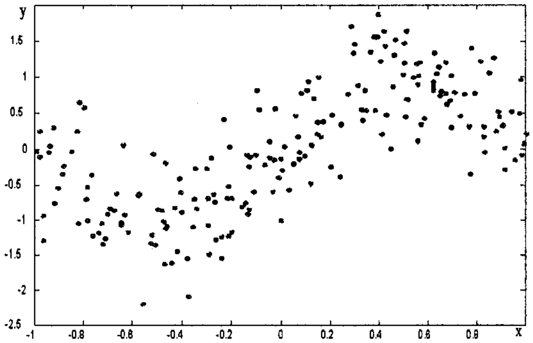
The prediction results of the neural networks with different combinations of hidden nodes and splits are summarized in tables 9.2–9.6.

The convergence characteristic at different α values is shown for selected splits in the figures 9.4-9.6.

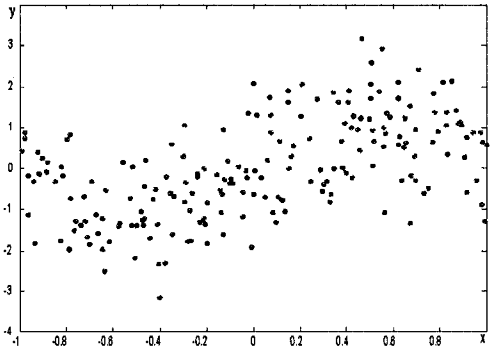
Experiment 1: Synthetic Datasets-1



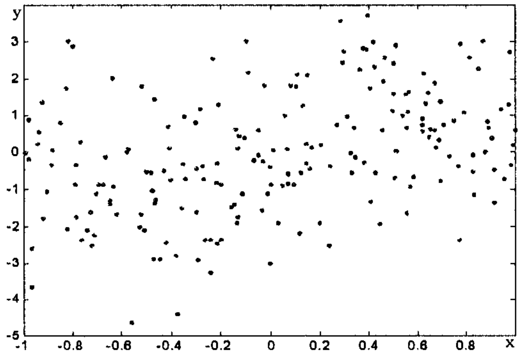
σ=0.25; site D



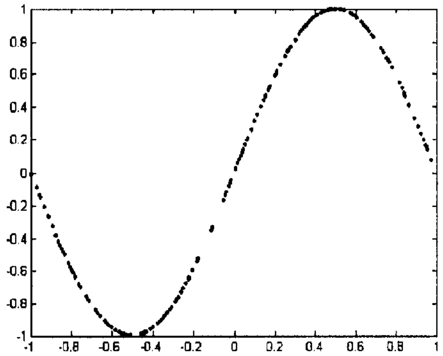
σ=0.5; site D



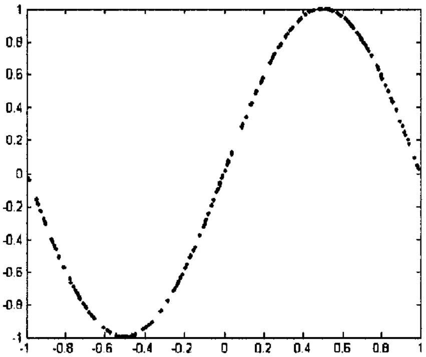
σ=1.0; site D



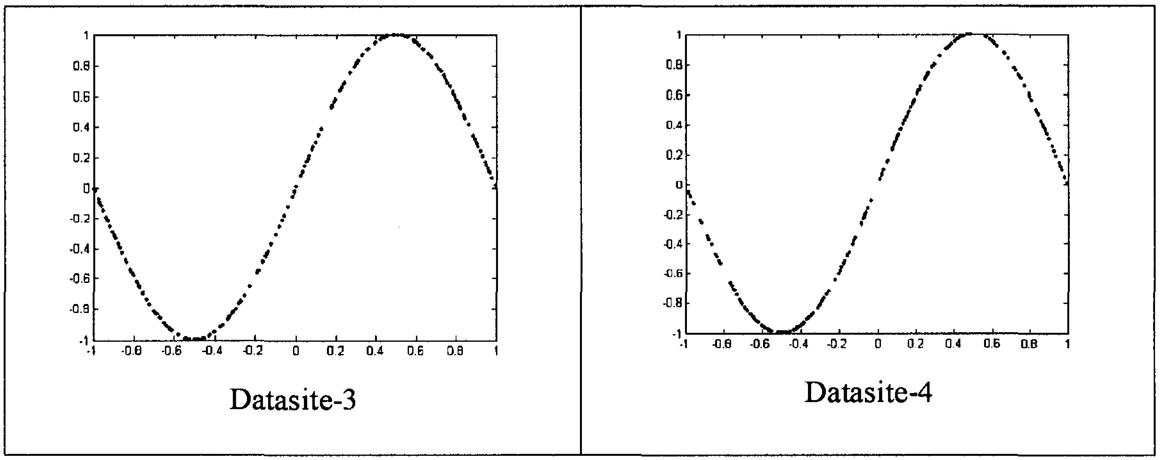
σ=1.5; site D



Datasite-1



Datasite-2



Datasite-3

Datasite-4

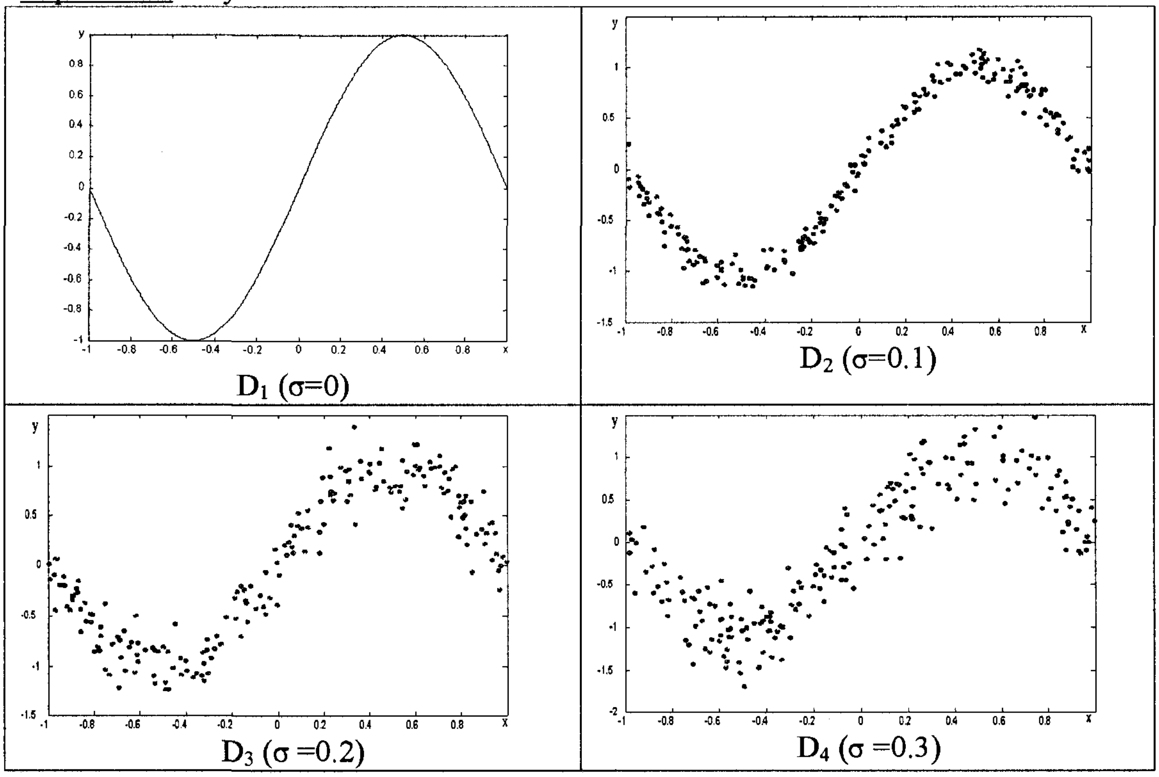
Figure 9.2 Synthetic dataset D and D[1] through D[4] used in Experiment 1; No noise(σ=0); all datasets consist of 200 patterns.

Table 9.2 Values of the global performance index for RBFNs: P=4, C=3.

	σ =0	σ=0.25	σ=0.5	σ=1.0	σ=1.5
G(α=0)	4.692	4.804	4.922	5.784	6.83
G(α_{opt})	2.381	2.447	2.617	3.262	4.51
Ratio	0.493	0.491	0.468	0.436	0.34
α_{opt}	1.021	0.986	0.975	0.904	0.9

We ran Experiment-1 for a synthetic 2-dimensional dataset that represents a sinusoid and contains 1000 patterns. The number of clusters and the number of auxiliary sites D[ii] were set to 3 and 4, respectively. As shown in Figure 9.2, sites D[1] through D[4] represent "ideal" models with no noise, while site D does not contain noise (σ=0) or is cluttered with small or large amounts of noise determined by varying values of σ. Note that for σ=1.0 and σ=1.5 the amount of noise added is quite large and the sinusoid pattern is not clearly recognizable. As expected, we observe that as the amount of noise on the D site gradually increases, the effect of the 4 "perfect" models on improving the D model slightly decreases (Tables 9.2). This is demonstrated by decreasing values of Ratio from 0.493 to 0.340 and slightly decreasing values of α from 1.021 to 0.900. In particular, this experiment shows that when the largest amount of noise (σ=1.5) is introduced on site D, the Ratio value drops quite substantially to 0.340. Again, for a large amount of noise, σ=1.5, the influence of other sites on "fixing" the D site diminishes and Ratio drops to 0.348.

Experiment 2: Synthetic Datasets-2



D$_1$ (σ=0)

D$_2$ (σ=0.1)

D$_3$ (σ =0.2)

D$_4$ (σ =0.3)

Figure 9.3 Synthetic datasets on sites D$_1$ through D$_4$ used in Experiment 2, where in each datasite, N=200.

σ determines a small level of noise added to the 4 sites. Five D sites for 5 different values of σ are the same as in Experiment 1.

Table 9.3 Values of the global performance index for RBFN: P=4, C=3.

	σ=0	σ=0.25	σ=0.5	σ=1.0	σ=1.5
G(α=0)	4.908	5.022	5.139	6.003	7.046
G(α_{opt})	2.583	2.65	2.819	3.465	4.713
Ratio	0.474	0.472	0.451	0.423	0.331
α_{opt}	1.035	0.976	0.986	0.914	0.91

Similar to Experiment 1, Experiment 2 was run on a synthetic 2-dimensional dataset that contains 1000 patterns and represents a sinusoid. As before, the number of clusters and the number of auxiliary sites D[ii] were set to 3 and 4, respectively. As seen in Figure 9.3, the level of noise on sites D[1] through D[4] is small and slightly increasing, while site D remains the same as in Experiment 1, i.e., it has either no noise or is contaminated with a considerable amount of noise determined by σ. Again, we observe that as the amount of noise on the D site increases, the effect of the 4 "quite good" models on improving the D model decreases (Table 9.3). For an RBFN this is demonstrated by decreasing values of Ratio from 0.474 to 0.331 and slightly decreasing values of α from 1.035 to 0.910.

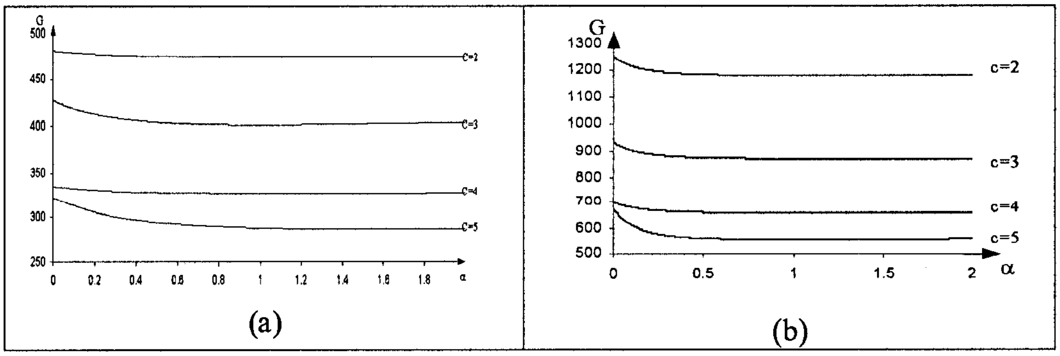
Machine Learning Datasets

We performed experiments on three selected datasets from the Machine Learning Repository. We demonstrate results for different numbers of clusters c varying from 2 to 5 and a number of sites changing from 1 to 5.

Experiment 3: Auto-mpg

Table 9.4 The values of the global performance index for RFBN: P=1–5; C=2–5

P\C	**2**	**3**	**4**	**5**
1: G(α=0)	480.887	426.886	334.975	322.437
G(α_{opt})	474.061	401.440	326.557	285.941
Ratio	0.014	0.060	0.025	**0.113**
α_{opt}	1.128	0.968	1.146	1.501
2: G(α=0)	1311.730	914.346	609.210	447.574
G(α_{opt})	855.546	641.818	546.224	424.634
Ratio	**0.348**	**0.298**	0.103	0.051
α_{opt}	1.491	1.861	2	0.903
3: G(α=0)	1252.120	932.811	700.637	673.026
G(α_{opt})	1181.990	869.457	656.682	554.116
Ratio	0.056	0.068	0.063	**0.177**
α_{opt}	1.219	1.249	1.347	0.969
4: G(α=0)	1634.130	1114.640	944.922	886.008
G(α_{opt})	1460.570	1011.040	896.693	775.992
Ratio	0.106	0.093	0.051	0.124
α_{opt}	0.983	1.661	0.408	0.739
5: G(α=0)	1971.180	1505.710	1321.930	1049.750
G(α_{opt})	1667.740	1263.730	1097.800	1002.050
Ratio	0.154	0.161	0.170	0.045
α_{opt}	0.96	2	0.764	0.408



(a)

(b)

Figure 9.4 Plots of performance index G vs. α for P=1 (a) and P=3 (b).

Analysis of the results for Auto-mpg data in Table 9.4 provides similar conclusions to those made for the Synthetic Datasets-2 (Experiment 2) in Table 9.3. Table 9.4 shows in bold several small or small-medium values of Ratio within [0.113, 0.348]. It appears that the same configurations of the data partitions and clusters have a low-medium or low effect on the model created on site D. For a higher number of partitions and 1 and 4 clusters, models existing on the auxiliary sites have a very low influence on the model created on site D. Having more than 3 partitions apparently introduces more noise because the number of patterns in the entire dataset is relatively small (equal to 398). The values of α, which minimize the G index, vary from the lowest of 0.408 to the highest of 2 and do not exhibit any easily discernable behaviour.

Experiment 4: Boston Housing

Table 9.5 Values of the global performance index for RFBN: P=1–5; C=2–5

P\C	**2**	**3**	**4**	**5**
1: G(α=0)	791.919	742.164	735.192	745.661
G(α_{opt})	774.573	733.830	715.810	727.730
Ratio	0.022	0.011	0.026	0.024
α_{opt}	1.503	0.794	1.978	0.339
2: G(α=0)	1167.540	1083.810	1168.830	1166.130
G(α_{opt})	1162.440	1081.880	1090.250	1096.400
Ratio	0.004	0.002	0.067	0.060
α_{opt}	1.04	0.891	0.751	1.147
3: G(α=0)	1530.650	1548.430	1534.910	1526.660
G(α_{opt})	1528.220	1472.820	1449.880	1436.150
Ratio	0.002	0.049	0.055	0.059
α_{opt}	1.157	0.708	0.446	0.443
4: G(α=0)	1943.610	1923.770	1968.390	1962.400
G(α_{opt})	1933.760	1847.020	1825.100	1786.930
Ratio	0.005	0.040	0.073	0.089
α_{opt}	0.58	1.829	0.416	2
5: G(α=0)	2358.530	2339.100	2593.630	2306.190
G(α_{opt})	2300.690	2261.300	2190.290	2094.800
Ratio	0.025	0.033	0.156	0.092
α_{opt}	1.989	1.074	2	2

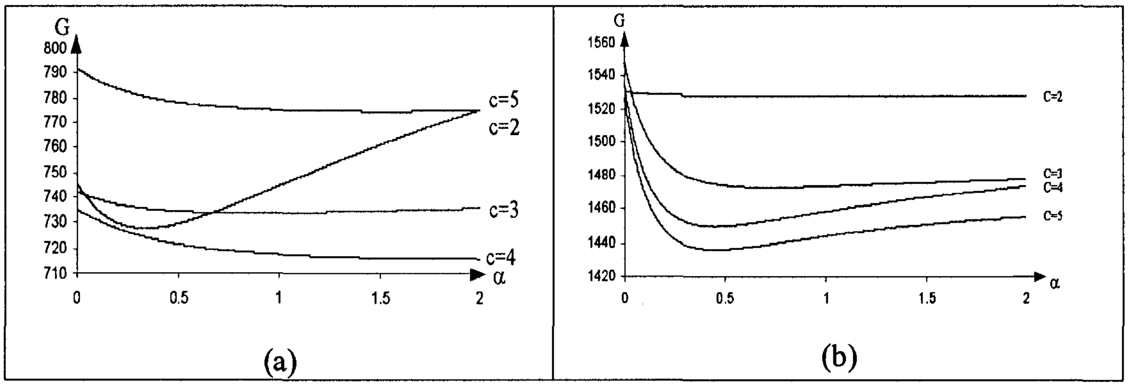


Figure 9.5 Plots of performance index G vs. α for P=1 (a) and P=3 (b).

The RBFN model was employed on the Boston Housing dataset with 506 patterns and 14 attributes. The findings mostly parallel results obtained for the Auto-mpg dataset. Some configurations of partitions and clusters, in terms of their counts, cause an increased and substantial effect of other sites on site D. One can conclude that when the number of dimensions in the dataset increases and the number of patterns is relatively small the RBFN method clearly has an effect on site D. Again, G does not seem to be very dependent on α; however, the collaboration is beneficial.

Experiment 5: Abalone

Table 9.6 Values of the global performance index for RFBN: P=1, 2, 5; C=2–5.

P\C	2	3	4	5
1: G(α=0)	87.233	65.319	47.052	46.060
G(α_{opt})	87.195	58.551	45.801	43.529
Ratio	0.000	0.104	0.027	0.055
α_{opt}	1.081	1.276	0.879	0.978
2: G(α=0)	137.335	91.480	87.192	83.471
G(α_{opt})	135.109	87.307	74.565	70.451
Ratio	0.016	0.046	**0.145**	**0.156**
α_{opt}	1.128	0.733	1.23	0.753
5: G(α=0)	280.586	189.951	192.595	166.728
G(α_{opt})	272.247	176.575	157.982	143.051
Ratio	0.030	0.070	**0.180**	**0.142**
α_{opt}	0.946	1.611	1.589	0.487

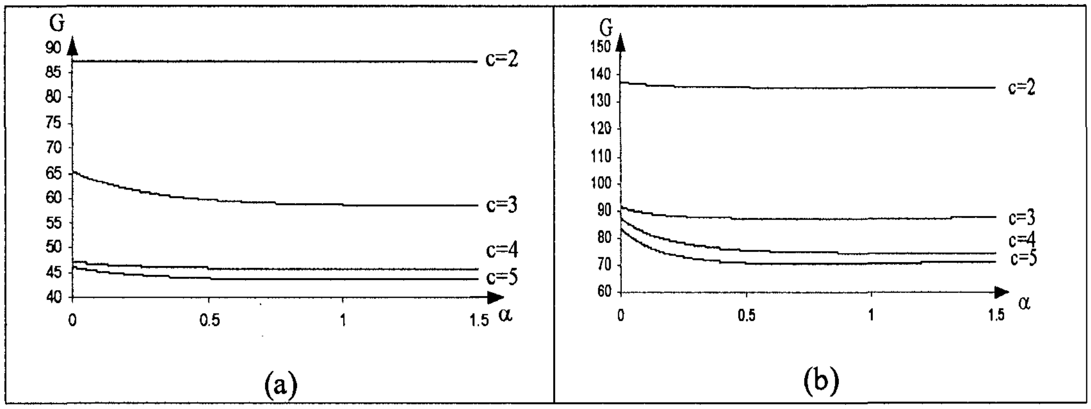


Figure 9.6 Plots of performance index G vs. α for P=1 (a) and P=2 (b).

The Abalone dataset has 4177 patterns and 9 dimensions. To some extent, the findings parallel results obtained from Auto-mpg and Boston Housing datasets. Again, some configurations of partitions and clusters, in terms of their counts, show a low effect of other sites on site D. The reader is referred to values in bold in Table 9.6. The influence of other sites on site D is relatively low for the RBFN models. This is evident when the number of dimensions in the dataset increases and the number of patterns is relatively large. Again, it is difficult to notice any easily interpretable pattern in the behaviour of α.

9.4 Conclusions

We present a methodology of system identification and modeling accomplished through a collaborative framework and knowledge-driven experience. Data-driven knowledge is realized by building models from currently available data, while knowledge-driven experience is based on the parameters of models built in the past. We construct and present conceptual and algorithmic frameworks for the reconciliation of these two essential sources of knowledge by considering gradient-based neural network models. In particular, we employ models based on RBFN. The experimental results show that knowledge-driven experience can enhance data-driven experience, i.e., models built in the past can "improve" the quality of models currently under construction.

In the future different model architectures could be examined, i.e., linear and nonlinear regression models could be combined. Neural network systems could be based on error back-propagation and fuzzy rule-based systems residing on the datasites could realize the exchange of parameters and cooperation between the models via high level information granules and prototypes. A much more challenging task would involve dealing with models built on different feature spaces, i.e., different model architectures could be reconciled as could slightly different feature spaces on which the models are built.

References

1. R. Agarwal, R. Srikant, Privacy-prserving data mining. *In Proc. Of the ACM SIGMOD Conference on Management of Data*, ACM Press, New York, May 2000, 439-450.

2. L. Breiman, Bagging predictors, *Machine Learning* , 24(3), 1996,123-140.

3. S. G. Cao, N. W. Rees, G. Feng, Analysis and design for a class of complex control systems, Part I and Part II, *Automatica*, 33, 6, 1997, 1017-1028 and 33, 6, 1997, 1029-1039.

4. K. H. Fasol, H. P. Jörgl, Principles of model building and identification, *Automatica*, 16, 5, 1980, 505-518.

5. A. Jain, R. Duin, J. Mao, Statistical pattem recognition: a review, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 2000, 4-37.

6. M. R. Genesereth, S. P. Ketchpel, Software agents, *Communications of the ACM*, 37(7), 1994, 48-53.

7. S. S. Haykin, Neural Networks: A comprehensive Foundation, *Macmillan*, New York, 1994.

8. M. Kumar, N. Stoll, R. Stoll, An energy-gain bounding approach to robust fuzzy identification,*Automatica*, 42, 5, 2006, 711-721.

9. W. Pedrycz, G. Vukovich, On elicitation of membership functions, *IEEE Trans. on Systems, Man, and Cybernetics, Part B,* 32, 2002, 761-767.

10. W. Pedrycz, *Knowledge-Based Clustering: From Data to Information Granules,* J. Wiley, Hoboken, NJ, 2005.

11. W. Pedrycz, F. Gomide, *Fuzzy Systems Engineering*, J. Wiley, Hoboken, NJ, 2007.

12. D. B. Skillicorn, S. M. McConnell, Distributed prediction from vertically partitioned data, *Journal of Parallel and Distributed computing*, 2007.

13. G. Tsoumakas, L. Angelis, I. Vlahavas, Clustering classifiers for knowledge discovery from physically distributed databases, *Data & knowledge Engineering*, 49(3), 2004, 223-242.

14. L. A. Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*, 90, 1997, 111-117.

15. L. A. Zadeh, Toward a generalized theory of uncertainty (GTU)- an outline, *Information Sciences*, 172, 2005, 1-40.

Chapter 10

Conclusions and Future Directions

This chapter highlights the conclusions made in this study and discusses some directions for future research.

10.1 Conclusions

This study proposes a new suite of collaborative frameworks using the idea of knowledge-based networks. We investigate two collaborative frameworks: collaborative clustering and experience-consistent. The collaborative clustering framework deals with clustering problems. The experience-consistent framework handles regression and classification problems.

An important task of our research is to find consensus among datasites through a collaborative approach so that a global structure (model) of the datasites can be constructed.

In fuzzy collaborative clustering each site comes with its own partition matrix; direct comparison of two partition matrices is not feasible as we may not have a correspondence between rows. We have used the concept of proximity and a proximity matrix induced by a given partition matrix. It is essential to evaluate how consistent the resulting structures are. A viable method we have applied is to compare partition matrices to quantify the distance between them.

In experience-consistent modeling it is advantageous to not only consider currently available data, but also to actively exploit previously obtained knowledge. A previously constructed model can serve as a regularization mechanism for conditions that will be encountered quite often in various modeling platforms. A level of achieved consistency is expressed in terms of fuzzy sets of the regression parameters of the local models occurring in the conclusions of the rules. In other words, the form of the parameters gives rise to so-called fuzzy linear regression models. The optimization procedure applied helps to strike a sound balance between data-driven and knowledge-driven evidence. It is noted that the granularity of the fuzzy numbers of the parameters of the local regression models is helpful in quantifying the reconciled differences between models.

Experimental results presented in Chapters 5 to 9 reveal that both collaborative frameworks are efficient and can be used to the find global characteristics of the collaborating sites.

- In the case of fuzzy collaborative clustering (both horizontal and vertical modes) the results are compelling; in most cases the values of the proximity index were reduced around 40% in comparison to values reported without collaboration. The plots of prototypes before and after collaboration also show the effectiveness of the developed algorithm.

- In the case of the experience-consistent framework, for all models under study global objective functions were minimized after collaboration.

The study makes the following contributions:

Fuzzy collaborative clustering framework

In fuzzy collaborative clustering we have formulated the distributed computing problem entirely in the framework of Pedrycz's collaborative clustering model which is based on a fuzzy clustering (FCM), and we have extended the study in several respects:

- Previous work mainly focused on basic aspects of collaborative clustering; we have developed a new suite of collaborative clustering frameworks.

- We reformulate new horizontal and vertical clustering frameworks to support multiple phases of reconciliation of the collaborative process so as to obtain optimal global structures of the collaborating sites.

- We introduce a quantification method for evaluating the overall consistency achieved among datasites using a proximity-distance method. Experimental results reveal that the proximity-distance index is a suitable vehicle to quantify the collaboration and will help in selecting an optimal β.

- We present an evaluation of our approach in advanced constructs such as type-2 fuzzy sets.

In such a framework each site actively participates in collaboration for multiple phases, thus this method is suitable for dynamic applications. The multiple phase method also overcomes instabilities in the clusterings and improves the ability to find global structures present in collaborating datasites.

Experience-consistent framework

The experience-consistent framework deals with predictions in a distributed environment. The main contributions of this study are:

- We elaborate on key design issues of distributed methods relating to regression, fuzzy linear regression (rule-based model), linear classification (two-class model), and radial basis function networks (RBFN) for system identification.
- We introduce a variety of communication mechanisms for effective interactions between datasites.
- We evaluate our methods in advanced constructs such as type-2 fuzzy sets.

10.2 Future directions

Suggestions for future research in both collaborative frameworks are summarized below.

Fuzzy collaborative clustering framework

We have presented vertical and horizontal fuzzy collaborative clustering for the case in which the collaborative framework attempts to find global structure in the presence of privacy preserving constraints. We demonstrate both cases at the same and different levels of information granularity at collaborating sites.

- It would be worthwhile to study both vertical and horizontal frameworks for situations in which the intensity of a collaboration coefficient is computed dynamically, so that the impact of noisy datasites can be isolated during a collaboration process.
- Future research may investigate the use of a multi-objective optimization approach to capture the complex global structure of collaborating sites (collaborating datasites may have hybrid structure: simultaneous heterogeneous and homogeneous structures).
- A case in which each collaborating site has a different base learning algorithm could be investigated. This would generalize the framework, enabling us to cope with more real-world problems.

Experience-consistent framework

We present foundation frameworks for experience-consistent modeling. Based on these foundations, the following scenarios could be investigated:

- A two-class category classification of the model of experience-consistent framework could be generalized to handle multiclass problems.

- In fuzzy rule-based experience-consistent models we have implemented the model with the assumption that the same number of information granularity is present at all sites. In future, different levels of granularity present at interacting sites could be investigated.

- The proposed approach to the development of granular rule-based architectures could be refined by allowing for a different treatment of individual rule-based systems depending upon their temporal or spatial relationships with the original dataset D. Furthermore, the use of the developed scheme in case of other fuzzy models or neurofuzzy architectures could be considered.

- In RBFN experience-consistent model, a much more challenging task would involve dealing with models built on different feature spaces, i.e., the mechanism of reconciliation could be introduced, not only between different architectures of the models but also between slightly different feature spaces on which each model is built.

Theoretical and algorithmic approaches of collaborative computing investigated in this study can be used as a foundation for further research in the area of fuzzy distributed modeling.

Appendix A

Experience-consistent Regression model

Here we include detailed derivations of the optimal vector of parameters of the regression model in presence of the experience-consistent component which minimizes V. Let us rewrite V in an explicit manner

$$V = \sum_{\substack{x_k \in D \\ y_k \in D}} (a^T x_k + a_0 - y_k)^2 + \alpha \sum_{j=1}^{P} \sum_{\substack{x_k \in D \\ y_k \in D}} (a^T x_k + a_0 - a^T[j] x_k - a_0[j])^2$$

$$= \sum_{\substack{k=1 \\ x_k \in D \\ y_k \in D}}^{N} (\sum_{i=1}^{n} a_i x_{ki} + a_0 - y_k)^2 + \alpha \sum_{j=1}^{P} \sum_{\substack{k=1 \\ x_k \in D \\ y_k \in D}}^{N} (\sum_{i=1}^{n} a_i x_{ki} + a_0 - \sum_{i=1}^{n} a_i[j] x_{ki} - a_0[j])^2$$

For simplification purposes we include constant a$_0$ by accepting an augmented version of **x** and **a**. More specifically, we augment **a** by a$_0$ while the corresponding entry of **x** is set up to 1. In other words, we end with **a** and **x** to be two (n+1)-dimensional vectors. Subsequently taking the derivative of V we obtain

$$\frac{\partial V}{\partial a_s} = 2 \sum_{\substack{k=1 \\ x_k \in D \\ y_k \in D}}^{N} (\sum_{i=0}^{n} a_i x_{ki} - y_k) x_{ks} + 2\alpha \sum_{j=1}^{P} \sum_{\substack{k=1 \\ x_k \in D \\ y_k \in D}}^{N} (\sum_{i=0}^{n} a_i x_{ki} - \sum_{i=0}^{n} a_i[j] x_{ki}) x_{ks} = 0$$

$$= \sum_{\substack{k=1 \\ x_k \in D \\ y_k \in D}}^{N} \sum_{i=0}^{n} a_i x_{ki} x_{ks} - \sum_{\substack{k=1 \\ x_k \in D \\ y_k \in D}}^{N} y_k x_{ks} + \alpha \sum_{j=1}^{P} \sum_{k=1}^{N} \sum_{i=0}^{n} a_i x_{ki} x_{ks} - \alpha \sum_{j=1}^{P} \sum_{\substack{k=1 \\ x_k \in D \\ y_k \in D}}^{N} \sum_{i=0}^{n} a_i[j] x_{ki} x_{ks} = 0$$

Let us introduce the notation $Y_k[j] = \sum_{i=0}^{n} a_i[j].x_{ki}$. In the sequel we obtain

$$\sum_{\substack{k=1 \\ x_k \in D \\ y_k \in D}}^{N} \sum_{i=0}^{n} a_i x_{ki} x_{ks} - \sum_{\substack{k=1 \\ x_k \in D \\ y_k \in D}}^{N} y_k x_{ks} + \alpha \sum_{j=1}^{P} \sum_{\substack{k=1 \\ x_k \in D \\ y_k \in D}}^{N} \sum_{i=0}^{n} a_i x_{ki} x_{ks} - \alpha \sum_{j=1}^{P} \sum_{\substack{k=1 \\ x_k \in D \\ y_k \in D}}^{N} Y_k[j].x_{ks} = 0$$

Taking common factor out, we obtain

$$\sum_{\substack{k=1\\x_k\in D\\y_k\in D}}^{N}\left\{\left(\sum_{i=0}^{n}a_i x_{ki}+\alpha\sum_{j=1}^{P}1.\sum_{i=0}^{n}a_i x_{ki}\right)-\left(y_k+\sum_{j=1}^{P}Y_k[j]\right)\right\}.x_{ks}=0$$

Further simplification yields $y'_k=\left(y_k+\alpha\sum_{j=1}^{P}Y_k[j]\right)$ and we obtain

$$\sum_{\substack{k=1\\x_k\in D\\y_k\in D}}^{N}\left\{\sum_{i=0}^{n}a_i x_{ki}\left(1+\alpha(P)\right)-y'_k\right\}.x_{ks}=0$$

$$\sum_{i=0}^{n}a_i\sum_{\substack{k=1\\x_k\in D}}^{N}x_{sk}^{T}x_{ki}\left(1+\alpha(P)\right)-\sum_{\substack{k=1\\x_k\in D\\y_k\in D}}^{N}x_{sk}^{T}y'_k=0$$

Next we get

$$\left(1+\alpha(P)\right)\sum_{i=0}^{n}a_i\sum_{\substack{k=1\\x_k\in D\\y_k\in D}}^{N}x_{sk}^{T}x_{ki}-\sum_{\substack{k=1\\x_k\in D\\y_k\in D}}^{N}x_{sk}^{T}y'_k=0$$

$$\left(1+\alpha(P)\right)\sum_{i=0}^{n}a_i\sum_{\substack{k=1\\x_k\in D\\y_k\in D}}^{N}x_{sk}^{T}x_{ki}=\sum_{\substack{k=1\\x_k\in D\\y_k\in D}}^{N}x_{sk}^{T}y'_k$$

Rewriting the above expression in a matrix form, we obtain

$$a_{opt}^{T}=\frac{\left(X^{T}X\right)^{-1}.X^{T}Y}{(1+\alpha.P)}$$

Appendix B

RBF Base Model: Optimization Steps

Step 1: First, we apply FCM (for several iterations) on individual dataset to generate prototypes (i.e., number of neurons in hidden layer) and then compute partition matrix R$_i$ using Radial-basis approach as below.

$$R_i(x) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{v}_i\|^2}{2\sigma_i^{\,2}}\right).$$

Step 2: The consequent part is:

$$y_i = R_i w_i .$$

Step 3: The output of the overall model is:

$$\hat{y}_k = \sum_{i=1}^{C} R_i w_i ,$$

$$Q = \sum_{k=1}^{N} (y_k - \sum_{i=1}^{C} R_{ki} w_i)^2 ,$$

$$MSE = \frac{1}{N}\sum_{k=1}^{N} (y_k - \sum_{i=1}^{C} R_{ki} w_i)^2 .$$

We intend to minimize the squared error between ŷ and y by following the objective function using a gradient method to adjust w$_i$.

$$Q = (y - \hat{y})^2 ,$$

$$\hat{y} = \sum_{i=1}^{C} \hat{y}_i ,$$

$$y_i = R_i w_i ,$$

$$w_i(\text{new}) = w_i(\text{old}) + \Delta w_i ,$$

$$\Delta w_i = -\eta \frac{\partial Q}{\partial w_i}, \text{ where } \eta \text{ is a learning rate,}$$

$$= -\eta \frac{\partial Q}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_i},$$

$$= -\eta \frac{\partial Q}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial w_i},$$

1) $\dfrac{\partial Q}{\partial \hat{y}} = \dfrac{\partial}{\partial \hat{y}}\left[(y-\hat{y})^2\right] = -2(y-\hat{y})$,

2) $\dfrac{\partial \hat{y}}{\partial \hat{y}_i} = \dfrac{\partial}{\partial \hat{y}_i}\left[\sum_{i=1}^{C} \hat{y}_i\right] = 1$,

3) $\dfrac{\partial \hat{y}_i}{\partial w_i} = \dfrac{\partial}{\partial w_i}\left[R_i w_i\right] = R_i$,

$\Delta w_i = 2\eta(y-\hat{y})R_i$.

w_i is updated by the following equation.

$w_i(\text{new}) = w_i(\text{old}) + 2\eta(y-\hat{y})R_i$.

Appendix C

Optimization Scheme: Radial Basis Function Networks(RBFN) Model

$$V = \sum_{\substack{k=1 \\ x_k, \text{t arg et}_k \in D}}^{N} \left(\sum_{i=1}^{C} w_i R_{ki}(x_k) - \text{t arg et}_k\right)^2 + \alpha \sum_{ii=1}^{P} \sum_{\substack{k=1 \\ x_k \in D}}^{N} \left(\sum_{i=1}^{C} w_i R_{ki}(x_k) - \sum_{i=1}^{C} w_i[ii] R_{ki}(x_k)\right)^2,$$

where, $R_{ki}(x_k) = U_{ki}(x_k)$ that is determined as in Appendix-A , $x_k \in D$, and w_i is a D site connection vector between the hidden layer and the output layer. $w_i[ii]$ are connection parameters for the D[ii] site.

Let $y_k = \text{target}_k$,

$$\frac{\partial V}{\partial w_s} = \frac{\partial}{\partial w_s}\left[\sum_{\substack{k=1 \\ x_k, \text{t arg et}_k \in D}}^{N} \left(\sum_{i=1}^{C} w_i R_{ki}(x_k) - y_k\right)^2 + \alpha \sum_{ii=1}^{P} \sum_{\substack{k=1 \\ x_k \in D}}^{N} \left(\sum_{i=1}^{C} w_i R_{ki}(x_k) - \sum_{i=1}^{C} w_i[ii] R_{ki}(x_k)\right)^2\right] = 0$$

$$\frac{\partial V}{\partial w_s} = 2\sum_{\substack{k=1 \\ x_k \in D}}^{N} \left(\sum_{i=1}^{C} w_i R_{ki}(x_k) - y_k\right) R_{ks}(x_k) + 2\alpha \sum_{ii=1}^{P} \sum_{\substack{k=1 \\ x_k \in D}}^{N} \left(\sum_{i=1}^{C} w_i R_{ki}(x_k) - \sum_{i=1}^{C} w_i[ii] R_{ki}(x_k)\right).R_{ks}(x_k) = 0$$

$$\nabla V(w_s) = \sum_{\substack{k=1 \\ x_k \in D}}^{N} \sum_{i=1}^{C} w_i R_{ki}(x_k) R_{ks}(x_k) - \sum_{\substack{k=1 \\ x_k \in D}}^{N} y_k R_{ks}(x_k) + \alpha \sum_{ii=1}^{P} \sum_{\substack{k=1 \\ x_k \in D}}^{N} \sum_{i=1}^{C} w_i R_{ki}(x_k) R_{ks}(x_k)$$

$$-\alpha \sum_{ii=1}^{P} \sum_{\substack{k=1 \\ x_k \in D}}^{N} \sum_{i=1}^{C} w_i[ii] R_{ki}(x_k) R_{ks}(x_k) = 0$$

Now update the w_i (new)

$$w_i(\text{new}) = w_i(\text{old}) - \eta \nabla V(w_i).$$

Appendix D

Computational Steps of experience-consistent RBFN Model:

Step 1: Run FCM on all datasites and R_i is computed as mentioned in Appendix-B.

Step 2: Compute $w_{i(opt)}$ connection parameters minimizing V (optimization scheme as shown above) for $\alpha = 0.0$ to 2.0 with step 0.001.

$$V = \sum_{\substack{k=1 \\ x_k, target_k \in D}}^{N} \left(\sum_{i=1}^{C} w_i R_{ki}(x_k) - target_k\right)^2 + \alpha \sum_{ii=1}^{P} \sum_{\substack{k=1 \\ x_k \in D}}^{N} \left(\sum_{i=1}^{C} w_i R_{ki}(x_k) - \sum_{i=1}^{C} w_i[ii] R_{ki}(x_k)\right)^2,$$

where: $R_{ki}(x_k) = U_{ki}(x_k)$, $x_k \in D$, and w_i is a D site connection vector between the hidden layer and the output layer. $w_i[ii]$ are connection parameters for the D[ii] site.

$w_i(new) = w_i(old) - \eta \nabla V(w_i)$, compute as in Appendix-C.

w_i is updated iteratively so there is no further change in $\nabla V(w_i)$.

Step 3: G index computed on $\mathbf{w}_i$(opt):

$$G = \frac{1}{N} \sum_{x_k, target_k \in D} \left(\sum_{i=1}^{C} w_i(opt) R_{ik}(x_k) - target_k\right)^2 + \frac{1}{N_1} \sum_{x_k, target_k \in D[1]} \left(\left(\sum_{i=1}^{C} w_i(opt) R_{ik}(x_k[1])\right) - target_k\right)^2$$

$$+ \ldots\ldots + \frac{1}{N_P} \sum_{x_k, target_k \in D[P]} \left(\left(\sum_{i=1}^{C} w_i(opt) R_{ik}(x_k[P])\right) - target_k\right)^2$$

Step 4: Iterate α from 0 to 2 with step (0.001) for steps 2 and 3.