Fuzzy Computing for Data Mining

KAORU HIROTA, MEMBER, IEEE, AND WITOLD PEDRYCZ, FELLOW, IEEE

Invited Paper

The study is devoted to linguistic data mining, an endeavor that exploits the concepts, constructs, and mechanisms of fuzzy set theory. The roles of information granules, information granulation, and the techniques therein are discussed in detail. Particular attention is given to the manner in which these information granules are represented as fuzzy sets and manipulated according to the main mechanisms of fuzzy sets. We introduce unsupervised learning (clustering) where optimization is supported by the linguistic granules of context, thereby giving rise to so-called context-sensitive fuzzy clustering. The combination of neuro, evolutionary, and granular computing in the context of data mining is explored. Detailed numerical experiments using well-known datasets are also included and analyzed.

***Keywords*—** *Context-sensitive fuzzy clustering, data mining, fuzzy sets, granular computing, information granules, knowledge discovery, linguistic labels, unsupervised learning.*

I. INTRODUCTION

Data mining (DM) involves searching for stable, meaningful, easily interpretable patterns in databases [6], [7], [11], [12], [15], [20]. It emerged in the late 1980's in response to the difficulties in interpreting and understanding important associations stored in large databases. DM is an immensely heterogeneous research area that embraces techniques and ideas that stem from probability and statistics, neurocomputing, rough sets, fuzzy sets, data visualization, databases, and so forth. In spite of such a profound diversity, the focal point is constant: to reveal patterns that are not only meaningful but also easily comprehensible. The requirement forces us to represent data and use algorithms that are conducted at a certain level of information granularity, rather than being confined exclusively to tedious number crunching.

People do not always comprehend numbers well, but people do understand information granules. By information granules we refer to collections of data that by consequence of their similarity, resemblance, or operational cohesion can be assembled or associated meaningfully. Such encapsulation or granulation is completed so that we can better comprehend the underlying phenomenon and/or generate more efficient processing.

Interestingly, information granules tend to dominate all DM pursuits [55]. When constructed appropriately, they are easily understood, carry sufficient conceptual substance, and help indicate interesting relationships that are present within the available data. Here we concentrate on the technology of fuzzy sets in DM because this provides a highly intuitive and appealing presentation to the end user. We revisit the ideas of unsupervised learning (learning without exemplars) which are enhanced by domain-specific knowledge. The resulting context-based clustering becomes a useful tool for DM. Moreover, the contexts introduced imply a certain modularization effect that can enhance computational efficiency. The study is illustrated by some selected experimental studies.

The material is organized as follows. First we provide a concise introduction to DM. We raise the fundamental conceptual and algorithms issues, identify main classes of tasks, and highlight a number of long-term pursuits of DM. Next, we concentrate on the problem of information granularity and indicate its central role in DM. In the sequel, we embark on more algorithmic facets of granular computing and show how it can be realized in terms of context-based fuzzy clustering. This also leads us to some insights into the nature of DM in comparison with some generic mechanisms of databases, including generic query methods.

II. DM: MAKING SENSE OF DATA

Everyday business and industry are faced with a flood of data. In fact, this is the most evident sign of the ongoing information revolution. Information is an important commodity. It comes with a genuine challenge. Just to name a few of the commensurate problems consider that:

1) WalMart completes around 20 million transactions per day;

Manuscript received March 30, 1998; revised April 30, 1999. This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

K. Hirota is with the Interdisciplinary Graduate School of Science and Engineering, Department of Computational Intelligence and Systems Science, Tokyo Institute of Technology, Yokohama 226 Japan.

W. Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton T6G 2G7 Canada.

Publisher Item Identifier S 0018-9219(99)06913-3.

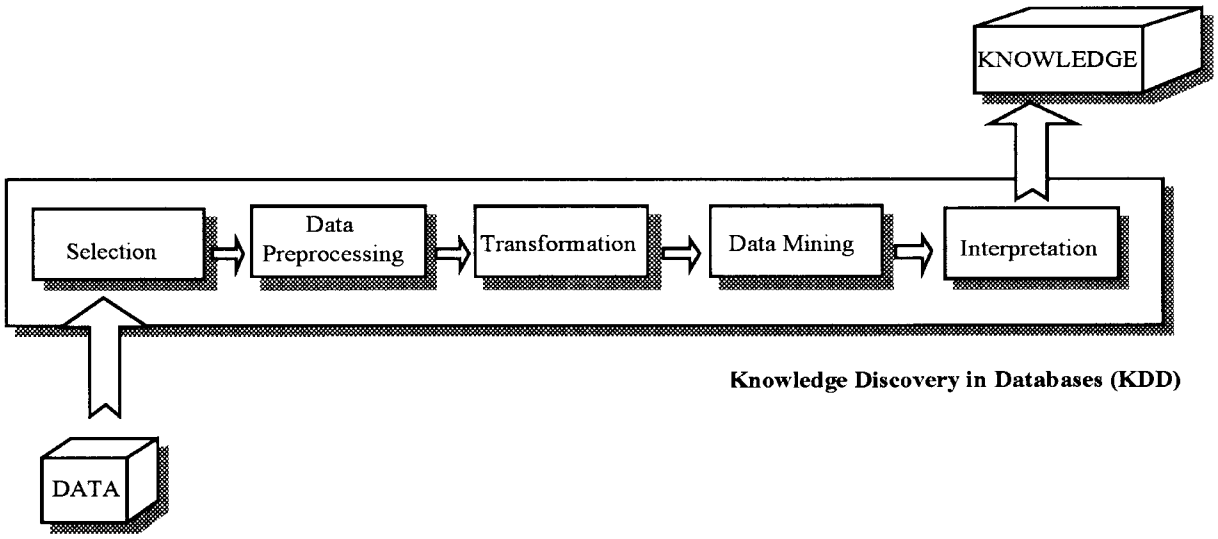


Fig. 1. KDD—a general architecture and its main phases.

2) the planned NASA earth observing system to be launched in 1999 will generate 50 Gb of image data per hour;
3) the rapidly expanding information superhighway will require advanced tools (intelligent agents) for mining (or even finding) data.

Indisputably, we live in a society that is data rich and knowledge poor. Meaningful efforts are required to distill and interpret revealing relationships. Quite descriptively, a mission of knowledge discovery is to make sense of data.

The term knowledge discovery in databases (KDD) is commonly defined as being "concerned with identifying interesting patterns and describing them in a concise and meaningful manner" [15]. If we are primarily concerned with the process of revealing useful patterns within databases (and the ensuing optimization machinery including a specialized query language), then we refer to this activity as DM. The panoply of current methods for knowledge discovery, especially those aimed at DM, is impressive. It ranges from data visualization to more profound approaches that hinge on statistics, neurocomputing, set theory, machine learning, evolutionary computation, and rough sets. For a recent overview of knowledge discovery see [7], [13], and [20].

Here we first provide a brief yet comprehensive overview of both knowledge discovery and DM, revisiting the fundamental concepts and resulting architectures. This makes the paper self contained with respect to the underlying concepts, methods, and algorithms of DM pursuits.

A. A General Architecture of Knowledge Discovery and DM

Knowledge discovery in databases concerns nontrivial processes of identifying patterns in data that are of value to the user. This means that the patterns are valid, novel, potentially useful, and easily understandable. The general process of KDD is outlined in Fig. 1 (see also [11] and [12]). As illustrated, the process of KDD is accomplished in a number of essential phases:

1) selection;
2) data preprocessing;
3) transformation;
4) DM;
5) interpretation.

We begin by identifying and understanding the domain of application, identifying the goal of the activity [as expressed by the user(s)], and collecting prior domain knowledge about the problem. This includes a detailed description of data and their ensuing databases (their format, access mechanisms, etc.). These activities give rise to a more detailed understanding of the problem and an identification of the essential variables that are believed to be crucial. They also provide some common sense qualitative hints as to the general relationships that occur in the problem. This phase of KDD leads to the formation of a target dataset (database) taken from the application domain by choosing a subset of available data (sampled data) or a subset of variables in the dataset. Data preprocessing and "cleaning" is then applied to remove noise and outliers, identify time sequences, and handle missing variables. Subsequently, we proceed with data reduction and projection (transformation). This concerns finding useful features that effectively reduce the dimensionality of the data. The DM activities follow this phase and focus on the identification of patterns. The format and specificity (level of detail) of the data as well as the methods pursued all depend on the main goal of KDD. The back end of KDD concentrates on the evaluation and interpretation of the DM results. Here too, this is facilitated by certain tools for the user [30].

KDD is usually iterative and requires considerable user interaction. This results in many feedback loops at different depths (levels of detail) in one or more phases of the KDD process.

There are different ways to formulate the DM problem. In particular, identifying the level of hierarchy at which to work is important. Consider several typical scenarios.

1) Tell me something interesting about the data.
2) Find interesting associations in the data.
3) Describe the data in terms of some concise functional dependencies that exist between variables.

These three categories constitute a hierarchy of problems with respect to their generality. The first alone (the most general) is an ideal DM process or "architecture" for which one should eventually strive. Unfortunately, this goal is difficult to achieve. What might be interesting to the user is not obvious. Patterns that are revealed (established) by the DM system may not be relevant to the user's purpose. The patterns could just as well be trivial and well known, even though they are well supported by the experimental evidence conveyed by the database under question. Moreover, the most suitable representation (rules, temporal patterns, correlations) for the problem at hand is unknown. Finally, the desired level of detail is also generally unknown *a priori*.

In comparison to this first category of DM activities, finding interesting associations in data forms another important but less general and more manageable pursuit. The primary target here is to establish and quantify relationships between variables that are encountered in the database. Two common ways for quantifying associations include correlations (i.e., correlation coefficients) and relations (being either two-valued relations or fuzzy relations).

Describing data in the form of some functional dependencies results in a more detailed class of patterns in data. These dependencies could be linear or nonlinear. If they are linear, then we may call on regression methods along with a vast number of other modeling/identification schemes. If they are nonlinear then neural networks are interesting models that provide adjustable mapping functions. When developing functional dependencies, we should be cognizant that they may result through the search for causal relations and coincidental relations.

It should be stressed, and unfortunately has not been underscored strongly enough in DM, that there is a fundamental difference between associations and functional dependencies. Associations are direction-free constructs that capture relationships between variables, but they do not make any explicit assertion as to this direction, i.e., what is implied by what. In contrast, functional dependencies are mappings from many variables to another variable in the problem. They necessarily provide a specific direction of the mapping by stating that a certain variable is implied (predicted) by the others. On the algorithmic side, the tools of correlation analysis are used to quantify associations. Among the tools for functional dependencies one may mention neural networks as providers of nonlinear and highly adjustable parametric mappings. A reasonable way to follow is to start developing associations and afterwards proceed with constructing more detailed functional dependencies.

All nontrivial DM tasks are integrally linked with the notion of "interestingness" that arises as an important design factor to be considered in any implementation. Any meaningful quantification of the measure of how interesting something is must be nontrivial. Striving for discovering interesting patterns is regarded as a focal point of the design pursuits that can be accomplished through the feedback loop that involves the group of potential users of the DM system. One of the ways to raise the level of interestingness is by adjusting the granularity of information and usage of the resulting information granules as generic building blocks around which all DM activities tend to revolve.

DM is subsumed by the overall process of KDD. It concentrates primarily on the algorithmic issues of revealing patterns in data. The front end of the KDD includes activities that are inherent to databases and database mechanisms, including various access means (e.g., query languages) and their optimization. These database-driven activities also appear under different names such as "data warehousing," and "online analytical processing" (OLAP). The back end of KDD is associated with various visualization tools.

B. Main Technologies of DM and Their Synergy

A number of essential information technologies contribute to the concepts of DM and their related architectures. The very nature of DM stipulates a synergistic use of such technologies. Those are the same synergistic mechanisms that have driven progress in the area of computational intelligence (CI) [36]. The list of key technologies includes three primary entries:

1) neurocomputing;
2) evolutionary computing;
3) granular computing (fuzzy and rough sets).

These technologies are rarely used in their pure format in system design. In most applications, systems are designed based on an interaction between fuzzy or rough sets, neural networks, and evolutionary methods. Based on the synergy between the technologies, we encountered such important categories as neurofuzzy systems, evolutionary neural networks, and so forth.

Any taxonomy that describes the design of CI systems can exhibit various faces. There are various criteria to quantify potential facets of the symbiosis that occurs between techniques of CI. Fig. 2 positions granular computing, neural networks, and evolutionary methods in a two-dimensional representation of time complexity and level of prior domain knowledge that is available up front. These are two important factors that are instrumental in identifying the most promising symbiotic links. Data (most often numeric readings) and pieces of prior domain knowledge are the two essential sources used in the construction of CI systems. Various technologies exhibit different abilities of representing and processing data and knowledge. Granular computing, and in particular fuzzy sets, is useful at the knowledge side. Neurocomputing is appropriate at the data side. Table 1 provides a matrix that indicates the forms of computation and their hybrids.

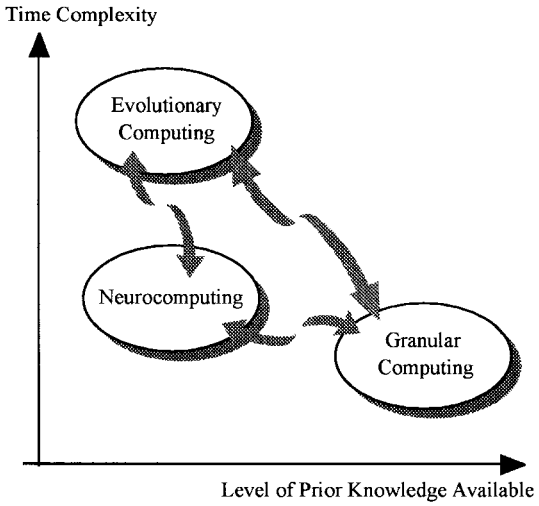


Fig. 2. The technologies of CI, their distribution in the two-dimensional space of knowledge "tidbits" and time complexity, along with their main links.

C. Main Classes of DM Models

Along with the multifaceted nature of the computational technologies that underlie DM, there are a multitude of DM models. The list that follows outlines a series of representative examples.

1) *Associations:* These are rules (conditional dependencies and statements) of the form $A \rightarrow B$, where A and B are sets of items of interest, both being subsets of a certain universe of discourse ($\mathcal{T}$). An association rule is an implication $A \rightarrow B$ where $A \in \mathcal{T}$ and $B \in \mathcal{T}$ and $A \cap B = \varnothing$. There are two main characteristics of an association rule.
   a) The confidence level λ if $\lambda\%$ of transactions $\mathcal{D}$ contain both A and B (where by a transaction we mean a set of elements of $\mathcal{T}$; similarly $\mathcal{D}$ stands for the collection of transactions).
   b) The support σ in $\mathcal{D}$ if $\sigma\%$ of transactions contain the union of A and B, $A \cup B$.

   There are a number of interesting generalizations of associations [46], [48], [51]. These include taxonomies (*is-a* hierarchy) over the items discussed in the problem. The associations may include predicates (ϕ, γ) of the variables, say

$$\phi(A, B, C) \rightarrow \gamma(D, E)$$

   etc.

   The associations are also generalized by incorporating time factor, cf. [44].
2) *Classification:* Based on models of patterns in the database, all of the existing data are mapped to predefined set categories. There are many methods for constructing these categories that vary as to their method of design, performance, learning, etc. There are numerous examples of classifiers, including linear discriminant functions, piecewise linear classifiers, nearest-neighbor classifiers, decision trees (ID3, CART, etc.) [4], [40], [41].
3) *Regression:* A mapping is learned with data that transforms input variables into a real-valued prediction for the dependent variable(s). As such, regression models are aimed at a broad range of approximation tasks.
4) *Clustering:* Classes (groups) of data are developed based on their similarities and differences [21]. There are many clustering methods, including hierarchical- and objective-function-based algorithms. In contrast to classification and regression, this approach does not treat class labels and therefore becomes more demanding conceptually.
5) *Summarization:* This is concerned with finding a compact data description. A contingency table provides well-known example of this approach. Concept learning [14] also falls under the same category of summarization, whereby we mean a logic expression (predicate) such as equivalence, similarity, etc. Concept learning is often used in the discovery of empirical laws [57].

D. Fundamental Long-Term Pursuits of DM

Several fundamental interests in DM include the following.

1) *User-orientation of DM activities:* This issue comes under the general umbrella of human–machine interaction. It concerns aspects of visualization [9] and projecting data on lower-dimensional spaces.
2) *Efficient implementation of DM algorithms:* With the growing size and dimensionality of problems, it becomes necessary to investigate new ways of efficient computing. This also embraces a notion of scalability. We expect that the DM algorithms should be able to scale up efficiently as the size of the problem grows. As shown in [19], any method whose computational complexity exceeds $o(n^{1.5})$ (where n is the number of items of data) becomes highly unlikely to be applicable to large problems.
3) *Parallelization of DM methods:* Parallelizing DM tasks is one of the most efficient ways for making large-scale DM feasible. As observed in [6], several generic DM mechanisms are inherently parallel. For instance, when using decision trees, each branch of the tree can be viewed as a separate task (task parallelization). Training sets can be partitioned across various processors where each is assigned to a single node (data parallelization).
4) *Multistrategy approach to DM:* This facet has been emphasized in many different ways. The various patterns that one is looking for call for diverse strategies to be exploited in tandem [5], [47], [56]. This is again a strong argument in favor of computational intelligence being regarded as a conceptual backbone of DM. Further, an evaluation of DM methods involves a series of performance criteria [29].

Table 1
Main Features and Arising Synergy Between Neural, Evolutionary, and Granular Computing Environments; the Table Describes Ways in Which Each Specific Technology (Located in Consecutive Rows of the Table) Augments the Other Ones (Those Identified in Successive Columns of the Table)

	Neural Computing	**Evoutionary Computing (EC)**	**Granular Computing**
Neural Computing	Computing paradigm that exploits highly distributed and massively parallel processing. Basic processing units (artificial neurons) are arranged into multilayer computing structures (neural networks). Neural networks exhibit significant learning abilities (required plasticity comes with connections of the neurons). Approximation abilities of neural networks are quantified via a *universal* approximation theorem	Neural networks cooperate with evolutionary computing at two distinct levels of problem optimization: *global*, where EC identifies "promising" search regions and *local* more refined neurocomputing techniques	Neural enhancement of learning abilities (plasticity) of constructs of granular computing. Constructs of granular computing act as blueprints of the solution, whereas neural mechanisms are used in their further refinement based on available experimental data. Commonly arise as *neurofuzzy* systems, underlining symbiotic links between fuzzy sets and neural networks
Evolutionary Computing (EC)	Evolution of neural network architectures (e.g., modification of their topologies, number of layers and neurons). It is also used for parametric optimization of neural networks (connections and topology)	*Population-based* optimization approach exploiting various mechanisms of evolutionary activities (variation and selection). Essential to handling complex tasks involving optimization (evolution) of their structure as well as parameters	Global (structure and parameter-oriented) optimization of granular constructs required to cope with complex multivariable and highly nonlinear characteristics of granular systems (e.g., evolutionary optimization of fuzzy rule-based systems)
Granular Computing	Augments neural networks by knowledge-based components either at the architectural level (such as specialized logic-oriented networks giving rise to highly heterogeneous networks) or the learning level (supporting metalearning mechanisms). Metalearning articulates basic learning procedures in terms of adjustable (variable) learning rates (e.g., realized via fuzzy rule-based systems)	Granular computing incorporates domain (prior) knowledge into encoding schemes of EC (e.g., fuzzy encoding) as well as helps establish efficient computing environment (such as modifiable mutation and crossover rates)	Concentrates on formation and processing of *information granules* (information granulation). Realized through the use of fuzzy sets, rough sets, interval analysis, etc. Information granules help concentrate on meaningful chunks of information and reduce the complexity of ensuing processing pursuits (via resulting modularization)

E. Main Properties of DM Pursuits

The required methodology and tools for DM should exhibit some particular features to support the underlying process. It is worth elaborating on the notion of "interestingness" as being the central feature of an DM endeavor. It entails several essential constituents.

1) *Validity:* This property pertains to the significance of the knowledge that has been discovered.
2) *Novelty:* This describes the degree to which the discovered pattern(s) deviate from prior knowledge.
3) *Usefulness:* This relates the findings of the knowledge discovery to the goals of the user, especially in terms of the impact that these findings may have on decisions to be made. This is strongly related to the notion of "interestingness" [45].
4) *Simplicity:* This is primarily concerned with the aspects of syntactic complexity of the presentation of a finding. Greater simplicity promotes significant ease of interpretation.
5) *Generality:* This entails the fraction of the population of data to which a particular finding refers.

All DM pursuits are highly user oriented. In spite of some level of automation, ultimately there is always a user who decides on the character of the resulting DM, its depth and focus, main directions to be taken, etc. The final results of DM need to be interpreted with ease. The compactness of the results, as well as the ease of interpretation, call for treating data with appropriate granularization, rather than simple number crunching. On a technical note, the interaction with the user arises as a certain form of a weak and indirect supervision of the overall search mechanism exploited in the DM process.

While the intent of this section was to provide a general overview of DM as a coherent research endeavor with well-defined and distinct features, we must admit that many issues have barely been tackled. At the same time, it is apparent that the idea of information granularity (and computing at the level of granularity) permeates the entire activity.

III. Granular Computing

Granular computing is geared toward representing and processing information in generic components that help organize, conceptualize, and utilize or reveal knowledge about the problem at hand in an efficient and computationally effective manner. A suitable granulation helps identify important patterns in a siege of numeric data.

Interestingly, the idea of information granulation has existed for a long time, even though it has been manifested only in some specific and limited ways. For instance, an effect of temporal granulation occurs in analog-to-digital (A/D) conversion equipped with an averaging window: one uniformly granulates an incoming signal over uniform time series. An effect of spatial granulation occurs quite evidently in image processing, especially when we are concerned with image compression. On the technical side, the granulation mechanism is inherently linked with information compression and its quality. Mostly, this compression is lossy. The choice of the information granules in terms of their size and distribution can affect the level of losses.

There are a number of conceptual vehicles that construct, manage, and process information granules.

1) *Set theory:* With their basic conceptual skeleton of sets and relations, set theory is used to encapsulate individual elements. Sets gave rise to interval analysis [28] that plays a dominant role in computing with numerical hypercubes and numerical intervals, in particular. Set-theoretic approaches are also encountered in many optimization problems.
2) *Fuzzy sets:* These constructs, introduced by Zadeh [53]–[55], emerge as an interesting augmentation of set theory that helps resolve dilemmas inherently associated with a dichotomization (yes/no) problem associated with the use of sets. By admitting continuous rather than abrupt boundaries between complete belongingness and complete exclusion, fuzzy sets capture a notion of partial membership of an element to the granule in question. This is a dominant concept that permeates most of the advanced descriptors that we encounter in the real world. These include common-sense notion (tall individuals, low inflation, steady income) as well as very specific technical terms (ill-defined matrix, small negative error in a control loop, medium power dissipation).
3) *Rough sets:* These were introduced [31] in order to treat a lack of complete discrimination between classes. They are most commonly applied to information systems and DM.
4) *Random sets:* These sets [26] form a cornerstone of mathematical morphology and have been used frequently in image processing.
5) *Probability theory:* Probability density functions (pdf's) are another interesting example of information granules that can be used to the describe the likelihood of class intervals. In classification problems, a conditional pdf (conditioned on a given class) is an information granule specific to that class.

Each of the above methodologies of information granules has it own research agenda, application areas, and open questions. In many cases they interact rather than compete. In the remainder of this study we select a single methodology of fuzzy sets and discuss its further pursuits in the setting of DM.

A. Fuzzy Sets as Linguistic Granules

A fuzzy set can be regarded as an elastic constraint imposed on the elements from a universe of discourse [32], [34], [38], [39], [53]. By admitting a certain form of elasticity (flexibility) when defining concepts and introducing various mechanisms of fuzzy logic, we can capture the essence of various notions that are encountered in everyday life. For instance, the terms such as low interest rates or high levels of pollution are highly descriptive and meaningful; often these terms are more useful than precise descriptions, such as an interest rate of 6.275%, or 2.4 parts/mm of pollutants. Conceptually, fuzzy sets help alleviate problems with the classification of elements of a boundary nature by allowing for a notion of partial membership to a category. Algorithmically, fuzzy sets make the problems continuous.

Let us underline an important enhancement that is inherent to fuzzy sets. By their very nature, crisp sets are nondifferentiable constructs. Their usage reduces the utility of gradient-based optimization. By consequence, we usually resort to some other types of optimization tools such as random search or evolutionary computation that can provide global optimization and do not require derivative information. Fuzzy sets introduce a welcome aspect of continuity to the problem. On the operational side of the technology of fuzzy sets, we are provided with a vast arsenal of methods that support all facets of computing with fuzzy sets. Operations on fuzzy sets, linguistic modifiers (e.g., very cold), linguistic approximation (e.g., about three meters), and fuzzy arithmetic are a few among the basic computational means that are available.

Fuzzy sets are a backbone of many real-world applications. The success is evident. The industrial facet of this technology is well documented with many practical systems [17]. Hirota [18] provides a comprehensive overview of the advancements in the theory and applications of fuzzy sets.

In what follows, we highlight two points that are predominant in many applications. We elaborate on the aspect of information granularity conveyed by fuzzy sets and a concept of a frame of cognition.

1) Information Granularity of Fuzzy Sets: Defining information granularity helps answer questions as to the information content that resides within a given linguistic granule. Specificity and cardinality of fuzzy sets are most relevant in this regard. An introduction of such measures is motivated by the need for quantifying a level of difficulty (or hesitation) when picking up a single element in the universe of discourse that is regarded as a reasonable representative of the fuzzy set. Two limit cases are intuitively easy to handle.

1) If the fuzzy set is of a degenerate form, namely it is already a single element, $A = \{x_0\}$, there is no hesitation in selecting x_0 as an excellent (the only) representative of A.
2) If A covers almost the entire universe of discourse and contains many elements with membership equal to 1.0, then the choice of only a single element gives rise to a great deal of hesitation.

In the first instance, the fuzzy set is very specific, whereas the specificity of the fuzzy set occurring in the second situation is zero. The specificity measure [50], [51] of a fuzzy set A defined over a certain universe of discourse $\boldsymbol{X}$, described as Sp(A), is a nonnegative number such that:

1) Sp(A) $= 1$ if and only if there exists only one element of $\boldsymbol{X}$ for which A assumes 1 while the remaining membership values are equal zero;
2) if $A(x) = 0$ for all elements of $\boldsymbol{X}$ then Sp(A) $= 0$;
3) if $A_1 \supset A_2$ then Sp(A_1) $\leq$ Sp(A_2).

In [46] the specificity measure is defined as the integral

$$\text{Sp}(A) = \int_0^{\alpha_{\max}} \frac{1}{\text{card}(A_\alpha)} d\alpha$$

where $\alpha_{\max}$ is the maximal value of the membership function A and card(A) denotes the cardinality of the α-cut of A, that is A_α. If we confine attention to normal fuzzy sets (viz. those sets whose maximal membership values attain 1), then a standard sigma count

$$\sigma(A) = \int_{\boldsymbol{X}} A(x)\, dx$$

could serve as a plausible measure of granularity, meaning that it effectively summarizes the number of the elements embraced (at least partially) by the given fuzzy set. Note that the sigma count is inversely related to the specificity measure (higher values of the sigma count of a fuzzy set imply lower values of its specificity measure).

2) The Frame of Cognition: So far we have discussed a single fuzzy set and proposed scalar characterizations, but what really matters in most fuzzy set applications are the families of fuzzy sets. We usually refer the these as a frame of cognition. This notion emerges in fuzzy modeling, control, classification, etc. Primarily, any use of fuzzy sets calls for some form of interfacing with a real-world process. Generally, the frame consists of several normal fuzzy sets, also described as linguistic labels, that are used as basic reference points for fuzzy information processing. Sometimes, in order to emphasize their focal role in this processing, they are referred to as linguistic landmarks. When the aspects of fuzzy information processing need to be emphasized, we may refer to these fuzzy sets as a fuzzy codebook, a concept widely exploited in information coding and its transmission. By adjusting the granularity of the labels we can easily implement the principle of incompatibility [54]. The principle itself expresses that as the complexity of the system increases, its model will exhibit two highly conflicting and impossible-to-achieve characteristics: meaningfulness and precision. When the model becomes too precise, it becomes meaningless. A suitable balance has to be struck by admitting a level of precision that makes the model relevant. By changing the size of the information granules, we can easily cover a broad spectrum that ranges from qualitative form (symbols) up to that of the numerical character with the highest possible granularity.

Let us now move to a more formal definition. A frame of cognition [32], [33]

$$\mathcal{A} = \{A_1, A_2, \ldots, A_c\}$$

is a collection of the fuzzy sets that is defined in the same universe of discourse $\boldsymbol{X}$ and satisfies the following conditions.

a) Coverage: $\mathcal{A}$ covers $\boldsymbol{X}$, that is, any element of $\boldsymbol{X}$ belongs to at least one label of $\mathcal{A}$. More precisely, this requirement can be written in the form

$$\underset{x \in \boldsymbol{X}}{\forall} \quad \underset{i=1,2,\ldots,c}{\exists} \quad A_i(x) > 0.$$

The notion of coverage emphasizes that the universe of discourse $\boldsymbol{X}$ becomes represented by the collection of the linguistic terms. Being more stringent, we may demand an ϵ-level of coverage of $\boldsymbol{X}$, that formalizes in the following form:

$$\underset{x \in \boldsymbol{X}}{\forall} \quad \underset{i=1,2,\ldots,c}{\exists} \quad A_i(x) > \epsilon$$

where $\epsilon \in [0, 1]$ stands for the assumed coverage level. This simply means that any element of $\boldsymbol{X}$ belongs to at least one label to a degree not less than ϵ. Otherwise, we can regard this label as a representative of this element to a nonzero extent. The condition of coverage assures us that each element of $\boldsymbol{X}$ is sufficiently represented by $\mathcal{A}$. Moreover, if the membership functions sum up to one over

$\boldsymbol{X}$

$$\underset{x \in \boldsymbol{X}}{\forall} \quad \sum_{i=1}^{c} A_i(x) = 1$$

then the frame of cognition is referred to as a fuzzy partition.

b) Semantic soundness of $\mathcal{A}$: This condition translates into a general requirement of a linguistic "interpretability" of its elements. Especially, we may pose a few more conditions that characterize this notion in more detail (see also [39]).

1) A_i's are unimodal and normal fuzzy sets. In this way they identify the regions of $\boldsymbol{X}$ that are semantically equivalent with the linguistic terms.
2) A_i's are sufficiently disjoint. This requirement assures that the terms are sufficiently distinct and therefore become linguistically meaningful.
3) The number of the elements of $\mathcal{A}$ is usually quite low. Some psychological findings (cf. [27]) suggest 7 ± 2 linguistic terms to constitute an upper limit for the cardinality of the frame of cognition when being perceived in the sense of a basic vocabulary of linguistic terms. Again, these numbers change in the case of a visual memory (where this number is 4 ± 2 items). In general, these numbers are quite low.

The above features are given in a descriptive rather than formal format and should be treated as a collection of useful guidelines rather than a series of strict definitions.

DM calls for a multitude of activities that depend on the type of user. For instance, a corporate report usually requires pieces of knowledge about associations between various factors (variables) collected at a highly general level. They help in gaining a global overview of a problem, identifying the most crucial relationships, and undertaking some strategic decisions. At the other end of the spectrum arise far more specific situations in which we require detailed yet very local information. What is common to these two decision scenarios (and many others) is a concept of information granularity, which concerns the issue of summarizing the information (compression). Fuzzy sets, as well as set theory to some extent, support this essential feature. They can be regarded as conceptual "filters" that focus on a specific level of detail that can then be searched for patterns within a database.

Consider a few examples of fuzzy sets as shown in Fig. 3. They illustrate the underlying use of the concept of information granularity. For instance, the fuzzy set in the upper part of Fig. 3(a) is far more specific (detailed) than the one displayed at the bottom. In the latter case, we are not concerned about details (and, in fact, they become *hidden* in the description of interest).

There remains an aspect of expressing information granularity in a quantitative way. One can consider a sigma count (being an example of an energy measure of fuzziness) as a good option in the case of normal fuzzy sets [21], [36], [38]. More generally, for subnormal fuzzy sets (i.e., these with the maximal membership value not reaching 1.0), one can deal with the specificity measure. Following the semantics of fuzzy sets, we easily construct hierarchies of concepts starting with a very specific and detailed description and ending with general ones. The process of generalization and specialization is illustrated in Fig. 4. In the first instance, we use a standard logical operation that may lead to expressions of the form

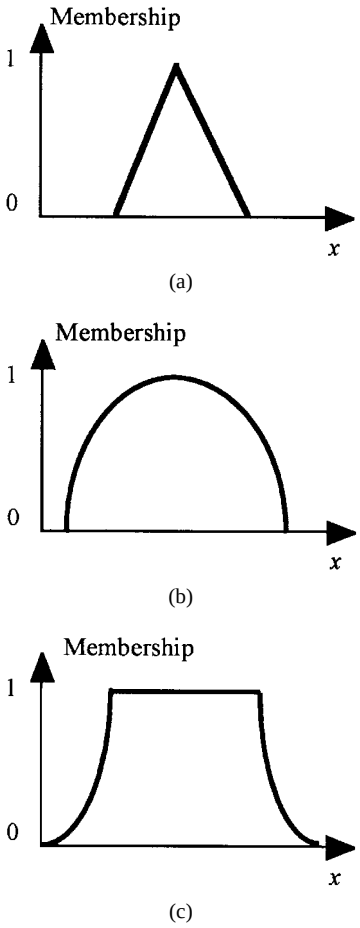


Fig. 3. Fuzzy sets and an effect of information granularity associated with them; fuzzy sets ordered (a)-(c) from most specific to least specific information granules.

$$A_1 \text{ or } A_2.$$

The result of the OR operation is a fuzzy set of lower granularity. In the second case, we apply a linguistic modifier of fuzzification (more or less)

$$\text{more or less } A_2.$$

The contrast intensification operation has an opposite effect on the original fuzzy sets leading to its specification (refinement), say

$$\text{very } A_3.$$

A similar effect of increasing information granularity can be achieved by applying the AND operation while starting from a union of several fuzzy sets. Note, however, that the AND operation yields a subnormal fuzzy set.

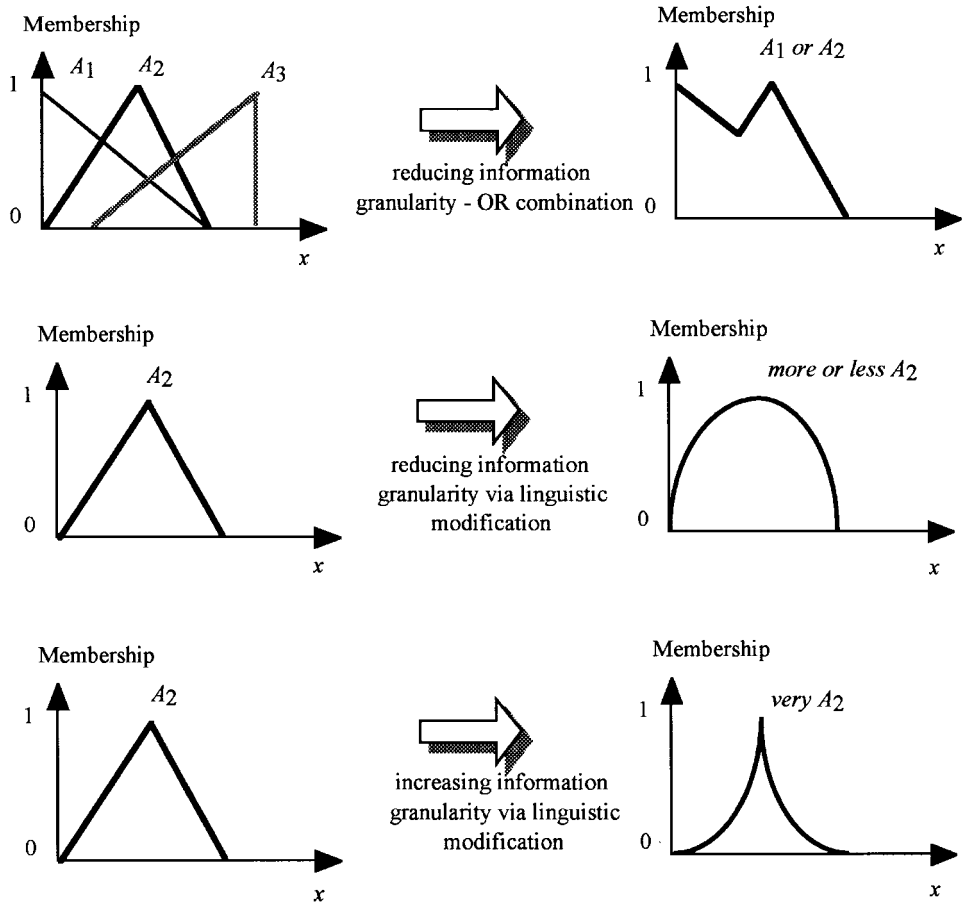


Fig. 4. An effect of generalization and specialization (refinement) producing information entities of lower or higher granularity through OR combination and the use of linguistic modifiers.

More importantly, using fuzzy sets offers the potential for "chunking" data at many scales (associated with several nested universes of discourse) that make it possible to pyramid the constructs (Fig. 5). In other words, we may use the same collections of fuzzy sets (linguistic labels) that become redefined at successively subsumed universes of discourse (spaces). This greatly improves the overall process of concentrating successively on some regions of interest and exploring them in more depth if necessary.

B. Linguistic Granules and Associations as Blueprints of Numeric Constructs

Linguistic granules (and information granules in general) serve two important purposes.

1) They help establish sound and meaningful chunks of information that provide a background for further refinements.
2) They support modularization of the dataset, which reduces the level of required computing that is necessary to reveal detailed relationships at a numeric rather than linguistic level.

Fig. 6 illustrates a number of possible follow-ups that are founded in linguistic granules; they include correlation analysis, regression models, and neural networks. In all these cases, information granules serve as a design blueprint by producing meaningful entities and revealing relationships between them. Subsequently, these may be refined to develop more detailed relationships within a realm of the individual information granules. This also means that the ensuing models become local, being confined to the boundaries of the given information granule.

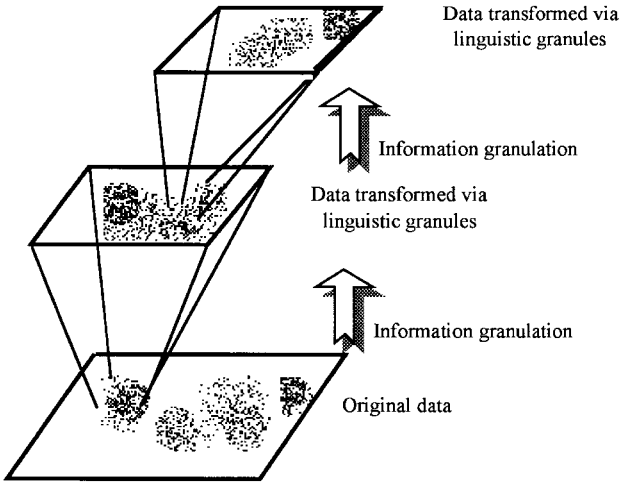


Fig. 5. Realization of a pyramid architecture of DM; introduced information granules of different level of granularity give rise to the effect of focusing.

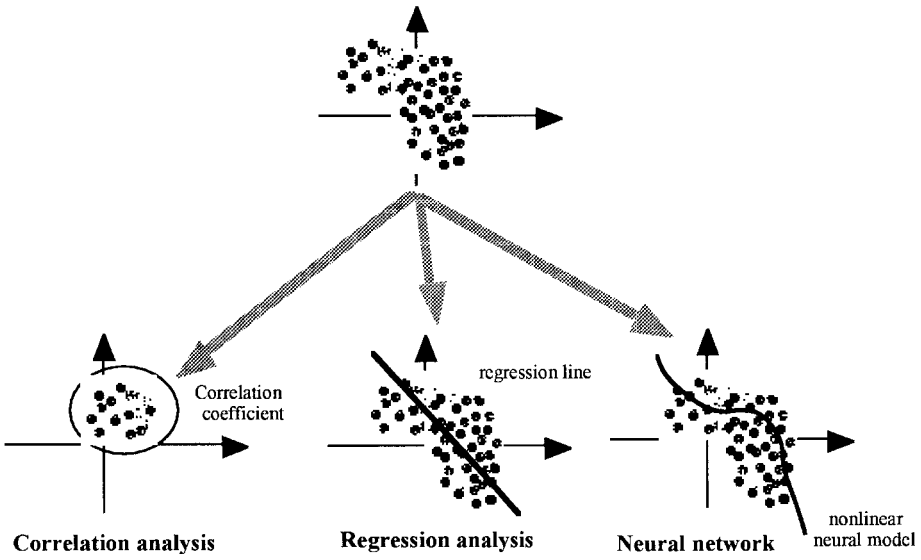


Fig. 6. Refinements of associations between linguistic granules realized with the aid of correlation analysis, regression models, and neural networks. The data points with nonzero membership values of the specific linguistic granule are further quantified by a single correlation coefficient and captured by a linear regression model or a nonlinear model realized as a neural network.

More specifically, correlation analysis quantifies dependencies between the data points that have been invoked by a certain linguistic granule. It is well known that correlation analysis addresses only a linear relationship between variables. The resulting correlation coefficient quantifies the strength of this linear association. The focal effect provided by the linguistic terms contributes to the increased values of the correlation coefficient. The same correlation coefficient, if computed over all available data, often assumes far lower values. Linguistic granules can be helpful in building a standard linear regression model where the computations take into consideration data points that are weighted by their levels of adherence to the already established linguistic granules. Similarly, the same weighting phenomenon of the individual data is applicable when designing neural networks. Observe that in both cases, the linguistic granules provide a regularization effect and eliminate or reduce the impact of outliers on the performance of the final model.

IV. Fuzzy Clustering in DM

In this section, we discuss a role of unsupervised learning (clustering) in the problem of DM. A highly enlightening and appealing characterization of the clustering approach or grouping is offered in [23]: "cluster analysis is the art of finding groups in data." This emphasizes that the primary thrust of clustering is to arrange a collection of data into a small number of groups (clusters) so that similar elements are allocated to the same group. The elements (patterns) that are quite disparate should be placed into separate categories. The literature on this subject is enormously rich; the reader may refer to classic references [1], [10], [16], [21]. One recent publication concentrates on knowledge-based approaches [2].

It is of utmost importance to position clustering techniques as a viable methodology of DM. Does clustering live up to the expectations raised in the setting of DM? In order to answer this crucial question, we should reiterate the main postulate of DM.

The proactive role of a potential user is visible in the DM process. While largely autonomous, the overall procedure is guided generally by a user who is interested in different ways in which the data can be examined. There are several detailed conceptual and operational facets, including the following.

1) Information granularity at which all mechanisms of DM are active. This granularity could be (and usually is) highly diversified in terms of its level. In regions of particular interest, attention can be paid to minute details that in turn dictate a high degree of granularity (eventually to the numeric level). Otherwise, the regions of low interest call for an allocation of relatively coarse (linguistic) information granules. The variable level of information granularity supports the idea of interestingness (see Section II) and leads to its efficient implementation.
2) Transparency of a generated summary of the main associations revealed through DM. Here the transparency is viewed in terms of the ease of understandability of the summary as well as its relevancy. Again, the role of information granulation becomes apparent.

These two considerations suggest that clustering algorithms are to be embedded in the auxiliary framework that includes these DM requirements. In the following discussion, we elaborate on context-oriented fuzzy clustering. The choice is dictated primarily by conceptual simplicity along with an associated algorithmic efficiency.

V. Context-Oriented Fuzzy Clustering

To illustrate how clustering, and fuzzy clustering in particular, plays a role in DM, let us consider a relational

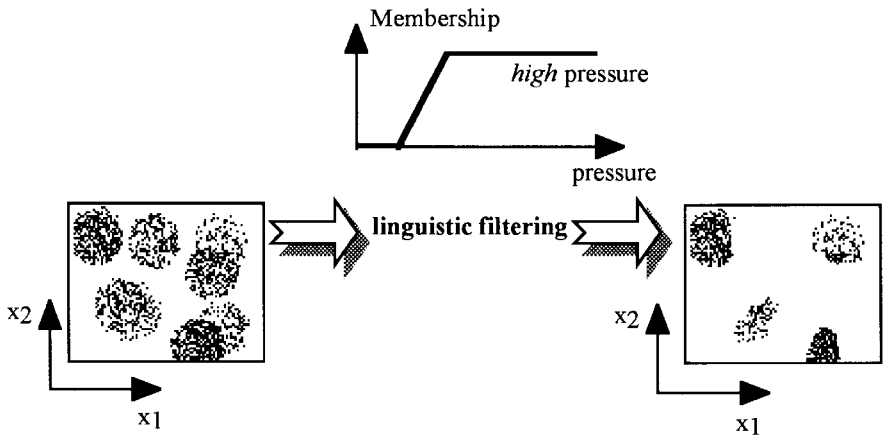


Fig. 7. The use of linguistic context (high pressure) in data filtering.

table (array) $\mathcal{X}$ comprising objects regarded as vectors of real numbers. We are interested in revealing (discovering) a structure and eventually quantifying functional dependencies that manifest throughout this table. The focal nature of DM is achieved by specifying linguistic terms prior to launching any detailed analysis and executing computationally intensive algorithms. While there is a great diversity of DM processes, we highlight only a few most representative and interesting scenarios:

Let us consider one of the attributes of interest (call it a context variable) and define therein a fuzzy set (linguistic term of focus) such that

$$A\colon \boldsymbol{Y} \rightarrow [0,1]$$

where $\boldsymbol{Y}$ stands for a universe of discourse of this attribute (variable). The problem transforms as follows:

$$\text{reveal structure in } \mathcal{X} \text{ in context } \mathcal{A}$$

where the context of DM is established as

$$\mathcal{A} = \{A\colon \boldsymbol{X} \rightarrow [0,1]\}.$$

The essence of such clustering is portrayed in Fig. 7. If we confine attention to one of the variables as a context variable (say, pressure) over which one defines a collection of linguistic terms (information granules), this particular choice sheds light on some section of the entire dataset that become of interest in light of the assumed context. Some regions of data are also practically eliminated from any further analysis under the auspices of the particular information granule of the context variable. While Fig. 7 emphasizes the concept itself, further details are exemplified through Fig. 8.

Note that the selected information granule (context) directly impacts the resulting data to be examined. The context can be regarded as a window (or focal point) of DM. The introduced linguistic context provides a certain tagging of the data. Fig. 8 illustrates this effect. The fuzzy set of context is defined in the form of the following exponential membership function:

$$\mathcal{A}(x) = \begin{cases} \exp\left(-\dfrac{100-x}{100}\right) \\ 1 \quad \text{if } x > 100. \end{cases}$$

The problem of DM reads as follows:

$$\text{reveal structure in } \mathcal{X} \text{ in context } \{\text{pressure} = \text{high}\}.$$

Similarly, if we may be interested in characterizing customers of medium or high disposable income, the resulting clustering task would then read as follows:

$$\text{reveal structure in market database in context } \{\text{disposable income} = \text{medium or high}\}.$$

Several attributes can form the composite context. For instance, let A and B be two fuzzy sets defined in $\boldsymbol{Y}$ and $\boldsymbol{Z}$, respectively. Then any composite context $\mathcal{A}$ is formed as a Cartesian product of A and B

$$\mathcal{A} = A \times B$$

that is

$$\mathcal{A}(y,z) = \min(A(y), \boldsymbol{B}(z)).$$

Similarly, we may arrive at the problem formulated as

$$\text{reveal structure in } \mathcal{X} \text{ in context } \{\text{pressure} = \text{small and temperature} = \text{medium}\}.$$

In addition to the two basic forms of the linguistic contexts, there are a number of interesting extensions, see Fig. 9.

The examples below illustrate each of these contexts.

1) *Composite logical context:* Pressure is small and temperature is low or humidity is medium.
2) *Composite relational context:* Prices of product "a" and discount prices of product "b" are similar.
3) *Composite regression context:* Error of linear regression model $x_i = f(x_j, a)$ is negative small.

It is instructive to recall that the clustering problem of the form

$$\text{reveal structure in } \mathcal{X}$$

is context free and comes exactly in the same format as commonly studied in the standard domain of data clustering.

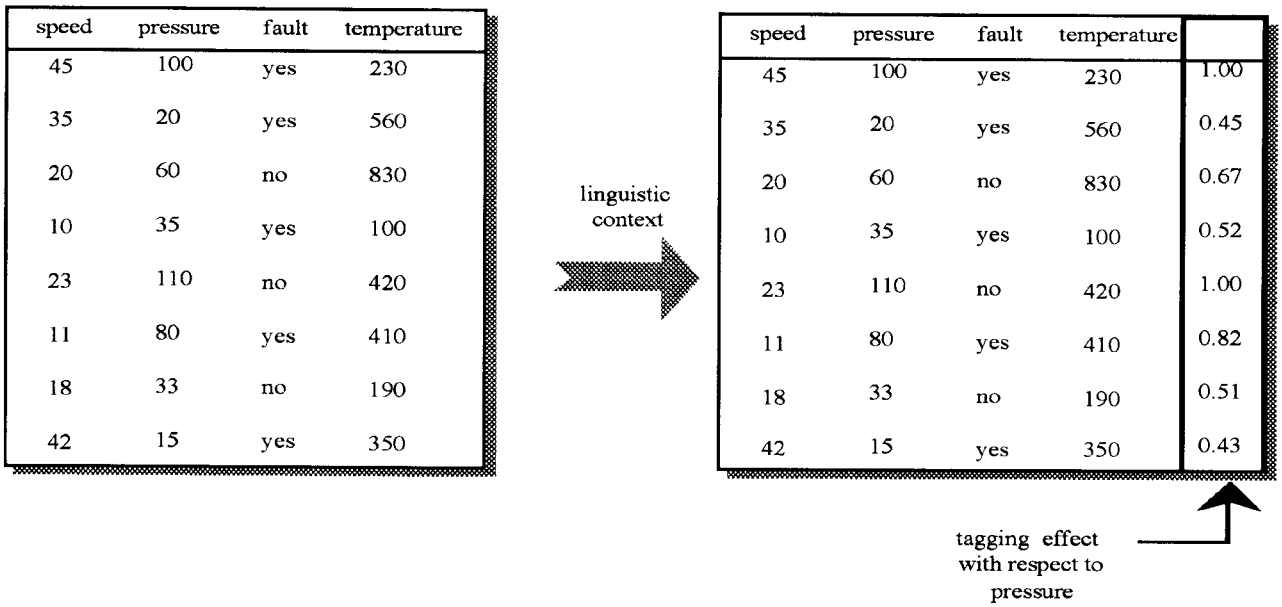


Fig. 8. Data tagging with the use of the fuzzy set of context; note an effect of reducing the dataset to be clustered when some data elements with low or zero tagging values (in this case with respect to "pressure") can be dropped.

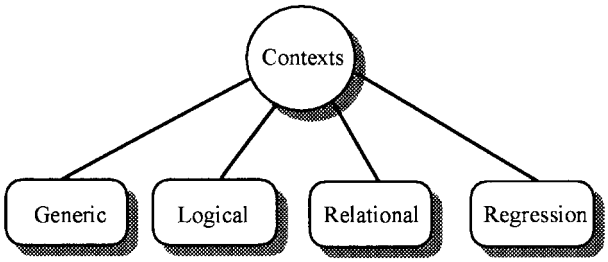


Fig. 9. A taxonomy of linguistic contexts exploited in DM that distinguishes between several categories of contexts: generic contexts describe a single linguistic entity; logical contexts combine a number of generic contexts through the use of logic operations (AND, OR, NOT); relational contexts develop in the form of some relations between linguistic terms; and regression contexts concern properties (e.g., errors) of some already constructed regression models.

A. The Algorithm

The conditioning aspect (context sensitivity) of the clustering mechanism is introduced into the algorithm by considering the conditioning variable (context) given the values $f_1, f_2, \ldots, f_N$ on the corresponding patterns. More specifically, f_k describes a level of involvement of $\boldsymbol{x}_k$ in the assumed context, $f_k = \mathcal{A}(k)$. In other words, $\mathcal{A}$ acts as a DM filter (or a focal element or a data window) by focusing attention on some specific subsets of data. The way in which f_k can be associated with or allocated among the computed membership values of $\boldsymbol{x}_k$, say $u_{1k}, u_{2k}, \cdots, u_{ck}$, is not unique. Two possibilities are worth exploring.

1) We admit f_k to be distributed additively across the entries of the kth column of the partition matrix, meaning that

$$\sum_{i=1}^{c} u_{ik} = f_k \qquad k = 1, 2, \ldots, N.$$

2) We request that the maximum of the membership values within the corresponding column equals f_k

$$\max_{i=1,\,2,\cdots,\,c} u_{ik} = f_k \qquad k = 1, 2, \ldots, N.$$

We confine attention to the first manner of distribution of the conditioning variable. It is in rapport with most constraints encountered in the standard fuzzy c-means (FCM) clustering method and its variants. Bearing this in mind, we modify the requirements for the partition matrices and define the family

$$\mathcal{U}(\boldsymbol{f}) = \left\{ u_{ik} \in [0,1] \middle| \sum_{i=1}^{c} u_{ik} = f_k \quad \forall_k \quad \text{and} \quad 0 < \sum_{k=1}^{N} u_{ik} < N \quad \forall_i \right\}.$$

Thus the standard normalization condition where the membership values sum up to one is replaced by the involvement (conditioning) constraint. The optimization problem is now reformulated accordingly [35]–[37]

$$\min_{U, \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_c} Q$$

subject to

$$U \in \mathcal{U}(f)$$

Let us proceed with deriving a complete solution to this optimization problem. Essentially, it can be divided into two separate subproblems:

1) optimization of the partition matrix U;
2) optimization of the prototypes

Table 2

Given: The number of clusters (c). Select the distance function $\|.\|$, termination criterion e (>0) and initialize partition matrix $U \in \mathcal{U}$. Select the value of the fuzzification parameter "m" (the default is $m = 2.0$)

1. Calculate the centers (prototypes) of the clusters

$$v_i = \frac{\sum_{k=1}^{N} u_{ik}^m x_k}{\sum_{k=1}^{N} u_{ik}^m}$$

$i = 1, 2, ..., c$

2. Update the partition matrix

$$u_{ik} = \frac{f_k}{\sum_{j=1}^{c} \left(\frac{\| x_k - v_i \|}{\| x_k - v_j \|} \right)^{2/(m-1)}}$$

$i = 1, 2, ..., c, j=1, 2, ..., N$

3. Compare U' to U, if termination criterion $\|U' - U\| < e$ is satisfied then stop, else return to step (1) and proceed with computing by setting up U equal to U'

Result: partition matrix and prototypes

As these tasks can be handled independently, we start with the partition matrix. Moreover, we notice that each column of U can be optimized independently, so let us fix the index of the data point (k) and reformulate the resulting problem

$$\min_{U} \sum_{i=1}^{c} u_{ik}^m d_{ik}^2$$

subject to

$$\sum_{i=1}^{c} u_{ik} = f_k$$

(in other words, having the fixed data index, we have to solve "N" independent optimization problems). To be more concise, we have introduced the notation d_{ik} to describe the distance between the pattern and the prototype, namely $d_{ik}^2 = \|\boldsymbol{x}_k - \boldsymbol{v}_i\|^2$.

As the above is an example of optimization with constraints, we can easily convert this into unconstrained optimization by using the technique of Lagrange multipliers.

The overall algorithm is summarized as a sequence of steps shown in Table 2. There are two important design components of the clustering method: 1) the distance function $\|\cdot\|$ being a primordial component of the minimized objective function and 2) the fuzzification parameter (m). The distance function articulates a notion of similarity (or dissimilarity) between two elements in the data space. The typical variants concern Euclidean, Hamming, and Tschebyschev distance functions. The Euclidean distance is the most commonly used. The Hamming distance promotes some important robustness features. The values of the fuzzification factor become reflected in the form of the clusters being produced (or, equivalently, the form of membership function). One can observe that with increasing values of "m" there is a profound rippling effect where the membership functions tend to exhibit more local minima. For lower values of the fuzzification factor, the resulting membership functions tend to resemble characteristic functions of sets, meaning that we are getting less elements with intermediate membership values. Simply, the results become localized around zero or one.

The context $\mathcal{A}$ has a profound effect on the performance of clustering. If $\boldsymbol{f} < \boldsymbol{f}'$, then the population of patterns involved in grouping and placed under context $\boldsymbol{f}'$ is lower. Similarly, the number of eventual clusters could be lowered as well. The above inclusion relation between the contexts holds if the context fuzzy sets are made more specific or if the contexts consist of more constraints (focal points). In the first case we get $\mathcal{A} \subset \mathcal{A}'$, where $\boldsymbol{f}$ is implied by $\mathcal{A}$ and $\boldsymbol{f}'$ by $\mathcal{A}'$. In the latter, the ensuing $\boldsymbol{f}$ is associated with $A \times B \times C$ and $\boldsymbol{f}'$ comes with $A \subset B$; here again $\mathcal{A} \subset \mathcal{A}'$.

Let us underline that the context of clustering plays an important role in discovering knowledge nuggets—rare yet essential pieces of information. Without any direction imposed by the user, such knowledge nuggets could be easily washed away in a mass of useless but frequent (and thus statistically meaningful) data. The filtering of data accomplished by the context prevents this from happening.

One should emphasize that the membership values of contexts do not sum up to one; a similar phenomenon can be witnessed in possibilistic clustering [25] and clustering with noisy clusters [8]. One should stress, however, that the origin of these two departures from the original constraint is completely different.

B. Quantification of the Associations Between Information Granules

Context-based clustering leaves us with the number of contexts and induced clusters. The links (associations)

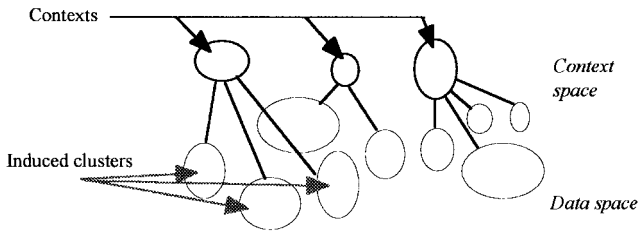


Fig. 10. Linguistic contexts and induced clusters—the formation of the basic associations.

between these entities are assumed by the method but not quantified at all. What we are left with is a structure depicted in Fig. 10. The figure shows a web of links between the contexts (defined in the context space) and a series of induced clusters (those being located in the data space). Note, however, that these links have not been quantified. Some of them could be far more meaningful than others.

The manner in which further more detailed quantification of the associations are created is left for further developments. The following method is anticipated. The use of the standard Boolean confusion matrix in the development of the associations. In this case one admits a simple threshold criterion by assigning successive data to the induced clusters and the respective contexts by taking into consideration the highest membership grades. This is the simplest possible criterion that leads to the standard confusion matrix. Each row of the matrix denotes an induced cluster, whereas the columns describe the contexts. The threshold criterion allocates the data across the matrix. Counting the number of elements in each row provides a score to the associated context-induced cluster. If the nonzero number of occurrences happens only in the single entry defined by this specific context and not otherwise, then the association concerns only the context under consideration. It could well be that there are some other nonzero entries in this row, meaning that the discussed induced cluster expanded too far and embraced some auxiliary contexts. All the obtained associations can be ordered by inspecting the entries of this contingency table.

While this method can be utilized as a basic vehicle aimed at the evaluation of the quality of the associations and produce some of their pruning, this approach does not discriminate between the data points that are very close to the centers of the prototypes and those that are quite peripheral to the prototypes of the induced clusters or/and the contexts themselves. No matter where the data are located, they contribute equally to the counting procedure applied to the contingency table. This, however, could be very restrictive, especially in light of the continuous boundaries between the resulting constructs. To alleviate this deficiency, we generalize the contingency table by counting the levels of strength of the respective induced clusters and the pertinent context. In the simplest case, the entries of the contingency table can be updated using the values of the products of the fuzzy sets or relations under consideration. The contingency table generalized in this way does not focus on the counting of events (coincidences) but concentrates primarily upon the activation levels of the associations obtained by the available data. As before, one can order the associations by inspecting the entries of the table. The association with only one nonzero entry in the row that is situated at the respective context and a high value of this particular element of the contingency matrix assumes a high score. This approach does not take into consideration the number of occurrences but counts a total mass of activation of the coincidences between the clusters and the contexts.

There is also another alternative approach that attempts to strike a balance between the overall level of activation and the number of binary occurrences of the highest activations of the entities (clusters and context). One simple method takes these two matrices and determines their ratio. More specifically, we divide the continuous version of the contingency table by its Boolean counterpart. The entries of the new matrix formed in this way represent an average level of coincidence between the clusters and the respective context. As before, the associations can be easily ordered based on the distribution of the entries of the corresponding row of the matrix. More specifically, in spite of the form of the matrix, the following index can serve as an indicator of the relevance of the association:

$$\kappa = \frac{\text{sum of entries of the rows corresponding to the context}}{\text{sum of all entries}}.$$

If κ assumes high values, then the association is regarded as highly relevant. This occurs when there are no other nonzero entries in this row (such nonzero entries tend to reduce the value of κ) and the respective entry is high enough. One could have a highly focused association with no activation of some other contexts but with very low values of the entry; this also contributes to the overall low performance of the association.

Once the associations have been ordered, only the most significant can be revealed as the result of mining of dataset.

Finally, note that the mining activities have been performed at a certain level of information granularity and as such do not allow introducing more details without further computation. In other words, what we have is a collection of meaningful associations Fig. 11 that can be treated as general patterns

$$(\text{induced cluster} \longleftrightarrow \text{context}).$$

Any speculations about the internal details of this association are beyond the discussion carried out in the conceptual realm discussed here. In fact, by imposing a certain level of granularity, our intent was to avoid getting into such details. Regardless, if at some point of further analysis the numerical details need to be revealed, one has to pursue numerically oriented computing of the relationships within the specific entities involved at this level of building the patterns within data.

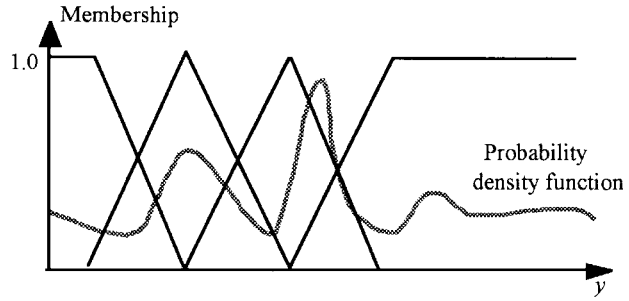


Fig. 11. Triangular fuzzy sets of context and the equalization problem—also indicated is a pdf of the context variable y.

The computations of their membership functions result directly from the assumed clustering model. Thus we derive

$$u_i = \frac{1}{\sum_{j=1}^{c} \left(\frac{\|x - v_i\|}{\|x - v_j\|} \right)^{2/(m-1)}} \qquad j = 1, 2, \ldots, c$$

with the same distance function as encountered in the original method.

Interestingly enough, the patterns have nothing to do with any specific direction—what has been revealed are just plain associations between the relations and the context.

VI. Algorithmic and Application-Driven Aspects of Context-Based Clustering

Context-based fuzzy clustering possesses interesting properties. First, it happens to be computationally efficient where the efficiency is gained through the introduction of a modularization effect.

A. Modularization Effect and Its Computational Efficiency

There is an interesting issue of computational efficiency provided by forming linguistic contexts and clustering the data falling within such a context. We show that this type of redefining the problem of clustering and executing it for selected portions of the entire database pays off in terms of efficiency. More specifically, we will be concerned with the computational effort associated with a single iteration of the clustering method. In order to quantify this effect, consider the problem of clustering N data points in c clusters. The overall computing effort involves calculations of the distance functions between the patterns and the prototypes. Not considering any other operations, one can easily conclude that the computations of each entry of U, say u_{ik}, requires $(1+c)*c$ operations for computing the distances. For the entire partition matrix we end up having $(1+c)*c*N$ computations of the distances.

Let us assume that the clustering exploits p contexts and, in the sequel, each context activates the same fraction of all the patterns, that is, N/p. This is a simplified yet quite reasonable assumption. Furthermore, we select c' clusters per each context where $c' = c/p$. Again, this is a rational assumption as we finally get the same number of the clusters, no matter whether we cluster all of the data at once or proceed with the consecutive contexts. Following the same motivation, the computational overhead required for clustering based on a particular context equals

$$\left(1 + \frac{c}{p}\right) \frac{c}{p} \frac{N}{p}.$$

Considering that this computational effort has to be multiplied by the number of the contexts identified in the clustering problem, we obtain the expression

$$p\left(1 + \frac{c}{p}\right) \frac{c}{p} \frac{N}{p} = \left(1 + \frac{c}{p}\right) c \frac{N}{p}$$

which is still less than the computational effort for the context-free clustering.

The ratio

$$\rho = \frac{\left(1 + \frac{c}{p}\right) c \frac{N}{p}}{(1+c)cN} = \frac{1 + \frac{c}{p}}{p(1+c)}$$

can serve as an indicator of the reduction of the computing effort due to the introduction of the linguistic contexts and the resulting modularization of the clustering problem.

The resulting savings can be substantial. For instance, we partition the data into 40 clusters. Furthermore, assume that we introduce eight contexts (and for each of them we cluster the pertinent data into five clusters). Under such circumstances one obtains

$$\rho = \frac{1}{8} \frac{(1 + \frac{40}{8})}{(1+40)} = 0.1786$$

indicating that the context-based clustering accounts only for around 17.9% of the total effort being spent when carrying out the context-free clustering. This ratio is even lower if we increase the number of the contexts, say up to ten (thus building four clusters in each of the contexts); now ρ attains around 12.2% of the original computing effort. Note, however, that these figures could be too low as we are dealing only with a single iteration of the method. Thus for running the algorithm for each context, this savings should be reduced by the factor equal to the number of contexts assumed in the method. Importantly enough, the clustering for the individual context may terminate in fewer iterations than the clustering including all the data.

B. Determination of Fuzzy Sets of Contexts

The selection of the fuzzy sets of contexts as well as their number is induced by the nature of the problem of DM. These fuzzy sets can be completed based on the preferences of the user. While this is valid to a high degree, one should become aware of some implications stemming from the choice of the linguistic terms being made at the very beginning of the overall cycle of DM. First, note that the granularity of the fuzzy set of context

activates a certain subset of the entire database. If the context becomes very narrow (of high granularity), it could have easily happened that there will not be enough data to be clustered. This setting of the context could be done on purpose (say, in order to focus search for patterns on some specific cases), but it could be a result of an improper selection of the contexts. We have to remember that in any case our contexts (intentions) are confronted with the data (facts) and if there are not enough of them, no strong conclusions can be derived and supported in light of the existing experimental evidence. Having this in mind, the simplest possible criterion would be to look at the sigma count of the data activated by the specific context. If this count does not exceed an assumed threshold, the context needs to be changed (expanded). More specifically, if $\boldsymbol{f}$ denotes one of the contexts to be utilized, the associated sigma count of the fuzzy set of context ($\boldsymbol{f}$) is defined as

$$\sigma(\boldsymbol{f}) = \sum_{k=1}^{N} \boldsymbol{f}(y_k).$$

Then if this becomes lower than the pre-assigned threshold, we need to revisit the context and make it more general.

One may think of an equalization of the linguistic contexts and make them activate the same fraction of the database. Let us assume that we deal with r contexts, $\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_r$. For each of them we determine its sigma count, $\sigma(\boldsymbol{f}_1)$, $\sigma(\boldsymbol{f}_2), \ldots, \sigma(\boldsymbol{f}_r)$ and modify the contexts accordingly to make these values equal. In general, the context defined over the region of the context variable of low pdf becomes broader (as we have to accumulate enough membership values within the respective fuzzy set). On the other hand, for the regions where the pdf is high, the resulting fuzzy sets of contexts could be made relatively narrow. This tendency seems to be intuitively well justified.

When it comes to the algorithmic aspect, one can simplify the problem by looking into triangular fuzzy sets of context with 1/2 overlap between two successive fuzzy sets, as illustrated in Fig. 11 (the first and the last fuzzy set are described by trapezoidal membership functions). Then the parameters of the membership functions can be easily determined in a systematic way by moving toward higher values of the argument and computing the sigma count of the resulting fuzzy set.

For the uniform pdf, we end up with a uniform distribution of the membership (which fully adheres to our intuitive findings). The linguistic equalization arising in this way assures us that the linguistic terms are equally meaningful as being supported by the experimental data to the same extent.

C. Context-Based Clustering and Databases

Context-based clustering carries some interesting resemblances to standard queries in databases. Moreover, it nicely generalizes the concept of a query that could be better described as a metaquery. In the standard querying process, one formulates a query and the mechanisms of database help retrieve all pertinent records from the database that respond to the formulated request. Obviously, for the query of the type: "find all customers who have recently bought a Ford Contour and are of middle age" (assuming that the linguistic term "middle" has already been defined) the database retrieval mechanisms will produce a long (and perhaps in some cases useless) list of such individuals. The expectations are that we will be provided with a concise and meaningful characterization (description) of this specific sector of the car market. This, in fact, is what the discussed clustering method does (refer to Fig. 12). The generalized metaquery is just the imposed context while the characterization comes in the form of the induced clusters.

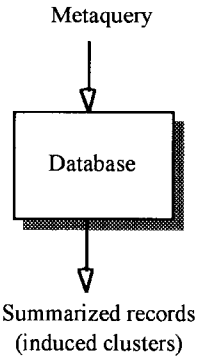


Fig. 12. Context-based clustering as a process of summarization in a database.

It is advantageous to underline the usage of queries of different character and a way in which the results of information retrieval are presented to the end user.

1) A precise query and an enumeration of objects that match precisely this query. The query arises as a statement of the form

   $$X \text{ is } a \text{ and } Y \text{ is } b \text{ or } Z \text{ is } c$$

   where "a," "b," "c," etc. are precise values of some logic predicates. The objects are enumerated in the form of a complete list of pertinent items retrieved from the given database.
2) A linguistic query and an enumeration of objects that match this query to a nonzero level of match. This alternative is often studied in the realm of fuzzy databases with a number of fundamental findings. The query comes in the form

   $$X \text{ is } A \text{ and } Y \text{ is } B \text{ or } Z \text{ is } C$$

   where now A, B, and C are fuzzy sets being the linguistic values of the corresponding predicates. The objects are retrieved and presented as a list of items coming with a nonzero degree of match. In comparison to 1), this approach is more flexible by admitting queries that involve linguistic concepts and accept items tagged by the property articulated in the original query. This membership tagging helps us establish an order at which retrieved items can be ranked.
3) The context-based clustering comes as a direct extension of the previous approach. As before, we

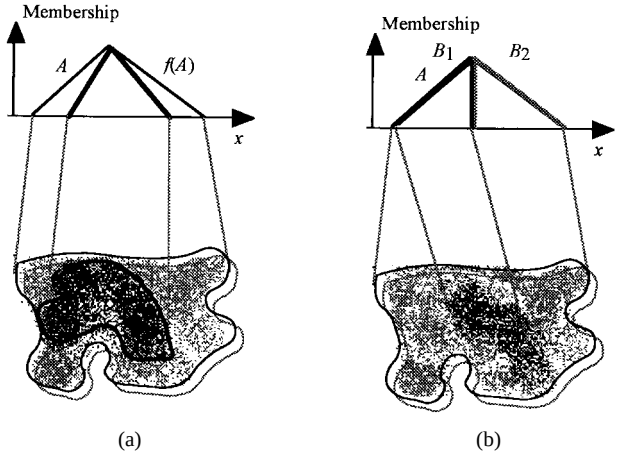


Fig. 13. Two ways of context refinement: (a) through the use of specificity—enhancing increasing linguistic modifier $f(A)$ and (b) through defining a family of linguistic granules subsumed by the original context. The shadowed portion of data are those elements that are activated (filtered) by the corresponding fuzzy set of context.

admit linguistic queries (that are fuzzy contexts). The results of retrieval are provided in a summarized (condensed) form of the linguistic granules generated by the context-based fuzzy clustering.

D. Hierarchical DM Through Context Refinement

Context-based clustering supports hierarchical activities of DM directly. We start with a number of user-defined contexts (information granules) that orient the overall DM pursuit. Once the induced information granules have been generated, the end user has an ability to analyze them. If some of them are too general and not overly meaningful, the previous context that has been used originally can be refined or split into a number of linguistic entities. In the first case (see Fig. 13), we get $f(A)$, where $f(A) \subset A$ and $f(A)$ comes from a family of specificity-increasing linguistic modifiers such as *very*. In the second case, we may refine A and express it as a union of more specific contexts. Thus, $B_1, B_2, \ldots, B_p$ are subsumed in the original context A.

VII. Numerical Studies

In this section, we concentrate on two selected examples and carry out a complete analysis that highlights the key features of the clustering approach to DM. These studies rely on widely available datasets that are already used in many studies in machine learning and DM.

Example 1: The discussed dataset (called auto mpg) comes from the repository of machine learning datasets at University of California, Irvine (see http://ftp.ics.edu/pub/machine-learning-databases/). It consists of a series of car makes (American, European, and Japanese). The makes of the vehicles are characterized by nine features used in their description include fuel consumption (in miles per gallon), a number of cylinders, displacement, horse power, weight, acceleration, model year, and origin (United States, Europe, Japan). The entire dataset includes 392 items, 248 of which are U.S. vehicles, 78 come from Japan, and 66 are European makes. Thus the dataset exhibits a significant diversity that potentially makes our DM pursuits meaningful. A short excerpt from this dataset is shown in Fig. 14. The origin of the vehicles are encoded as follows: 1-United States; 2-Europe; 3-Japan.

The specific goal of DM here is to characterize (describe) classes of vehicles with regard to their economy (fuel consumption). Given a descriptor of fuel efficiency, say medium efficiency, the task reads as

— describe cars of medium efficiency.

Importantly, the notion of fuel efficiency needs to be quantified first. In fact, this quantification (granularization) has to be provided by a user who is interested in his/her particular goals of DM. When talking about the economy of the vehicle, it naturally leads us to accept the first variable (fuel consumption) as the context variable and then complete clustering in the space of the remaining variables (except the names of the cars). The granularity of the context variable is established via trapezoidal fuzzy sets with the membership functions of the form

$$\mathcal{T}(y, -1, 0, 10, 20)$$
$$\mathcal{T}(y, 10, 20, 20, 30)$$
$$\mathcal{T}(y, 20, 30, 30, 40)$$
$$\mathcal{T}(y, 30, 40, 50, 80)$$

where, as usual, the parameters denote the characteristic points of the piecewise membership functions of these fuzzy sets (see Fig. 15).

When the two intermediate parameters are the same, the result is a triangular fuzzy set.

The first one, $\mathcal{T}(y, -1, 0, 10, 20)$, can be regarded as a descriptor of vehicles of low efficiency while the last one [namely, $\mathcal{T}(y, 30, 40, 50, 80)$] characterizes vehicles of high fuel economy. The two intermediate categories characterized by $\mathcal{T}(y, 10, 20, 20, 30)$ and $\mathcal{T}(y, 20, 30, 30, 40)$ treat vehicles of medium fuel consumption. These linguistic fuzzy labels have been used to capture the meaning of the vehicles of some specific and meaningful nature. If necessary, these linguistic labels could be easily revised and modified according to the interest of the user as well as the detailed analysis of the previously obtained results. We should stress that the labels have not been optimized to meet some criteria discussed before (as, for instance, the equalization one). To illustrate that, the histogram of the context variable is shown in Fig. 16.

The calculations reveal the values of the sigma count of the respective fuzzy labels as outlined in Table 3. Thus, it becomes apparent (as expected by eyeballing the histogram) that some linguistic terms (the second and third) are quite dominant.

The clustering is carried out for five clusters per context so, finally, we end up with 20 different associations between

29.0	4	85.00	52.00	2035	22.20	76	1	chevrolet chevette
24.5	4	98.00	60.00	2164	22.10	76	1	chevrolet woody
29.0	4	90.00	70.00	1937	14.20	76	2	vw rabbit
33.0	4	91.00	53.00	1795	17.40	76	3	honda civic
20.0	6	225.00	100.00	3651	17.70	76	1	dodge aspen se
18.0	6	250.00	78.00	3574	21.00	76	1	ford granada ghia
18.5	6	250.00	110.00	3645	16.20	76	1	pontiac ventura sj
17.5	6	258.00	95.00	3193	17.80	76	1	amc pacer d/l
29.5	4	97.00	71.00	1825	12.20	76	2	volkswagen rabbit
32.0	4	85.00	70.00	1990	17.00	76	3	datsun b-210
28.0	4	97.00	75.00	2155	16.40	76	3	toyota corolla
26.5	4	140.00	72.00	2565	13.60	76	1	ford pinto
20.0	4	130.00	102.00	3150	15.70	76	2	volvo 245
13.0	8	318.00	150.00	3940	13.20	76	1	plymouth volare premier
19.0	4	120.00	88.00	3270	21.90	76	2	peugeot 504
19.0	6	156.00	108.00	2930	15.50	76	3	toyota mark ii
16.5	6	168.00	120.00	3820	16.70	76	2	mercedes-benz 280s

Fig. 14. A excerpt from auto mpg dataset. The columns of the table denotefuel consumption, number of cylinders, displacement, horse power, weight, acceleration, model year, and origin (United States-1, Europe-2, Japan-3).

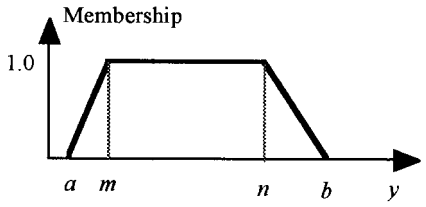


Fig. 15. A class of trapezoidal fuzzy sets (fuzzy numbers) $T(y, a, m, n, b)$.

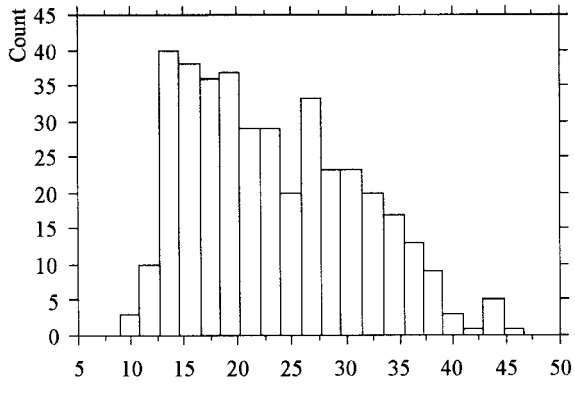


Fig. 16. Distribution of the values of the context variable (mpg).

Table 3

context	σ-count
$context_1$	64.83
$context_2$	168.04
$context_3$	118.98
$context_4$	38.15

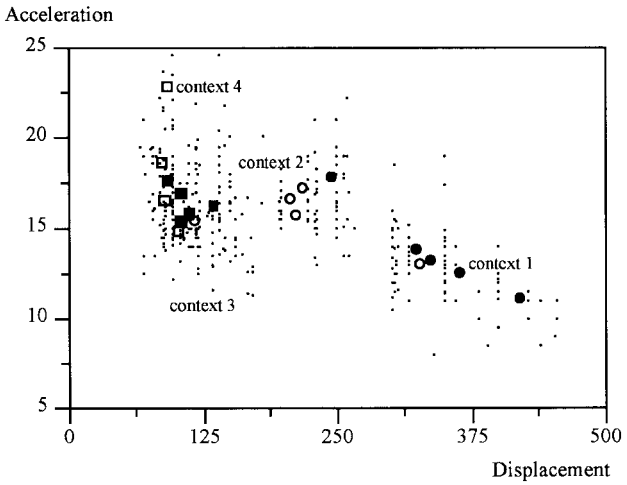


Fig. 17. Prototypes generated by the linguistic contexts identified in the DM problem.

the resulting linguistic granules. The fuzzification parameter (m) was set to two. (This particular value is the most commonly used.) The resulting prototypes are summarized in (1) at the bottom of the next page. Based on their values, one can easily generate the corresponding membership functions of the linguistic terms; each row describes an individual prototype (as we have five prototypes per context). Obviously, some coordinates of the prototypes (such as the number of cylinders) need to be rounded off to the nearest integer.

The plots of the prototypes in a two-dimensional space of acceleration and displacement are illustrated in Fig. 17. There is some overlap between prototypes induced by the successive contexts. This indicates that there is an interaction between these information granules. The overlap is particularly high in case of contexts 3 and 4 that concerns vehicles characterized by rather high fuel efficiency.

But even at this numeric level one can reveal a series of interesting facts, for instance:

1) when it comes to low fuel economy, large and heavy American cars dominate this category;
2) Japanese cars are placed in the fourth category with horsepower in the range of 88–100, four cylinder engines, and a weight of about 2 tons.

By projecting the prototypes on the respective coordinates (variables), the descriptors of the individual classes can be visualized in terms of the corresponding membership functions (see Fig. 18).

We emphasize that the descriptors obtained in this way are condensed and easy to comprehend. They are also user-driven and highly interactive; by changing the contexts the user can conveniently affect a point of view at the data.

The number of clusters required for DM is addressed by looking backward at the context reconstruction. The results are shown in Fig. 19, where the discrete membership functions of the contexts are contrasted with the sum of the membership values of the resulting clusters induced by the method. In general, the reflected linguistic terms tend to overlap to a somewhat higher extent than the original contexts. Moreover, the one-to-one character of the mapping has not been preserved, meaning that for a single membership value of the original context there is a series of the grades of membership resulting from the induced clusters.

Table 4

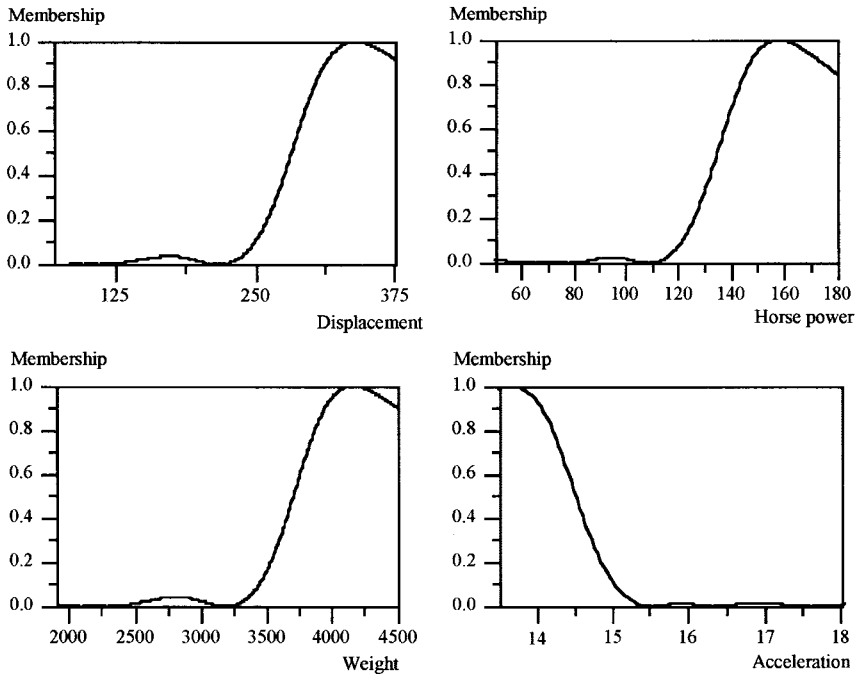
	$class_1$	$class_2$	$class_3$	$class_4$
$cluster_1$	10	8	0	0
$cluster_2$	10	1	0	0
$cluster_3$	17	5	0	0
$cluster_4$	21	1	0	0
$cluster_5$	4	13	0	0

Table 5

	$class_1$	$class_2$	$class_3$	$class_4$
$cluster_1$	0	24	7	0
$cluster_2$	7	20	0	0
$cluster_3$	0	10	2	0
$cluster_4$	0	25	1	0
$cluster_5$	0	25	3	1

Finally, one can examine the relevance of the links between the contexts and induced fuzzy sets. The relevance is quantified in the form of the confusion matrices (Tables 4–7) that summarize the results of associations between the contexts and induced clusters. We use the maximum of membership criterion categorizing the links based on the highest values of the membership grades. The columns of the confusion matrices correspond with the individual contexts while the rows describe the clusters induced by the corresponding matrices. Note that in several cases we have encountered some misclassified data points. This, however, is unavoidable because the created categories naturally overlap.

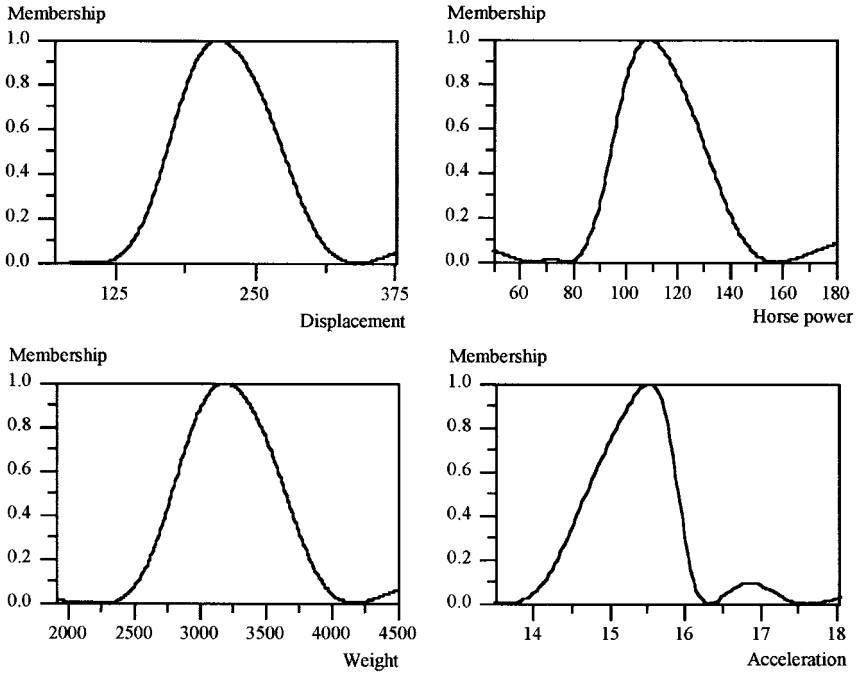
$$
\begin{array}{lllllll}
\text{context}_1 & & & & & & \\
\text{no. cyl.} & \text{displ.} & \text{horse power} & \text{weight} & \text{acceler} & \text{model yr.} & \text{origin} \\
7.9597 & 324.0681 & 147.4904 & 4141.4565 & 13.8119 & 74.7960 & 1.0077 \\
7.9935 & 421.9014 & 206.9132 & 4576.3774 & 11.0116 & 71.7837 & 1.0015 \\
7.9754 & 338.9013 & 156.0435 & 4179.1138 & 13.1936 & 72.4731 & 1.0045 \\
7.9851 & 364.9198 & 175.4141 & 4402.2515 & 12.4981 & 72.0487 & 1.0031 \\
6.0845 & 246.5758 & 101.6843 & 3509.3506 & 17.7591 & 74.4538 & 1.0451 \\
\text{context}_2 & & & & & & \\
4.0874 & 117.2117 & 95.2903 & 2561.4968 & 15.3389 & 74.0668 & 2.3533 \\
7.9295 & 328.2322 & 149.5903 & 3978.1433 & 12.9293 & 75.3593 & 1.0127 \\
5.6790 & 207.2347 & 99.2230 & 3134.1489 & 16.5716 & 75.9661 & 1.1594 \\
5.7946 & 212.4101 & 98.3907 & 2967.3010 & 15.6952 & 72.7827 & 1.0960 \\
5.9282 & 219.8767 & 99.9425 & 3295.7715 & 17.1119 & 77.6124 & 1.0705 \\
\text{context}_3 & & & & & & \\
4.1583 & 113.9076 & 78.9360 & 2294.1370 & 15.7431 & 76.9066 & 1.5600 \\
4.1166 & 135.1387 & 85.3163 & 2584.2524 & 16.1681 & 80.4332 & 1.0853 \\
4.0465 & 93.6141 & 72.0283 & 2087.6772 & 17.5755 & 73.4383 & 2.4799 \\
4.0475 & 105.1029 & 73.0323 & 2239.2920 & 16.8227 & 79.4058 & 2.8950 \\
4.0955 & 105.7471 & 78.8554 & 2214.5576 & 15.3149 & 76.0530 & 1.9204 \\
\text{context}_4 & & & & & & \\
4.0219 & 90.2672 & 65.1666 & 1979.7280 & 16.5085 & 80.6769 & 2.9150 \\
4.0585 & 105.6941 & 69.6008 & 2095.6660 & 15.2831 & 80.6091 & 1.0754 \\
4.0386 & 103.6010 & 75.0548 & 2111.0115 & 14.7648 & 80.5556 & 2.7140 \\
4.0186 & 88.0564 & 65.0256 & 2063.3794 & 18.5730 & 79.6490 & 2.9457 \\
4.0252 & 92.8853 & 49.6662 & 2170.7195 & 22.7944 & 80.0778 & 1.9977
\end{array}
\tag{1}
$$



year : 1972, 1975

origin: USA

(a)

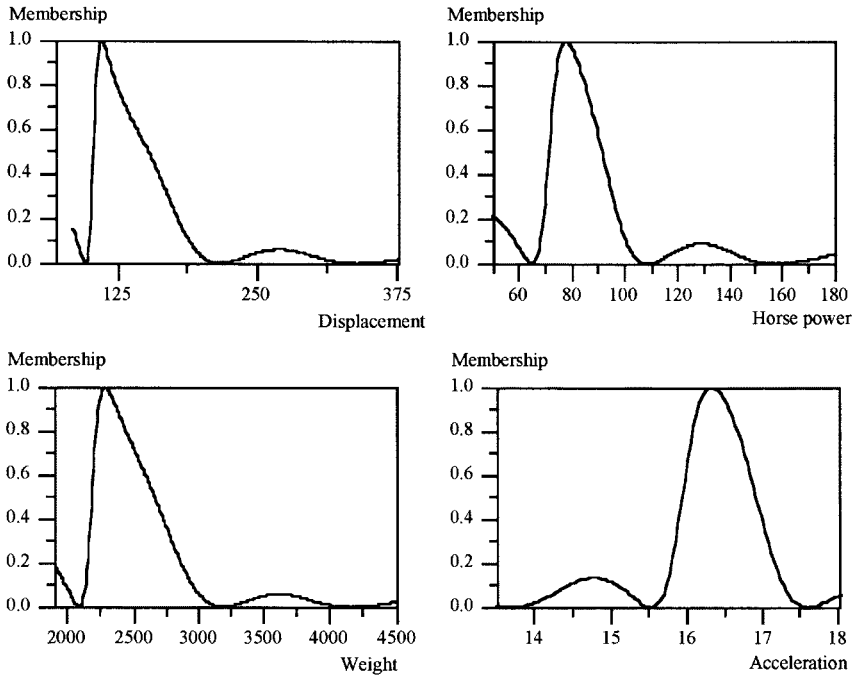


year : 1973, 1974, 1975, 1976, 1977

origin: USA, Europe

(b)

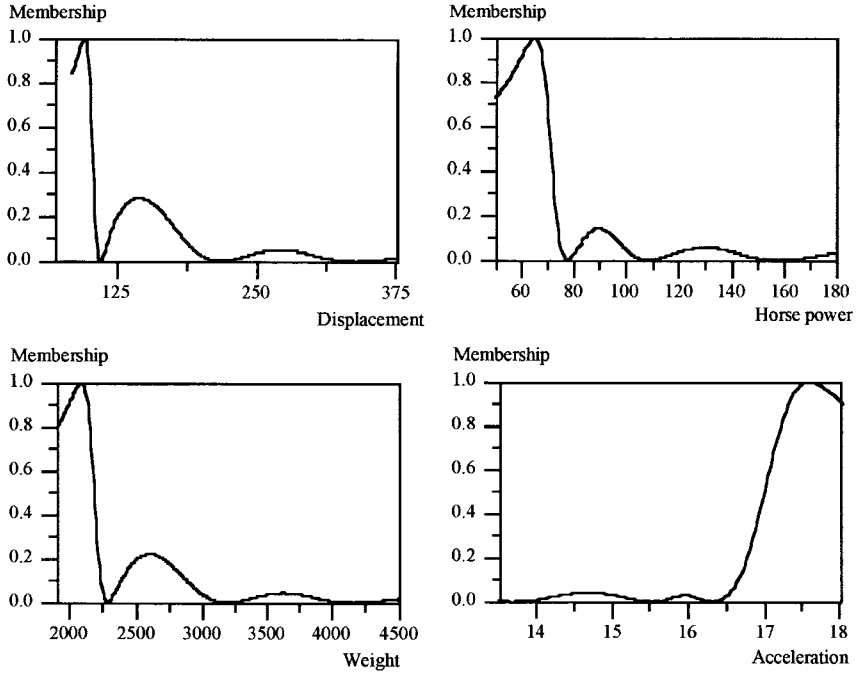
Fig. 18. Linguistic descriptors of the vehicles of various fuel efficiency (linguistic contexts): (a) context_1 and (b) context_2. As two variables assume discrete values (origin of the vehicle and its year), these are indicated at the bottom of each descriptor. For illustrative purposes, the prototypes within the same context are averaged.



year : 1976, 1977, 1980, 1973

origin: USA, Europe, Japan

(c)



year : 1980, 1981

origin: Japan, Europe, USA

(d)

Fig. 18. *(Continued.)* Linguistic descriptors of the vehicles of various fuel efficiency (linguistic contexts): (c) $context_3$ and (d) $context_4$. As two variables assume discrete values (origin of the vehicle and its year), these are indicated at the bottom of each descriptor. For illustrative purposes, the prototypes within the same context are averaged.

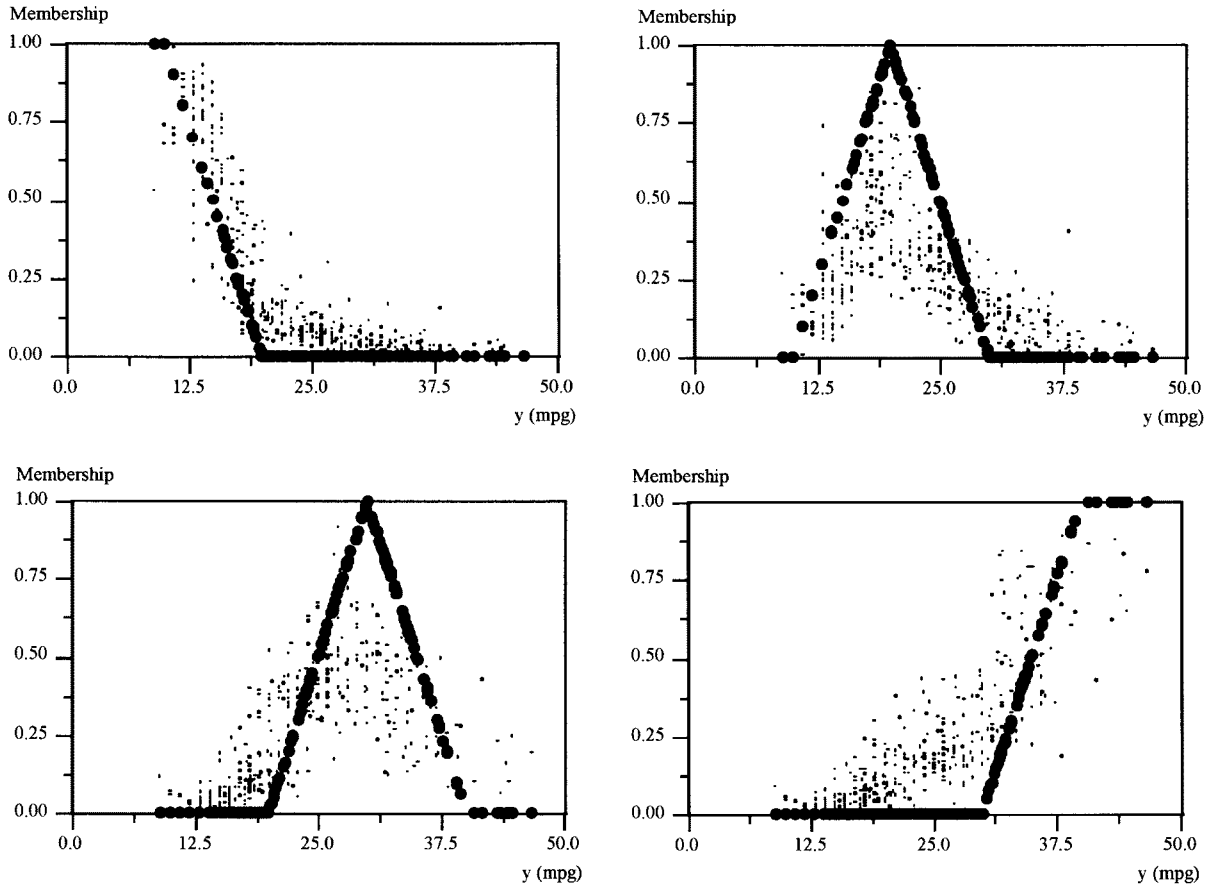


Fig. 19. Original contexts and reflected membership values that result from the induced clusters; small dots represent experimental data while the original membership functions are marked by large dots.

Table 6

	$class_1$	$class_2$	$class_3$	$class_4$
$cluster_1$	0	12	10	0
$cluster_2$	0	6	19	0
$cluster_3$	0	5	19	1
$cluster_4$	0	0	9	1
$cluster_5$	0	5	14	1

Table 7

	$class_1$	$class_2$	$class_3$	$class_4$
$cluster_1$	0	0	5	5
$cluster_2$	0	0	9	6
$cluster_3$	0	4	12	7
$cluster_4$	0	0	6	6
$cluster_5$	0	3	5	5

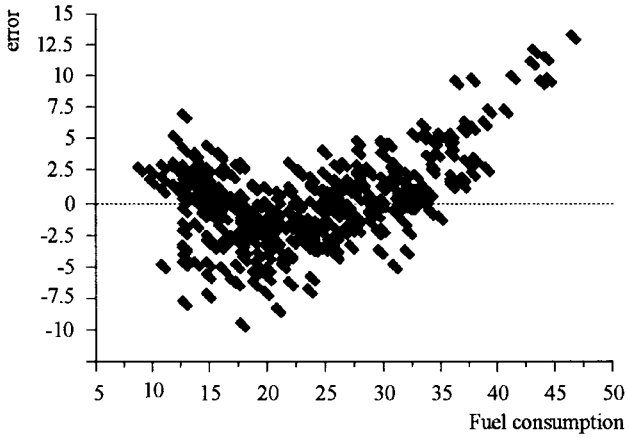


Fig. 20. Error of the regression model; illustrated are errors (differences between the model and data) shown *vis a vis* dependent variable.

Finally, it is instructive to contrast this granular approach with some well-known and standard techniques such as regression models. The models of this form associate (relate) independent variables and a dependent variable in the form of a linear multivariable relationship. The parameters of the regression model are derived with the use of the standard minimum square error procedure. The regression line is a compact representation of data (more precisely, their approximation). Regression models come with inevitable approximation errors (see Fig. 20). The errors strongly depend upon the nonlinear character of the dataset. The more nonlinear the data are, the more profound the approximation error when using a linear model. This becomes visible in Fig. 20, where the linear regression model fails to approximate fuel consumption of vehicles of high fuel efficiency.

It is worth underlining that the regression model lacks transparency: the only representation we arrive at is a vector of the parameters of the model. They are difficult to interpret and visualize especially when dealing with highly dimensional data.

Example 2: This example concerns the performance of various models of computers. The data describe various makes of computers by using some basic hardware characteristics and summarize their performance through a single numeric index. The features of the patterns used therein are indicated as follows:

MYCT	machine cycle time in nanoseconds;
MMIN	minimum main memory in kilobytes;
MMAX	maximum main memory in kilobytes;
CACHE	cache memory in kilobytes;
CHMIN	minimum channels in units;
CHMAX	maximum channels in units;
PERF	relative performance.

To illustrate a variety of the computers under study, an excerpt of this dataset is shown in the following:

amdahl,470v/7,29,8000,32000,32,8,32,269
amdahl,470v/7a,29,8000,32000,32,8,32,220
amdahl,470v/7b,29,8000,32000,32,8,32,172
amdahl,470v/7c,29,8000,16000,32,8,16,132
amdahl,470v/b,26,8000,32000,64,8,32,318
amdahl,580-5840,23,16000,32000,64,16,32,367
...
sperry,80/6,180,512,4000,0,1,3,21,
sperry,80/8,124,1000,8000,0,1,8,42
sperry,90/80-model-3,98,1000,8000,32,2,8,46
sratus,32,125,2000,8000,0,2,14,52
wang,vs-100,480,512,8000,32,0,0,67.

The first two columns of the dataset identify the make of the computer (say, amdahl) along with its specific type (e.g., 470v/7). For instance, the first computer is characterized by the value of MYCT equal to 29, MMIN of 8000, MMAX of 32000, etc. We complete the context-based clustering by defining contexts in the space of the relative performance. This allows us to discriminate between several linguistic categories of the computers with respect to their performance and characterize such categories of machines. We distinguish four classes (contexts) of the performance and describe them by trapezoidal or triangular membership functions. We start with the computers of low performance, sweep through the machines of medium performance, and end up with the computers of high performance. More specifically, the corresponding membership functions capturing such categories are defined as

low performance $\mathcal{T}(x, 0, 0, 10, 20)$
$\mathcal{T}(x, 10, 20, 150, 250)$
$\mathcal{T}(x, 150, 250, 400, 500)$
high performance T(x, 400, 500, 2000, 2100).

The experiments are carried out for three clusters per each context. As in the first experiment, the fuzzification factor is equal to two. First, we list the results by showing the prototypes of the individual contexts (note that we deal with a six-dimensional space of the parameters of the computers):

$\mathcal{T}(x, 0, 0, 10, 20)$
prototype 1
223.43 555.56 1829.82 2.35 1.07 3.92
prototype 2
844.72 544.81 3624.26 0.16 0.99 2.64
prototype 3
1213.89 645.31 1491.56 0.00 0.70 0.72
$\mathcal{T}(x, 10, 20, 150, 250)$
prototype 1
131.08 1634.28 7510.06 10.62 2.43 11.58
prototype 2
396.53 990.99 5571.41 5.93 1.80 9.35
prototype 3
88.20 3491.64 14 788.85 34.78 5.24 19.96
$\mathcal{T}(x, 150, 250, 400, 500)$
prototype 1
50.31 3071.15 31 762.39 112.58 50.99 102.67
prototype 2
39.06 2377.62 9471.44 126.23 11.04 29.85
prototype 3
34.65 8113.07 30 338.65 52.14 9.57 25.94
$\mathcal{T}(x, 400, 500, 2000, 2100)$
prototype 1
29.94 8160.12 63 610.66 113.23 12.08 173.02
prototype 2
28.95 15 996.53 36 633.18 105.92 15.15 29.27
prototype 3
23.36 30 565.78 62 270.66 132.00 30.56 60.71.

The resulting linguistic labels in the space machine cycle and maximum main memory associated with the computers of low and high performance are shown in Fig. 21. The distribution of the data in the two-dimensional space is illustrated in Fig. 22. It becomes obvious that the computers described as those of high performance exhibit quite centralized distribution of linguistic terms characterizing a length of machine cycle (all three clusters overlap quite substantially and are located in the range not exceeding 35 ns). An opposite effect is visible for the size of the memory; here high performance computers come with memories starting from 30 000 kb.

VIII. Conclusions

Making sense of data by searching for stable, meaningful, easily interpretable patterns is a genuine challenge that confronts all DM techniques. While DM techniques may originate from different schools of thought and at the same time may adhere to some general methodological avenues, such techniques need to address seriously the requirements stemming from the main requirement of DM. As revealed

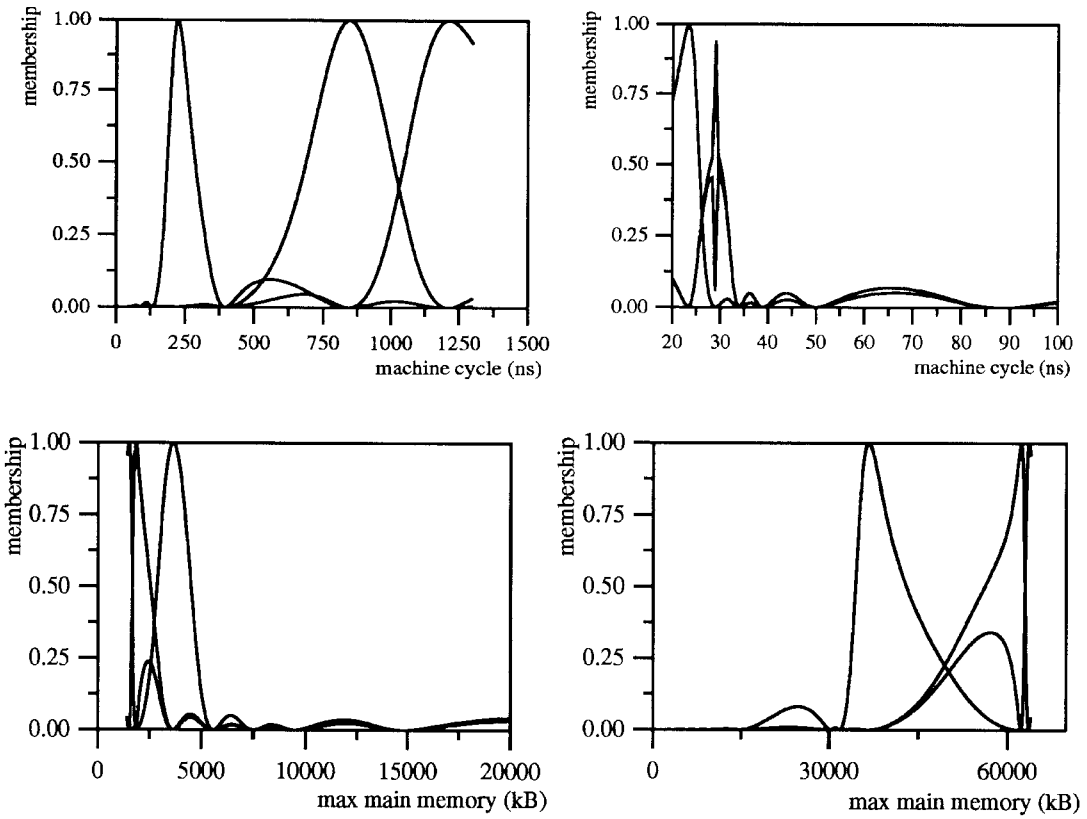


Fig. 21. Linguistic terms associated with the computers of low and high performance.

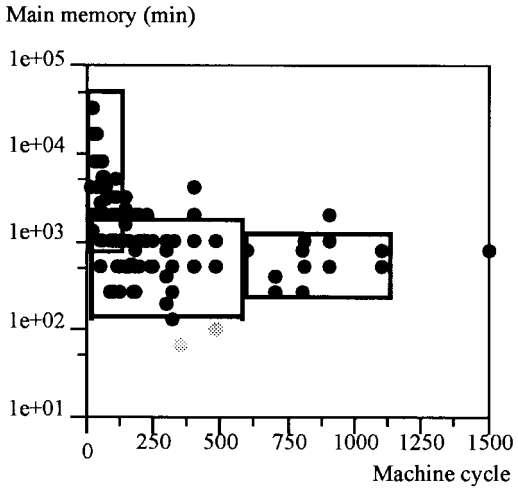


Fig. 22. Clusters of computers visualized in the space of machine cycle—minimum main memory; observe an overlap between the categories of the machines. The groups found in the data are marked by overlapping boxes.

by the study, information granulation helps cope with a mass of detailed data encountered in databases. This study has also emphasized and exemplified the role of granular computing as one of the cornerstones of DM that realizes a quest for patterns that are transparent to the end user. Fuzzy sets appear to be one of the attractive alternatives in this regard: they focus on representing and modeling concepts with gradual boundaries (linguistic terms) that easily appeal to the end-user as well as result in a robust computing environment. We have discussed the underlying principles in more detail by analyzing and quantifying the notions of information granularity as well as introducing some associated ideas of information generality and specificity. We have studied the ideas of unsupervised learning enriched by domain knowledge conveyed in terms of linguistic contexts that help focus on revealing the most essential relationships within the datasets. The resulting context-based clustering not only becomes a useful DM tool but computationally is far more efficient than the standard tools of fuzzy clustering. This efficiency comes with the modularization effect being introduced by the use of the linguistic contexts. The experimental studies using widely accessible datasets highly justify the use of fuzzy sets as a suitable information granulation vehicle supporting DM.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for helpful comments. They also extend their thanks to D. B. Fogel for the constructive suggestions and discussions from which they greatly benefited.

REFERENCES

[1] M. R. Anderberg, *Cluster Analysis for Applications*. New York: Academic, 1973.

[2] E. Backer, *Computer-Assisted Reasoning in Cluster Analysis*. New York: Prentice-Hall, 1995.

[3] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum, 1981.

[4] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Belmont, CA: Wadsworth, 1984.

[5] C. Brunk, J. Kelly, and R. Kohavi, "MineSet: An integrated system for data mining," in *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining*, Newport Beach, CA, Aug. 14–17, 1997. Menlo Park, CA: AAAI Press, pp. 135–138.

[6] J. Chattratichat, "Large scale data mining: challenges and responses," in *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining*, Newport Beach, CA, Aug. 14–17, 1997. Menlo Park, CA: AAAI Press, pp. 143–146.

[7] *Commun. ACM (Special Issue on Data Mining)*, p. 11, 1996.

[8] R. Dave, "Characterization and detection of noise in clustering," *Pattern Recognition Lett.*, vol. 12, pp. 657–664, 1992.

[9] M. Derthick, J. Kolojejchick, and S. F. Roth, "An interactive visualization environment for data exploration," in *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining*, Newport Beach, CA, Aug. 14–17, 1997, pp. 2–9.

[10] B. S. Everitt, *Cluster Analysis*. Berlin, Germany: Heinemann, 1974.

[11] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Mag.*, vol. 17, pp. 37–54, 1996.

[12] ____,"The KDD process for extracting useful knowledge from volumes of data," *Commun. ACM*, vol. 39, pp. 27–41, 1996.

[13] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds., *Advances in Knowledge Discovery and Data Mining*. Menlo Park, CA: AAAI Press, 1996.

[14] D. H. Fisher, "Knowledge acquisition via incremental learning," *Machine Learning*, vol. 2, pp. 139–172, 1987.

[15] W. Frawley, G. Piatetsky-Shapiro, and C. Matheus, "Knowledge discovery in databases: An overview," in *Knowledge Discovery in Databases*, G. Piatetsky-Shapiro and W. Frawley, Eds. Menlo Park, CA: AAAI Press, 1991, pp. 1–7.

[16] J. A. Hartigan, *Clustering Algorithms*. New York: Wiley, 1975.

[17] K. Hirota, *Industrial Applications of Fuzzy Technology*. Berlin, Germany: Springer Verlag, 1993.

[18] ____,*Industrial Applications of Fuzzy Technology in the World*. Singapore: World Scientific, 1995.

[19] P. J. Huber, "From large to huge: A statisticians reaction to KDD and DM," in *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining*, Newport Beach, CA, Aug. 14–17, 1997. Menlo Park, CA: AAAI Press, pp. 304–308.

[20] *Int. J. Intell. Syst. (Special Issue on Knowledge Discovery in Data- and Knowledge Bases)*, vol. 7, no. 7, 1992.

[21] A. K. Jain and Dubes, *Algorithms for Clustering Data*. New York: Wiley, 1988.

[22] A. Kandel, *Fuzzy Mathematical Techniques with Applications*. Menlo Park, CA: Addison-Wesley, 1986.

[23] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data*. New York: Wiley, 1990.

[24] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty and Information*. Englewood Cliffs, NJ: Prentice-Hall, 1988.

[25] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 98–110, 1993.

[26] G. Matheron, *Random Sets and Integral Geometry*. New York: Wiley, 1975.

[27] G. A. Miller, "The magical number seven plus or minus two: Some limits on our capacity for processing information," *Psychol. Rev.*, vol. 63, pp. 81–97, 1956.

[28] R. E. Moore, *Interval Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1966.

[29] G. Nakhaezadeh and A. Schnabl, "Development of multi-criteria metrics for evaluation of data mining algorithms," in *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining*, Newport Beach, CA, Aug. 14–17, 1997. Menlo Park, CA: AAAI Press, pp. 37–42.

[30] A. Papantonakis and P. J. H. King, "Syntax and semantics of GQL, a graphical query language," *J. Visual Languages Comput.*, vol. 6, pp. 3–25, 1995.

[31] Z. Pawlak, "Rough sets," *Int. J. Comput. Inform. Sci.*, vol. 11, pp. 341–356, 1982.

[32] W. Pedrycz, "Fuzzy sets framework for development of perception perspective," *Fuzzy Sets Syst.*, vol. 37, pp. 123–137, 1990.

[33] ____, "Selected issues of frame of knowledge representation realized by means of linguistic labels," *Int. J. Intell. Syst.*, vol. 7, pp. 155–170, 1992.

[34] ____, *Fuzzy Sets Engineering*. Boca Raton, FL: CRC Press, 1995.

[35] ____, *Conditional Fuzzy C—Means, Pattern Recognition Letters*, vol. 17, no. 3, pp. 625–632, Mar. 1996.

[36] ____, *Computational Intelligence: An Introduction*. Boca Raton, FL: CRC Press, 1997.

[37] ____, "Conditional fuzzy clustering in the design of radial basis function neural networks," *IEEE Trans. Neural Networks*, vol. 9, pp. 601–612, July 1998.

[38] W. Pedrycz and F. Gomide, *An Introduction to Fuzzy Sets: Analysis and Design*. Cambridge, MA: MIT Press, 1998.

[39] W. Pedrycz and J. V. de Oliveira, "Optimization of fuzzy relational models," in *Proc. 5th IFSA World Congr.*, vol. 2, Seoul, South Korea, 1993, pp. 1187–1190.

[40] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, Jan. 1986.

[41] ____, "Simplifying decision trees," *Int. J. Man-Machine Studies*, vol. 27, pp. 221–234, 1987.

[42] ____, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1993.

[43] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic, 1982.

[44] Y. Shahar, "A framework for knowledge-based temporal abstraction," *Artificial Intell.*, vol. 90, pp. 79–133, 1997.

[45] A. Silberschatz and A. Tuzhilin, "On subjective measures of interestingness in knowledge discovery," in *Proc. 1st Int. Conf. Knowledge Discovery and Data Mining*. Menlo Park, CA: AAAI Press, 1995, pp. 275–281.

[46] R. Srikant, Q. Vu, and R. Agrawal, "Mining association rules with item constraints," in *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining*, Newport Beach, CA, Aug. 14–17, 1997. Menlo Park, CA: AAAI Press, pp. 67–73.

[47] S. Stoflo, A. L. Prodromidis, S. Tselepis, W. Lee, D. W. Fan, and P. K. Chan, "JAM: Java agents for meta-learning over distributed databases," in *Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining*, Newport Beach, CA, Aug. 14–17, 1997. Menlo Park, CA: AAAI Press, pp. 74–77.

[48] H. Toivonen, "Sampling large databases for association rules," in *Proc. 22nd Int. Conf. Very Large Databases*, 1996, pp. 134–145.

[49] R. R. Yager, "Measuring tranquility and anxiety in decision making: An application of fuzzy sets," *Int. J. Gen. Syst.*, vol. 8, pp. 139–146, 1982.

[50] ____, "Entropy and specificity in a mathematical theory of evidence," *Int. J. Gen. Syst.*, vol. 9, pp. 249–260, 1983.

[51] K. Yoda, T. Fukuda, and Y. Morimoto, "Computing optimized rectilinear regions for association rules," in *Proc. 3rd Int. Conf. Knowledge Discovery and Data Mining*, Newport Beach, CA, Aug. 14–17, 1997. Menlo Park, CA: AAAI Press, pp. 96–103.

[52] Y. Wang and A. K. C. Wong, "Representing discovered patterns using attributed hypergraph," in *Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining*, Portland, OR, Aug. 2–4, 1996. Menlo Park, CA: AAAI Press, pp. 283–286.

[53] L. A. Zadeh, *Fuzzy sets, Inform. Control*, vol. 8, pp. 338–353, 1965.

[54] ____, "The concept of a linguistic variable and its application to approximate reasoning," *Inform. Sci.*, vol. 8, pp. 199–249, 1987.

[55] ____, "Fuzzy sets and information granularity," in *Advances in Fuzzy Set Theory and Applications*, M. M. Gupta, R. K. Ragade, and R. R. Yager, Eds. Amsterdam, The Netherlands: North Holland, 1979, pp. 3–18.

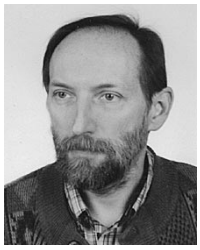
[56] N. Zhong and S. Ohsuga, "Toward a multi-strategy and cooperative discovery system," in *Proc. 1st Int. Conf. Knowledge Discovery and Data Minining*. Menlo Park, CA: AAAI Press, 1995, pp. 337–342.

[57] J. Zytkow, "Automated discovery of empirical laws," *Fundamenta Informaticae*, vol. 27, pp. 299–318, 1996.

Kaoru Hirota (Member, IEEE) was born in Japan on January 6, 1950. He received the B.E., M.E., and Dr. E. degrees in electronics from Tokyo Institute of Technology, Tokyo, Japan, in 1974, 1976, and 1979, respectively.

From 1979 to 1982, he was with the Sagami Institute of Technology, Fujisawa, Japan. From 1982 to 1985, he was with the College of Engineering, Hosei University, Tokyo, Japan. Since 1995, he has been with the Interdisciplinary Graduate School of Science and Technology, Tokyo Institute of Technology, Yokohama, Japan. He is now a Department Head Professor of the Department of Computational Intelligence and Systems Science. His research interests include fuzzy systems, intelligent robot, image understanding, expert systems, hardware implementation, and multimedia intelligent communication.

Dr. Hirota was an Associate Editor of IEEE TRANSACTIONS ON FUZZY SYSTEMS from 1993 to 1995 and of IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS from 1996 to the present. He is a Senior Associate Editor of the *International Journal of Information Sciences Applications* and Editor-in-Chief of the *International Journal of Advanced Computational Intelligence*. He is a member of the International Fuzzy Systems Association (IFSA) and served as its Vice President from 1991 to 1993 and as Treasurer from 1997 to the present. He is also a member of the Japan Society for Fuzzy Theory and Systems (SOFT) and served as its Vice President from 1995 to 1997.



Witold Pedrycz (Fellow, IEEE) is Professor and Director of Computer Engineering and Software Engineering in the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Canada. He is actively pursuing research in computational intelligence, fuzzy modeling, knowledge discovery and data mining, fuzzy control, including fuzzy controllers, pattern recognition, knowledge-based neural networks, and relational computation. He has published numerous papers in the area of applied fuzzy sets as well research monographs: *Fuzzy Control and Fuzzy Systems* (Research Study Press, 1988 and Wiley, 1993); *Fuzzy Relation Equations and Their Applications to Knowledge Engineering* (Kluwer, 1988); *Fuzzy Sets Engineering* (CRC Press, 1995); *Computational Intelligence: An Introduction* (CRC Press, 1997); *Fuzzy Sets: Analysis and Design* (MIT Press, 1998); and *Data Mining Techniques* (Kluwer, 1998). He is also one of the Editors-in-Chief of the *Handbook of Fuzzy Computation* (Oxford/Inst. Phys., 1998).

Dr. Pedrycz is a member of many program committees of international conferences and has served on editorial boards of journals on fuzzy set technology and neurocomputing (IEEE TRANSACTIONS ON FUZZY SYSTEMS, IEEE TRANSACTIONS ON SYSTEMS MAN AND CYBERNETICS, *Fuzzy Sets and Systems*), soft computing (*Soft Computing Research Journal*), intelligent Manufacturing (*Journal of Intelligent Manufacturing*), and pattern recognition (*Pattern Recognition Letters*).