Data Mining in Soft Computing Framework: A Survey

Sushmita Mitra, *Senior Member, IEEE*, Sankar K. Pal, *Fellow, IEEE*, and Pabitra Mitra

Abstract—**The present article provides a survey of the available literature on data mining using soft computing. A categorization has been provided based on the different soft computing tools and their hybridizations used, the data mining function implemented, and the preference criterion selected by the model. The utility of the different soft computing methodologies is highlighted. Generally fuzzy sets are suitable for handling the issues related to understandability of patterns, incomplete/noisy data, mixed media information and human interaction, and can provide approximate solutions faster. Neural networks are nonparametric, robust, and exhibit good learning and generalization capabilities in data-rich environments. Genetic algorithms provide efficient search algorithms to select a model, from mixed media data, based on some preference criterion/objective function. Rough sets are suitable for handling different types of uncertainty in data. Some challenges to data mining and the application of soft computing methodologies are indicated. An extensive bibliography is also included.**

Index Terms—**Fuzzy logic, genetic algorithms, knowledge discovery, neural networks, neuro-fuzzy computing, rough sets, rule extraction.**

I. Introduction

THE digital revolution has made digitized information easy to capture and fairly inexpensive to store [1], [2]. With the development of computer hardware and software and the rapid computerization of business, huge amount of data have been collected and stored in databases. The rate at which such data is stored is growing at a phenomenal rate. As a result, traditional *ad hoc* mixtures of statistical techniques and data management tools are no longer adequate for analyzing this vast collection of data. Several domains where large volumes of data are stored in centralized or distributed databases include the following.

- Financial Investment: Stock indexes and prices, interest rates, credit card data, fraud detection [3].
- Health Care: Several diagnostic information stored by hospital management systems [4].
- Manufacturing and Production: Process optimization and trouble shooting [5].
- Telecommunication network: Calling patterns and fault management systems.
- Scientific Domain: Astronomical observations [6], genomic data, biological data.
- The World Wide Web [7].

Raw data is rarely of direct benefit. Its true value is predicated on the ability to extract information useful for decision support or exploration, and understanding the phenomenon governing the data source. In most domains, data analysis was traditionally a manual process. One or more analysts would become intimately familiar with the data and, with the help of statistical techniques, provide summaries and generate reports. In effect, the analyst acted as a sophisticated query processor. However, such an approach rapidly breaks down as the size of data grows and the number of dimensions increases. Databases containing number of data in the order 10^9 and dimension in the order of 10^3 are becoming increasingly common. When the scale of data manipulation, exploration and inferencing goes beyond human capacities, people look to computing technologies for automating the process.

All these have prompted the need for intelligent data analysis methodologies, which could discover useful knowledge from data. The term KDD refers to the overall process of *knowledge discovery in databases*. *Data mining* is a particular step in this process, involving the application of specific algorithms for extracting patterns (models) from data. The additional steps in the KDD process, such as data preparation, data selection, data cleaning, incorporation of appropriate prior knowledge, and proper interpretation of the results of mining, ensures that useful knowledge is derived from the data.

The subject of KDD has evolved, and continues to evolve, from the intersection of research from such fields as databases, machine learning, pattern recognition, statistics, artificial intelligence, reasoning with uncertainties, knowledge acquisition for expert systems, data visualization, machine discovery, and high-performance computing. KDD systems incorporate theories, algorithms, and methods from all these fields. Many successful applications have been reported from varied sectors such as marketing, finance, banking, manufacturing, and telecommunications. Database theories and tools provide the necessary infrastructure to store, access and manipulate data. *Data warehousing* [2], a recently popularized term, refers to the current business trends in collecting and cleaning transactional data, and making them available for analysis and decision support. A good overview of KDD can be found in Ref. [8], [9].

Fields concerned with inferring models from data include statistical pattern recognition, applied statistics, machine learning and neural computing. A natural question that arises is: how is KDD different from those fields? KDD focuses on the overall process of knowledge discovery from large volumes of data, including the storage and accessing of such data, scaling of algorithms to massive data sets, interpretation and visualization of results, and the modeling and support of the overall human machine interaction.

Data mining is a form of knowledge discovery essential for solving problems in a specific domain. Individual data sets may be gathered and studied collectively for purposes other than those for which they were originally created. New knowledge may be obtained in the process while eliminating one of the largest costs, *viz.*, data collection. Medical data, for example, often exists in vast quantities in an unstructured format. The application of data mining can facilitate systematic analysis in such cases. Medical data, however, requires a large amount of preprocessing in order to be useful. Here numeric and textual information may be interspersed, different symbols can be used with the same meaning, redundancy often exists in data, erroneous/misspelled medical terms are common, and the data is frequently rather sparse. A robust preprocessing system is required in order to extract any kind of knowledge from even medium-sized medical data sets. The data must not only be cleaned of errors and redundancy, but organized in a fashion that makes sense to the problem.

Soft computing is a consortium of methodologies that works synergistically and provides, in one form or another, flexible information processing capability for handling real-life ambiguous situations [10]. Its aim is to exploit the tolerance for imprecision, uncertainty, approximate reasoning, and partial truth in order to achieve tractability, robustness, and low-cost solutions. The guiding principle is to devise methods of computation that lead to an acceptable solution at low cost by seeking for an approximate solution to an imprecisely/precisely formulated problem [11].

Soft computing methodologies (involving fuzzy sets, neural networks, genetic algorithms, and rough sets) are most widely applied in the data mining step of the overall KDD process. Fuzzy sets provide a natural framework for the process in dealing with uncertainty. Neural networks and rough sets are widely used for classification and rule generation. Genetic algorithms (GAs) are involved in various optimization and search processes, like query optimization and template selection. Other approaches like case based reasoning [5] and decision trees [12], [13] are also widely used to solve data mining problems.

The present article provides an overview of the available literature on data mining, that is scarce, in the soft computing framework. Section II describes the basic notions of knowledge discovery in databases, and data mining. Some challenges are highlighted. This is followed in Section III by a survey explaining the role of the aforesaid soft computing tools and their hybridizations, categorized on the basis of the different data mining functions implemented and the preference criterion selected by the model. The utility and applicability of the different soft computing methodologies is highlighted. Section IV concludes the article. Some challenges to data mining and the application of soft computing methodologies are also indicated.

II. Knowledge Discovery and Data Mining

KDD is defined as *the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data* [8], [14]. *Data* is a set of facts F, and a *pattern* is an expression E in a language L describing the facts in a subset F_E of F. E is called a pattern if it is simpler than the enumeration of all facts in F_E. A measure of certainty, measuring the *validity* of discovered patterns, is a function C mapping expressions in L to a partially or totally ordered measure space M_C. An expression E in L about a subset $F_E \subset F$ can be assigned a certainty measure $c = C(E, F)$. *Novelty* of patterns can be measured by a function $N(E, F)$ with respect to changes in data or knowledge. Patterns should potentially lead to some *useful* actions, as measured by some utility function $u = U(E, F)$ mapping expressions in L to a partially or totally ordered measure space M_U. The goal of KDD is to make patterns *understandable* to humans. This is measured by a function $s = S(E, F)$ mapping expressions E in L to a partially or totally ordered measure space M_S.

Interestingness of a pattern combines validity, novelty, usefulness, and understandability, and can be expressed as $i = I(E, F, C, N, U, S)$ which maps expressions in L to a measure space M_I. A pattern $E \in L$ is called *knowledge* if for some user-specified threshold $i \in M_I$, $I(E, F, C, N, U, S) > i$ [8]. One can select some thresholds $c \in M_C$, $s \in M_S$, and $u \in M_u$, and term a pattern E knowledge

$$\text{iff} \quad C(E, F) > c, \quad \text{and } S(E, F) > s, \quad \text{and } U(E, F) > u. \tag{1}$$

The role of interestingness is to threshold the huge number of discovered patterns and report only those which may be of some use. There are two approaches to designing a measure of interestingness of a pattern, *viz.*, objective and subjective. The former uses the structure of the pattern and is generally used for computing *rule interestingness*. However often it fails to capture all the complexities of the pattern discovery process. The *subjective* approach, on the other hand, depends additionally on the *user* who examines the pattern. Two major reasons why a pattern is interesting from the subjective (user-oriented) point of view are as follows [15].

- *Unexpectedness*: when it is 'surprising' to the user.
- *Actionability*: when the user can act on it to her/his advantage.

Though both these concepts are important is has often been observed that actionability and unexpectedness are correlated. In literature, unexpectedness is often defined in terms of the dissimilarity of a discovered pattern from a vocabulary provided by the user.

As an example, consider a database of student evaluations of different courses offered at some university. This can be defined as EVALUATE (TERM, YEAR, COURSE, SECTION, INSTRUCTOR, INSTRUCT_RATING, COURSE_ RATING). We describe two patterns that are interesting in terms of actionability and unexpectedness, respectively. The pattern that Professor X is consistently getting the overall INSTRUCT_RATING below overall COURSE_RATING can be of interest to the chairperson because this shows that Professor X has room for improvement. If, on the other hand, in most of the course evaluations the overall INSTRUCT_RATING is higher than COURSE_RATING and it turns out that in most of Professor X's rating overall INSTRUCT_RATING is lower than COURSE_RATING, then such a pattern is unexpected and hence interesting.

Data mining is a step in the KDD process consisting of a particular enumeration of patterns E_j over the data, subject to some

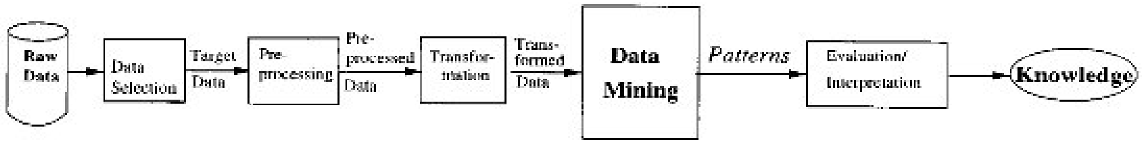

Fig. 1. The KDD process.

computational limitations. It uses *historical* data to discover regularities and improve future decisions [16]. The data can consist of (say) a collection of time series descriptions that can be learned to predict later events in the series.

A. KDD Process

The overall KDD process is outlined in Fig. 1. It is interactive and iterative involving, more or less, the following steps [17].

1) Understanding the application domain: includes relevant prior knowledge and goals of the application.
2) Extracting the target data set: includes selecting a data set or focusing on a subset of variables.
3) Data cleaning and preprocessing: includes basic operations, such as noise removal and handling of missing data. Data from real-world sources are often erroneous, incomplete, and inconsistent, perhaps due to operation error or system implementation flaws. Such low quality data needs to be cleaned prior to data mining.
4) Data integration: includes integrating multiple, heterogeneous data sources.
5) Data reduction and projection: includes finding useful features to represent the data (depending on the goal of the task) and using dimensionality reduction or transformation methods.
6) Choosing the function of data mining: includes deciding the purpose of the model derived by the data mining algorithm (e.g., summarization, classification, regression, clustering, web mining, image retrieval, discovering association rules and functional dependencies, rule extraction, or a combination of these).
7) Choosing the data mining algorithm(s): includes selecting method(s) to be used for searching patterns in data, such as deciding on which model and parameters may be appropriate.
8) Data mining: includes searching for patterns of interest in a particular representational form or a set of such representations.
9) Interpretation: includes interpreting the discovered patterns, as well as the possible visualization of the extracted patterns. One can analyze the patterns automatically or semiautomatically to identify the truly interesting/useful patterns for the user.
10) Using discovered knowledge: includes incorporating this knowledge into the performance system, taking actions based on knowledge.

B. Data Mining

KDD refers to the overall process of turning low-level data into high-level knowledge. An important step in the KDD process is data mining. Data mining is an interdisciplinary field with a general goal of predicting outcomes and uncovering relationships in data. It uses automated tools employing sophisticated algorithms to discover hidden patterns, associations, anomalies and/or structure from large amounts of data stored in data warehouses or other information repositories. Data mining tasks can be descriptive, i.e., discovering interesting patterns describing the data, and predictive, i.e., predicting the behavior of the model based on available data.

Data mining involves fitting models to or determining patterns from observed data. The fitted models play the role of inferred knowledge. Deciding whether the model reflects useful knowledge or not is a part of the overall KDD process for which subjective human judgment is usually required. Typically, a data mining algorithm constitutes some combination of the following three components.

- The model: The function of the model (e.g., classification, clustering) and its representational form (e.g., linear discriminants, neural networks). A model contains parameters that are to be determined from the data.
- The preference criterion: A basis for preference of one model or set of parameters over another, depending on the given data. The criterion is usually some form of goodness-of-fit function of the model to the data, perhaps tempered by a smoothing term to avoid overfitting, or generating a model with too many degrees of freedom to be constrained by the given data.
- The search algorithm: The specification of an algorithm for finding particular models and parameters, given the data, model(s), and a preference criterion.

A particular data mining algorithm is usually an instantiation of the model/preference/search components. The more common model functions in current data mining practice include the following.

1) Classification [18]–[22]: classifies a data item into one of several predefined categorical classes.
2) Regression [8], [23]–[25]: maps a data item to a real-valued prediction variable.
3) Clustering [26]–[33]: maps a data item into one of several clusters, where clusters are natural groupings of data items based on similarity metrics or probability density models.
4) Rule generation [34]–[41]: extracts classification rules from the data.
5) Discovering association rules [42]–[45]: describes association relationship among different attributes.
6) Summarization [46]–[49]: provides a compact description for a subset of data.
7) Dependency modeling [50], [51]: describes significant dependencies among variables.

8) Sequence analysis [52], [53]: models sequential patterns, like time-series analysis. The goal is to model the states of the process generating the sequence or to extract and report deviation and trends over time.

The rapid growth of interest in data mining is due to the 1) falling cost of large storage devices and increasing ease of collecting data over networks; 2) development of robust and efficient machine learning algorithms to process this data; and 3) falling cost of computational power, enabling use of computationally intensive methods for data analysis [16].

The notion of scalability relates to the efficient processing of such large data sets, while generating from them the best possible models. The most commonly cited reason for scaling up is that increasing the size of the training set often increases the accuracy of learned classification models. In many cases, the degradation in accuracy when learning from smaller samples stems from overfitting, presence of noise, and existence of large number of features. Again, scaling up to very large data sets implies that fast learning algorithms must be developed. However, rather than speeding up a slow algorithm, the issue is more of turning an impracticable algorithm into a feasible one. A large number of examples introduces potential problems with both time and space complexity. Finally, the goal of the learning (say, classification accuracy) must not be substantially sacrificed by a scaling algorithm. The three main approaches to scaling up include [54] the following:

- designing a fast algorithm: reducing asymptotic complexity, optimizing the search and representation, finding approximate solutions, or taking advantage of the task's inherent parallelism;
- partitioning the data: dividing the data into subsets (based on instances or features), learning from one or more of the selected subsets, and possibly combining the results;
- using a relational representation: addresses data that cannot feasibly be treated as a single flat file.

The first generation of data mining algorithms has been demonstrated to be of significant value across a variety of real-world applications. But these work best for problems involving a large set of data collected into a single database, where the data is described by numeric or symbolic features. Here the data invariably does not contain text and image features interleaved with these features, and is carefully and cleanly collected with a particular decision-making task in mind.

Development of new generation algorithms is expected to encompass more diverse sources and types of data that will support mixed-initiative data mining, where human experts collaborate with the computer to form hypotheses and test them. The main challenges to the data mining procedure involve the following:

1) *Massive data sets and high dimensionality*. Huge data sets create combinatorially explosive search space for model induction, and increase the chances that a data mining algorithm will find spurious patterns that are not generally valid. Possible solutions include robust and efficient algorithms, sampling approximation methods and parallel processing.
2) *User interaction and prior knowledge*. Data mining is inherently an interactive and iterative process. Users may interact at various stages, and domain knowledge may be used either in the form of a high-level specification of the model, or at a more detailed level. Visualization of the extracted model is also desirable.
3) *Overfitting and assessing the statistical significance*. Data sets used for mining are usually huge and available from distributed sources. As a result, often the presence of spurious data points leads to overfitting of the models. Regularization and resampling methodologies need to be emphasized for model design.
4) *Understandability of patterns*. It is necessary to make the discoveries more understandable to humans. Possible solutions include rule structuring, natural language representation, and the visualization of data and knowledge.
5) *Nonstandard and incomplete data*. The data can be missing and/or noisy.
6) *Mixed media data*. Learning from data that is represented by a combination of various media, like (say) numeric, symbolic, images and text.
7) *Management of changing data and knowledge*. Rapidly changing data, in a database that is modified/deleted/augmented, may make previously discovered patterns invalid. Possible solutions include incremental methods for updating the patterns.
8) *Integration*. Data mining tools are often only a part of the entire decision making system. It is desirable that they integrate smoothly, both with the database and the final decision making procedure.

III. Soft Computing for Data Mining

Recently various soft computing methodologies have been applied to handle the different challenges posed by data mining. The main constituents of soft computing, at this juncture, include fuzzy logic, neural networks, genetic algorithms, and rough sets. Each of them contributes a distinct methodology for addressing problems in its domain. This is done in a cooperative, rather than a competitive, manner. The result is a more intelligent and robust system providing a human-interpretable, low cost, approximate solution, as compared to traditional techniques.

Let us first describe the roles and significance of the individual soft computing tools and their hybridizations, followed by the various systems developed for handling the different functional aspects of data mining. A suitable preference criterion is often optimized during mining. It may be mentioned that there is no universally best data mining method; choosing particular soft computing tool(s) or some combination with traditional methods is entirely dependent on the particular application and requires human interaction to decide on the suitability of an approach.

A. Fuzzy Sets

The modeling of imprecise and qualitative knowledge, as well as the transmission and handling of uncertainty at various stages are possible through the use of fuzzy sets. Fuzzy logic is capable of supporting, to a reasonable extent, human type reasoning in *natural* form. It is the earliest and most widely reported constituent of soft computing. The development of fuzzy logic has led to the emergence of soft computing. In this section we provide a glimpse of the available literature pertaining to the use of fuzzy sets in data mining.

Knowledge discovery in databases is mainly concerned with identifying interesting patterns and describing them in a concise and meaningful manner [8]. Fuzzy models can be said to represent a prudent and user-oriented sifting of data, qualitative observations and calibration of commonsense rules in an attempt to establish meaningful and useful relationships between system variables [55]. Despite a growing versatility of knowledge discovery systems, there is an important component of human interaction that is inherent to any process of knowledge representation, manipulation, and processing. Fuzzy sets are inherently inclined toward coping with linguistic domain knowledge and producing more interpretable solutions.

The notion of *interestingness*, which encompasses several features such as validity, novelty, usefulness, and simplicity, can be quantified through fuzzy sets. Fuzzy dissimilarity of a discovered pattern with a user-defined vocabulary has been used as a measure of this interestingness [56]. As an extension to the above methodology *unexpectedness* can also be defined in terms of a *belief system*, where if a belief b is based on previous evidence ξ then $d(b \mid \xi)$ denotes the degree of belief b. In soft belief systems, a weight w_i is attached to each belief b_i. The degree of a belief may be measured with conditional probability, Dempster-Shafer belief function or frequency of the raw data. Here, the interestingness of a pattern E relative to a belief system B and evidence ξ may be formally defined as

$$I(E, B, \xi) = \sum_{b_i \in B} w_i |d(b_i \mid E, \xi) - d(b_i \mid \xi)|. \qquad (2)$$

This definition of interestingness measures the amount by which the degrees of belief change as a result of a new pattern E.

There is a growing indisputable role of fuzzy set technology in the realm of data mining [57]. Various data browsers have been implemented using fuzzy set theory [58]. Analysis of real-world data in data mining often necessitates simultaneous dealing with different types of variables, *viz.*, categorical/symbolic data and numerical data. Nauck [59] has developed a learning algorithm that creates *mixed* fuzzy rules involving both categorical and numeric attributes. Pedrycz [55] discusses some constructive and fuzzy set-driven computational vehicles of knowledge discovery, and establishes the relationship between data mining and fuzzy modeling. The role of fuzzy sets is categorized below based on the different functions of data mining that are modeled.

1) Clustering: Data mining aims at sifting through large volumes of data in order to reveal useful information in the form of new relationships, patterns, or clusters, for decision-making by a user [60]. Fuzzy sets support a focused search, specified in linguistic terms, through data. They also help discover dependencies between the data in qualitative/semi-qualitative format. In data mining, one is typically interested in a focused discovery of structure and an eventual quantification of functional dependencies existing therein. This helps prevent searching for meaningless or trivial patterns in a database. Researchers have developed fuzzy clustering algorithms for this purpose [26]. Russell and Lodwick [27] have explored fuzzy clustering methods for mining telecommunications customer and prospect databases to gain residential and business customer market share. Pedrycz has designed fuzzy clustering algorithms [28] using 1) contextual information and 2) induced linguistic space for better focusing of the search procedure in KDD.

Achieving focus is important in data mining because there are too many attributes and values to be considered and can result in combinatoric explosion. Most unsupervised data mining approaches try to achieve attribute focus by first recognizing the most interesting features. Mazlack [61] suggests a converse approach of progressively reducing the data set by partitioning and eliminating the least important attributes to reduce intraitem dissonance within the partitions. A *soft* focus is used to handle both crisp and imprecise data. It works by progressive reduction of cognitive dissonance, leading to an increase in useful information. The objective is to generate cohesive and comprehensible information *nuggets* by sifting out uninteresting attributes. A combined distance metric takes care of different types of attributes simultaneously, thus avoiding any taxonomic structure. Noncrisp values are handled by granularization followed by partitioning.

Increased granularity reduces attribute distinctiveness, resulting in loss of useful information, while finer grains lead to partitioning difficulty. Soft granules can be defined in terms of membership functions. *Granular computing* [62] is useful in finding meaningful patterns in data by expressing and processing chunks of information (granules). These are regarded as essential entities in all cognitive pursuits geared toward establishing meaningful patterns in data. The concept of granular computing allows one to concentrate all computational effort on some specific and problem-oriented subsets of a complete database. It also helps split an overall computing effort into several subtasks, leading to a *modularization* effect.

2) Association Rules: An important area of data mining research deals with the discovery of *association rules* [42]. An association rule describes an interesting association relationship among different attributes. A Boolean association involves binary attributes, a generalized association involves attributes that are hierarchically related, and a quantitative association involves attributes that can take on quantitative or categorical values. The use of fuzzy techniques has been considered to be one of the key components of data mining systems because of the affinity with human knowledge representation [63]. Wei and Chen [43] have mined generalized association rules with fuzzy taxonomic structures. A crisp taxonomy assumes that a child belongs to its ancestor with degree one. A fuzzy taxonomy is represented as a directed acyclic graph, each of whose edges represents a fuzzy *IS-A* relationship with degree μ $(0 \leq \mu \leq 1)$. The partial belonging of an item in a taxonomy is taken into account while computing the degrees of support and confidence.

Au and Chan [44] utilize an *adjusted difference* between observed and expected frequency counts of attributes for discovering fuzzy association rules in relational databases. Instead of dividing quantitative attributes into fixed intervals, they employ linguistic terms to represent the revealed regularities and exceptions. Here no user-supplied thresholds are required, and quantitative values can be directly inferred from the rules. The linguistic representation leads to the discovery of *natural* and more understandable rules. The algorithm allows one to discover both *positive* and *negative* rules, and can deal with fuzzy class boundaries as well as missing values in databases. The use of fuzzy techniques buries the boundaries of adjacent intervals of numeric quantities, resulting in resilience to noises such as inaccuracies in physical measurements of real life entities. The effectiveness of the algorithm was demonstrated on a transactional database of a PBX system and a database concerning industrial enterprises in mainland China.

3) Functional Dependencies: Fuzzy logic has been used for analyzing inference based on functional dependencies (FDs), between variables, in database relations. Fuzzy inference generalizes both imprecise (set-valued) and precise inference. Similarly, fuzzy relational databases generalize their classical and imprecise counterparts by supporting fuzzy information storage and retrieval [50]. Inference analysis is performed using a special abstract model which maintains vital links to classical, imprecise and fuzzy relational database models. These links increase the utility of the inference formalism in practical applications involving "catalytic inference analysis," including knowledge discovery and database security. FDs are an interesting notion from a knowledge discovery standpoint since they allow one to express, in a condensed form, some properties of the real world which are valid on a given database. These properties can then be used in various applications such as reverse engineering or query optimization. Bosc *et al.* [51] use a data mining algorithm to extract/discover extended FDs, represented by gradual rules composed of linguistic variables.

4) Data Summarization: Summary discovery is one of the major components of knowledge discovery in databases. This provides the user with comprehensive information for grasping the essence from a large amount of information in a database. Fuzzy set theory is also used for data summarization [46]. Typically, fuzzy sets are used for an interactive top-down summary discovery process which utilizes fuzzy *IS-A* hierarchies as domain knowledge. Here generalized tuples are used as a representational form of a database summary including fuzzy concepts. By virtue of fuzzy *IS-A* hierarchies, where fuzzy *IS-A* relationships common in actual domains are naturally expressed, the discovery process comes up with more accurate database summaries.

Linguistic summaries of large sets of data are derived as linguistically quantified propositions with a degree of validity [47]. This corresponds to the preference criterion involved in the mining task. The system consists of a summarizer (like, *young*), a quantity in agreement (like, *most*), and the truth/validity (say, 0.7). Single-attribute simple summarizers often need to be extended for some confluence of attribute values, implying combinatorial problems due to the huge number (all possible combinations) of summaries involved and the determination of the most appropriate/valid one.

It is found that often the most interesting linguistic summaries are nontrivial and human-consistent concepts, involving complicated combinations of attributes. In practice, this cannot be generated automatically and *human assistance/interaction* is required. Kacprzyk and Zadrozny [48] have developed *FQUERY*, a fuzzy querying add-on for *Access*, for an interactive linguistic summarization using *natural* terms and *comprehensible* quantifiers. It supports various fuzzy elements in queries, including interval attributes with membership for matching in a fuzzy relation and importance coefficients. First the user has to formulate a set of linguistic summaries of interest. The system then retrieves records from the database and calculates the validity of each summary. Finally, a most appropriate linguistic summary is selected. The scheme has also been used for fuzzy querying over the Internet, using a WWW browser like Microsoft Explorer or Netscape Navigator. The definition of fuzzy values, fuzzy relations, and linguistic quantifiers is via Java applets.

Chiang *et al.* [52] have used fuzzy linguistic summary for mining time series data. The system provides human interaction, in the form of a graphic display tool, to help users premine a database and determine what knowledge could be discovered. The model is used to predict the on-line utilization ranks of different resources, including CPU and real storage.

5) Web Application: Mining typical user profiles and URL associations from the vast amount of access logs is an important component of Web personalization, that deals with tailoring a user's interaction with the Web information space based on information about him/her. Nasraoui *et al.* [64] have defined a *user session* as a temporally compact sequence of Web accesses by a user and used a dissimilarity measure between two Web sessions to capture the organization of a Web site. Their goal is to categorize these sessions using Web mining.

6) Image Retrieval: Recent increase in the size of *multimedia* information repositories, consisting of mixed media data, has made content-based image retrieval (CBIR) an active research area [65]. Unlike traditional database techniques which retrieve images based on exact matching of keywords, CBIR systems represent the information content of an image by visual features such as color, texture, and shape, and retrieve images based on similarity of features. Frigui [66] has developed an *interactive* and *iterative* image retrieval system that takes into account the *subjectivity* of human perception of visual content. The feature relevance weights are learned from the user's positive and negative feedback, and the Choquet integral is used as a dissimilarity measure. The smooth transition in the user's feedback is modeled by continuous fuzzy membership functions. Medasani and Krishnapuram [67] have designed a fuzzy approach to handle complex linguistic queries consisting of multiple attributes. Such queries are usually more *natural, user-friendly*, and *interpretable* for image retrieval. The degree to which an image satisfies an attribute is given by the membership value of the feature vector corresponding to the image in the membership function for the attribute. Fuzzy connectives are used to combine the degrees of satisfaction of multiple attributes in a complex query to arrive at an overall degree of satisfaction while ranking images for retrieval.

B. Neural Networks

Neural networks were earlier thought to be unsuitable for data mining because of their inherent *black-box* nature. No information was available from them in symbolic form, suitable for verification or interpretation by humans. Recently there has been widespread activity aimed at redressing this situation, by extracting the embedded knowledge in trained networks in the form of symbolic rules [34]. This serves to identify the attributes that, either individually or in a combination, are the most significant determinants of the decision or classification. Unlike fuzzy sets, the main contribution of neural nets toward data mining stems from rule extraction and clustering.

1) Rule Extraction: In general, the primary input to a connectionist rule extraction algorithm is a representation of the trained neural network, in terms of its nodes, links and sometimes the data set. One or more hidden and output units are used to automatically derive the rules, which may later be combined and simplified to arrive at a more comprehensible rule set. These rules can also provide new insights into the application domain. The use of neural nets helps in 1) incorporating parallelism and 2) tackling optimization problems in the data domain. The models are usually suitable in *data-rich* environments.

Typically a network is first trained to achieve the required accuracy rate. Redundant connections of the network are then removed using a pruning algorithm. The link weights and activation values of the hidden units in the network are analyzed, and classification rules are generated [34], [35].

2) Rule Evaluation: Here we provide some quantitative measures to evaluate the performance of the generated rules [68]. This relates to the *preference criteria/goodness of fit* chosen for the rules. Let N be an $l \times l$ matrix whose (i, j)th element n_{ij} indicates the number of patterns actually belonging to class i, but classified as class j.

- *Accuracy*: It is the correct classification percentage, provided by the rules on a test set defined as $(n_{ic}/n_i) \cdot 100$, where n_i is equal to the number of points in class i and n_{ic} of these points are correctly classified.
- *User's accuracy*: If n'_i points are found to be classified into class i, then the user's accuracy (U) is defined as $U = n_{ic}/n'_i$.
- *Kappa*: The kappa value for class $i (K_i)$ is defined as

$$K_i = \frac{n \cdot n_{ic} - n_i \cdot n'_i}{n \cdot n'_i - n_i \cdot n'_i}. \tag{3}$$

  The numerator and denominator of overall kappa are obtained by summing the respective numerators and denominators of K_i separately over all classes.
- *Fidelity*: It is measured as the percentage of the test set for which network and the rulebase output agree [68].
- *Confusion*: This measure quantifies the goal that the "*confusion should be restricted within minimum number of classes*". Let $\hat{n}_{ij}$ be the mean of all n_{ij} for $i \neq j$. Then [68]

$$\mathrm{Conf} = \frac{\mathrm{Card}\{n_{ij} : n_{ij} \geq \hat{n}_{ij}, i \neq j\}}{l} \tag{4}$$

  for an l class problem.
- *Coverage*: The percentage of examples from a test set for which no rules are fired is used as a measure of the uncovered region. A rulebase having a smaller uncovered region is superior.
- *Rulebase size*: This is measured in terms of the number of rules. The lower its value, the more compact is the rulebase.
- *Computational complexity*: This is measured in terms of the CPU time required.
- *Confidence*: The confidence of the rules is defined by a confidence factor cf. We have [68]

$$cf_j = \inf_{j:\ \text{all nodes in the path}} \frac{(\Sigma_i w_{ji} - \theta_j)}{\Sigma_i w_{ji}} \tag{5}$$

  where w_{ji} is the ith incoming link weight to node j and θ_j is its threshold.

3) Clustering and Self Organization: One of the big challenges of data mining is the organization and retrieval of documents from archives. Kohonen *et al.* [31] have demonstrated the utility of a huge self-organizing map (SOM) with more than one million nodes to partition a little less than seven million patent abstracts where the documents are represented by 500-dimensional feature vectors. Vesanto *et al.* [32] employ a step-wise strategy by partitioning the data with a SOM, followed by its clustering. Alahakoon *et al.* [33] perform hierarchical clustering of SOMs, based on a spread factor which is independent of the dimensionality of the data.

Shalvi and DeClaris [29] have designed a data mining technique, combining Kohonen's self-organizing neural network with data visualization, for clustering a set of pathological data containing information regarding the patients' drugs, topographies (body locations) and morphologies (physiological abnormalities). König [69] has combined SOM and Sammon's nonlinear mapping for reducing the dimension of data representation for visualization purposes.

4) Regression: Neural networks have also been used for a variety of classification and regression tasks [23]. Time series prediction has been attempted by Lee and Liu [53]. They have employed a neural oscillatory elastic graph matching model with hybrid radial basis functions for tropical cyclone identification and tracking.

C. Neuro-Fuzzy Computing

Neuro-fuzzy computation [11] is one of the most popular hybridizations widely reported in literature. It comprises a judicious integration of the merits of neural and fuzzy approaches, enabling one to build more intelligent decision-making systems. This incorporates the generic advantages of artificial neural networks like massive parallelism, robustness, and learning in *data-rich* environments into the system. The modeling of imprecise and qualitative knowledge in natural/linguistic terms as well as the transmission of uncertainty are possible through the use of fuzzy logic. Besides these generic advantages, the neuro-fuzzy approach also provides the corresponding application specific merits as highlighted earlier.

The rule generation aspect of neural networks is utilized to extract more *natural* rules from fuzzy neural networks [36]. The fuzzy MLP [18] and fuzzy Kohonen network [19] have been used for linguistic rule generation and inferencing. Here the input, besides being in quantitative, linguistic, or set forms, or a combination of these, can also be incomplete. The components of the input vector consist of membership values to the overlapping partitions of linguistic properties *low, medium,* and *high* corresponding to each input feature. Output decision is provided in terms of class membership values. The block diagram of a fuzzy neural network is depicted in Fig. 2.

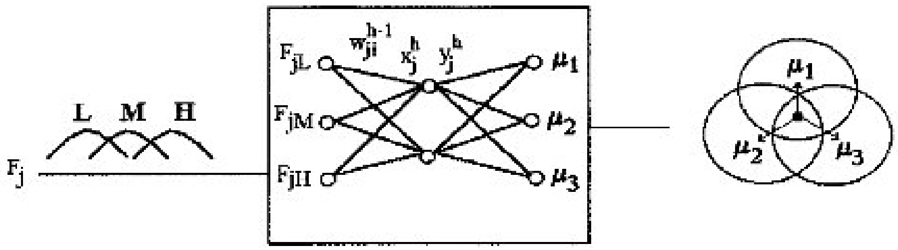

Fig. 2. Block diagram of a fuzzy neural network.

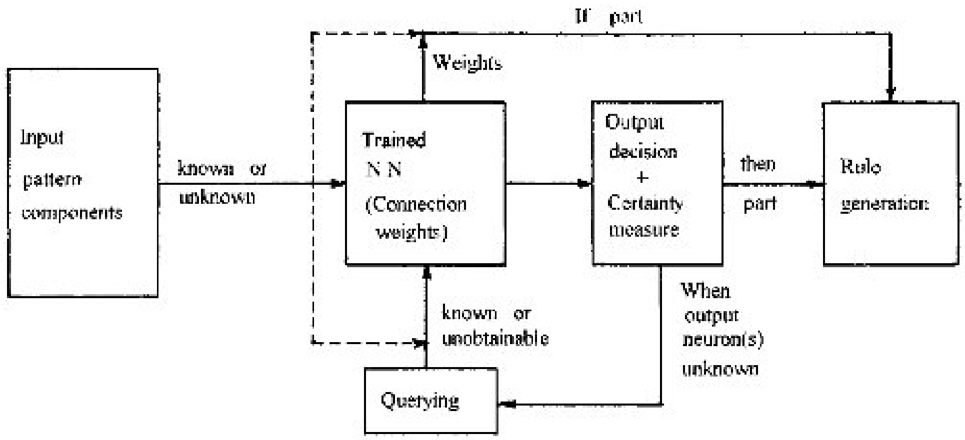

Fig. 3. Block diagram of inferencing and rule generation phases.

The models are capable of

- inferencing based on complete and/or partial information;
- querying the user for unknown input variables that are key to reaching a decision;
- producing justification for inferences in the form of IF-THEN rules.

The connection weights and node activation values of the trained network are used in the process. A *certainty factor* determines the confidence in an output decision. Note that this certainty refers to the preference criterion for the extracted rules, and is different from the notion of certain patterns of (1). Fig. 3, [18] gives an overall view of the various stages involved in the process of inferencing and rule generation.

Zhang *et al.* [41] have designed a granular neural network to deal with numerical-linguistic data fusion and granular knowledge discovery in numerical-linguistic databases. The network is capable of learning internal granular relations between input and output and predicting new relations. Low-level granular data can be compressed to generate high-level granular knowledge in the form of rules.

A neuro-fuzzy knowledge-based network by Mitra *et al.* [20] is capable of generating both *positive* and *negative* rules in linguistic form to justify any decision reached. In the absence of positive information regarding the belonging of a pattern to class C_k, the complementary information about the pattern not belonging to class $C_{k'}$ is used for generating the negative rules. The *a priori* class information and the distribution of pattern points in the feature space are taken into account while encoding the crude *domain knowledge* from the data set among the connection weights. Fuzzy intervals and linguistic sets are used in the process. The network topology is automatically determined, followed by refinement using growing and/or pruning of links and nodes. The knowledge-based network converges earlier, resulting in more meaningful rules.

D. Genetic Algorithms

GAs are adaptive, robust, efficient, and global search methods, suitable in situations where the search space is large. They optimize a *fitness function*, corresponding to the preference criterion of data mining, to arrive at an optimal solution using certain genetic operators. Knowledge discovery systems have been developed using genetic programming concepts [70], [71]. The *MASSON* system [72], where intentional information is extracted for a given set of objects, is popular. The problem addressed is to find common characteristics of a set of objects in an object-oriented database. Genetic programming is used to automatically generate, evaluate, and select object-oriented queries. GAs are also used for several other purposes like fusion of multiple data types in *multimedia* databases, and automated program generation for mining multimedia data [73].

However, the literature in the domain of GA-based data mining is not as rich as that of fuzzy sets. We provide below a categorization of few such interesting systems based on the functions modeled.

1) Regression: Besides discovering human-interpretable patterns data mining also encompasses prediction [8], where some variables or attributes in the database are used to determine unknown or future values of other variables of interest. The traditional weighted average or linear multiregression models for prediction require a basic assumption that there is no interaction among the attributes. GAs, on the other hand, are able to handle attribute interaction in a better manner. Xu *et* *al.*[24] have designed a multi-input–single-output system using a nonlinear integral. An adaptive GA is used for learning the nonlinear multiregression from a set of training data.

Noda *et al.* [25] use GAs to discover *interesting* rules in a dependence modeling task, where different rules can predict different goal attributes. Generally attributes with high information gain are good predictors of a class when considered individually. However attributes with low information gain could become more relevant when attribute interactions are taken into account. This phenomenon is associated with rule interestingness. The degree of interestingness of the consequent is computed based on the relative frequency of the value being predicted by it. In other words, the rarer the value of a goal attribute, the more interesting a rule it predicts. The authors attempt to discover a few interesting rules (knowledge nuggets) instead of a large set of accurate (but not necessarily interesting) rules.

2) Association Rules: Lopes *et al.* [45] evolve association rules of IF C THEN P type, which provide a high degree of accuracy and coverage. While the *accuracy* of a rule measures its degree of confidence, its *coverage* is interpreted as the comprehensive inclusion of all the records that satisfy the rule. Hence $\mathrm{Accuracy} = (|C \bigcap P|)/(|C \bigcap P| + |C \bigcap \overline{P}|)$ and $\mathrm{Coverage} = (|C \bigcap P|)/(|C \bigcap P|) + (|\overline{C} \bigcap P|)$ are defined. Note that quantitative measures for rule evaluation have been discussed in Section III-B2, with reference to neural networks.

E. Rough Sets

The theory of rough sets [74] has emerged as a major mathematical tool for managing uncertainty that arises from granularity in the domain of discourse, i.e., from the indiscernibility between objects in a set, and has proved to be useful in a variety of KDD processes. It offers mathematical tools to discover hidden patterns in data and therefore its importance, as far as data mining is concerned, can in no way be overlooked. A fundamental principle of a rough set-based learning system is to discover redundancies and dependencies between the given features of a problem to be classified. It approximates a given concept from below and from above, using *lower* and *upper approximations*. Fig. 4 provides a schematic diagram of a rough set.

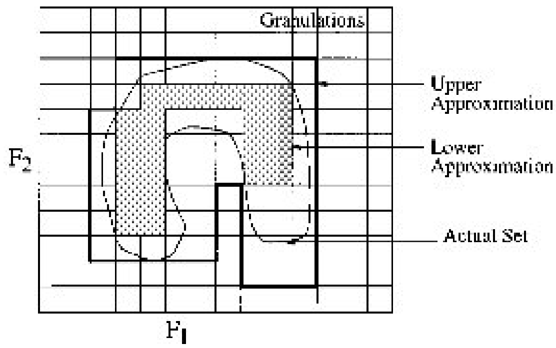

Fig. 4. Lower and upper approximations in a rough set.

A rough set learning algorithm can be used to obtain a set of rules in IF-THEN form, from a *decision table*. The rough set method provides an effective tool for extracting knowledge from databases. Here one first creates a knowledge base, classifying objects and attributes within the created decision tables. Then a knowledge discovery process is initiated to remove some undesirable attributes. Finally the data dependency is analyzed, in the reduced database, to find the minimal subset of attributes called *reduct*.

Rough set applications to data mining generally proceed along the following directions.

1) *Decision rule induction from attribute value table* [37]–[40]. Most of these methods are based on generation of discernibility matrices and reducts.
2) *Data filtration by template generation* [75]. This mainly involves extracting elementary blocks from data based on equivalence relation. Genetic algorithms are also sometimes used in this stage for searching, so that the methodologies can be used for large data sets.

Besides these, reduction of memory and computational requirements for rule generation, and working on dynamic databases [40] are also considered.

Some of the rough set-based systems developed for data mining include 1) the KDD-R system based on the variable precision rough set (VPRS) model [76]; and 2) the rule induction system based on learning from examples based on rough set theory (LERS) [77]. LERS has been extended in [78] to handle *missing* attributes using the closest fit.

F. Other Hybridizations

Banerjee *et al.* [21] have used a *rough-neuro-fuzzy* integration to design a knowledge-based system, where the theory of rough sets is utilized for extracting domain knowledge. In the said rough-fuzzy MLP, the extracted crude domain knowledge is encoded among the connection weights. Rules are generated from a decision table by computing relative reducts. The network topology is automatically determined and the dependency factors of these rules are encoded as the initial connection weights. The hidden nodes model the conjuncts in the antecedent part of a rule, while the output nodes model the disjuncts. Various other *rough-fuzzy* hybridizations for intelligent system design are reported in [79].

A promising direction in mining a huge dataset is to 1) partition it; 2) develop classifiers for each module; and 3) combine the results. A modular approach has been pursued [22], [68], [80] to combine the knowledge-based rough-fuzzy MLP subnetworks/modules generated for each class, using GAs. Fig. 5 depicts the knowledge flow for the entire process. An l-class classification problem is split into l two-class problems. Dependency rules are extracted directly from real-valued attribute table consisting of fuzzy membership values by adaptively applying a threshold. The final network is evolved using a GA with restricted mutation operator, in a novel *rough-neuro-fuzzy-genetic* framework. The *divide and conquer* strategy, followed by evolutionary optimization, is found to enhance the performance of the network.

George and Srikanth [49] have used a *fuzzy-genetic* integration, where GAs are applied to determine the most appropriate data summary. Kiem and Phuc [30] have developed a *rough-neuro-genetic* hybridization for discovering conceptual clusters from a large database.

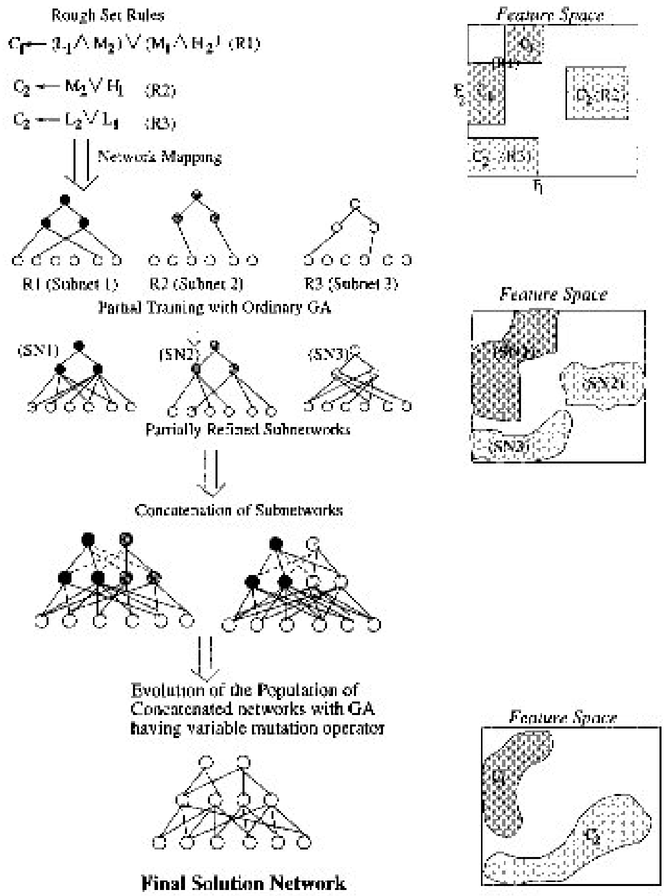

Fig. 5. Knowledge flow in a modular rough-neuro-fuzzy-genetic system.

IV. Conclusion and Discussion

Current research in data mining mainly focuses on the discovery algorithm and visualization techniques. There is a growing awareness that, in practice, it is easy to discover a huge number of patterns in a database where most of these patterns are actually obvious, redundant, and useless or uninteresting to the user. To prevent the user from being overwhelmed by a large number of uninteresting patterns, techniques are needed to identify only the useful/interesting patterns and present them to the user.

Soft computing methodologies, involving fuzzy sets, neural networks, genetic algorithms, rough sets, and their hybridizations, have recently been used to solve data mining problems. They strive to provide approximate solutions at low cost, thereby speeding up the process. A categorization has been provided based on the different soft computing tools and their hybridizations used, the mining function implemented, and the preference criterion selected by the model.

Fuzzy sets, which constitute the oldest component of soft computing, are suitable for handling the issues related to understandability of patterns, incomplete/noisy data, mixed media information and human interaction, and can provide approximate solutions faster. They have been mainly used in clustering, discovering association rules and functional dependencies, summarization, time series analysis, web applications, and image retrieval.

Neural networks are suitable in data-rich environments and are typically used for extracting embedded knowledge in the form of rules, quantitative evaluation of these rules, clustering, self-organization, classification and regression. They have an advantage, over other types of machine learning algorithms, for scaling [81].

Neuro-fuzzy hybridization exploits the characteristics of both neural networks and fuzzy sets in generating natural/linguistic rules, handling imprecise and mixed mode data, and modeling highly nonlinear decision boundaries. Domain knowledge, in natural form, can be encoded in the network for improved performance.

Genetic algorithms provide efficient search algorithms to select a model, from mixed media data, based on some preference criterion/objective function. They have been employed in regression and in discovering association rules. Rough sets are suitable for handling different types of uncertainty in data and have been mainly utilized for extracting knowledge in the form of rules.

Other hybridizations typically enjoy the generic and application-specific merits of the individual soft computing tools that they integrate. Data mining functions modeled by such systems include rule extraction, data summarization, clustering, incorporation of domain knowledge, and partitioning. It is to be noted that the notion of partitioning, i.e., the modular approach, provides an effective direction for scaling up algorithms and speeding up convergence. Case-based reasoning (CBR), a novel AI problem solving paradigm, has recently drawn the attention of both soft computing and data mining communities. A profile of potential applications is available in [82].

Some of the challenges to the use of these methodologies include the following.

- Scalability problem to extremely large heterogeneous databases spread over multiple files, possibly in different disks or across the web in different geographical locations. Often combining such data in a single very large file may be infeasible.
- Feature evaluation and dimensionality reduction to improve prediction accuracy. Some recent work in this direction is available in [83]–[86].
- Choice of metrics and evaluation techniques to handle dynamic changes in data.
- Incorporation of domain knowledge and user interaction.
- Quantitative evaluation of performance.
- Efficient integration of soft computing tools. In this connection the computational theory of perceptions, as explained by Zadeh [87], needs attention.

Recently, several commercial data mining tools have been developed based on soft computing methodologies. These include Data Mining Suite, using fuzzy logic; Braincell, Cognos 4Thought and IBM Intelligent Miners for Data, using neural networks; and Nuggets, using GAs.

Since the databases to be mined are often very large, parallel algorithms are desirable [88]. However, one has to explore a tradeoff between computation, communication, memory usage, synchronization, and the use of problem-specific information to select a suitable parallel algorithm for data mining. One can also also partition the data appropriately and distribute the subsets to multiple processors, learning concept descriptions in parallel, and then combining them. This corresponds to loosely coupled collections of otherwise independent algorithms, and is termed *distributed data mining* [89].

References

[1] U. Fayyad and R. Uthurusamy, "Data mining and knowledge discovery in databases," *Commun. ACM*, vol. 39, pp. 24–27, 1996.

[2] W. H. Inmon, "The data warehouse and data mining," *Commun. ACM*, vol. 39, pp. 49–50, 1996.

[3] J. A. Major and D. R. Riedinger, "EFD—A hybrid knowledge statistical-based system for the detection of fraud," *Int. J. Intell. Syst.*, vol. 7, pp. 687–703, 1992.

[4] R. L. Blum, *Discovery and Representation of Causal Relationships From a Large Time-Oriented Clinical Database: The RX Project*. New York: Spinger-Verlag, 1982, vol. 19. of *Lecture Notes in Medical Informatics*.

[5] R. Heider, Troubleshooting CFM 56-3 Engines for the Boeing 737—Using CBR and Data-Mining, Spinger-Verlag, New York, vol. 1168, pp. 512–523, 1996. *Lecture Notes in Computer Science*.

[6] U. Fayyad, D. Haussler, and P. Stolorz, "Mining scientific data," *Commun. ACM*, vol. 39, pp. 51–57, 1996.

[7] O. Etzioni, "The world-wide web: Quagmire or goldmine?," *Commun. ACM*, vol. 39, pp. 65–68, 1996.

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

[9] "Special issue on knowledge discovery in data- and knowledge bases," *Int. J. Intell. Syst.*, vol. 7, 1992.

[10] L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing," *Commun. ACM*, vol. 37, pp. 77–84, 1994.

[11] S. K. Pal and S. Mitra, *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing*. New York: Wiley, 1999.

[12] J. Furnkranz, J. Petrak, and R. Trappl, "Knowledge discovery in international conflict databases," *Applied Artificial Intelligence*, vol. 11, pp. 91–118, 1997.

[13] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann, 1993.

[14] K. J. Cios, W. Pedrycz, and R. Swiniarski, *Data Mining Methods for Knowledge Discovery*. Dordrecht, The Netherlands: Kluwer, 1998.

[15] A. Silberschatz and A. Tuzhilin, "What makes patterns interesting in knowledge discovery systems," *IEEE Trans. Knowledge Data Eng.*, vol. 8, pp. 970–974, 1996.

[16] T. M. Mitchell, "Machine learning and data mining," *Commun. ACM*, vol. 42, no. 11, 1999.

[17] U. Fayyad, G. P. Shapiro, and P. Smyth, "The KDD process for extracting useful knowledge from volumes of data," *Commun. ACM*, vol. 39, pp. 27–34, 1996.

[18] S. Mitra and S. K. Pal, "Fuzzy multi-layer perceptron, inferencing and rule generation," *IEEE Trans. Neural Networks*, vol. 6, pp. 51–63, 1995.

[19] ——, "Fuzzy self organization, inferencing and rule generation," *IEEE Trans. Syst., Man, Cybern. A*, vol. 26, pp. 608–620, 1996.

[20] S. Mitra, R. K. De, and S. K. Pal, "Knowledge-based fuzzy MLP for classification and rule generation," *IEEE Trans. Neural Networks*, vol. 8, pp. 1338–1350, 1997.

[21] M. Banerjee, S. Mitra, and S. K. Pal, "Rough fuzzy MLP: Knowledge encoding and classification," *IEEE Trans. Neural Networks*, vol. 9, pp. 1203–1216, 1998.

[22] S. Mitra, P. Mitra, and S. K. Pal, "Evolutionary modular design of rough knowledge-based network using fuzzy attributes," *Neurocomput.*, vol. 36, pp. 45–66, 2001.

[23] V. Ciesielski and G. Palstra, "Using a hybrid neural/expert system for database mining in market survey data," in *Proc. Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*. Portland, OR, Aug. 2–4, 1996, p. 38.

[24] K. Xu, Z. Wang, and K. S. Leung, "Using a new type of nonlinear integral for multiregression: An application of evolutionary algorithms in data mining," *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, pp. 2326–2331, Oct. 1998.

[25] E. Noda, A. A. Freitas, and H. S. Lopes, "Discovering interesting prediction rules with a genetic algorithm," *Proc. IEEE Congr. Evolutionary Comput. CEC '99*, pp. 1322–1329, July 1999.

[26] I. B. Turksen, "Fuzzy data mining and expert system development," *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, pp. 2057–2061, Oct. 1998.

[27] S. Russell and W. Lodwick, "Fuzzy clustering in data mining for telco database marketing campaigns," in *Proc. NAFIPS 99*, New York, NY, June 1999, pp. 720–726.

[28] W. Pedrycz, "Conditional fuzzy c-means," *Pattern Recognition Lett.*, vol. 17, pp. 625–632, 1996.

[29] D. Shalvi and N. De Claris, "Unsupervised neural network approach to medical data mining techniques," *Proc. IEEE Int. Joint Conf. Neural Networks*, pp. 171–176, May 1998.

[30] H. Kiem and D. Phuc, "Using rough genetic and Kohonen's neural network for conceptual cluster discovery in data mining," in *Proc. RSFDGrC'99*, Yamaguchi, Japan, Nov. 1999, pp. 448–452.

[31] T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paatero, and A. Saarela, "Self organization of a massive document collection," *IEEE Trans. Neural Networks*, vol. 11, pp. 574–585, 2000.

[32] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Neural Networks*, vol. 11, pp. 586–600, 2000.

[33] D. Alahakoon, S. K. Halgamuge, and B. Srinivasan, "Dynamic self organizing maps with controlled growth for knowledge discovery," *IEEE Trans. Neural Networks*, vol. 11, pp. 601–614, 2000.

[34] A. B. Tickle, R. Andrews, M. Golea, and J. Diederich, "The truth will come to light: Directions and challenges in extracting the knowledge embedded within trained artificial neural networks," *IEEE Trans. Neural Networks*, vol. 9, pp. 1057–1068, 1998.

[35] H. J. Lu, R. Setiono, and H. Liu, "Effective data mining using neural networks," *IEEE Trans. Knowledge Data Eng.*, vol. 8, pp. 957–961, 1996.

[36] S. Mitra and Y. Hayashi, "Neuro-fuzzy rule generation: Survey in soft computing framework," *IEEE Trans. Neural Networks*, vol. 11, pp. 748–768, 2000.

[37] T. Mollestad and A. Skowron, "A rough set framework for data mining of propositional default rules," in *Lecture Notes Comput. Sci.*, 1996, vol. 1079, pp. 448–457.

[38] X. Hu and N. Cercone, "Mining knowledge rules from databases: A rough set approach," in *Proc. 12th Int. Conf. Data Eng.*. Washington, DC, Feb. 1996, pp. 96–105.

[39] A. Skowron, "Extracting laws from decision tables—A rough set approach," *Comput. Intell.*, vol. 11, pp. 371–388, 1995.

[40] N. Shan and W. Ziarko, "Data-based acquisition and incremental modification of classification rules," *Comput. Intell.*, vol. 11, pp. 357–370, 1995.

[41] Y. Q. Zhang, M. D. Fraser, R. A. Gagliano, and A. Kandel, "Granular neural networks for numerical-linguistic data fusion and knowldege discovery," *IEEE Trans. Neural Networks*, vol. 11, pp. 658–667, 2000.

[42] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *Proc. 1993 ACM SIGMOD Int. Conf. Management Data*, Washington, DC, May 1993, pp. 207–216.

[43] Q. Wei and G. Chen, "Mining generalized association rules with fuzzy taxonomic structures," in *Proc. NAFIPS 99*, New York, June 1999, pp. 477–481.

[44] W. H. Au and K. C. C. Chan, "An effective algorithm for discovering fuzzy rules in relational databases," *Proc. IEEE Int. Conf. Fuzzy Syst. FUZZ IEEE 98*, pp. 1314–1319, May 1998.

[45] C. Lopes, M. Pacheco, M. Vellasco, and E. Passos, "Rule-evolver: An evolutionary approach for data mining," in *Proc. RSFDGrC'99*, Yamaguchi, Japan, Nov. 1999, pp. 458–462.

[46] D. H. Lee and M. H. Kim, "Database summarization using fuzzy ISA hierarchies," *IEEE Trans. Syst., Man, Cybern. B*, vol. 27, pp. 68–78, 1997.

[47] R. R. Yager, "On linguistic summaries of data," in *Knowledge Discovery in Databases*, W. Frawley and G. Piatetsky-Shapiro, Eds. Menlo Park, CA: AAAI/MIT Press, 1991, pp. 347–363.

[48] J. Kacprzyk and S. Zadrozny, "Data mining via linguistic summaries of data: An interactive approach," in *Proc. IIZUKA 98*, Fukuoka, Japan, Oct. 1998, pp. 668–671.

[49] R. George and R. Srikanth, "Data summarization using genetic algorithms and fuzzy logic," in *Genetic Algorithms and Soft Computing*, F. Herrera and J. L. Verdegay, Eds. Heidelberg, Germany: Physica-Verlag, 1996, pp. 599–611.

[50] J. Hale and S. Shenoi, "Analyzing FD inference in relational databases," *Data Knowledge Eng.*, vol. 18, pp. 167–183, 1996.

[51] P. Bosc, O. Pivert, and L. Ughetto, "Database mining for the discovery of extended functional dependencies," in *Proc. NAFIPS 99*, New York, June 1999, pp. 580–584.

[52] D. A. Chiang, L. R. Chow, and Y. F. Wang, "Mining time series data by a fuzzy linguistic summary system," *Fuzzy Sets Syst.*, vol. 112, pp. 419–432, 2000.

[53] R. S. T. Lee and J. N. K. Liu, "Tropical cyclone identification and tracking system using integrated neural oscillatory leastic graph matching and hybrid RBF network track mining techniques," *IEEE Trans. Neural Networks*, vol. 11, pp. 680–689, 2000.

[54] F. Provost and V. Kolluri, "A survey of methods for scaling up inductive algorithms," *Data Mining Knowledge Discovery*, vol. 2, pp. 131–169, 1999.

[55] W. Pedrycz, "Fuzzy set technology in knowledge discovery," *Fuzzy Sets Syst.*, vol. 98, pp. 279–290, 1998.

[56] B. Liu, W. Hsu, L. F. Mun, and H. Y. Lee, "Finding interesting patterns using user expectation," *IEEE Trans. Knowledge Data Eng.*, vol. 11, pp. 817–832, 1999.
[57] R. R. Yager, "Database discovery using fuzzy sets," *Int. J. Intell. Syst.*, vol. 11, pp. 691–712, 1996.
[58] J. F. Baldwin, "Knowledge from data using fuzzy methods," *Pattern Recognition Lett.*, vol. 17, pp. 593–600, 1996.
[59] D. Nauck, "Using symbolic data in neuro-fuzzy classification," in *Proc. NAFIPS 99*, New York, June 1999, pp. 536–540.
[60] P. Piatetsky-Shapiro and W. J. Frawley, Eds., *Knowledge Discovery in Databases*. Menlo Park, CA: AAAI/MIT Press, 1991.
[61] L. J. Mazlack, "Softly focusing on data," in *Proc. NAFIPS 99*, New York, June 1999, pp. 700–704.
[62] L. A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets Syst.*, vol. 19, pp. 111–127, 1997.
[63] A. Maeda, H. Ashida, Y. Taniguchi, and Y. Takahashi, "Data mining system using fuzzy rule induction," *Proc. IEEE Int. Conf. Fuzzy Syst. FUZZ IEEE 95*, pp. 45–46, Mar. 1995.
[64] O. Nasraoui, R. Krishnapuram, and A. Joshi, "Relational clustering based on a new robust estimator with application to web mining," in *Proc. NAFIPS 99*, New York, June 1999, pp. 705–709.
[65] S. K. Pal, A. Ghosh, and M. K. Kundu, Eds., *Soft Computing for Image Processing*. Heidelberg, Germany: Physica-Verlag, 2000.
[66] H. Frigui, "Adaptive image retrieval using the fuzzy integral," in *Proc. NAFIPS 99*, New York, June 1999, pp. 575–579.
[67] S. Medasani and R. Krishnapuram, "A fuzzy approach to complex linguistic query based image retrieval," in *Proc. NAFIPS 99*, New York, June 1999, pp. 590–594.
[68] S. K. Pal, S. Mitra, and P. Mitra, "Rough fuzzy MLP: Modular evolution, rule generation and evaluation," *IEEE Trans. Knowledge Data Eng.*, 2001. To appear.
[69] A. Koenig, "Interactive visualization and analysis of hierarchical neural projections for data mining," *IEEE Trans. Neural Networks*, vol. 11, pp. 615–624, 2000.
[70] I. W. Flockhart and N. J. Radcliffe, "A genetic algorithm-based approach to data mining," in *Proc. 2nd Int. Conf. Knowledge Discovery Data Mining (KDD-96)*. Portland, OR, Aug. 2–4, 1996, p. 299.
[71] M. L. Raymer, W. F. Punch, E. D. Goodman, and L. A. Kuhn, "Genetic programming for improved data mining: An application to the biochemistry of protein interactions," in *Proc. 1st Annu. Conf. Genetic Programming 1996*, Stanford Univ., CA, July 28–31, 1996, pp. 375–380.
[72] T. Ryu and C. F. Eick, "MASSON: Discovering commonalties in collection of objects using genetic programming," in *Proc. 1st Annu. Conf. Genetic Programming 1996*, Stanford Univ., CA, July 28–31, 1996, pp. 200–208.
[73] A. Teller and M. Veloso, "Program evolution for data mining," *Int. J. Expert Syst.*, vol. 8, pp. 216–236, 1995.
[74] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning about Data*. Dordrecht, : Kluwer, 1991.
[75] L. Polkowski and A. Skowron, *Rough Sets in Knowledge Discovery 1 and 2*. Heidelberg, Germany: Physica-Verlag, 1998.
[76] W. Ziarko and N. Shan, "KDD-R: A comprehensive system for knowledge discovery in databases using rough sets," in *Proc. 3rd Int. Workshop Rough Sets Soft Comput. RSSC'94*, 1994, pp. 164–173.
[77] J. W. Grzymala-Busse, "LERS—A knowledge discovery system," in *Rough Sets in Knowledge Discovery 2, Applications, Case Studies and Software Systems*, L. Polkowski and A. Skowron, Eds. Heidelberg, Germany: Physica-Verlag, 1998, pp. 562–565.
[78] J. W. Grzymala-Busse, W. J. Grzymala-Busse, and L. K. Goodwin, "A closest fit approach to missing attribute values in preterm birth data," in *Proc. RSFDGrC'99*, Yamaguchi, Japan, Nov. 1999, pp. 405–413.
[79] S. K. Pal and A. Skowron, Eds., *Rough Fuzzy Hybridization: A New Trend in Decision Making*, Singapore: Springer-Verlag, 1999.
[80] P. Mitra, S. Mitra, and S. K. Pal, "Staging of cervical cancer with soft computing," *IEEE Trans. Biomed. Eng.*, vol. 47, pp. 934–940, 2000.
[81] Y. Bengio, J. M. Buhmann, M. Embrechts, and J. M. Zurada, "Introduction to the special issue on neural networks for data mining and knowledge discovery," *IEEE Trans. Neural Networks*, vol. 11, pp. 545–549, 2000.
[82] S. K. Pal, T. S. Dillon, and D. S. Yeung, Eds., *Soft Computing Case Based Reasoning*. London, U.K.: Springer-Verlag, 2001.
[83] S. Bengio and Y. Bengio, "Taking on the curse of dimensionality in joint distribution using neural networks," *IEEE Trans. Neural Networks*, vol. 11, pp. 550–557, 2000.
[84] R. Kewley, M. Embrechta, and C. Breneman, "Data strip mining for the virtual design of pharmaceuticals with neural networks," *IEEE Trans. Neural Networks*, vol. 11, pp. 668–679, 2000.
[85] C. K. Shin, S. J. Yu, U. T. Yun, and H. K. Kim, "A hybrid approach of neural network and memory based learning to data mining," *IEEE Trans. Neural Networks*, vol. 11, pp. 637–646, 2000.
[86] S. K. Pal, R. K. De, and J. Basak, "Unsupervised feature evaluation: A neuro-fuzzy approach," *IEEE Trans. Neural Networks*, vol. 11, pp. 366–376, 2000.
[87] L. A. Zadeh, "A new direction in AI: Toward a computational theory of perceptions," *AI Mag.*, vol. 22, pp. 73–84, 2001.
[88] R. Agrawal and J. C. Shafer, "Parallel mining of association rules," *IEEE Trans. Knowledge Data Eng.*, vol. 8, pp. 962–969, 1996.
[89] H. Kargupta and P. Chan, *Advances in Distributed and Paralell Knowledge Discovery*. Cambridge, MA: MIT Press, 2000.

Sushmita Mitra (M'99–SM'00) received the B.Sc. (Hons.) degree in physics and the B.Tech. and M. Tech. degrees in computer science from the University of Calcutta, Calcutta, India, in 1984, 1987, and 1989, respectively, and the Ph.D. degree in computer science from the Indian Statistical Institute, Calcutta, in 1995.

From 1992 to 1994, she was with the European Laboratory for Intelligent Techniques Engineering, Aachen, Germany, as a *German Academic Exchange Service (DAAD)* Fellow. She joined the Indian Statistical Institute, Calcutta, in 1989 and has been an Associate Professor since 1995. She is a coauthor of the book *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing Paradigm* (New York: Wiley, 1999), and has about 50 research publications. She was a Visiting Professor at Meiji University, Japan, in 1999. Her research interests include data mining, pattern recognition, fuzzy sets, artificial intelligence, neural networks and soft computing.

Dr. Mitra was a recipient of the National Talent Search Scholarship (1978–83) from the National Council for Educational Research and Training, India, the IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award in 1994, and CIMPA-INRIA-UNESCO Fellowship in 1996.

Sankar K. Pal (M'81–SM'84–F'93) received the M.Tech. and Ph.D. degrees in radio physics and electronics in 1974 and 1979, respectively, from the University of Calcutta, Calcutta, India. In 1982, he received another Ph.D. degree in electrical engineering along with DIC from Imperial College, University of London.

He is a Professor and Distinguished Scientist at the Indian Statistical Institute, Calcutta. He is also the Founding Head of Machine Intelligence Unit. He worked at the University of California, Berkeley, and the University of Maryland, College Park during 1986 to 1987 as a Fulbright Postdoctoral Visiting Fellow; at the NASA Johnson Space Center, Houston, Texas during 1990 to 1992 and 1994 as a Guest Investigator under the NRC-NASA Senior Research Associateship program; and at the Hong Kong Polytechnic University, Hong Kong in 1999 as a Visiting Professor. He served as a Distinguished Visitor of IEEE Computer Society (USA) for the Asia-Pacific Region during 1997 to 1999. His research interests includes pattern recognition, image processing, data mining, soft computing, neural nets, genetic algorithms, and fuzzy systems. He is a co-author/co-editor of eight books including *A Fuzzy Mathematical Approach to Pattern Recognition*, (New York: Wiley, 1986) and *Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing* (New York: Wiley, 1999); and has about 300 research publications.

Prof. Pal is a Fellow of the IEEE, Third World Academy of Sciences, Italy, and all the four National Academies for Science/Engineering in India. He has received the 1990 S. S. Bhatnagar Prize, the 1993 Jawaharlal Nehru Fellowship, the 1993 Vikram Sarabhai Research Award, the 1993 NASA Tech Brief Award, the 1994 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award, the 1995 NASA Patent Application Award, 1997 IETE—Ram Lal Wadhwa Gold Medal, the 1998 Om Bhasin Foundation Award, the 1999 G. D. Birla Award for Scientific Research, the 2000 Khwarizmi International Award (first winner) from the Islamic Republic of Iran, and the 2001 Syed Husain Zaheer Medal from Indian National Science Academy. He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS (1994 to 1998), *Pattern Recognition Letters*, the *International Journal of Pattern Recognition and Artificial Intelligence, Neurocomputing, Applied Intelligence, Information Sciences, Fuzzy Sets and Systems*, and *Fundamenta Informaticae*; a Member, Executive Advisory Editorial Board, IEEE TRANSACTIONS ON FUZZY SYSTEMS, the *International Journal on Image and Graphics*, and the *International Journal of Approximate Reasoning*; and a Guest Editor of many journals including the IEEE COMPUTER.

Pabitra Mitra received the B.Tech. in electrical engineering from the Indian Insitute of Technology, Kharagpur, in 1996.

He worked as a scientist with the Center for Artificial Intelligence and Robotics, Bangalore, India. Currently he is a Senior Research Fellow of Indian Statistical Institute, Calcuuta. His research interests are in the area of data mining and knowledge discovery, pattern recognition, learning theory, and soft computing.