Fuzzy Data Analysis: Challenges and Perspectives

Rudolf Kruse, Christian Borgelt, and Detlef Nauck
Dept. of Knowledge Processing and Language Engineering
Otto-von-Guericke-University of Magdeburg
D-39106 Magdeburg, Germany
(E-mail: rudolf.kruse@cs.uni-magdeburg.de)

Abstract

In meeting the challenges that resulted from the explosion of collected, stored, and transferred data, *Knowledge Discovery in Databases* or *Data Mining* has emerged as a new research area. However, the approaches studied in this area have mainly been oriented at highly structured and precise data. In addition, the goal to obtain understandable results is often neglected. Therefore we suggest to concentrate on *Information Mining*, i.e., the analysis of heterogeneous information sources with the prominent aim of producing comprehensible results. Since the aim of fuzzy technology has always been to model linguistic information and to achieve understandable solutions, we expect it to play an important role in information mining.

1. Introduction: A View of Information Mining

Due to modern information technology, which produces ever more powerful computers every year, it is possible today to collect, store, transfer, and combine huge amounts of data at very low costs. Thus an ever-increasing number of companies and scientific and governmental institutions can afford to build up large archives of documents and other data like numbers, tables, images, and sounds. However, exploiting the information contained in these archives in an intelligent way turns out to be fairly difficult. In contrast to the abundance of data there is a lack of tools that can transform these data into useful information and knowledge. Although a user often has a vague understanding of his data and their meaning—he can usually formulate hypotheses and guess dependencies—, he rarely knows

- where to find the "interesting" or "relevant" pieces of information,
- whether these pieces of information support his hypotheses and models,
- whether (other) interesting phenomena are hidden in the data,
- which methods are best suited to find the needed pieces of information in a fast and reliable way,
- how the data can be translated into human notions that are appropriate for the context in which they are needed.

In reply to these challenges a new area of research has emerged, which has been named "Knowledge Discovery in Databases" or "Data Mining". Although the standard definition of knowledge discovery and data mining [7] only speaks of discovery in *data*, thus not restricting the type and the organization of the data to work on, it has to be admitted that research up to now concentrated on highly structured data. Usually a minimal requirement is relational data. Most methods (e.g. classical methods like decision trees and neural networks) even demand as input a single uniform table, i.e., a set of tuples of attribute values. It is obvious, however, that this paradigm is hardly adequate for mining image or sound data or even textual descriptions, since it is inappropriate to see such data as, say, tuples of picture elements. Although such data can often be treated successfully by transforming them into structured tables using feature extraction, it is not hard to see that methods are needed which yield, for example, descriptions of what an image depicts, and other methods which can make use of such descriptions e.g. for retrieval purposes.

Another important point to be made is the following: The fact that pure neural networks are often seen as data mining methods, although their learning result (matrices of numbers) is hardly interpretable, shows that in contrast to the standard definition the goal of *understandable* patterns is often neglected. Of course, there are applications where comprehensible results are not needed and, for example, the prediction accuracy of a classifier is the only criterion of success. Therefore interpretable results should not be seen as a *conditio sine qua non*. However, our own experience—gathered in several cooperations with industry—is that modern technologies are accepted more readily, if the methods applied are easy to understand and the results can be checked against human intuition. In addition, if we want to gain insight into a domain, training, for instance, a neural network is not of much help.

Therefore we suggest to concentrate on *information mining*, which we see as an extension of data mining and which can be defined in analogy to the KDD definition given in [7] as follows:

> Information mining is the non-trivial process of identifying valid, novel, potentially useful, and *understandable* patterns in *heterogeneous information sources*.

The term *information* is thus meant to indicate two things: In the first place, it points out that the heterogeneous sources to mine can already provide *information*, understood as expert background knowledge, textual descriptions, images and sounds etc., and not only raw data. Secondly, it emphasizes that the results must be *comprehensible* ("must provide a user with information"), so that a user can check their plausibility and can get insight into the domain the data comes from.

For research this results in the challenges

- to develop theories and scalable techniques that can extract knowledge from large, dynamic, multi-relational, and multi-medial information sources,
- to close the semantic gap between structured data and human notions and concepts, i.e., to be able to translate computer representations into human notions and concepts and vice versa.

The goal of fuzzy systems has always been to model human expert knowledge and to produce systems that are easy to understand. Therefore we expect fuzzy systems technology to play a prominent role in the quest to meet these challenges. In the following we try to point out how fuzzy techniques can help to do information mining.

2. Strengths of Fuzzy Set Models

Although there is still some philosophical discussion going on whether a (symbolic) language is necessary for consciousness and thinking abilities, it is undisputed that language is a humans most effective tool to structure his experience and to model his environment. Therefore, in order to represent the background knowledge of human experts and to arrive at understandable data mining results, it is absolutely necessary to model linguistic terms and do what Zadeh so pointedly called *computing with words* [17].

A fundamental property of linguistic terms is their inherent vagueness, i.e., they have "fuzzy" boundaries: For each linguistic term there usually are some phenomena to which it can clearly be applied and some others, which can not be described using this term. But in between these phenomena there lies a "penumbra" of phenomena for which it is not definite whether the term is applicable or not. Well-known examples include the terms *pile of sand* (which is the basis of the classic *sorites* paradox) and *bald*. In both cases no precise number of hairs or grains of sand, respectively, can be given which separates the situations in which the terms are applicable from those in which they are not.

The reason for this inherent vagueness is that for practical purposes full precision is not necessary and may even be a waste of resources. To quote an example by Wittgenstein [16]: The request "Please stay around here!" is, of course, inexact. It would be more precise to draw a line on the ground, or, because the line has a certain width and thus would still not be fully exact, to use a color boundary. But this precision would be entirely pointless, since the inexact request can be expected to work absolutely fine.

Fuzzy set theory provides excellent means to model the "fuzzy" boundaries of linguistic terms by introducing gradual memberships. In contrast to classical set theory, in which an object or a case either is a member of a given set (defined, e.g., by some property) or not, fuzzy set theory makes it possible that an object or a case belongs to a set only to a certain degree, thus modeling the penumbra of the linguistic term describing the property that defines the set.

Interpretations of membership degrees include *similarity*, *preference*, and *uncertainty*: They can state how similar an object or case is to a prototypical one, they can indicate preferences between suboptimal solutions to a problem, or they can model uncertainty about the true situation, if this situation is described in imprecise terms. Drawing on Wittgenstein's example as an illustration, we may say that the locations "around here" are (for example, w.r.t. the person being in sight or calling distance) sufficiently similar to "here", so that the request works fine. Or we may say that it would be preferred, if the person stayed exactly "here", but some deviation from "here" would still be acceptable. Finally, if we tell someone to stay "around here" and then go away, we are uncertain about the exact location this person is in at a given moment. It is obvious that all of these interpretations are needed in applications and thus it is not surprising that they all have proven useful for solving practical problems. They also turned out to be worth considering when non-linguistic, but imprecise, i.e., set-valued information has to be modeled.

In general, due to their closeness to human reasoning, solutions obtained using fuzzy approaches are easy to understand and to apply. Due to these strengths, fuzzy systems are the method of choice, if linguistic, vague, or imprecise information has to be modeled.

3. Fuzzy Set Methods in Data Mining

The research in knowledge discovery in databases and data mining has lead to a large number of suggestions for a general model of the knowledge discovery process. A recent suggestion for such a model, which can be expected to have considerable impact, since it is backed by several large companies like NCR and DaimlerChrysler, is the CRISP-DM model (CRoss Industry Standard Process for Data Mining) [5].

The basic structure of this process model is depicted in figure 1. The circle indicates that data mining is essentially a circular process, in which the evaluation of the results can trigger a re-execution of the data prepa-

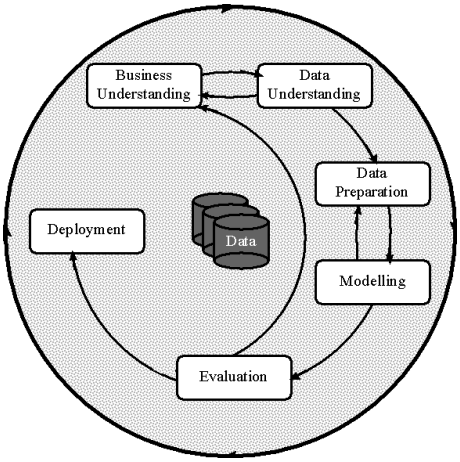

Figure 1: The CRISP-DM Model.

ration and model generation steps. In this process, fuzzy set methods can profitably be applied in several phases:

The *business understanding* and *data understanding* phases are usually strongly human centered and only little automation can be achieved here. These phases serve mainly to define the goals of the knowledge discovery project, to estimate its potential benefit, and to identify and collect the necessary data. In addition, background domain knowledge and meta-knowledge about the data is gathered. In these phases, fuzzy set methods can be used to formulate, for instance, the background domain knowledge in vague terms, but still in a form that can be used in a subsequent modeling phase. Furthermore, fuzzy database queries are useful to find the data needed and to check whether it may be useful to take additional, related data into account.

In the *data preparation* step, the gathered data is cleaned, transformed and maybe properly scaled to produce the input for the modeling techniques. In this step fuzzy methods may, for example, be used to detect outliers, e.g., by *fuzzy clustering* the data [3, 10] and then finding those data points that are far away from the cluster prototypes.

The *modeling* phase, in which models are constructed from the data in order, for instance, to predict future developments or to build classifiers, can, of course, benefit most from fuzzy data analysis approaches. These approaches can be divided into two classes. The first class, *fuzzy data* analysis [12], consists of approaches that analyze fuzzy data—data derived from imprecise measurement instruments or from the descriptions of human domain experts. An example from our own research is the induction of *possibilistic graphical models* [4] from data which complements the induction of the well-known probabilistic graphical models. The second class, fuzzy *data analysis* [1], consists of methods that use fuzzy techniques to structure and analyze crisp data, for instance, *fuzzy clustering* for data segmentation and rule generation and *neuro-fuzzy systems* for rule generation.

In the *evaluation* phase, in which the results are tested and their quality is assessed, the usefulness of fuzzy modeling methods becomes most obvious. Since they yield interpretable systems, they can easily be checked for plausibility against the intuition and expectations of human experts. In addition, the results can provide new insights into the domain under consideration, in contrast to, e.g., pure neural networks, which are black boxes.

To illustrate the usefulness of fuzzy data analysis approaches, in the following sections we discuss two topics in a little more detail: generating fuzzy rules from data and learning possibilistic graphical models.

4. Rule Generation and Information Fusion with Neuro-Fuzzy-Systems

In order to use fuzzy systems in data analysis, it must be possible to induce fuzzy rules from data. To describe a fuzzy system completely we need to determine a rule base (structure) and fuzzy partitions (parameters) for all variables. The data driven induction of fuzzy systems by simple heuristics based on local computations is usually called *neuro-fuzzy* [13]. If we apply such techniques, we must be aware of the trade-off between precision and interpretability. A fuzzy solution is not only judged for its accuracy, but also—if not especially—for its simplicity and readability. The user of the fuzzy system must be able to comprehend the rule base.

Important points for the interpretability of a fuzzy system are that

- there are only few fuzzy rules in the rule base,
- there are only few variables used in each rule,
- the variables are partitioned by few meaningful fuzzy set,
- no linguistic label is represented by more than one fuzzy set.

There are several ways to induce the structure of a fuzzy system. Cluster-oriented and hyperbox-oriented approaches to fuzzy rule learning create rules and fuzzy sets at the same time. Structure-oriented approaches need initial fuzzy partitions to create a rule base [14].

Cluster-oriented rule learning approaches are based on fuzzy cluster analysis [3, 10], i.e., the learning process is unsupervised. Hyperbox-oriented approaches use a supervised learning algorithm that tries to cover the training data by overlapping hyperboxes [2]. Fuzzy rules are created in both approaches by projection of clusters or hyperboxes. The main problem of both approaches is that each generated fuzzy rule uses individual membership functions and thus the rule base is hard to interpret. Cluster-oriented approaches additionally suffer from a loss of information and can only determine an

appropriate number of rules, if they are iterated with different fixed rule base sizes.

Structure-oriented approaches avoid all these drawbacks, because they do not search for (hyperellipsoidal or hyperrectangular) clusters in the data space. By providing (initial) fuzzy sets before fuzzy rules are created the data space is structured by a multidimensional fuzzy grid. A rule base is created by selecting those grid cells that contain data. This can be done in a single pass through the training data. This way of learning fuzzy rules was suggested in [15]. Extended versions were used in the neuro-fuzzy classification system NEFCLASS [13]. NEFCLASS uses a performances measure for the detected fuzzy rules. Thus the size of the rule base can be determined automatically by adding rules ordered by their performance until all training data is covered. The performance measure is also used to compute the best consequent for each rule.

The number of fuzzy rules can also be restricted by including only the best rules in the rule base. It is also possible to use pruning methods to reduce the number of rules and the number of variables used by the rules. In order to obtain meaningful fuzzy partitions, it is better to create rule bases by structure-oriented learning than by cluster-oriented or by hyperbox-oriented rule learning. The latter two approaches create individual fuzzy sets for each rule and thus provide less interpretable solutions. Structure-oriented methods allow the user to provide appropriate fuzzy partitions in advance such that all rules share the same fuzzy sets. Thus the induced rule base can be interpreted well.

After the rule base of a fuzzy system has been generated, we must usually train the membership function in order to improve the performance. In NEFCLASS, for example, the fuzzy sets are tuned by a simple backpropagation-like procedure. The algorithm does not use gradient-descent, because the degree of fulfillment of a fuzzy rule is determined by the minimum and non-continuous membership functions may be used. Instead a simple heuristics is used that results in shifting the fuzzy sets and in enlarging or reducing their support.

The main idea of NEFCLASS is to create comprehensible fuzzy classifiers, by ensuring that fuzzy sets cannot be modified arbitrarily during learning. Constraints can be applied in order to make sure that the fuzzy sets still fit their linguistic labels after learning. For the sake of interpretability we do not want adjacent fuzzy sets to exchange positions, we want the fuzzy sets to overlap appropriately, etc.

The most recent JAVA implementation of the NEFCLASS approach to generate fuzzy classifiers from data has the following features:

- structure-oriented fuzzy rule learning,
- automatic determination of the number of rules,
- treatment of missing values (without imputation),
- the ability to use data with both numeric *and* symbolic attributes,
- constrained fuzzy set learning, and
- automatic pruning strategies.

The tool is called NEFCLASS-J and can be obtained at `http://fuzzy.cs.uni-magdeburg.de`.

If neuro-fuzzy methods are used in information mining, it is useful to consider their capabilities in fusing information from different sources. Information fusion refers to the acquisition, processing, and merging of information originating from multiple sources to provide a better insight and understanding of the phenomena under consideration. There are several levels of information fusion. Fusion may take place at the level of data acquisition, data pre-processing, data or knowledge representation, or at the model or decision making level. On lower levels where raw data is involved, the term (sensor) *data fusion* is preferred. Some aspects of information fusion can be implemented by NEFCLASS. For a conceptual and comparative study of fusion strategies in various calculi of uncertainty see [9, 6].

If a fuzzy classifier is created based on a supervised learning problem $\tilde{\mathcal{L}}$, then the most common way is to provide a data set, where each pattern is labeled—ideally with its correct class. That is, we assume that each pattern belongs to one class only. Sometimes it is not possible to determine this class correctly due to a lack of information. Instead of a crisp classification it would also be possible to label each pattern with a vector of membership degrees. This requires that a vague classification is obtained in some way for the training patterns, e.g. by partially contradicting expert opinions.

Training patterns with fuzzy classifications are one way to implement information fusion with neuro-fuzzy systems. If we assume that a group of n experts provide partially contradicting classifications for a set of training data we can fuse the expert opinions into fuzzy sets that describe the classification for each training pattern. According to the context model, we can view the experts as different observation contexts [11]. The training then reflects fusion of expert opinions on the data set level. Due to the capabilities of its learning algorithms NEFCLASS can handle such training data in the process of creating a fuzzy classifier.

Another aspect of information fusion that is implemented by NEFCLASS is to integrate expert knowledge in form of fuzzy rules and information obtained from data. If prior knowledge about the classification problem is available, then the rule base of the fuzzy classifier can be initialized with suitable fuzzy rules before rule learning is invoked to complete the rule base. If the algorithm creates a rule from data that contradicts with an expert rule then we can

- always prefer the expert rule,
- always prefer the learned rule, or

- select the rule with the higher performance value.

In NEFCLASS we determine the performance of all rules over the training data and in case of contradiction the better rule prevails. This reflects fusion of expert opinions and observations.

Because NEFCLASS is able to resolve conflicts between rules based on rule performance, it is also able to fuse expert opinions on the fuzzy rule level. Rule bases from different experts can be entered as prior knowledge. They will be fused into one rule base and contradictions are resolved automatically by deleting from each pair of contradicting rules the rule with lower performance.

After all contradictions between expert rules and rules learned from data were resolved, usually not all rules can be included into the rule base, because its size is limited by some criterion. In this case we must decide whether

- to include expert rules in any case, or
- to include rules by descending performances values.

The decision depends on the trust we have in the experts knowledge and in the training data. A mixed approach can be used, e.g. include the best expert rules and then use the best learned rules to complete the rule base.

A similar decision must be made, when the rule base is pruned after training, i.e. is it acceptable to remove an expert rule during pruning, or must such rules remain in the rule base. In NEFCLASS expert rules and rules induced from data are not treated differently.

5. Dependency Analysis with Possibilistic Graphical Models

Since reasoning in multi-dimensional domains tends to be infeasible in the domains as a whole—and the more so, if uncertainty and imprecision are involved—decomposition techniques, that reduce the reasoning process to computations in lower-dimensional subspaces, have become very popular. In the field of graphical modeling, *decomposition* is based on dependence and independence relations between the attributes or variables that are used to describe the domain under consideration. The structure of these dependence and independence relations are represented as a graph (hence the name graphical models), in which each node stands for an attribute and each edge for a direct dependence between two attributes. The precise set of dependence and (conditional) independence statements that hold in the modeled domain can be read from the graph using simple graph theoretic criteria, for instance, d-separation, if the graph is a directed one, or simple separation, if the graph is undirected.

The conditional independence graph (as it is also called) is, however, only the *qualitative* or *structural component* of a graphical model. To do reasoning, it has to be enhanced by a *quantitative component* that provides confidence information about the different points of the underlying domain. This information can often be represented as a distribution function on the underlying domain, for example, a probability distribution, a possibility distribution, a mass distribution etc. W.r.t. this quantitative component, the conditional independence graph describes a *factorization* of the distribution function on the domain as a whole into conditional or marginal distribution functions on lower-dimensional subspaces.

Graphical models make reasoning much more efficient, because propagating the evidential information about the values of some attributes to the unobserved ones and computing the marginal distributions for the unobserved attributes can be implemented by locally communicating node and edge processors in the conditional independence graph.

For some time the standard approach to construct a graphical model has been to let a human domain expert specify the dependency structure of the considered domain. This provided the conditional independence graph. Then the human domain expert had to estimate the necessary conditional or marginal distribution functions, which then formed the quantitative component of the graphical model. This approach, however, can be tedious and time consuming, especially, if the domain under consideration is large. In addition, it may be impossible to carry it out, if no or only vague knowledge is available about the dependence and independence relations that hold in the domain to be modeled. Therefore recent research has concentrated on learning graphical models from databases of sample cases.

Due to the origin of graphical modeling research in probabilistic reasoning, the most widely known methods are, of course, learning algorithms for Bayesian or Markov networks. However, these approaches—as probabilistic approaches do in general—suffer from certain deficiencies, if imprecise information, understood as set-valued data, has to be taken into account. For this reason recently possibilistic graphical models also gained some attention [4], for which learning algorithms have been developed in analogy to the probabilistic case. These methods can be used to do dependency analysis, even if the data to analyze is highly imprecise and thus offer interesting perspectives for future research.

We have implemented these methods as a plug-in for the well-known data mining tool *Clementine* (ISL/SPSS). Its probabilistic version is currently used at DaimlerChrysler for fault analysis.

6. Concluding Remarks

In knowledge discovery and data mining as it is, there is a tendency to focus on purely data-driven approaches

in a first step. More model-based approaches are only used in the refinement phases (which in industry are often not necessary, because the first successful approach wins—and the winner takes all). However, to arrive at truly useful results, we must take background knowledge and, in general, non-numeric information into account and we must concentrate on comprehensible models.

The complexity of the learning task, obviously, leads to a problem: When learning from information, one must choose between (often quantitative) methods that achieve good performance and (often qualitative) models that explain what is going on to a user. This is another good example of Zadeh's principle of the incompatibility between precision and meaning. Of course, precision and high performance are important goals. However, in the most successful fuzzy applications in industry such as intelligent control and pattern classification, the introduction of fuzzy sets was motivated by the need for more human-friendly computerized devices that help a user to formulate his knowledge and to clarify, to process, to retrieve, and to exploit the available information in a most simple way. In order to achieve this user-friendliness, often certain (limited) reductions in performance and solution quality are accepted.

So the question is: What is a good solution from the point of view of a user in the field of information mining? Of course, correctness, completeness, and efficiency are important, but in order to manage systems that are more and more complex, there is a constantly growing demand to keep the solutions conceptually simple and understandable. This calls for a formal theory of utility in which the simplicity of a system is taken into account. Unfortunately such a theory is extremely hard to come by, because for complex domains it is difficult to measure the degree of simplicity and it is even more difficult to assess the gain achieved by making a system simpler. Nevertheless, this is a lasting challenge for the fuzzy community to meet.

References

[1] H. Bandemer and W. Näther. *Fuzzy Data Analysis.* Kluwer, Dordrecht, Netherlands 1992

[2] M. Berthold and K.P. Huber. Constructing Fuzzy Graphs from Examples. *Int. J. Intelligent Data Analysis* 3, 1999. An electronic journal available from `http://www.elsevier.com/locate/ida`

[3] J.C. Bezdek, J. Keller, R. Krishnapuram, and N. Pal. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing.* Series: The Handbooks on Fuzzy Sets. Kluwer, Norwell, MA, USA 1998

[4] C. Borgelt, J. Gebhardt, and R. Kruse. Chapter F1.2: Inference Methods. In: E. Ruspini, P. Bonissone, and W. Pedrycz, eds. *Handbook of Fuzzy Computation.* Institute of Physics Publishing Ltd., Bristol, United Kingdom 1998

[5] P. Chapman, J. Clinton, T. Khabaza, T. Reinartz, and R. Wirth. *The CRISP-DM Process Model.* available from `http://www.ncr.dk/CRISP/`

[6] D. Dubois, H. Prade, and R.R. Yager. Merging Fuzzy Information. In: J.C. Bezdek, D. Dubois, and H. Prade, eds. *Approximate Reasoning and Fuzzy Information Systems, (Series: Handbook of Fuzzy Sets)*, 335–402. Kluwer, Dordrecht, Netherlands 1999

[7] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds. *Advances in Knowledge Discovery and Data Mining.* AAAI Press / MIT Press, Cambridge, MA, USA 1996

[8] J. Gebhardt, M.A. Gil, and R. Kruse. Fuzzy Set Theoretic Methods in Statistics. In: R. Slowinski (Hrsg.): *Fuzzy Sets in Decision Analysis, Operations Research and Statistics*, 311–347. The Handbook of Fuzzy Sets Series, Kluwer, Boston 1998

[9] J. Gebhardt and R. Kruse. Parallel Combination of Information Sources. In: D. Gabbay and P. Smets, eds. *Handbook of Defeasible Reasoning and Uncertainty Management Systems, Vol. 3: Belief Change*, 329–375. Kluwer, Dordrecht, Netherlands 1998

[10] F. Höppner, F. Klawonn, R. Kruse, and T. Runkler. *Fuzzy Cluster Analysis.* J. Wiley & Sons, Chichester, England 1999

[11] R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of Fuzzy Systems.* J. Wiley & Sons, Chichester, England 1994

[12] R. Kruse and K.D. Meyer. *Statistics with Vague Data.* Reidel, Dordrecht, Netherlands 1987

[13] D. Nauck, F. Klawonn, and R. Kruse. *Foundations of Neuro-Fuzzy Systems.* J. Wiley & Sons, Chichester, England 1997

[14] D. Nauck and R. Kruse. Chapter D.2: Neuro-fuzzy Systems. In: E. Ruspini, P. Bonissone, and W. Pedrycz, eds. *Handbook of Fuzzy Computation.* Institute of Physics Publishing Ltd., Bristol, United Kingdom 1998

[15] L.-X. Wang and J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Trans. Syst., Man, Cybern.* 22:1414–1227. IEEE Press, Piscataway, NJ, USA 1992

[16] L. Wittgenstein. *Philosophical Investigations.* Prentice Hall, Englewood Cliffs, NJ, USA 1973 (first published 1953)

[17] L.A. Zadeh. Fuzzy Logic = Computing With Words. *IEEE Transactions on Fuzzy Systems* 4:103–111. IEEE Press, Piscataway, NJ, USA 1996