To appear in Fuzzy Sets and Systems

Fuzzy Methods in Machine Learning and Data Mining: Status and Prospects

Eyke Hüllermeier

University of Magdeburg, Faculty of Computer Science
Universitätsplatz 2, 39106 Magdeburg, Germany
eyke.huellermeier@iti.cs.uni-magdeburg.de

Abstract

Over the past years, methods for the automated induction of models and the extraction of interesting patterns from empirical data have attracted considerable attention in the fuzzy set community. This paper briefly reviews some typical applications and highlights potential contributions that fuzzy set theory can make to machine learning, data mining, and related fields. The paper concludes with a critical consideration of recent developments and some suggestions for future research directions.

1 Introduction

Aspects of knowledge representation and reasoning have dominated research in fuzzy set theory (FST) for a long time, at least in that part of the theory which lends itself to intelligent systems design and applications in artificial intelligence (AI). Yet, problems of automated learning and knowledge acquisition have more and more come to the fore during the recent years. This is not very surprising in view of the fact that the "knowledge acquisition bottleneck" seems to remain one of the key problems in the design of intelligent and knowledge-based systems. Indeed, experience has shown that a purely knowledge-driven approach, which aims at formalizing problem-relevant human expert knowledge, is difficult, intricate, and tedious. More often than not, it does not even lead to fully satisfying results. Consequently, a kind of *data-driven* adaptation of fuzzy systems is often worthwhile. In fact, such a "tuning" even suggests itself since, in many applications, data is readily available. Indeed, recent research has shown that the traditional knowledge-driven approach can be complemented by a data-driven one in a reasonable way. In the extreme case, the former is even completely replaced by the latter. For example, several approaches in which fuzzy models (e.g. fuzzy rule bases) are learned from data in a fully automated way have already been developed [4].

In addition to this internal shift within fuzzy systems research, an external development has further amplified the aforementioned trends. This development is is the great interest that the field of *knowledge discovery in databases* (KDD) has attracted in diverse research communities in recent years. As a response to the progress in digital data acquisition and storage technology, along with the limited human capabilities in analyzing and exploiting large amounts of data, this field has recently emerged as a new research discipline, lying at the intersection of statistics, machine learning, data management, and other areas. According to a widely accepted definition, KDD refers to the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable structure in data [27]. The central step within the overall KDD process is *data mining*, the application of computational techniques to the task of finding patterns and models in data. Meanwhile, KDD has established itself as a new, independent research field, including its own journals and conferences.

The aim of this paper is to convey an impression of the current status and prospects of FST in machine learning, data mining, and related fields. After a brief introduction to these fields (section 2), we present a collection of typical applications of FST (section 3). The examples are representative but are not intended to provide a comprehensive review of the literature. In section 4, we try to highlight in a more systematic way the potential contributions that FST can make to machine learning and data mining. Finally, we conclude with a critical consideration of recent developments and some suggestions for future research directions in section 5.

2 Machine Learning, Data Mining, and Related Fields

The automated learning of models from empirical data is a central theme in several research disciplines, ranging from classical (inferential) statistics to more recent fields such as machine learning. Model induction may serve different purposes, such as accurate *prediction* of future observations or intelligible *description* of dependencies between variables in the domain under investigation, among other things. Typically, a model induction process involves the following steps:

- data acquisition

- data preparation (cleaning, transforming, selecting, scaling, ...)
- model induction
- model interpretation and validation
- model application

A common distinction of performance tasks in empirical[1] machine learning is supervised learning (e.g. classification and regression), unsupervised learning (e.g. clustering) and reinforcement learning. Throughout the paper, we shall focus on the first two performance tasks that have attracted much more attention in the FST community than the latter one.

In unsupervised learning, the learning algorithm is simply provided with a set of data. The latter typically consists of data points $z \in \mathcal{Z}$, where $\mathcal{Z}$ is the Cartesian product of the domains of a fixed set of attributes. That is, an observation z is described in terms of a feature vector. Roughly speaking, the goal in unsupervised learning is to discover any kind of structure in the data, such as properties of the distribution, relationships between data entities, or dependencies between attributes. This includes, e.g., nonparametric features such as modes, gaps, or clusters in the data (section 3.1), as well as interesting patterns like those discovered in association analysis (section 3.4).

The setting of supervised learning proceeds from a predefined division of the data space into an input space $\mathcal{X}$ and an output space $\mathcal{Y}$. Assuming a dependency between the input attributes and the output, the former is considered as the *predictive* part of an instance description (like the regressor variables in regression analysis), whereas the latter corresponds to the target to be predicted (e.g. the dependent variable in regression). The learning algorithm is provided with a set of labeled examples $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Again, the inputs x are typically feature vectors. A distinction between different types of performance tasks is made according to the structure of the output space $\mathcal{Y}$. Even though problems involving output spaces of a richer structure have been considered recently (e.g. so-called ranking problems [33,30]), $\mathcal{Y}$ is typically a one-dimensional space. In particular, the output is a categorical attribute (i.e., $\mathcal{Y}$ is a nominal scale) in *classification*. Here, the goal is to generalize beyond the examples given by inducing a model that represents a complete mapping from the input space to the output space (a hypothetical classification function). The model itself can be represented by means of different formalisms such as, e.g., threshold concepts or logical conjunctions. In *regression*, the output is a numerical variable, hence the goal is to induce a real-valued mapping

[1] Here, *empirical* learning is used as an antonym to *analytical* learning. Roughly speaking, analytical learning systems do not require external inputs, whereas such inputs are essential for empirical learning systems. An example of analytical learning is speedup learning.

$\mathcal{X} \rightarrow \mathcal{Y}$ that approximates an underlying (functional or probabilistic) relation between $\mathcal{X}$ and $\mathcal{Y}$ well in a specific sense. So-called *ordinal regression* is in-between regression and classification: the output is measured on an ordinal scale.

As can be seen, supervised machine learning puts special emphasis on induction as a performance task. Moreover, apart from the *efficiency* of the induced model, the *predictive accuracy* of that model is the most important quality criterion. The latter refers to the ability to make accurate predictions of outputs for so far unseen inputs. The predictive accuracy of a model $h : \mathcal{X} \rightarrow \mathcal{Y}$ is typically measured in terms of the *expected loss*, i.e., the expected value of $\ell(y, h(x))$, where $\ell(\cdot)$ is a loss function $\mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ (and (x, y) an example drawn at random according to an underlying probability measure over $\mathcal{X} \times \mathcal{Y}$.[2])

Data mining has a somewhat different focus. Here, other aspects such as, e.g., the *understandability*, gain in importance. In fact, the goal in data mining is not necessarily to induce *global* models of the system under consideration (e.g. in the form of a functional relation between input and output variables) or to recover some underlying data generating process, but rather to discover *local* patterns of interest, e.g. very frequent (hence typical) or very rare (hence atypical) events. Data mining is of a more explanatory nature, and patterns discovered in a data set are usually of a *descriptive* rather than of a *predictive* nature. Data Mining also puts special emphasis on the analysis of very large data sets and, hence, on aspects of scalability and efficiency.

Despite these slightly different goals, the typical KDD process has much in common with the process of inductive reasoning as outlined above, except for the fact that the former can be (and indeed often is) circular in the sense that the data mining results will retroact on the acquisition, selection, and preparation of the data, possibly initiating a repeated pass with modified data, analysis tools, or queries. A typical KDD process may comprise the following steps:

- data cleaning
- data integration (combination of multiple sources)
- data selection
- data transformation (into a form suitable for the analysis)
- data mining
- evaluation of patterns
- knowledge presentation

[2] Since this measure is normally unknown, the expected loss is approximated by the *empirical* loss in practice, i.e., the average loss on a test data set.

Recently, the interest in data mining has shifted from the analysis of large but homogeneous data sets (relational tables) to the analysis of more complex and heterogeneous information sources, such as e.g. texts, images, audio and video data, and the term *information mining* has been coined to describe a KDD process focused on this type of information sources [43].

There are several other fields that are closely related to machine learning and data mining such as, e.g., classical statistics and various forms of data analysis (distinguished by adjectives like multivariate, exploratory, Bayesian, intelligent, ...) Needless to say, it is impossible to set a clear boundary between these fields. Subsequently, we shall simply subsume them under the heading "machine learning and data mining" (ML&DM), [3] understood in a wide sense as the application of computational methods and algorithms for extracting models and patterns from potentially very large data sets.

3 Typical Applications of Fuzzy Set Theory

The tools and technologies that have been developed in FST have the potential to support all of the steps that comprise a process of model induction or knowledge discovery. In particular, FST can already be used in the data selection and preparation phase, e.g., for modeling vague data in terms of fuzzy sets [60], to "condense" several crisp observations into a single fuzzy one, or to create fuzzy summaries of the data [44]. As the data to be analyzed thus becomes fuzzy, one subsequently faces a problem of *fuzzy data* analysis [5].

The problem of analyzing fuzzy data can be approached in at least two principally different ways. First, standard methods of data analysis can be extended in a rather generic way by means of an extension principle. For example, the functional relation between the data points and the coefficients of a linear regression function can be extended to the case of fuzzy data, where the observations are described in terms of fuzzy sets. Thus, the coefficients become fuzzy as well. A second, often more sophisticated approach is based on embedding the data into more complex mathematical spaces, such as fuzzy metric spaces [18], and to carry out data analysis in these spaces [19].

If fuzzy methods are not used in the data preparation phase, they can still be

[3] Our distinction between machine learning and data mining can roughly be seen as a "modern" or extended distinction between descriptive and inductive statistics. We note, however, that this view is not an opinio communis. For example, some people have an even more general view of data mining that includes machine learning as a special case.

employed in a later stage in order to analyze the original data. Thus, it is not the data to be analyzed that is fuzzy, but rather the methods used for analyzing the data (in the sense of resorting to tools from FST). In the following, we shall focus on this type of fuzzy data analysis (where the adjective "fuzzy" refers to the term *analysis*, not to the term *data*), which is predominant in ML&DM. In fact, our brief review of typical applications of FST (including possibility theory) is more oriented toward these fields than toward methods from classical statistics and multivariate data analysis (such as linear regression).

3.1 Fuzzy Cluster Analysis

Clustering methods are among the most important unsupervised learning techniques. In data analysis, they are often routinely applied as one of the first steps in order to convey a rough idea of the structure of a data set. Clustering refers to the process of grouping a collection of objects into classes or "clusters" such that objects within the same class are *similar* in a certain sense, and objects from different classes are dissimilar. In addition, the goal is sometimes to arrange the clusters into a natural hierarchy (hierarchical clustering). Also, cluster analysis can be used as a form of descriptive statistics, showing whether or not the data consists of a set of distinct subgroups.

As input, clustering algorithms typically assume information about the (pairwise) similarity between objects e.g. in the form of a *proximity matrix*. Usually, objects are described in terms of a set of measurements from which similarity degrees between pairs of objects are derived, using a kind of similarity or distance measure. There are basically three types of clustering algorithms: *Mixture modeling* assumes an underlying probabilistic model, namely that the data were generated by a probability density function, which is a mixture of component density functions. *Combinatorial algorithms* do not assume such a model. Instead, they proceed from an objective function to be maximized and approach the problem of clustering as one of combinatorial optimization. So called *mode-seekers* are somewhat similar to mixture models. However, they take a non-parametric perspective and try to estimate modes of the component density functions directly. Clusters are then formed by looking at the closeness of the objects to these modes that serve as cluster centers. The popular K-means algorithm is maybe the most popular representative of this class.

In standard clustering, each object is assigned to one cluster in an unequivocal way. Consequently, the individual clusters are separated by sharp boundaries. In practice, such boundaries are often not very natural or even counterintuitive. Rather, the boundary of single clusters and the transition between different clusters are usually “smooth” rather than abrupt. This is the main motivation underlying fuzzy extensions to clustering algorithms [34]. In fuzzy clustering an object may belong to different clusters at the same time, at least to some extent, and the degree to which it belongs to a particular cluster is expressed in terms of a membership degree. The membership functions of the different clusters (defined on the set of observed points) is usually assumed to form a partition of unity. This version, often called probabilistic clustering, can be generalized further by weakening this constraint: In possibilistic clustering, the sum of membership degrees is constrained to be *at least* 1 [42]. Fuzzy clustering has proved to be extremely useful in practice and is now routinely applied also outside the fuzzy community (e.g. in recent bioinformatics applications [31]).

3.2 Learning Fuzzy Rule Bases

The most frequent application of FST in machine learning is the induction or the adaptation of rule-based models. This is hardly astonishing, since rule-based models have always been a cornerstone of fuzzy systems and a central aspect of research in the field, not only in ML&DM but also in many other subfields, notably approximate reasoning and fuzzy control. (The terms *fuzzy system* and *fuzzy rule base* are sometimes even used synonymously.)

Fuzzy rule bases can represent both classification and regression functions, and different types of fuzzy models have been used for these purposes. In order to realize a regression function, a fuzzy system is usually wrapped in a “fuzzifier” and a “defuzzifier”: The former maps a crisp input to a fuzzy one, which is then processed by the fuzzy system, and the latter maps the (fuzzy) output of the fuzzy system back to a crisp value. For so-called Takagi-Sugeno models, which are quite popular for modeling regression functions, the defuzzification step is unnecessary, since these models output crisp values directly.

In the case of classification, the consequent of single rules such as

$$\text{IF } (\texttt{size} \in \texttt{TALL}) \text{ AND } (\texttt{weight} \in \texttt{LIGHT}) \text{ THEN } (\texttt{class} = A)$$

is usually a class assignment (i.e. a singleton fuzzy set).[4] Evaluating a rule base (à la Mamdani-Assilan) thus becomes trivial and simply amounts to “maximum matching”, that is, searching the maximally supporting rule for each class. Thus, much of the appealing interpolation and approximation properties of fuzzy inference gets lost, and fuzziness only means that rules can be

[4] More generally, a rule consequent can suggest different classes with different degrees of certainty.

activated to a certain degree. There are, however, alternative methods which combine the predictions of several rules into a classification of the query [12].

A plethora of strategies has been developed for inducing a fuzzy rule base from the data given. Refraining from a detailed exposition, we only point out one chief difference between these strategies. This difference concerns the way in which individual rules or their condition parts are learned. One possibility is to (successively) identify regions in the input space that seem to be qualified to form the (the extension of) a condition part of a rule. This can be done by looking for clusters using clustering algorithms (see above), or by identifying hyperboxes in the manner of so-called *covering* (separate and conquer) algorithms [29]. By projecting the regions thus obtained onto the various dimensions of the input space, rule antecedents of the form "$X \in A$" are obtained, where X is an individual attribute and A is a fuzzy set (the projection of the fuzzy region). The condition part of the rule is then given by the conjunction of these antecedents. This approach is relatively flexible, though it suffers from the disadvantage that each rule makes use of its own fuzzy sets. Thus, the complete rule base might be difficult to interpret.

An alternative is to proceed from a fixed fuzzy partition for each attribute, i.e., a regular "fuzzy grid" of the input space, and to consider each cell of this grid as a potential antecedent part of a rule [61]. This approach is advantageous from an interpretability point of view. On the other hand, it is less flexible and may produce inaccurate models when the one-dimensional partitions define a multi-dimensional grid that does not reflect the structure of the data.

Especially important in the field of fuzzy rule learning are hybrid methods that combine FST with other methodologies, notably evolutionary algorithms and neural networks. For example, evolutionary algorithms are often used in order to optimize ("tune") a fuzzy rule base or for searching the space of potential rule bases in a (more or less) systematic way [13]. Quite interesting are also *neuro-fuzzy* methods [47]. For example, one idea is to encode a fuzzy system as a neural network and to apply standard methods (like backpropagation) in order train such a network. This way, neuro-fuzzy systems combine the representational advantages of fuzzy systems with the flexibility and adaptivity of neural networks.

3.3 Fuzzy Decision Tree Induction

Decision tree induction, well-known examples of which include the ID3 algorithm [50] and its successor C4.5 and C5.0 [51] as well as the CART system [8], are among the most widely applied supervised machine learning techniques.

The basic principle underlying most decision tree learners is that of partitioning the set of given examples in a recursive manner until the subsets of examples thus obtained are “homogeneous” enough (with respect to an output attribute, which is typically a class label). Each inner node of a decision tree defines a partition of the examples assigned to that node. This is done by classifying elements according to the value of a specific input attribute. The latter is selected according to a measure of effectiveness in improving the homogeneity of the resulting partition, thereby supporting the overall objective of constructing a small (simple) tree. The recursive partitioning process terminates if a stopping condition is satisfied, in which case the node becomes a leaf of the tree.

Once the decision tree has been constructed, each path can be considered as a rule. The antecedent part of a rule is a conjunction of constraints on attribute values and the conclusion part determines a value for the class variable. New examples are then classified on the basis of these rules, i.e., by looking at the class label of the leaf node the attribute values of which match the description of the example. Decision tree induction can thus be seen as a special form of rule induction, where the potential rule bases are restricted to those having a tree-like hierarchical structure. Since decision trees are derived in a top-down fashion by means of a heuristic (greedy) strategy, the class of potential models is searched in a rather efficient way, though optimality cannot be guaranteed.

Fuzzy variants of decision tree induction have been developed for quite a while (e.g. [62,40]) and seem to remain a topic of interest even today (see [48] for a recent approach and a comprehensive overview of research in this field). In fact, these approaches provide a typical example for the “fuzzification” of standard machine learning methods. In the case of decision trees, it is primarily the “crisp” thresholds used for defining splitting predicates (constraints), such as e.g. $\texttt{size} \leq 181$, at inner nodes that have been criticized: Such thresholds lead to hard decision boundaries in the input space, which means that a slight variation of an attribute (e.g. $\texttt{size} = 182$ instead of $\texttt{size} = 181$) can entail a completely different classification of an object (e.g. of a person characterized by size, weight, gender, ...) Moreover, the learning process becomes unstable in the sense that a slight variation of the training examples can change the induced decision tree drastically.[5]

In order to make the decision boundaries “soft”, an obvious is idea to apply fuzzy predicates at the inner nodes of a decision tree, such as e.g. $\texttt{size} \in \texttt{TALL}$, where $\texttt{TALL}$ is a fuzzy set (rather than an interval). In other words, a fuzzy

[5] Decision tree induction is known to have high variance, a property that has recently been exploited in connection with ensemble learning techniques.

partition instead of a crisp one is used for the splitting attribute (here `size`) at an inner node. Since an example can satisfy a fuzzy predicate to a certain degree, the examples are partitioned in a fuzzy manner as well. That is, an object is not assigned to exactly one successor node in a unique way, but perhaps to several successors with a certain degree. For example, a person whose size is 181 cm could be an element of the `TALL`-group to the degree, say, 0.7 and of the complementary group to the degree 0.3.

The above idea of "soft recursive partitioning" has been realized in different ways. Moreover, the problems entailed by corresponding fuzzy extensions have been investigated. For example, how can splitting measures like entropy, originally defined for ordinary sets of examples, be extended to fuzzy sets of examples [14]? Or, how can a new object be classified by a fuzzy decision tree?

3.4 Fuzzy Association Analysis

As mentioned above, methods for learning dependencies in the data that are expressed in terms of (IF–THEN) rules are quite popular. Rules of this form are also extracted from data in so-called association analysis, by now one of the most frequently applied data mining techniques. As an important difference between association rules and rule-based models for classification and regression, we note that the former are of a *descriptive* nature. Moreover, association rules are typically considered in isolation, i.e., as a particular type of local pattern [6], and not as an integral part of a rule base.

Association rules provide a means for representing dependencies between attribute values of objects (data records) stored in a database. [7] Typically, an association involves two sets of binary attributes (features), $\mathcal{A}$ and $\mathcal{B}$. Then, the intended meaning of a rule "IF $\mathcal{A}$ THEN $\mathcal{B}$" is that an object having all the features in $\mathcal{A}$ is likely to have all the features in $\mathcal{B}$ as well. In order to decide whether a potential association is interesting or at least well-supported by the data, it is evaluated by several quality measures. Standard measures include the *support* of a rule, i.e. the number of objects in the database satisfying both the condition $\mathcal{A}$ and the conclusion $\mathcal{B}$, and its *confidence*, i.e. the relative frequency of objects satisfying $\mathcal{B}$ among those satisfying $\mathcal{A}$. Then, the problem is to find all associations the support and confidence of which is

[6] Even though dependencies between association rules are of course important. For example, if one rule is more general than another one, only the first one should be reported to the user of a data mining system.

[7] In a sense, association analysis can be seen as an extension of statistical correlation analysis.

above a (user-specified) threshold. Since the number of candidate rules grows exponentially with the number of attributes, this problem is computationally complex [1,54].

Association rules of the above type are often employed in the context of market-basket analysis, where an object is a purchase and features are associated with products or items. In this context, the association

$$\text{IF } \{\texttt{paper}, \texttt{envelopes}\} \text{ THEN } \{\texttt{stamps}\}$$

suggests, for example, that a purchase containing paper and envelopes is likely to contain stamps as well.

A generalization of binary association rules is motivated by the fact that a database is usually not restricted to binary attributes but also contains attributes with values ranging on (completely) ordered scales, such as numerical or ordered categorical attributes. In *quantitative association analysis*, rule antecedents and consequents are specified in terms of subsets of attribute values, typically in the form of intervals. For example, “Employees at the age of 30 to 40 have incomes between \$50,000 and \$70,000”.

The use of fuzzy sets in connection with association analysis has been proposed by numerous authors (see [10,16] for recent overviews), with motivations closely resembling those in the case of rule learning and decision tree induction. Again, by allowing for “soft” rather than crisp boundaries of intervals, fuzzy sets can avoid certain undesirable threshold effects [57], this time concerning the quality measures of association rules rather than the classification of objects. Moreover, identifying fuzzy sets with linguistic terms allows for a comprehensible and user-friendly presentation of rules discovered in a database. Example: “Middle-aged employees receive considerable incomes.”

Many standard techniques for association rule mining have been transferred to the fuzzy case, sometimes in a rather ad-hoc manner. Indeed, publications on this topic are often more concerned with issues of data preprocessing, e.g. the problem of finding good fuzzy partitions for the quantitative attributes, rather than the rule mining process itself. Still, more theoretically-oriented research has recently been started [23]. For example, the existence of different types of fuzzy rules [24] suggests that fuzzy associations might be interpreted in different ways and, hence, that the evaluation of an association cannot be independent of its interpretation. In particular, one can raise the question which generalized logical operators can reasonably be applied in order to evaluate fuzzy associations, e.g, whether the antecedent part and the condition part should be combined in a conjunctive way (à la Mamdani rules) or by means of a generalized implication (as in implication-based fuzzy rules) [35]. Moreover, since standard evaluation measures for association rules can be generalized in many ways, it is interesting to investigate properties of particular generalizations and to look for an axiomatic basis that supports the choice of specific measures [23].

3.5 Fuzzy Methods in Case-Based Learning

Case-based or instance-based learning algorithms have been applied successfully in fields such as machine learning and pattern recognition during the recent years [3,15]. The case-based learning (CBL) paradigm is also of central importance in case-based reasoning (CBR), a problem solving methodology that goes beyond the standard prediction problems of classification and regression that are usually considered in machine learning [52,41].

As the term suggests, in CBL special importance is attached to the concept of a *case*. A case or an instance can be thought of as a single experience, such as a pattern (along with its classification) in pattern recognition or a problem (along with a solution) in CBR. Rather than inducing a global model (theory) from the data and using this model for further reasoning, as inductive, model-based machine learning methods typically do, CBL systems simply store the data itself. The processing of the data is deferred until a prediction (or some other type of query) is actually requested, a property that qualifies CBL as a *lazy* learning method [2]. Predictions are then derived by combining the information provided by the stored examples, primarily by those cases that are *similar* to the new query. In fact, the major assumption underlying CBL is a commonsense principle suggesting that “similar problems have similar solutions”. This “similarity hypothesis” serves as a basic inference paradigm in various domains of application. For example, in a classification context, it translates into the assertion that “similar objects have similar class labels”.

Similarity-based inference has also been a topic of interest in FST, which is hardly astonishing since similarity is one of the main semantics of fuzzy membership degrees [53,56]. Along these lines, a close connection between case-based learning and fuzzy rule-based reasoning has been established in [21,22]. Here, the aforementioned “similarity hypothesis” has been formalized within the framework of fuzzy rules. More precisely, each observed case, consisting of a problem x (e.g. an object) and an associated solution y (e.g. a class label), is encoded in terms of a so-called *possibility rule*: If a query x_0 is similar to x, then its solution y_0 is *possibly* similar to y. On the basis of such rules, case-based inference can be realized as a special type of fuzzy set-based approximate reasoning. Note that this “possibilistic” variant of the similarity hypothesis takes the heuristic and, hence, uncertain character of similarity-based inference into consideration.

A possibilistic variant of the well-known k-nearest neighbor classifier, which constitutes the core of the family of CBL algorithms, has been presented in [37]. Among other things, this paper emphasizes the ability of possibility theory to represent partial ignorance as a special advantage in comparison to probabilistic approaches. In fact, this point seems to be of critical importance in case-based learning, where the reliability of a classification strongly depends on the existence of cases that are similar to the query. Consider the extreme situation where the case library does not contain any case similar to the query. In other words, nothing is known about the query. This situation of complete ignorance can be represented in an adequate way in possibility theory (either by the possibility distribution $\pi \equiv 1$, or by the distribution $\delta \equiv 0$, depending on whether possibility is interpreted as "potential possibility" or "evidential support"). In probability theory, complete ignorance is usually modeled by the uniform distribution, as suggested by the principle of insufficient reason. In our situation, this distribution is given by $p \equiv 1/c$, where c is the number of potential class labels. The adequacy of this distribution as a representation of ignorance has been called into question [25]. In particular, one cannot distinguish between complete ignorance and the situation where one can be quite sure that the class labels are indeed equi-probable, since the distribution $p \equiv 1/c$ has been derived from a large enough number of similar cases. From a knowledge representational point of view, this clearly shows the advantage of *absolute* (possibilistic) over *relative* (probabilistic) degrees of support in CBL. For example, telling a patient that your experience does not allow any statement concerning his prospect of survival is very different from telling him that his chance is fifty-fifty.

The use of OWA-operators as generalized aggregation operators in case-based learning has been proposed in [63]. In fact, there are several types of aggregation problems that arise in CBL. One of these problems concerns the derivation of a global degree of similarity between cases: Typically, one proceeds from *local* similarity degrees pertaining to individual (one-dimensional) attributes of a case (e.g. the size of a person), since specifying the similarity with respect to a single aspect is much easier than providing a global similarity directly. Rather, a global degree of similarity is derived indirectly, by aggregating the local similarity degrees. Usually, this is done by means of a simple linear combination, and this is where OWA-operators provide an interesting, more flexible alternative. A second aggregation problem in CBL concerns the combination of the evidences in favor of different class labels that come from the neighbors of the query case. In [38], it is argued that cases retrieved from a case library must not be considered as independent information sources, as implicitly done by most case-based learning methods. To take interdependencies between the neighbored cases into account, a new inference principle is developed that combines potentially interacting pieces of evidence by means of the (discrete) Choquet-integral. This method can be seen as a generalization of weighted nearest neighbor estimation.

3.6 Possibilistic Networks

So-called graphical models, including Bayesian networks [49] and Markov networks [45], have been studied intensively in recent years. The very idea of such models is to represent a high-dimensional probability distribution (defined on the Cartesian product of the domains of all attributes under consideration) in an efficient way, namely by factorizing it into several low-dimensional conditional or marginal distributions. This is accomplished by exploiting certain independence relations between subsets of attributes. The graphical model itself represents just these (in-)dependence relations: A node corresponds to an attribute, and an edge represents a direct dependency. Efficient algorithms for propagating evidence in graphical models have been developed to obtain the conditional distribution of certain attributes given the values of other variables.

Moreover, methods for learning graphical models from data have been devised. A graphical model basically consists of two components, a *qualitative* and a *quantitative* one. The former consists of the dependence and independence relations (as represented by the graph), the latter of the associated (conditional, low-dimensional) probability distributions. Correspondingly, there are two types of learning methods: those which assume the qualitative part to be given and, hence, learn only the probability distributions, and those which aim at learning the complete model, including the independence relations.

By their very nature, graphical models of the above kind provide a suitable means for representing *probabilistic* uncertainty. However, they cannot easily deal with other types of uncertainty such as imprecision or incompleteness. This has motivated the development of *possibilistic networks* as a possibilistic counterpart to probabilistic networks [6]. This approach relies upon possibility theory as an underlying uncertainty calculus, which makes it particularly suitable for dealing with imprecise data (in the form of set-valued specifications of attribute values). In this approach, the interpretation of possibility distributions is based on the so-called context model [32], hence possibility degrees are considered as a kind of upper probability.

4 Potential Contributions of Fuzzy Set Theory

4.1 Graduality

The ability to represent gradual concepts and fuzzy properties in a thorough way is one of the key features of fuzzy sets. This aspect is also of primary importance in the context of ML&DM.

In machine learning, for example, the formal problem of *concept learning* has received a great deal of attention. A concept is usually identified with its extension, that is a subset C of an underlying set (universe) U of objects. For example, C might be the concept "dog" whose extension is the set of dogs presently alive, a subset of all creatures on earth. The goal of (machine) learning is to induce an *intensional* description of a concept from a set of (positive and negative) examples, that is a characterization of a concept in terms of its properties (a dog has four legs and a tail, it can bark, ...). Now, it is widely recognized that most natural concepts have non-sharp boundaries. To illustrate, consider concepts like woods, river, lake, hill, street, house, or chair. Obviously, these concepts are vague or fuzzy, in that one cannot unequivocally say whether or not a certain collection of trees should be called a wood, whether a certain building is really a house, and so on. Rather, one will usually agree only to a certain extent that an object belongs to a concept. Thus, an obvious idea is to induce *fuzzy concepts*, that are formally identified by a fuzzy rather than a crisp subset of U. Fuzzy concepts can be characterized in terms of fuzzy predicates (properties) which are combined by means of generalized logical connectives. In fact, one should recognize that graduality is not only advantageous for expressing the concept itself, but also for modeling the qualifying properties. For example, a "firm ground" is a characteristic property of a street, and this property is obviously of a fuzzy nature (hence it should be formalized accordingly).

Likewise, in data mining, the patterns of interest are often vague and have boundaries that are non-sharp in the sense of FST. To illustrate, consider the concept of a "peak": It is usually not possible to decide in an unequivocal way whether a timely ordered sequence of measurements (e.g. the expression profile of a gene in a microarray experiment, to mention one of the topical application areas of fuzzy data mining) has a "peak" (a particular kind of pattern) or not. Rather, there is a gradual transition between having a peak and not having a peak. Taking graduality into account is also important if one must decide whether a certain property is frequent among a set of objects, e.g., whether a pattern occurs frequently in a data set. In fact, if the pattern is specified in an overly restrictive manner, it might easily happen that none of the objects matches the specification, even though many of them can be seen as approximate matches. In such cases, the pattern might still be considered as “well-supported” by the data.

Unfortunately, the representation of graduality is often foiled in machine learning applications, especially in connection with the learning of predictive models. In such applications, a fuzzy prediction is usually not desired, rather one is forced to come up with a definite final decision. Classification is an obvious example: Usually, a decision in favor of one particular class label has to be made, even if the object under consideration seems to have partial membership in several classes simultaneously. This is the case both in theory and practice: In practice, the bottom line is the course of action (e.g. the choice among a set of applicants) one takes on the basis of a prediction, not the prediction itself. In theory, a problem concerns the performance evaluation of a fuzzy classifier:[8] The standard benchmark data sets (e.g. those from the UCI repository or the StatLib archive[9]) have crisp rather than fuzzy labels. Moreover, a fuzzy classifier cannot be compared with a standard (non-fuzzy) classifier unless it eventually outputs crisp predictions.

Needless to say, if a fuzzy predictor is supplemented with a “defuzzification” mechanism (like a winner-takes-all strategy in classification), many of its merits are lost. In the classification setting, for instance, a defuzzified fuzzy classifier does again produce hard decision boundaries in the input space. Thereby, it is actually reduced to a standard classifier.

Here is an example often encountered in the literature: Suppose the premise of a classification rule to be a conjunction of antecedents of the form $x_i \in A_i$, where x_i is an attribute value and A_i a fuzzy set, and let the rules be combined in a disjunctive way. Moreover, let the consequent of a rule be simply a class assignment. If the standard minimum and maximum operators are used, respectively, as a generalized logical conjunction and disjunction, it is easy to see that the classifier thus obtained induces axis-parallel decision boundaries in the input space, and that the same boundaries can be produced by means of interval-based instead of fuzzy rules.

If a classifier is solely evaluated on the basis of its predictive accuracy, then all that matters is the decision boundaries it produces in the input space. Since a defuzzified fuzzy classifier does not produce a decision boundary that is principally different from the boundaries produced by alternative classifiers (such as decision trees or neural networks), fuzzy machine learning methods don’t

[8] The same problem occurs for probabilistic classifiers.

[9] `http://www.ics.uci.edu/˜mlearn`, `http://stat.cmu.edu/`

have much to offer with regard to generalization performance. And indeed, fuzzy approaches to classification do usually *not* improve predictive accuracy.

Let us finally note that “graduality” is of course not reserved to fuzzy methods. Rather, it is inherently present also in many standard learning methods. Consider, for example, a concept learner (binary classifier) $c : \mathcal{X} \rightarrow [0, 1]$ the output of which is a number in the unit interval, expressing a kind of “propensity” of an input x to the concept under consideration. Classifiers of such kind abound, a typical example is a multilayer perceptron. In order to extend such classifiers to multi-class problems (involving more than two classes), one common approach is to apply a one-against-all strategy: For each class y, a separate classifier $c_y(\cdot)$ is trained which considers that class as the concept to be learned and, hence, instances of all other classes as negative examples. The prediction for a new input x is then given by the class that maximizes $c_y(x)$. Now, it is of course tempting to consider the $c_y(x)$ as (estimated) membership degrees and, consequently, the collection $\{c_y(x) \,|\, y \in \mathcal{Y}\}$ of these estimations as a fuzzy classification.

4.2 Interpretability

A primary motivation for the development of fuzzy sets was to provide an interface between a numerical scale and a symbolic scale which is usually composed of linguistic terms. Thus, fuzzy sets have the capability to interface quantitative patterns with qualitative knowledge structures expressed in terms of natural language. This makes the application of fuzzy technology very appealing from a knowledge representational point of view. For example, it allows association rules discovered in a database to be presented in a linguistic and hence comprehensible way. In fact, the user-friendly representation of models and patterns is often emphasized as one of the key features of fuzzy methods.

The use of linguistic modeling techniques does also produce some disadvantages, however. A first problem concerns the interpretation of fuzzy models: Linguistic terms and, hence, models are highly subjective and context-dependent. It is true that the imprecision of natural language is not necessarily harmful and can even be advantageous. [10] A fuzzy controller, for example, can be quite insensitive to the concrete mathematical translation of a linguistic model. One should realize, however, that in fuzzy control the information flows in a reverse direction: The linguistic model is not the end product, as in ML&DM, it rather stands at the beginning.

[10] See Zadeh’s famous principle of incompatibility between precision and meaning.

It is of course possible to disambiguate a model by complementing it with the semantics of the fuzzy concepts it involves (including the specification of membership functions). Then, however, the complete model, consisting of a qualitative (linguistic) and a quantitative part, becomes cumbersome and will not be easily understandable. This can be contrasted with interval-based models, the most obvious alternative to fuzzy models: Even though such models do certainly have their shortcomings, they are at least objective and not prone to context-dependency.

Another possibility to guarantee transparency of a fuzzy model is to let a user of a data mining system specify all fuzzy concepts by hand, including the fuzzy partitions for all of the variables involved in the study under consideration. This is rarely done, however, mainly for two reasons. Firstly, the job is of course tedious and cumbersome if the number of variables is large. Secondly, much flexibility for model adaptation is lost, because it is by no means guaranteed that accurate predictive models or interesting patterns can be found on the basis of the fuzzy partitions as pre-specified by the user. In fact, in most methods the fuzzy partitions are rather *adapted* to the data in an optimal way, so as to maximize the model accuracy or the interestingness of patterns.

A second problem with regard to transparency concerns the complexity of models. A rule-based classifier consisting of, say, 40 rules each of which has a condition part with 5-7 antecedents, will hardly be comprehensible as a whole, even if the various ingredients might be well understandable. Now, since models that are simple, e.g. in the sense of including only a few attributes or a few rules, will often not be accurate at the same time, there is obviously a conflict between accuracy and understandability and, hence, the need to find a tradeoff between these criteria [9].

In fact, this tradeoff concerns not only the size of models, but also other measures that are commonly employed in order to improve model accuracy. In connection with rule-based models, for example, the *weighting* of individual rules can often help to increase the predictive accuracy. On the other hand, the interpretation of a set of weighted rules is anything but trivial.

4.3 Robustness

It is often claimed that fuzzy methods are more robust than non-fuzzy methods. Of course, the term “robustness” can refer to many things, e.g., to the sensitivity of an induction method towards violations of the model assumptions.[11] In connection with fuzzy methods, the most relevant type of robustness concerns sensitivity towards variations of the data. Roughly speaking, a learning or data mining method is considered robust if a small variation of the observed data does hardly alter the induced model or the evaluation of a pattern.[12]

A common argument supporting the claim that fuzzy models are in this sense more robust than non-fuzzy models refers to the already mentioned "boundary effect", which occurs in various variants and is arguably an obvious drawback of interval-based methods. In fact, it is not difficult to construct convincing demonstrations of this effect: In association analysis (cf. section 3.4), for example, a small shift of the boundary of an interval can have a drastic effect on the support of a fuzzy association rule if many data points are located near the boundary. This effect is alleviated when using fuzzy sets instead of intervals.

Unfortunately, such examples are often purely artificial and, hence, of limited practical relevance. Moreover, there is no clear conception of the concrete meaning of *robustness*. Needless to say, without a formal definition of robustness, i.e., certain types of robustness measures, one cannot argue convincingly that one data mining method is more robust than another one. For example, it makes a great difference whether robustness is understood as a kind of *expected* or a kind of *worst-case* sensitivity: It is true that a shifting of data points can have a stronger effect on, say, the support of an interval-based association rule than on the support of a fuzzy association. However, if the data points are not located at the boundary region of the intervals, it can also happen that the former is not affected at all, whereas a fuzzy rule is almost always affected at least to some extent (since the "boundary" of a fuzzy interval is much wider than that of a standard interval). Consequently, if robustness is defined as a kind of *average* rather than *maximal* sensitivity, the fuzzy approach might not be more robust than the non-fuzzy one.

4.4 Representation of Uncertainty

Machine learning is inseparably connected with uncertainty. To begin with, the data presented to learning algorithms is imprecise, incomplete or noisy most of the time, a problem that can badly mislead a learning procedure. But even if observations are perfect, the generalization beyond that data, the

[11] This type of sensitivity is of special interest in robust statistics.

[12] Note that we speak about robustness of the learning algorithm (that takes a set of data as input and outputs a model), not about robustness of the induced model (that takes instances as input and outputs, say, a classification).

process of induction, is still afflicted with uncertainty. For example, observed data can generally be explained by more than one candidate theory, which means that one can never be sure of the truth of a particular model.

Fuzzy sets and possibility theory have made important contributions to the representation and processing of uncertainty. In ML&DM, like in other fields, related uncertainty formalisms can complement probability theory in a reasonable way, because not all types of uncertainty relevant to machine learning are probabilistic and because other formalisms are more expressive than probability.

To illustrate the first point, consider the problem of inductive reasoning as indicated above: In machine learning, a model is often induced from a set of data on the basis of a *heuristic* principle of inductive inference such as, e.g., the well-known Occams's razor. As one can never be sure of the truth of the particular model suggested by the heuristic principle, it seems reasonable to specify a kind of *likelihood* for all potential candidate models. This is done, e.g., in Bayesian approaches, where the likelihood of models is characterized in terms of a posterior probability distribution (probability of models given the data). One can argue, however, that the uncertainty produced by heuristic inference principles such as Occam's razor is not necessarily of a probabilistic nature and, for example, that the derivation of a *possibility distribution* over the model space is a viable alternative. This idea has been suggested in [36] in connection with decision tree induction: Instead of learning a single decision tree, a possibility distribution over the class of all potential trees is derived on the basis of a possibilistic variant of Occam's razor.

The second point, concerning the limited expressivity of probability distributions, has been illustrated nicely in section 3.5, where we have argued that possibility distributions are more suitable for representing partial ignorance in case-based learning. In a similar way, possibility theory is used for modeling incomplete and missing data in possibilistic networks, as outlined in section 3.6.

Finally, we note that apart from possibility theory, other formalisms can be used to model various forms of uncertainty and incomplete information in learning from data. For example, belief functions have been extensively employed in this connection (e.g. [17,26]).

4.5 Incorporation of Background Knowledge

Roughly speaking, inductive learning can be seen as searching the space of candidate hypotheses for a most suitable model. The corresponding search process, regardless whether it is carried out in an explicit or implicit way, is usually "biased" in various ways, and each bias usually originates from a sort of background knowledge. For example, the *representation bias* restricts the hypothesis space to certain types of input-output relations such as, e.g., linear or polynomial relationships. Incorporating background knowledge is extremely important, because the data by itself would be totally meaningless if considered from an "unbiased" point of view [46].

Fuzzy set-based modeling techniques provide a convenient tool for making expert knowledge accessible to computational methods and, hence, to incorporate background knowledge in the learning process. Here, we briefly outline two possibilities.

One very obvious approach is to combine rule-based modeling and learning. For example, an expert can describe an input-output relation in terms of a fuzzy rule base (as in fuzzy control). Afterwards, the membership functions specifying the linguistic terms that have been employed by the expert can be adapted to the data in an optimal way.[13] In other words, the expert specifies the rough structure of the rule-based model, while the fine-tuning is done in a data-driven way. Let us note that specifying the structure of a model first and adapting that structure to the data afterwards is a general strategy for combining knowledge-based and data-driven modeling, which is not reserved to rule-based models; it is used, for example, in graphical models (cf. section 3.6) as well.

An alternative approach, called constraint-regularized learning, aims at exploiting fuzzy set-based modeling techniques within the context of the regularization (penalization) framework of inductive learning [39]. Here, the idea is to express vague, partial knowledge about an input-output relation in terms of fuzzy constraints and to let such constraints play the role of a penalty term within the regularization approach. Thus, an optimal model is one that achieves an optimal tradeoff between fitting the data and satisfying the constraints.

[13] Here, the expert knowledge implements a kind of *search bias*, as it determines the starting point of the search process and, hence, the local optimum in the space of models that will eventually be found.

4.6 Generalized Aggregation Operators

Many ML&DM methods make use of logical and arithmetical operators for representing relationships between attributes in models and patterns. In decision tree induction, for example, each inner node represents an equality or an inequality predicate, and these predicates are combined in a conjunctive way along a path of a tree. In nearest neighbor classification, each neighbor provides a certain amount of evidence in favor of the class it belongs to. To make a final decision, this evidence must be aggregated either way, which in the standard approach is done by simply adding them up.

Now, a large repertoire of generalized logical (e.g. t-norms and t-conorms) and arithmetical (e.g. Choquet- and Sugeno-integral) operators have been developed in FST and related fields. Thus, a straightforward way to extend standard learning methods consists of replacing standard operators by their generalized versions. In fact, several examples of this idea have been presented in previous sections.

The general effect of such generalizations is to make models more flexible. For example, while a standard decision tree can only produce axis-parallel decision boundaries, these boundaries can become non-axis-parallel for fuzzy decision trees where predicates are combined by means of a t-norm. Now, it is well-known that learning from empirical data will be most successful if the model class under consideration has just the right flexibility, since both over- and underfitting of a model can best be avoided in that case. Therefore, the question whether a fuzzy generalization will pay off cannot be answered in general: If the original (non-fuzzy) hypothesis space is not flexible enough, the fuzzy version will probably be superior. On the other hand, if the former is already flexible enough, a fuzzification might come along with a danger of overfitting.

5 Conclusions

All things considered, it is beyond question that FST has the potential to contribute to machine learning and data mining in various ways. In fact, the previous sections have shown that substantial contributions have already been made. Yet, our remarks also suggest that much scope for further developments is still left. According to our opinion, however, it is very important to focus on the right issues, that is to say, to concentrate more on the strengths and distinctive features of FST.

In particular, we doubt that FST will be very conducive to *generalization performance* and *model accuracy*, albeit the latter is still the dominant quality criterion in machine learning research. This somewhat sceptical view has at least two reasons: Firstly, after several years of intensive research the field of machine learning has reached a somewhat mature state, and a large repertoire of quite sophisticated learning algorithms is now available. Regarding predictive accuracy, a significant improvement of the current quality level can hardly be expected.

Secondly, and perhaps more importantly, FST does not seem to offer fundamentally new concepts or induction principles for the design of learning algorithms, comparable, e.g., to the ideas of *resampling* and *ensemble learning* [20] (like bagging [7] and boosting [28]) or the idea of *margin maximization* underlying kernel-based learning methods [55], that might raise hope for an improved generalization performance. As mentioned above, even though fuzzifying standard learning methods, e.g. by using fuzzy partitions of numeric attributes or generalized logical and arithmetic operators, can have an effect on the decision boundaries of a classifier or the regression function produced in the case of numeric prediction, the gain in predictive accuracy is mostly insignificant.

In this connection, we also like to question a current research trend in the FST community. It seems that the shift from (knowledge-driven) modeling to (data-driven) learning, as signified in section 1, comes along with a tendency to view fuzzy systems as pure function approximators. In fact, in many recent publications fuzzy sets simply serve as a special kind of basis or kernel function.[14] Thus, there is a high danger of losing sight of the original ideas and intentions of FST, and to produce another type of "black box" approach instead. Truly, renaming a basis function a "fuzzy set" does not mean that a model will suddenly become comprehensible.

Rather than suggesting new solutions to problems for which alternative methods from established fields such as, e.g., approximation theory, neural networks, and machine learning, will probably be more successful, more emphasis should be put on the distinguished features of FST. In this connection, let us highlight the following points:

1. FST has the potential to produce models that are more comprehensible, less complex, and more robust.
2. FST, in conjunction with possibility theory, can contribute considerably to the modeling and processing of various forms of uncertain and incomplete

[14] In [11], for example, a support vector machine is trained and then turned into a fuzzy rule base by identifying each support vector with a fuzzy rule.

information.

3. Fuzzy methods appear to be particularly useful for data pre- and post-processing.

Concerning the first point, our critical comments in previous sections have shown that, despite of the high potential, many questions are still open. For example, notions like “comprehensibility”, “simplicity”, or “robustness” still lack an underlying formal theory including a quantification of their intuitive meaning in terms of universally accepted measures. This is probably one of the reasons why model accuracy is still regarded as a more concrete and, hence, more important quality criterion. Anyway, we think that the tradeoff between accuracy on the one side and competitive criteria like interpretability, simplicity, and robustness on the other side is an issue of central importance and a problem to that FST can contribute in a substantial way. In fact, fuzzy information granulation appears to be an ideal tool for trading off accuracy against complexity and understandability of models. Of course, a necessary prerequisite for studying this tradeoff in a more rigorous way and, hence, a challenge for future research, is a better understanding and formalization of these alternative criteria.

The second point refers to an aspect that is of primary importance in ML&DM, and that has already been touched on in section 4.4. Meanwhile, the coexistence of various forms of uncertainty, not all of which can be adequately captured by probability theory, has been widely recognized. Still, in machine learning, and more generally in the AI community, fuzzy sets and related uncertainty calculi have not yet obtained a proper acceptance. This situation might be further impaired by the increasing popularity of probabilistic methodology, which in machine learning can mainly be ascribed to the success of *statistical learning theory* [59] as a solid foundation of empirical learning, and in AI to the general acceptance of the Bayesian framework for knowledge representation and reasoning under uncertainty. For the FST community, it is all the more important to show that alternative uncertainty formalisms can complement probability theory in a reasonable way.

Concerning the third point, we have the feeling that this research direction has not received enough attention so far. In fact, even though FST seems to be especially qualified for data pre- and postprocessing, e.g. for data summarization and reduction, approximation of complex and accurate models, or the (linguistic) presentation of data mining results, current research is still more focused on the inductive reasoning or data mining process itself. In this respect, we see a high potential for further developments, especially against the background of the current trend to analyze complex and heterogeneous information sources that are less structured than standard relational data tables.

Finally, there are some other research directions that are worth further exploration. For example, so far most of the work in the FST community has been *methodologically* oriented, focusing on the fuzzy extension of standard learning methods, whereas both the *experimental* validation and the *theoretical* analysis of fuzzy machine learning methods have received much less attention. As mentioned above, validating the predictive performance of a fuzzy method in an empirical way is not as easy, since fuzzy labels for comparison are rarely available in practice. What is a good fuzzy prediction? This question naturally arises if fuzzy predictions are not defuzzified, and it becomes even more intricate if predictions are expressed in terms of still more complex uncertainty formalisms such as, e.g., lower and upper possibility bounds, type-II fuzzy sets, or belief functions. Regarding theoretical analyses of fuzzy learning methods, it would be interesting to investigate whether fuzzy extensions are profitable from a theoretical point of view. For example, is it possible that a class of concepts is, say, PAC-learnable[15] by the fuzzy extension of a learning algorithm but not by the original version? Such results would of course be highly welcome as a formal justification of fuzzy learning methods.

Acknowledgements: The author is grateful to Thomas Sudkamp and Christophe Marsala for helpful comments and useful suggestions.

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proceedings of the 20th Conference on* VLDB, pages 487–499, Santiago, Chile, 1994.

[2] D.W. Aha, editor. *Lazy Learning.* Kluwer Academic Publ., 1997.

[3] D.W. Aha, D. Kibler, and M.K. Albert. Instance-based learning algorithms. *Machine Learning*, 6(1):37–66, 1991.

[4] R. Babuska. *Fuzzy Modeling for Control.* Kluwer Academic Publishers, Boston, 1998.

[5] H. Bandemer and W. Näther. *Fuzzy Data Analysis.* Kluwer Academic Publishers, Dordrecht, 1992.

[6] C. Borgelt and R. Kruse. *Graphical Models – Methods for Data Analysis and Mining.* Wiley, Chichester, 2002.

[7] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[15] The PAC (probably approximately correct) learning framework is a well-known formal model of inductive learning [58].

[8] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees.* Wadsworth International Group, Belmont, CA, 1984.

[9] J. Casillas, O. Cordon, F. Herrera, and L. Magdalena, editors. *Interpretability Issues in Fuzzy Modeling.* Studies in Fuzziness and Soft Computing. Springer-Verlag, Berlin, 2003.

[10] G. Chen, Q. Wei, E. Kerre, and G. Wets. Overview of fuzzy associations mining. In *Proc. ISIS–2003, 4th International Symposium on Advanced Intelligent Systems.* Jeju, Korea, September 2003.

[11] Y. Chen and JZ. Wang. Support vector learning for fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, 11(6):716–728, 2003.

[12] O. Cordon, MJ. del Jesus, and F. Herrera. Analyzing the reasoning mechanisms in fuzzy rule based classification systems. *Mathware & Soft Computing*, 5:321–332, 1998.

[13] O. Cordon, F. Gomide, F. Herrera, and F. Hoffmann anf L. Magdalena. Ten years of genetic fuzzy systems: current framework and new trends. *Fuzzy Sets and Systems*, 141(1):5–31, 2004.

[14] TH. Dang, B. Bouchon-Meunier, and C. Marsala. Measures of information for inductive learning. In *Proc. IPMU-2004*, Perugia, Italy, 2004.

[15] B.V. Dasarathy, editor. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques.* IEEE Computer Society Press, Los Alamitos, California, 1991.

[16] M. Delgado, N. Marin, D. Sanchez, and MA. Vila. Fuzzy association rules: general model and applications. *IEEE Transactions on Fuzzy Systems*, 11(2):214–225, 2003.

[17] T. Denoeux. A k-nearest neighbor classification rule based on Dempster-Shafer Theory. IEEE *Transactions on Systems, Man, and Cybernetics*, 25(5):804–813, 1995.

[18] P. Diamond and P. Kloeden. *Metric Spaces of Fuzzy Sets: Theory and Applications.* World Scientific, Singapur, 1994.

[19] P. Diamond and H. Tanaka. Fuzzy regression analysis. In R. Slowinski, editor, *Fuzzy Sets in Decision Analysis, Operations Research and Statistics*, pages 349–387. Kluwer, 1998.

[20] TG. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *First International Workshop on Multiple Classifier Systems*, volume 1857, pages 1–15. Springer-Verlag, 2000.

[21] D. Dubois, F. Esteva, P. Garcia, L. Godo, R. Lopez de Mantaras, and H. Prade. Fuzzy set modelling in case-based reasoning. *International Journal of Intelligent Systems*, 13:345–373, 1998.

[22] D. Dubois, E. Hüllermeier, and H. Prade. Fuzzy set-based methods in instance-based reasoning. IEEE *Transactions on Fuzzy Systems*, 10(3):322–332, 2002.

[23] D. Dubois, E. Hüllermeier, and H. Prade. A note on quality measures for fuzzy association rules. In *Proceedings* IFSA–03*, 10th International Fuzzy Systems Association World Congress*, LNAI 2715, pages 677–648, Istambul, July 2003. Springer-Verlag.

[24] D. Dubois and H. Prade. What are fuzzy rules and how to use them. *Fuzzy Sets and Systems*, 84:169–185, 1996.

[25] D. Dubois, H. Prade, and P. Smets. Representing partial ignorance. IEEE *Transactions on Systems, Man, and Cybernetics–Part A: Systems and Humans*, 26(3):361–377, 1996.

[26] Z. Elouedi, K. Mellouli, and P. Smets. Decision trees using the belief function theory. In *Proc. IPMU-2000*, volume I, pages 141–148, Madrid, 2000.

[27] UM. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery: An overview. In *Advances in Knowledge Discovery and Data Mining*. MIT Press, 1996.

[28] Y. Freund and RE. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[29] J Fürnkranz. Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1):3–54, 1999.

[30] J. Fürnkranz and E. Hüllermeier. Pairwise preference learning and ranking. In ECML–2003*, Proceedings 13th European Conference on Machine Learning*, Cavtat-Dubrovnik, Croatia, September 2003. Springer-Verlag.

[31] AP. Gasch and MB. Eisen. Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biology*, 3(11):1–22, 2002.

[32] J. Gebhardt and R. Kruse. A possibilistic interpretation of fuzzy sets by the context model. In *IEEE International Conference on Fuzzy Systems*, pages 1089–1096, 1992.

[33] S. Har-Peled, D. Roth, and D. Zimak. Constraint classification: a new approach to multiclass classification. In *Proceedings 13th Int. Conf. on Algorithmic Learning Theory*, pages 365–379, Lübeck, Germany, 2002. Springer.

[34] F. Höppner, F. Klawonn, F. Kruse, and T. Runkler. *Fuzzy Cluster Analysis.* Wiley, Chichester, 1999.

[35] E. Hüllermeier. Implication-based fuzzy association rules. In *Proceedings* PKDD–01*, 5th European Conference on Principles and Practice of Knowledge Discovery in Databases*, LNAI 2168, pages 241–252, Freiburg, Germany, September 2001. Springer-Verlag.

[36] E. Hüllermeier. Possibilistic induction in decision tree learning. In *Proceedings* ECML–02*, 13th European Conference on Machine Learning*, pages 173–184, Helsinki, Finland, August 2002. Springer-Verlag.

[37] E. Hüllermeier. Possibilistic instance-based learning. *Artificial Intelligence*, 148(1–2):335–383, 2003.

[38] E. Hüllermeier. Cho-k-NN: A method for combining interacting pieces of evidence in case-based learning. In *Proceedings* IJCAI–05*, 19th International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland, 2005.

[39] E. Hüllermeier, I. Renners, and A. Grauel. An evolutionary approach to constraint-regularized learning. *Mathware and Soft Computing*, 11(2–3):109–124, 2005.

[40] CZ. Janikow. Fuzzy decision trees: Issues and methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 28(1):1–14, 1998.

[41] J.L. Kolodner. *Case-based Reasoning.* Morgan Kaufmann, San Mateo, 1993.

[42] R. Krishnapuram and J.M. Keller. A possibilistic approach to clustering. IEEE *Transactions on Fuzzy Systems*, 1(2):98–110, 1993.

[43] R. Kruse and C. Borgelt. Information mining: Editorial. *Int. Journal of Approximate Reasoning*, 32:63–65, 2003.

[44] A. Laurent. Generating fuzzy summaries: a new approach based on fuzzy multidimensional databases. *Intelligent Data Analysis Journal*, 7(2):155–177, 2003.

[45] S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, 50(2):157–224, 1988.

[46] T.M. Mitchell. The need for biases in learning generalizations. Technical Report TR CBM–TR–117, Rutgers University, 1980.

[47] D. Nauck, F. Klawonn, and R. Kruse. *Foundations of Neuro-Fuzzy Systems.* Wiley and Sons, Chichester, UK, 1997.

[48] C. Olaru and L. Wehenkel. A complete fuzzy decision tree technique. *Fuzzy Sets and Systems*, 138(2), 2003.

[49] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, CA, 1988.

[50] J.R. Quinlan. J.r. quinlan. discovering rules by induction from large collections of examples. In D. Michie, editor, *Expert Systems in the Micro Electronic Age.* Edinburgh University Press, 1979.

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

[52] C.K. Riesbeck and R.C. Schank. *Inside Case-based Reasoning.* Hillsdale, New York, 1989.

[53] E.H. Ruspini. Possibility as similarity: The semantics of fuzzy logic. In P.P. Bonissone, H. Henrion, L.N. Kanal, and J.F. Lemmer, editors, *Uncertainty In Artificial Intelligence 6*. Elsevier Science Publisher, 1991.

[54] A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. In VLDB–95*, Proceedings of 21th International Conference on Very Large Data Bases*, pages 11–15, Zurich, September 1995.

[55] B. Schölkopf and AJ. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2001.

[56] T. Sudkamp. Similarity as a foundation for possibility. In *Proc. 9th IEEE Int. Conference on Fuzzy Systems*, pages 735–740, San Antonio, 2000.

[57] T. Sudkamp. Examples, counterexamples, and measuring fuzzy associations. *Fuzzy Sets and Systems*, 149(1), 2005.

[58] LG. Valiant. A theory of the learnable. *CACM*, 17(11):1134–1142, 1984.

[59] V.N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, second edition, 2000.

[60] R. Viertl. *Statistical Methods for Non-Precise Data.* CRC Press, Boca Raton, Florida, 1996.

[61] L.X. Wang and J.M. Mendel. Generating fuzzy rules by learning from examples. IEEE *Transactions on Systems, Man, and Cybernetics*, 22(6):1414–1427, 1992.

[62] R. Weber. Fuzzy-ID3: a class of methods for automatic knowledge acquisition. In *IIZUKA-92, Proc. of the 2nd Intl. Conf. on Fuzzy Logic*, volume 1, pages 265–268. 1992.

[63] R.R. Yager. Soft aggregation methods in case based reasoning. *Applied Intelligence*, 21:277–288, 2004.