Creating comprehensible regression models

Inductive learning and optimization of fuzzy regression trees using comprehensible fuzzy predicates

Mario Drobics · Johannes Himmelbauer

Abstract In this paper we will present a novel approach to data-driven fuzzy modeling which aims to create highly accurate but also easily comprehensible models. This is achieved by a three-stage approach which separates the definition of the underlying fuzzy sets, the learning of the initial fuzzy model, and finally a local or global optimization of the resulting model. The benefit of this approach is that it allows to use a language comprising of comprehensible fuzzy predicates and to incorporate expert knowledge by defining problem-specific fuzzy predicates. Furthermore, we achieve highly accurate results by applying a regularized optimization technique.

Keywords Inductive learning · Fuzzy regression trees · Regularization · Interpretability

1 Introduction

Fuzzy logic based systems can be used to gain insights into a complex system for which no analytical model exists. For many complex technical applications, the problem arises that no proper mathematical formulation can be found to describe the behavior of the respective system. The only available information might be a set of measurements taken from the system. Then the goal is to find a function \( f \) that models the inherent connection between the input parameters (settings and measurements) and the goal parameter (final parameter of interest) that is hidden in the data.

To find such a function \( f \), however, is not always the only objective. While statistical regression (Draper-Smith 1981) or neural networks (McClelland and Rumelhart 1986; Rumelhart and (McClelland 1986; Zurada 1992) allow to solve such kinds of machine learning problems, they leave the resulting function \( f \) as a black box, i.e. a plain function whose internals are difficult or impossible to comprehend. In many practical applications, however, qualitative insights into the structures of \( f \) are desirable. For such tasks, rule-based systems are most appropriate. They easily allow qualitative insight, since the function \( f \) is represented by logical rules in a close-to-natural-language manner. In the following, assume that we are not necessarily interested in the full function \( f \), but at least in significant bits of knowledge about \( f \) and their inherent structures, i.e. rules.

For the remaining, let us consider a data set \( \mathcal{X} \) of \( K \) samples

\[
\mathcal{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^K\},
\]

where each sample \( (i = 1, \ldots, K) \) has the same \((n + 1)\)-dimensional structure:

\[
\mathbf{x}^i = (x_1^i, \ldots, x_n^i, x_{n+1}^i)
\in X_1 \times \cdots \times X_n \times X_{n+1}
\]

The first \( n \) dimensions/variables are the inputs; the last dimension/variable \( n + 1 \) is the output under investigation. In the following, we refer to the \( r \)-th dimension \((r = 1, \ldots, n)\) as the \( r \)-th input attribute. The \( n + 1 \)-st dimension is called goal attribute. Ideally, the overall objective of this machine learning problem is then to
find a function

\[ f : X_1 \times \cdots \times X_n \rightarrow X_{n+1} \]  

such that the inherent connection between the input attributes and the goal attribute hidden in the data set is modeled as well as possible. Therefore, such machine learning problems can be regarded as a kind of data fitting.

To be able to handle numeric attributes in rule-based models, it is indispensable to define a discrete set of predicates for these kinds of attributes. If this quantization is done by means of partitions into crisp sets (intervals) as in traditional machine learning, small variations (e.g. noise) can cause large changes in the classification quality and unstable results. This entails the demand for admitting vagueness in the assignment of samples to predicates. Fuzzy sets (Zadeh 1965) perfectly solve this problem of artificial preciseness arising from sharp interval boundaries.

A second benefit of fuzzy logic systems like decision trees (Adamo 1980; Zeidler and Schlosser 1996) or rule-based methods (Baranyi et al.; Mikut et al. 2005) is, that they create not only a computational but also an interpretable model for \( f \). The resulting function helps the user to better understand the behavior of the system (Casillas et al. 2003). It turned out, however, that in many cases the simple application of methods for creating interpretable, computational models from data is not sufficient. There is often the need for higher accuracy, while preserving the interpretability of the systems. Consequently, several approaches were developed recently to optimize existing interpretable fuzzy systems (Burger et al. 2002; Regattieri Delgodox et al. 2001).

In this paper we will present an approach to compute semantically meaningful fuzzy sets a priori to the rule induction process. We will then present an inductive learning algorithm for fuzzy decision trees which uses the so obtained fuzzy predicates to create comprehensible fuzzy models from data. Finally, we present an approach to optimize such (Takagi-)Sugeno fuzzy systems via linear approximation of the consequences. We pay special attention to the stability of the solution and to preserving the fuzzy system’s interpretability. Applying variable selection is an additional contribution to reach this goal.

2 The underlying language

To define the underlying language for our fuzzy models, we have to consider the different types of input attributes that can occur. Basically, we can distinguish between three types of attributes:

- **Boolean categorical attributes**: The domain \( X_i \) is a structured finite set of labels, for instance, types of car engines (gasoline, diesel, hydrogen, electric) or classes of animals (birds, fish, mammals, etc.). The attribute values \( x^i_j \) are single elements of the label set \( X_i \).

- **Fuzzy categorical attributes**: There is again a unstructured finite set of labels, but with possible overlaps. Therefore, values of such kinds of variables may be fuzzy sets on this set of labels. For example, assume that we are given a finite set consisting of different grape varieties. Then blended wines (cuvees) cannot be assigned to single categories crisply.

- **Numerical attributes**: The underlying domain \( X_i \) is the set of real numbers or a subset thereof (e.g. an interval). The attribute values \( x^i_j \) are real numbers, e.g. pressures, temperatures, incomes, ratios, etc.

Note that Boolean categorical attributes are special cases of fuzzy categorical attributes, since any crisp label can be considered as a fuzzy set of labels, too.

Fuzzy predicates for categorical attributes, boolean or fuzzy, can be defined easily in a straightforward manner. Finding appropriate fuzzy predicates for numerical attributes, however, is often a subtle problem for which different approaches exist.

The simplest approach is to create fuzzy sets which form a partition for each dimension and which are evenly distributed over the data range or which have the same cardinality (equi-distance binning, or equi-frequent binning). Although this approach is sufficient for basic calculations, it has strong limitations with respect to accuracy. Furthermore, it turns out that for unequally distributed data, such fuzzy sets often conflict with the user’s intuition. To overcome these limitations, several approaches exist which try to fit the fuzzy sets to the given training data as well as possible. Most of these approaches use some kind of projection methods (Castellano et al. 2002; Klawonn and Kruse 1997) While fuzzy sets are typically defined over one dimension, fuzzy clustering methods can be used to identify higher-dimensional fuzzy sets (Hoeppner and Klawonn 2003; Yager and Filev 1994). Although the results of these two approaches are very promising, the resulting fuzzy sets do not always directly correspond to a linguistic expression – one of the main building blocks of fuzzy logic. Recent approaches in this direction use complex optimization techniques (Mikut et al. 2000) or create hierarchical fuzzy partitions Guillaume and Charnomordic (2004), requiring a high computational effort.

Alternatively, the fuzzy sets can also be computed ad hoc when creating the computational models. This approach has been applied in crisp algorithms like CART (Breiman et al. 1984) and C4.5 (Quinlan 1993) as well as