A hybrid genetic fuzzy neural network algorithm designed for classification problems involving several groups

Ralf Östermark*

Department of Business Administration, Åbo Akademi University, Henriksgatan 7, 20500 Åbo, Finland

Received September 1997; received in revised form February 1998

Abstract

We propose a multigroup classification algorithm based on a hybrid genetic fuzzy neural net (GFNN) framework. Recent results on evolutionary computation and fuzzy neural network methodology are combined to effectively adapt the membership functions of the fuzzifier and the defuzzifier to the data set. Separate membership functions are defined for each dimension in the fuzzifier and for each fuzzy output group in the defuzzifier. The signal inherent in the fuzzifier is aggregated by a suitable T-norm and transmitted to the defuzzifier. The defuzzifier aggregates the response, i.e., the predicted group membership, by a suitable conorm. If misclassifications occur during training, the membership functions of both the fuzzifier and the defuzzifier are adapted by a systematic, robust procedure. The algorithm is successfully tested with real economic data. In total, the GFNN performs as good as the best of the competing methods in our test. The results suggest economically meaningful interpretations. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Hybrid fuzzy neural networks; Fuzzy set theory; Evolutionary computation; Multigroup classification

1. Introduction

We propose an approximate classification method for multigroup problems using a hybrid genetic fuzzy neural net (GFNN) framework. We combine some recent results on genetic computation [27] and fuzzy set theory with results obtained with a flexible fuzzy neural network [28,16]. From a mathematical point of view, we consider the classification problem as a mixed-integer nonlinear programming problem.

To illustrate our approach, assume that \( m \) classes are given and that the class membership \( g_i, i = 1, \ldots, k \), of an object (for example, a firm) is a function of \( m \) observed variables \( g_i = f(x_1, x_2, \ldots, x_m) \). This setting is the starting point for abundant research in financial classification problems. However, mostly the functional relationship is plagued by anomalies such as missing observations, non-normality, non-linearity, asymmetric information on groups, etc. In the empirical test conducted subsequently, most of the observations in the database belong to one group only, whereby most of the methods tested will fail or produce trivial solutions. Since the economic consequences of classification errors can be extremely tangible, especially when trying to distinguish between healthy and bankrupt firms, the research problem has gained much attention in empirical research [29].

GFNN consists of four principal steps. Initially, we fuzzify the observed variables \((x_1, \ldots, x_m)\), i.e. we...
degrees of membership in each fuzzy output group. The adjustment of the membership functions in case of misclassification is carried out following the lines in [28], augmented by genetic manipulation as outlined in [27].

We tested the GFNN-algorithm on the same real economic data as in [28]. The algorithm performed acceptably well with the simple membership functions selected to represent the fuzzy signal-response mechanism. The estimated functions are consistent with economic intuition. The results demonstrate that genetic operators in conjunction with fuzzy neural network technology provide a robust computational devise for complicated economic classification problems.

We note that, especially for a small data set, the misclassification rate is a problematic metric for comparing performance. Therefore, future research may evaluate the usefulness of a cross-entropy or likelihood approach – as used in logistic regression – in connection with genetic fuzzy neural networks.

Our results suggest interesting paths for future research. Firstly, the specification of proper membership functions to be trained by the GFNN-algorithm is a crucial research issue. Thus, the precision of the algorithm might be improved by applying evolutionary computation techniques also in the identification stage, i.e., when specifying the membership functions of the fuzzifier and defuzzifier [24]. The estimation system could be supplemented by a superlayer, responsible for (stochastically) generating membership function types for the hybrid genetic fuzzy neural network. Such a system would also provide a rich variety of opportunities for improvement by parallel programming.

Appendix A

In classical dual-valued set theory, a subset $A \subset X$ can be defined by the mapping

$$\chi_A : X \rightarrow \{0, 1\}$$  \hspace{1cm} (A.1)

such that, for each element $x$, exactly one ordered pair $(x, \chi_A(x))$ is present.

A fuzzy set is described by the following:

**Definition A.1 (Zadeh [37])**. Let $X$ be a nonempty set. A fuzzy set $A$ in $X$ is characterized by its membership function

$$\mu_A : X \rightarrow [0, 1]$$

and $\mu_A(x)$ is interpreted as the degree of membership of element $x$ in fuzzy set $A$ for each $x \in X$.

**Definition A.2 (Intersection)**. The intersection of two fuzzy sets $A(x)$ and $B(x)$ is defined as

$$(A \cap B)(x) = \min\{A(x), B(x)\} = A(x) \land B(x), \ x \in X.$$  

**Definition A.3 (Union)**. The union $A(x)$ and $B(x)$ is defined as

$$(A \cup B)(x) = \max\{A(x), B(x)\} = A(x) \lor B(x), \ x \in X.$$  

Definitions A.2 and A.3 are directly extendable to multiple fuzzy sets. In order to make inferences on fuzzy sets, we invoke the well-known **extension principle** [37].

**Definition A.4**. Let $X = X_1 \times \cdots \times X_r$ be the cartesian product of crisp sets $X_i$ and let $A_i \subset X_i$ be fuzzy sets in $X_i$, $1 \leq i \leq r$. Then

$$A = A_1 \times \cdots \times A_r$$

$$\equiv \int_{X_1 \times \cdots \times X_r} \min(\mu_{A_1}(x_1), \ldots, \mu_{A_r}(x_r))$$

is a fuzzy set in $X$. Consider the mapping $f : X \rightarrow Y$. Using the extension principle, we can define $f(A)$ as a fuzzy subset on $Y$, such that

$$\mu_B(y) = \begin{cases} 
\sup_{(x_1, \ldots, x_r) \in X, \ y = f(x_1, \ldots, x_r)} (\min(\mu_{A_1}(x_1), \ldots, \mu_{A_r}(x_r))) \\
0 & \text{otherwise.}
\end{cases}$$

Definition A.4 extends the familiar operations on real numbers to fuzzy numbers.

**References**