9-2001

Nonparametric techniques to extract fuzzy rules for breast cancer diagnosis problem

Manish SARKAR
National University of Singapore

Tze-Yun LEONG
Singapore Management University, leongty@smu.edu.sg

DOI: https://doi.org/10.3233/978-1-60750-928-8-1394

MEDINFO 2001
V. Patel et al. (Eds)
Amsterdam: IOS Press

Nonparametric Techniques to Extract Fuzzy Rules for Breast Cancer Diagnosis Problem

Manish Sarkar and Tze-Yun Leong

Medical Computing Laboratory, Department of Computer Science
School of Computing, The National University of Singapore,
Lower Kent Ridge Road, Singapore: 119260

Abstract

This paper addresses breast cancer diagnosis problem as a pattern classification problem. Specifically, the problem is studied using Wisconsin-Madison breast cancer data set. Fuzzy rules are generated from the input-output relationship so that the diagnosis becomes easier and transparent for both patients and physicians. For each class, at least one training pattern is chosen as the prototype, provided (a) the maximum membership of the training pattern is in the given class, and (b) among all the training patterns, the neighborhood of this training pattern has the least fuzzy-rough uncertainty in the given class. Using the fuzzy-rough uncertainty, a cluster is constructed around each prototype. Finally, these clusters are interpreted as the fuzzy rules that relate the prognostic factors and the diagnosis results. The advantages of the proposed algorithm are, (a) there is no need to know the structure of the training data, (b) the number of fuzzy rules does not increase with the increase of the number of input dimensions, and (c) small number of fuzzy rules is generated. With the three generated fuzzy rules, 96.20% classification efficiency is achieved, which is comparable to other rule generation techniques.

Keywords:

Breast cancer, Wisconsin-Madison data, diagnosis, classification, nearest neighbors algorithm, fuzzy set, rough set, rule base, and clustering.

Introduction

To achieve better medical diagnostic results, we cast breast cancer diagnosis problem as a pattern classification problem. Many classification algorithms act as *black boxes*, i.e., for such classifiers we do not have any means to know why a particular diagnosis is offered to a new patient. Hence, the objective of this work is to apply machine-learning techniques to the Wisconsin-Madison breast cancer diagnosis problem [1] such that fuzzy diagnostic rules [2] are extracted. Both the physician and the patient can analyze this rule base to obtain the information why the particular diagnosis is selected.

The University of Wisconsin-Madison Hospital collected 699 samples using the *fine needle aspiration test* [1]. Each sample consists of the following ten attributes: (1) Patient's id, (2) clump thickness, (3) uniformity of cell size, (4) uniformity of cell shape, (5) marginal adhesion, (6) single epithelial cell size, (7) bare nuclei, (8) bland chromatin, (9) normal nucleoli, and (10) mitosis. Each sample is either benign or malignant. The objective of the fuzzy rules is to classify a new sample into any one of the two classes.

Using parametric and semiparametric classifiers, many researchers [3][6] have measured the performance of their classification algorithms on the Wisconsin-Madison breast cancer problem. The parametric and semiparametric classifiers need specific information about the structure of the data (e.g., the number of clusters or the number of hidden nodes) in the training set. In the breast cancer problem, where the input dimension is nine, it is difficult to collect the structural information necessary for constructing a parametric or semiparametric classifier. The nonparametric classifiers do not need the information about the structure of the training set. Hence, in our earlier work [4], nonparametric classification techniques like the conventional and fuzzy versions of the K-nearest neighbors (KNN) algorithms were adopted. The KNN algorithms produce better classification results than that of the other algorithms reported in the literature.

The classification performance of the KNN algorithm usually varies with the different values of K. When K is one, the class label of the test pattern is determined just based on the class label of the closest neighbor. This scheme suffers if the class label of the closest neighbor is corrupted by noise. On the other hand, the large value of K may increase the classification efficiency since there are more bodies of evidence to classify the test pattern. However, if the neighborhood is large, then the neighbors may belong to more than one class. It happens especially in the region where two classes overlap or noise is present. Thus, it may increase the confusion in assigning the class label to the test pattern. Therefore, the optimal value of K can only be found by a trade-off, which is currently achieved using trial and error procedures.

In the conventional KNN algorithm, all the neighbors receive equal importance. In the fuzzy KNN algorithm, the *importance* of a neighbor is determined based on the relative distance between the neighbor and the test pattern. Thus, a neighbor that is far away from the test pattern also receives considerable importance, although strictly speaking the importance associated with this neighbor should be close to zero no matter what the other neighbors are. Hence, in the *fuzzy-rough nearest neighbors* (FRNN) algorithm [5] this relative distance is modified to the absolute distance so that the close neighbors have significant influence on the test pattern, and this influence decreases as the distance increases. Instead of considering K closest training patterns as the neighbors, the FRNN algorithm considers all the training patterns as the neighbors with different degrees. The degree depends on the fuzzy typicality of the training pattern. Thus, the FRNN avoids the problem of choosing the optimal value of K.

Although the FRNN has good classification ability, it behaves like a black box. To avoid this problem, we employ the FRNN to cluster the training set in a nonparametric and supervised fashion. The output of the FRNN corresponding to a training pattern signifies the amount of *fuzzy-rough uncertainty* [2] associated with the region around the training pattern. If the maximum class membership of a training pattern is in the class C, and among all the training patterns, the neighborhood of this training pattern has the least fuzzy-rough uncertainty in the class C (it can be known from the FRNN output), then this training pattern is chosen as the prototype of the class. One or more than one such prototype is selected in a nonparametric fashion for each class. Using the fuzzy-rough uncertainty, a cluster is constructed around each prototype. These clusters are finally interpreted as the fuzzy rules that relate the input features and the output class labels.

Proposed Method

The proposed method consists of two parts: (1) Designing or training the fuzzy rule base, and (2) testing the rule base.

Rule Base Designing

It involves the following three steps (Figure 2):

Fuzzy-Rough nearest neighbors algorithm: The philosophy of the FRNN is to classify the test pattern based on the class label of all the training patterns. The training pattern that is the closest or most similar to the test pattern influences the decision most. From this angle, the principle of the FRNN is similar to that of the conventional KNN algorithm. But the difference is that in the KNN algorithm, the class label of the test pattern is decided based on the class labels of the K closest training patterns, and all these K training patterns are considered equally important. In contrast, in the FRNN, (a) class labels of all the training patterns are considered, but with different importance, and (b) class labels of the training patterns can be fuzzy. Another difference is that for any input test pattern the KNN algorithm produces some class label. If some absurd input is fed, then also the KNN algorithm provides some class label. On the other hand, in this case the FRNN has the capability to indicate that the input pattern does not belong to any output class. This property is known as *possibilistic* classification ability [2][5]. These three characteristics make the FRNN more attractive than the KNN counterparts.

In the FRNN, two different kinds of information are exploited: How similar the training patterns are to the test pattern, and the class label of these training patterns. The class label of the training patterns can be both conventional or fuzzy depending on the problem domain. Let $X \subseteq \mathbf{R}^N$ be the set of all possible input patterns, and $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\} \in X$ be a set of input training patterns for which the corresponding class labels are already known. When the test pattern $\boldsymbol{y} \in X$ is fed to the algorithm, the similarity between the test pattern $\boldsymbol{y}$ and the training pattern $\boldsymbol{x}_i$ is determined using $\exp(-d(\boldsymbol{y}, \boldsymbol{x}_i)^{1/(q-1)})$. Here $d(\boldsymbol{y}, \boldsymbol{x}_i) = \sum_{k=1}^{N} \frac{1}{\kappa_k}(y_k - x_{ik})^2$ is the squared, weighted, Euclidian distance between the test pattern and the training pattern, and $\boldsymbol{\kappa} = [\kappa_1, \kappa_2, ..., \kappa_N]' = [\frac{2}{n}\sum_{i=1}^{n}(y_1 - x_{i1})^2, ..., \frac{2}{n}\sum_{i=1}^{n}(y_N - x_{iN})^2]'$ is the average squared distance between the test pattern and all the training patterns. Thus, the similarities between $\boldsymbol{y}$ and all the training patterns are computed. Then the *fuzzy-rough ownership value* [4] of $\boldsymbol{y}$ for the cth class is computed as

$$o_y(c) = \frac{1}{n}\sum_{i=1}^{n} \mu_c(\boldsymbol{x}_i) \exp(-d(\boldsymbol{y}, \boldsymbol{x}_i)^{1/(q-1)}) \quad (1)$$

where $\mu_c(\boldsymbol{x}_i)$ is the initial class membership value assigned to the training pattern $\boldsymbol{x}_i$ for the class c. Here q determines the shape of the exponential. The role of q is quite similar to the *index of fuzziness* in the concentration and dilation operators [2], and the *index of fuzziness* in fuzzy C-means clustering algorithm [2]. Using the information supplied by the training patterns, for any test input, the output of the FRNN algorithm indicates the amount of fuzzy-roughness is present in the relationship between the input representation and the output class.

Next, we slightly modify the FRNN algorithm. Each training pattern is also considered as the test pattern. We seek to know how much fuzzy-rough ownership value at each training pattern is supported by the reaming training patterns. The fuzzy-rough ownership values of the training patterns represent two significant points: (a) how dense/compact the neighborhood of the training pattern is, and (b) whether the neighboring training patterns are from the same class. Thus, the fuzzy-rough ownership value aids us to locate the dense and *homogeneous* regions. Note that a region is called homogeneous if all patterns from the region belong to the same class. The fuzzy-rough ownership value is effective even if the output classes are overlapping or fuzzy. The high value of the fuzzy-rough ownership value

indicates that the neighboring region of the training pattern has less fuzzy-roughness, i.e., the neighboring region is compact and homogeneous. For each training pattern, the fuzzy-rough ownership value (o) and the corresponding value of κ are passed to the next step, where this information is exploited to cluster the input space in a nonparametric manner.

Clustering: The fuzzy-rough ownership values provide clues about the compact and homogeneous region in the training data. In this step, we choose the training patterns that represent the most compact and homogeneous region for each class. We follow a technique similar to *mountain clustering* [7]. For each class, all the training patterns are the candidates for the cluster centers. Without losing the generality, let us consider the cth class. The training pattern with the highest fuzzy-rough ownership value for the cth class is chosen as the first cluster center for the class c. Suppose it is the jth training pattern x_j with the spread κ. Note that the fuzzy-rough ownership values corresponding to noise and outliers will be close to zero, and hence it is less likely that they would be chosen as the cluster centers (see Figure 1). Then the fuzzy-rough ownership value of each training pattern, particularly the training pattern closest to the recent cluster center, is reduced so that two neighboring patterns cannot be the cluster centers. Specifically,

$$o_i(c) = \left[1 - \exp\left(-\sum_{k=1}^{N} \frac{1}{\kappa_k}(x_{jk} - x_{ik})^{2/(\tau-1)}\right)\right] o_i(c) \;\forall i \qquad (2)$$

Here $\tau \in (1, \infty)$ determines the importance we assign to the distance concept while subtracting values. Note that for all values of τ, $o_j(c) = 0$, and hence the jth training pattern cannot be selected as the next cluster center. Now the training pattern with the maximum updated fuzzy-rough ownership value is chosen as the next cluster center. Again, the fuzzy-rough ownership values of all the training patterns are updated. This process is repeated until the ratio of the fuzzy-rough ownership value of the recently chosen cluster center and the first cluster center becomes less than some threshold value. Adopting this method, for the c th class we obtain r_c clusters. Repeating this procedure for all the classes, we obtain total $\sum_{c=1}^{C} r_c$ clusters for all the classes. In the subsequent discussion, we represent the center, spread and the output values of the ith cluster for the cth class by z_i, v_i and $u_i(c)$, respectively.

Rule extraction: Each cluster corresponds to a fuzzy rule that relates a region in the input space to an output class. The right column in Table 1 contains the information obtained from the clustering, and the left column shows the corresponding interpretation in terms of the fuzzy rules. The ith rule of the cth class has the following format:

IF
  the 1st dimension is *around* z_{i1} AND
  ..... AND
  the Nth dimension is *around* z_{iN},
THEN
  the input belongs to cth class with confidence $u_i(c)$.

In the above rule, the fuzzy linguistic variable *around* for the kth dimension is defined as a Gaussian with center z_{ik} and spread $\sqrt{v_{ik}/2}$. The number of fuzzy rules for each class is equal to the number of clusters for that class (Table 1). Since the fuzzy rule base contains r_c fuzzy rules for the cth class, it contains total $\sum_{c=1}^{C} r_c$ fuzzy rules.

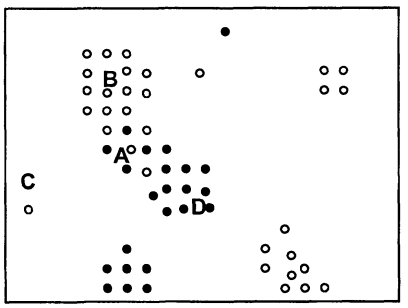

Figure 1 - In this 2-class hypothetical example, each training pattern has two dimensions. The training patterns are shown in white and dark circles to indicate that they belong to class 1 and class 2, respectively. The fuzzy-rough ownership value for the class 1 at the training pattern B is higher than that of at A, because close to B all the patterns belong to the same class, but close to A, the patterns belong to both the classes. Hence, compared to the pattern A, it is more likely that the patterns B (similarly pattern D) will be the cluster centers. The fuzzy-rough ownership value for the class 1 at the pattern C (which is an outlier) is low since no other training pattern is close to it. In this manner, the fuzzy-rough ownership value does not favor the pattern C to be a cluster center. Thus, the fuzzy-rough ownership values enable us to select the good cluster centers in a nonparametric fashion.

Table 1 - The relationship between the fuzzy rules and the corresponding fuzzy clusters.

Fuzzy Rules	**Fuzzy Clusters**
No. of fuzzy-rules =	No. of clusters
No. of input variables in IF part =	No. of dimension of each training pattern
Center of the linguistic variable along a dimension =	Cluster center along that dimension
Spread of each linguistic variable =	Spread of the corresponding cluster
Confidence factor of the THEN part =	Fuzzy-rough ownership value at the cluster center

INPUT:

Training data $\{x_i \mid i = 1, 2, \ldots, n\}$ with initial fuzzy class labels $\mu_c(x_i) \forall i, c$.

ALGORITHM:

(a) Fuzzy-Rough nearest neighbors algorithm

Initialize $o_j(c) = 0$ for $j = 1, 2, \ldots, n$ and $c = 1, 2, .., C$.

FOR $i = 1$ to n

FOR $j = 1$ to n

Compute $K_k = \frac{2}{n}\sum_{p=1}^{n}(x_{jk} - x_{pk})^2 \quad \forall k \in \{1, .., N\}$

Determine the squared weighted distance between x_j and x_i using $d(x_j, x_i) = \sum_{k=1}^{N}\frac{1}{K_k}(x_{jk} - x_{ik})^2$

FOR $c = 1$ to C

$$o_j(c) = o_j(c) + \frac{1}{n}\mu_c(x_i)\exp(-d(x_j, x_i)^{1/(q-1)})$$

END FOR

END FOR

END FOR

(b) Clustering

FOR $c = 1$ to C

$i = 1$

DO

$i = i + 1$

Choose the training pattern with maximum $o(c)$.

Record it as the new cluster center, and call it z_i.

Record K of the training pattern corresponding to the new cluster, and call it spread v_i.

Record $o(c)$ of the training pattern corresponding to the cluster center, and call it $u_i(c)$.

IF $i = 1$

$max = o(c)$

ENDIF

FOR $l = 1$ to n,

Update the fuzzy-rough ownership value of the training pattern using

$$o_l(c) = \left[1 - \exp(-\sum_{k=1}^{N}\frac{1}{K_k}(x_{jk} - x_{lk})^{2/(\tau-1)})\right] o_l(c)$$

ENDFOR

DOWHILE $(o(c)/max >$ THRESHOLD) AND $(i < n)$

ENDFOR

(c) Extraction of fuzzy rules

FOR $c = 1, 2, \ldots, C$

FOR $i = 1, 2, \ldots, r_c$

FOR $k = 1, 2, \ldots, N$

Consider the kth dimension of the ith cluster as the linguistic variable "*amount*" with the center z_{ik} and spread v_{ik}.

END FOR

Make confidence in the THEN part equal to $o_i(c)$.

Construct IF part of the ith fuzzy rule for the cth class using the linguistic variables and the confidence factor.

END FOR

END FOR

OUTPUT:

A fuzzy rule base consisting of $\sum_{c=1}^{C} r_c$ rules.

Figure 2 - Proposed training algorithm to generate fuzzy rules. Here C is the total number of classes, $q \in [1, \infty)$ is an index that controls the fuzziness while finding the similarity between two patterns, and $\tau \in (1, \infty)$ influences the updating of the fuzzy-rough ownership values. The constant THRESHOLD is used to control the number of clusters.

Rule Base Testing

The test pattern is fed to the fuzzy rule base. Based on the weighted distance between each rule and the test pattern, i.e., $d(y, z_i) = \sum_{k=1}^{N}\frac{1}{v_{ik}}(y_k - z_{ik})^2$, the IF part of each fuzzy rule fires. Depending on the firing strength, the THEN part of each rule is activated. The outputs (i.e., u) corresponding to the THEN parts of all the rules for a particular class (say cth class) are aggregated. Particularly,

$$out(c) = \frac{1}{r_c}\sum_{i=1}^{r_c} u_i(c)\exp(-d(y, z_i)^{1/(q-1)}) \quad (3)$$

The class label is determined based on the class that provides the maximum value for *out*. If no crisp class label is needed, then *out* can be used to indicate the belongingness of the input into different classes (Figure 3).

The advantages of the proposed algorithm are as follows: (a) it does not need to know the required number of rules, (b) it has the possibilistic classification ability, (c) it is fast to design and test, (d) unlike other parametric and semiparametric rule extraction techniques, the proposed method does not need any *a priori* structural information about the training data, (e) it is a simple algorithm, and (f) unlike feedforward neural networks with backpropagation learning, it does not have any convergence problem.

Results and Discussion

We preprocess the data as follows: The data set contains 16 samples each with one missing attribute. The 683 samples (339 malignant and 444 benign) are split randomly into a training set that consists of 119 malignant and 222 benign samples. The test set consists of the remaining 120 malignant and 222 benign samples.

Using q = 2, $\tau = 2$ and THRESHOLD = 0.8, we have obtained 1 rule for the class malignant and 2 rules for the class benign. One such extracted rule is

IF
the clump thickness is *around* 2 AND
the uniformity of cell size is *around* 1 AND
the uniformity of cell shape is *around* 1 AND
the marginal adhesion is *around* 1 AND
the single epithelial cell size is *around* 2 AND
the bare nuclei is *around* 1 AND
the bland chromatin is *around* 2 AND
the normal nucleoli is *around* 1 AND
the mitosis is *around* 1,
THEN
the output class is malignant with confidence 0.4.

The linguistic variable *around* is considered as a fuzzy number. The shape of the fuzzy number *around* for the clump thickness is shown in *Figure 4*.

INPUT:
(i) Fuzzy rule base generated in the training phase.
(ii) Test pattern y
ALGORITHM:
Initialize $out(c) = 0$ for all c
FOR $c = 1$ to C
FOR $i = 1$ to r_c
Determine the squared weighted distance between y and the IF part of the ith rule using

$$d(y, z_i) = \sum_{k=1}^{N} \frac{1}{v_{ik}} (y_k - z_{ik})^2$$

Calculate

$$out(c) = out(c) + \frac{1}{r_c} u_i(c) \exp(-d(y, z_i)^{1/(q-1)})$$

END FOR
END FOR
Class label of y is j where $out(j) = \max\{out(1), out(2), \ldots, out(c)\}$.
OUTPUT:
(i) Class label of y.
(ii) Class confidence values $out(c)$ for all c.

Figure 3 - Proposed testing algorithm.

If the constant THRESHOLD is decreased further, more rules are generated and the classification efficiency increases further. However, the increase of the number of rules makes it more difficult to understand the diagnosis process.

Although this paper reports the experimental results on the breast cancer problem, the same technique can be used for the other diagnosis problems where the inputs features are real numbers. This paper does not attempt to identify the prognostic factors that are more important for the diagnosis. In other words, the feature selection task is left for the future research.

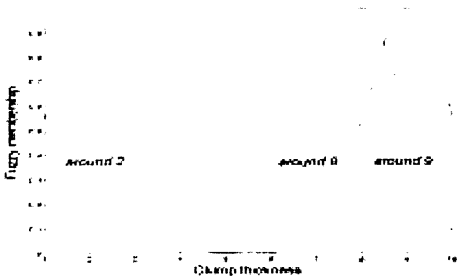

Figure 4 - The fuzzy numbers around 2, around 8 and around 9 are shown for the prognostic factor clump thickness.

Acknowledgement

A strategic research grant RP960351 from the National Science and Technology Board and the Ministry of Education, Singapore, has supported the work of this paper.

References

[1] Blake CL, and Merz CJ. UCI repository of machine learning databases, http://www.ics.uci.edu/~mlearn, 1998.

[2] Klir GS, and Yuan B. *Fuzzy Sets and Fuzzy Logic -- Theory and Applications*. Englewood Cliffs, NJ, Prentice Hall, 1995.

[3] Nauck D, and Kruse R. Obtaining interpretable fuzzy classification rules from medical data. *Artificial Intelligence in Medicine*, vol. 16, pp. 149-169, 1999.

[4] Sarkar M, and Leong TY. Application of *K*-nearest neighbors algorithm on breast cancer problem. *Proc. of American Medical Informatics Assoc. (AMIA) Symposium*, Los Angeles, USA, pp. 759-763, November 3-8, 2000.

[5] Sarkar M. Fuzzy-Rough nearest neighbors algorithm. *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, Nashville, Tennessee, USA, pp. 3556–3561, October 8-11, 2000.

[6] Setanio R. Generating concise and accurate classification rules for breast cancer diagnosis. *Artificial Intelligence in Medicine*, vol.18, pp. 205-219, 2000.

[7] Yager RR, and Filev DP. Approximate clustering via the mountain method. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 8, pp. 1279-1284, August, 1994.

Address for correspondence
manish, leongty}@comp.nus.edu.sg