Proceedings of the 2007 IEEE Symposium on Foundations of Computational Intelligence (FOCI 2007)

Goodness of Fit: Measures for a Fuzzy Classifier

Oliver Buchtala and Bernhard Sick
Faculty of Computer Science and Mathematics, University of Passau, Germany
e-mail: {buchtala, sick}@fmi.uni-passau.de

***Abstract*— The understandability of rule sets is an important issue in knowledge discovery, where classification rules, for example, are extracted from large data sets. An important criterion in this context is the goodness of fit of a given classifier, i.e., a measure that gives an quantitative answer to the question, how good a classifier fits to the data it has to classify. In this article we provide an appropriate measure for a Mamdani-type fuzzy classifier with Gaussians and singletons as membership functions, sum-prod inference, and height method for defuzzification. That is, goodness of fit must be measured for multivariate Gaussian mixture models. Therefore, we adopt conventional test methods for univariate, unimodal probability distributions (e.g., Kolmogorov-Smirnov for chi-square), provide a measure for the goodness of fit of our fuzzy classifier, and discuss its properties. In a second step we go even beyond this point by showing how this measure could be extended to an analysis tool that gives detailed hints which rules or which membership functions are not suitably realized.**

I. Introduction

The parameters of fuzzy classifiers (i.e., the parameters of membership functions and the rules) can be found in various ways:

- Unsupervised training methods (e.g., clustering techniques, [1]) require unlabeled data (samples).
- Supervised training methods without a continuously differentiable error function require labeled data (e.g., evolutionary algorithms [2] or iterative adaptation rules for certain neuro-fuzzy classifiers [3]).
- Supervised techniques with a continuously differentiable error function (e.g., first-order training methods such as Resilient Propagation [4] or second-order methods such as Scaled Conjugate Gradients [5]) also require labeled data but also a suitably defined fuzzy paradigm.
- Experts may also be able to adjust the parameters of a fuzzy classifier according to their needs [6].

One important motivation for the use of a fuzzy system is often the *interpretability* and *understandability* of the contained rules. The former says that the rules can easily be read by humans. This is always possible in the case of fuzzy rules (in contrast to rules embodied in a backpropagation neural network, for instance). The latter says that an application expert comprehends the meaning of each rule within the context of a given application. For a fuzzy rule, for example, this is easier if the premises of these fuzzy rules model clusters within the data and harder if the rules do not model the data, but the decision boundary between different classes. Fuzzy rules model the data, for example, if membership function are found by means of expectation maximization algorithms (e.g., [7]). They model the decision boundary, for example, if they were extracted from Support Vector Machine classifiers (e.g., [8], [9]). In a nutshell: There are many ways to build a fuzzy classifier but only a few result in actually *understandable* rule sets.

Here, we change the viewpoint: Given a fuzzy classifier (produced by a supervised training method, a human expert, or any other suitable technique), how can we measure how good this classifier does fit to the data it has to classify (so-called *goodness of fit*, GoF)? Given an appropriate solution for that problem we can also estimate whether the classifier can be understood by humans (cf. [10]) and it might even be possible to give hints which rules or which membership functions are not suitably realized. In order to achieve this goal we first define an appropriate Mamdani-type fuzzy classifier paradigm (Section II). Then we adopt methods from the field of statistics to measure GoF, demonstrate their properties, and discuss approaches to improve them (Section III). Finally, we summarize the major findings and give an outlook to our future work (Section IV).

II. Fuzzy Classifier Paradigm

In this section we define the fuzzy classifier paradigm for which we provide appropriate measures for GoF. Basically, this classifier is a Mamdani-type fuzzy system (FS) with Gaussians and singletons as membership functions, sum-prod inference, and height method for defuzzification.

We are given a rule set $\mathcal{H}$ as shown in Table I, consisting of rules $h = 1, \dots, H$ with the x_i $(i = 1, \dots, I)$ being linguistic input variables and the y_o $(o = 1, \dots, O)$ being linguistic output variables each corresponding to a class. The $\varphi_{(i,h)}$ are membership functions (linguistic terms) belonging to input variables and rules, realized by Gaussians as follows:

$$\varphi_{(i,h)} \stackrel{\text{def}}{=} e^{\left(-\frac{(c_{(i,h)}-x_i)^2}{r_{(i,h)}{}^2}\right)},$$

with $c_{(i,h)}, r_{(i,h)} \in \mathbb{R}$ being the parameters of the Gaussians (so-called center and radius). The $w_{(h,o)} \in [0;1]$ are membership functions (linguistic terms) belonging to rules and output variables which are realized by singletons. The values can be interpreted as class membership degrees.

The rule set is evaluated as follows:

- The evaluation of the premise of a single rule is realized by the product operator.
- An implication is realized by the product operator, too.
- The sum operator is taken to combine the rules.
- For defuzzification we apply the height method.

TABLE I
FUZZY RULE SET $\mathcal{H}$.

$$
\begin{array}{c}
\text{if } x_1 \text{ is } \varphi_{(1,1)} \ \ldots \text{ and } x_i \text{ is } \varphi_{(i,1)} \ldots \text{ and } x_I \text{ is } \varphi_{(I,1)} \text{ then } y_1 \text{ is } w_{(1,1)} \ \ldots \text{ and } y_o \text{ is } w_{(1,o)} \ \ldots \text{ and } y_O \text{ is } w_{(1,O)} \\
\vdots \qquad \vdots \\
\text{if } x_1 \text{ is } \varphi_{(1,h)} \ \ldots \text{ and } x_i \text{ is } \varphi_{(i,h)} \ldots \text{ and } x_I \text{ is } \varphi_{(I,h)} \text{ then } y_1 \text{ is } w_{(h,1)} \ \ldots \text{ and } y_o \text{ is } w_{(h,o)} \ \ldots \text{ and } y_O \text{ is } w_{(h,O)} \\
\vdots \qquad \vdots \\
\text{if } x_1 \text{ is } \varphi_{(1,H)} \ \ldots \text{ and } x_i \text{ is } \varphi_{(i,H)} \ldots \text{ and } x_I \text{ is } \varphi_{(I,H)} \text{ then } y_1 \text{ is } w_{(H,1)} \ \ldots \text{ and } y_o \text{ is } w_{(H,o)} \ \ldots \text{ and } y_O \text{ is } w_{(H,O)}
\end{array}
$$

- A winner-takes-all approach is used to decide on the class assignment of the input.

That is, we compute the output vector (class vector) $\mathbf{y}(k) \stackrel{\text{def}}{=} (y_1(k), ..., y_o(k), ..., y_O(k))$ with

$$y_o(\mathbf{x}(k)) = \sum_{h=1}^{H} w_{(h,o)} \cdot \frac{\varphi_h(\mathbf{x}(k))}{\sum_{h=1}^{H} \varphi_h(\mathbf{x}(k))},$$

where

$$\varphi_h(\mathbf{x}(k)) = \prod_{i=1}^{I} \varphi_{(i,h)}(x_i(k)) =$$

$$= \prod_{i=1}^{I} e^{\left(-\frac{(c_{(i,h)} - x_i(k))^2}{{r_{(i,h)}}^2}\right)},$$

the input vector $\mathbf{x}(k) \stackrel{\text{def}}{=} (x_1(k), ..., x_i(k), ..., x_I(k))$ (sample) and the number of the input vector $k = 1, \ldots, N$. Finally, we select the class o with the highest output value $y_o(\mathbf{x}(k))$.

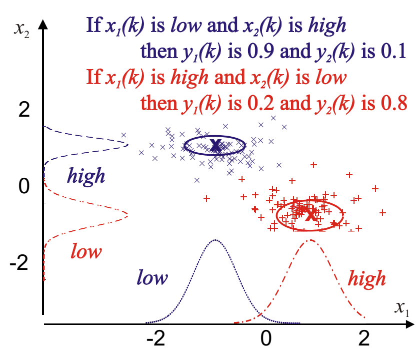

Fig. 1. Example of a fuzzy classifier consisting of two rules operating in a two-dimensional input space ($I = 2$ and $H = 2$).

The vectors $\mathbf{c}_h \stackrel{\text{def}}{=} (c_{(1,h)}, ..., c_{(i,h)}, ..., c_{(I,h)})$ and $\mathbf{r}_h \stackrel{\text{def}}{=} (r_{(1,h)}, ..., r_{(i,h)}, ..., r_{(I,h)})$ describe an axes-oriented hyperellipsoid in the input space of the fuzzy classifier. Thus, $\mathbf{c}_h$ can be regarded as a center of a hyperellipsoidal cluster – big x in Figure 1 – and $\mathbf{r}_h$ defines the shape of the cluster – ellipses in Figure 1. The activation of a rule describes the similarity of an input pattern $\mathbf{x}(k)$ and a center based on a certain matrix norm (Mahalanobis distance measure with diagonal covariance matrix).

The parameters of a fuzzy classifier can be determined by means of training algorithms such as gradient-based techniques or clustering techniques in combination with methods for the solution of linear least-squares (LLS) problems (see, e.g., [11] and our own work in [12], [13]). However, linguistic terms and rules could be defined by experts as well.

In accordance with the discussions and suggestions in [14], [15], [16] we defined this fuzzy classifier in a way such that it can also be interpreted as a neural network (radial basis function network, cf. [17], [18]). Thus, we gain the advantages of two worlds: Trainability of neural networks and interpretability – not yet understandability! – of FS. Here, we described the classifier as a Mamdani-type FS, but we could also see it as a special case of a Takagi-Sugeno-type FS.

III. MEASURES AND RESULTS

In this section we introduce a technique that can be used to evaluate the understandability of fuzzy rule systems (as defined above) by measuring the GoF. Formally, for a given set of rules $\mathcal{H} \stackrel{\text{def}}{=} \{h \mid h = 1, \ldots, H\}$ we want to determine a value $\eta_{\mathcal{F},\mathcal{H}}$ that represents the GoF. Here, we have chosen a statistical solution for this problem. $\mathcal{H}$ is interpreted as a statistical model of an underlying stochastic process X that produces the data being observed. Each rule premise describes a (local) function on the input space, which can be seen as a density function. Thus, when the SUM-operator is used for rule composition, the whole rule system defines a density function of a mixture model (in our case, a multivariate Gaussian mixture model). The usage of rules with Gaussian premises is motivated by the generalized central limit theorem: Processes with multi-causal origination tend to be normally distributed.

In the following, a short overview of classical GoF tests is given, which are typically applied to univariate and unimodal models. After that, a simple extension is introduced which allows to apply such a GoF test to multivariate, unimodal models, and some of its basic properties are demonstrated using simple examples. Then, a further improvement is proposed that gives even more detailed insight into the GoF of each rule. Finally, the method is extended to mixture models.

A. Statistical GoF Tests

In the literature there are basically two well-known statistical methods to determine the GoF of a given model (see, e.g., [19], [20]): the χ^2-test (chi-square) and the Kolmogorov-Smirnov (KS) statistics. The former is computed by partitioning the input space into bins. For each bin the actual number of data points and the expected number of data points are determined. The χ^2-test delivers a value that can then be used to discard the model (hypothesis) at a given level of significance. In general, the computation of the integral of the distribution function is very difficult or even impossible. Thus, this method is not appropriate in our case. The KS-test is based on the empirical cumulative distribution function (ECDF) of the data. The ECDF is compared to the cumulative distribution function $\mathcal{F}$ (CDF) of the assumed model. The statistic is computed as the maximum absolute difference between the ECDF and the CDF. A detailed description of this method can be found in [21], for instance. Algorithm 1 describes a more general variant of this approach, that allows weighted data points which we need for the methods introduced in a later section. Therefore, we define weights $\lambda_h(k)$ for each rule h and each pattern k that are probabilistic, i.e., they are positive and sum up to one. In the standard KS-test the data are weighted uniformly, i.e., $\lambda_h(k) \stackrel{\text{def}}{=} \frac{1}{N}$. It has to be emphasized that the KS-test can be applied to any arbitrary model for which the CDF is known.

Input: CDF $\mathcal{F}(x)$, data points $x(k)$, weights $\lambda_h(k)$, with $k \in 1, \dots, N$.
Output: KS-statistic $\eta_{\mathcal{F}}$.
Determine the permutation $\pi(k)$ such that $x(\pi(k))$ are in ascending order ($k \in 1, \dots, N$).
Set $E(0) \stackrel{\text{def}}{=} 0$.
for $k \in 1, \dots, N$ **do**
  Set $E(k) \stackrel{\text{def}}{=} E(k-1) + \lambda_h(\pi(k))$.
  Set $d_l(k) \stackrel{\text{def}}{=} \mathcal{F}(x(\pi(k))) - E(k-1)$.
  Set $d_u(k) \stackrel{\text{def}}{=} E(k) - \mathcal{F}(x(\pi(k)))$.
  Set $\eta_{\mathcal{F}}(k) \stackrel{\text{def}}{=} \max(d_l(k), d_u(k))$.
end
Set $\eta_{\mathcal{F}} \stackrel{\text{def}}{=} \max_{k \in 1,\dots,N} \eta_{\mathcal{F}}(k)$.

Algorithm 1: Computation of the KS-statistic $\eta_{\mathcal{F}}$ for weighted data.

Figure 2 shows the CDF of the normal distribution and the ECDF for 100 data points generated randomly with the same distribution. The KS-statistic represents the maximum absolute difference between these two curves.

To apply this test to our problem (multivariate Gaussian mixture models) there are two key problems that must be solved: First, we have to extend the statistic to be applicable to multivariate Gaussian distributions. Second, we must find a way to adopt the test to Gaussian mixture models.

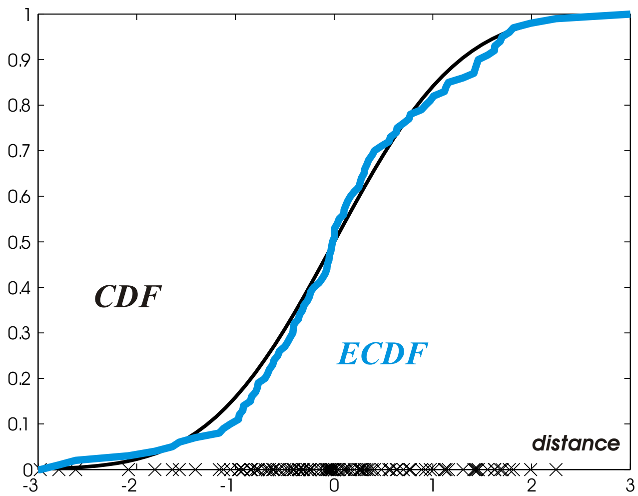

Fig. 2. CDF and ECDF for a normally distributed data set.

B. Testing Multivariate Normal Models

The basic idea of a test for the multivariate case based on the proposed KS-statistic is to find an appropriate mapping function $\psi : \mathbb{R}^I \mapsto \mathbb{R}$ and to perform the KS-test on the transformed univariate data. However, to apply the test it is necessary to know the distribution function of the transformed data.

KS-Test Using χ^2: A very useful transformation is motivated by the following property of the χ^2-distribution: Given data points $\mathbf{x}(k)$ with $k = 1, \dots, N$ that are multinormally distributed with parameters $\mu = (\mu_1, ..., \mu_I)$ and $\sigma = (\sigma_1, ..., \sigma_I)$, then the squared Mahalanobis distances of $\mathbf{x}(k)$ to μ are χ^2-distributed with parameter $I - 1$ (cf., e.g., [22]). In our case, the squared Mahalanobis distance for a rule h is defined by

$$d_h(k) \stackrel{\text{def}}{=} \sum_{i=1}^{I} \left(\frac{x_i(k) - c_{(i,h)}}{r_{(i,h)}} \right)^2 .$$

The CDF $\mathcal{F}_{\chi^2}$ can be computed using the Γ-function (see, e.g., [22]).

Input: Rule h with parameters $\mathbf{c}_h$ and $\mathbf{r}_h$, data points $\mathbf{x}(k)$, weights $\lambda_h(k)$, with $k \in 1, \dots, N$.
Output: KS-statistic $\eta_{\mathcal{F}_{\chi^2},h}$ for rule h.
for $k \in 1, \dots, N$ **do**
  Compute the squared Mahalanobis distance $d_h(k)$.
end
Compute the KS-statistic $\eta_{\mathcal{F}_{\chi^2},h}$ using Algorithm 1 for the values $d_h(k)$ with weights $\lambda_h(k)$ and CDF $\mathcal{F}_{\chi^2}$.

Algorithm 2: KS-statistic $\eta_{\mathcal{F}_{\chi^2},h}$ for a single rule using the χ^2-CDF.

Figure 3 demonstrates the test for a simple one-rule model (i.e., fuzzy classifier with one rule): There is almost no

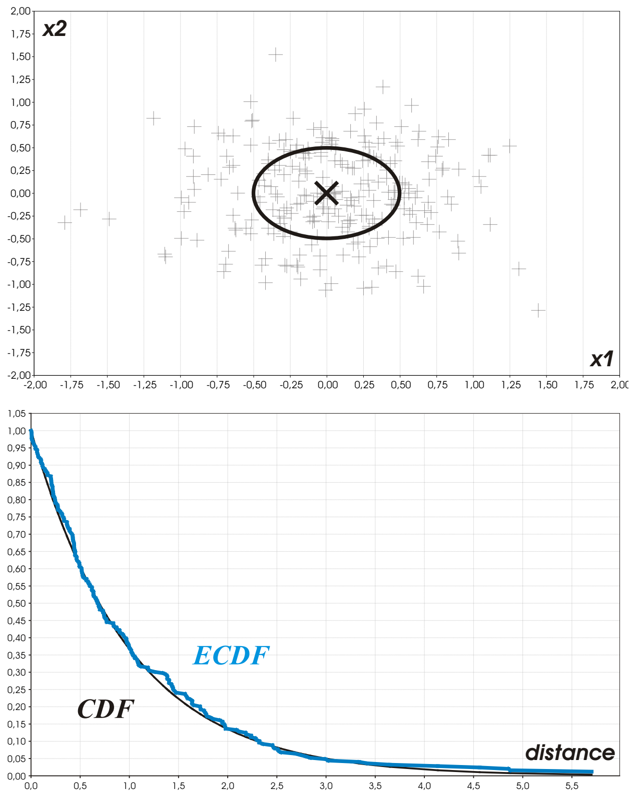

Fig. 3. Example 1: KS-test for a one-rule model that has a good data fit.

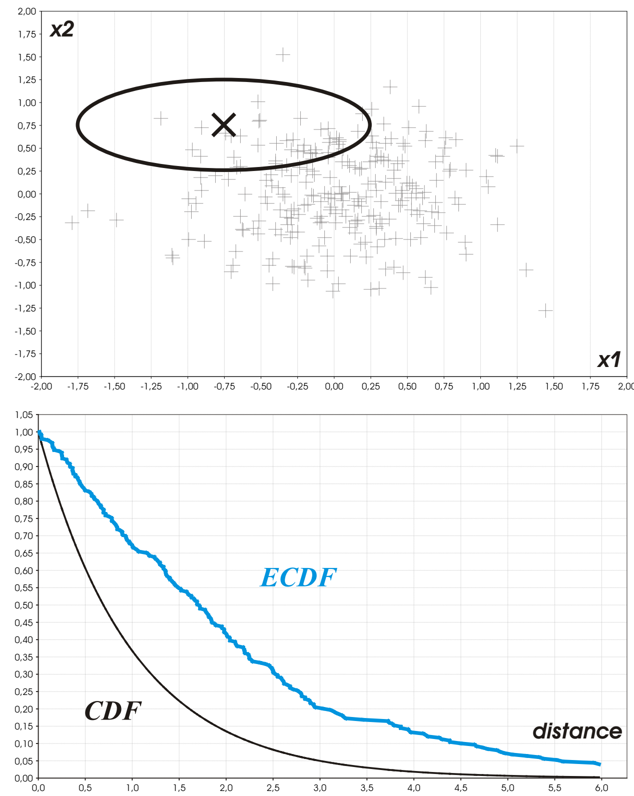

Fig. 4. Example 2: KS-test for a one-rule model that has a bad data fit.

difference between the CDF and the ECDF. By contrast, Figure 4 shows the test for a rule that does not fit the data very well.

One advantage of this method is that the determination of the squared Mahalanobis distance is a side-product of the rule activation and, therefore, has not to be computed explicitly. However, there is one major drawback of this approach which is illustrated in Figure 5. As the transformation is radial-symmetric, it is not possible to take the actual spatial distribution of the patterns correctly into account. Only the scalar distances to the prototype of a rule are considered.

Combination of Univariate KS-Tests: Another solution to the mapping problem is to use projections onto the main axes. In our case we assume the data to be uncorrelated in the different dimensions of the input space (i.e., we have diagonal covariance matrices). Thus, the projection of multivariate, normally distributed data results in univariate, normally distributed data and we can apply a KS-test using the CDF of a Gaussian distribution. To assess a complete rule, we compute the statistic for each of the input dimensions. The CDF $\mathcal{F}_{\mathcal{N}(\mu,\sigma)}$ of a normal distribution can be computed efficiently using the complementary error function (see, e.g., [22]). Algorithm 3 describes this computation more formally. The univariate squared Mahalanobis distance $d_{i,h}(k)$ for a rule h, an input dimension i and a pattern $x_i(k)$ is defined by

$$d_{i,h}(k) \stackrel{\text{def}}{=} \left(\frac{x_i(k) - c_{(i,h)}}{r_{(i,h)}}\right)^2 .$$

Input: Rule h with parameters $\mathbf{c}_h$ and $\mathbf{r}_h$, data points $\mathbf{x}(k)$, weights $\lambda_h(k)$ with $k \in 1, \ldots, N$.
Output: KS-statistic $\eta_{\mathcal{F}_{\mathcal{N}(\mathbf{c}_h,\mathbf{r}_h)},h}$ for rule h, KS-statistic $\eta^{(i)}_{\mathcal{F}_{\mathcal{N}(\mathbf{c}_h,\mathbf{r}_h)},h}$ for rule h and each input dimension $i \in 1, \ldots, I$.
for $i \in 1, \ldots, I$ **do**
  for $k \in 1, \ldots, N$ **do**
    Compute the (univariate) Mahalanobis squared distance $d_{i,h}(k)$.
  end
  Compute the KS-statistcs $\eta^{(i)}_{\mathcal{F}_{\mathcal{N}(\mathbf{c}_h,\mathbf{r}_h)},h}$ using Algorithm 1 for values $d_{i,h}(k)$ with weights $\lambda_h(k)$ and CDF $\mathcal{F}_{\mathcal{N}(\mathbf{c}_h,\mathbf{r}_h)}$.
end
Set $\eta_{\mathcal{F}_{\mathcal{N}(\mathbf{c}_h,\mathbf{r}_h)},h} \stackrel{\text{def}}{=} \max_{i\in 1,\ldots,I} \eta^{(i)}_{\mathcal{F}_{\mathcal{N}(\mathbf{c}_h,\mathbf{r}_h)},h}$.

Algorithm 3: KS-statistic $\eta_{\mathcal{F}_{\mathcal{N}(\mathbf{c}_h,\mathbf{r}_h)},h}$ for a single rule h.

Figure 6 shows this test applied to example 3 (cf. Figure

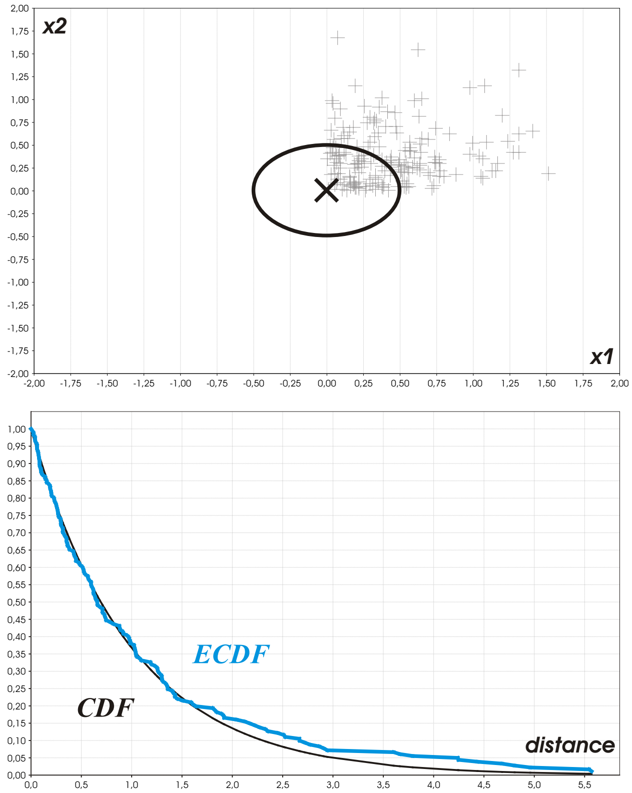

Fig. 5. Example 3a: The χ^2 based test does not reveal the spatial anomaly.

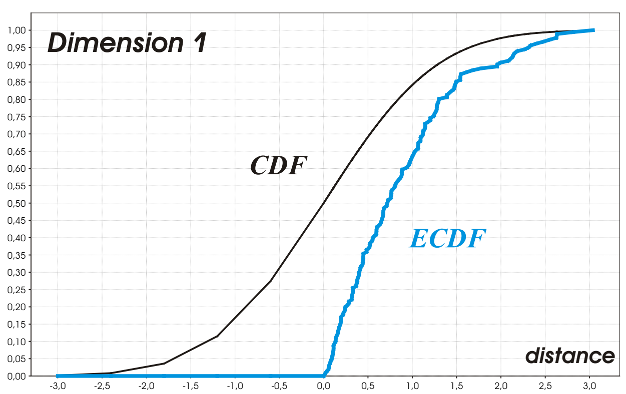

Fig. 6. Example 3b: The combination of univariate tests allows the detection of the spatial anomaly.

5). Here, the degenerated spatial distribution is represented by the curves. A major advantage of this method can be seen in Figure 7: A more detailed assessment of rules is possible. Certain weaknesses of a given model can be identified by the characteristics of the ECDF (bad radius, bad center) and, moreover, this information is available for each input dimension separately.

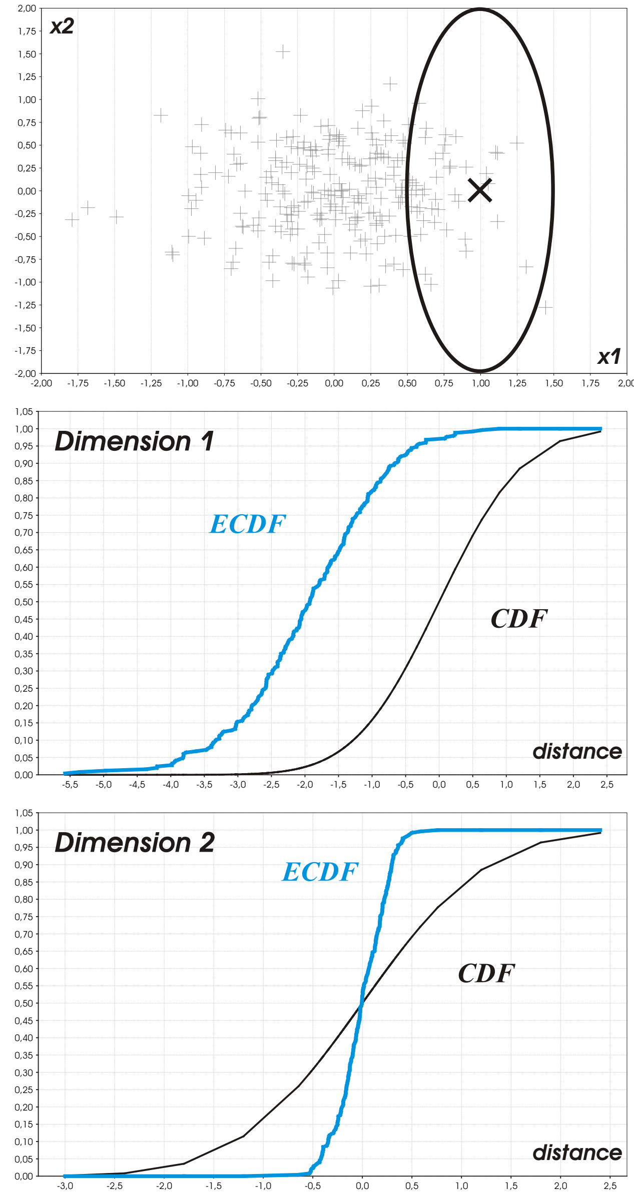

Fig. 7. Example 4: The combination of univariate tests reveals more detailed information about single parameters: Bad center in input dimension 1 and too large radius in input dimension 2.

C. GoF Tests for Mixture Models

To allow the application of the proposed statistics to Gaussian mixture models we introduce the following method which is motivated by techniques known from fuzzy clustering (see, e.g., [23]). The basic idea is to decompose the problem of testing a mixture model into several tests of unimodal models. This is done by associating the data points to the different rules. As displayed in Figure 8, it is possible that different unimodal distributions may be very close to each other and their data may overlap. To preserve the spatial distribution,

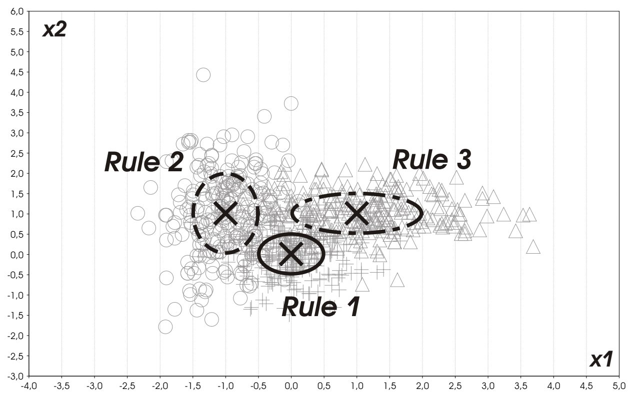

Fig. 8. Example 5: Mixture model containing very close rules.

a "fuzzy" association of data to rules is necessary. For each pair of pattern $\mathbf{x}(k)$ and rule h a weight $\lambda_h(k)$ is computed using Algorithm 4 where $\lambda_h(k)$ can be seen as the probability that a pattern $\mathbf{x}(k)$ is associated to a rule h in presence of all rules $1, \ldots, H$. Then, the proposed statistics are applied to each rule separately. The computation of a test for a complete mixture model (i.e., a whole rule set) is defined by Algorithm 5. Figure 9 shows the ECDF and the CDF for Rule 1 for input dimensions 1 and 2.

Input: Rule set $\mathcal{H}$, data points $\mathbf{x}(k)$ with $k \in 1, \ldots, N$.
Output: Weights $\lambda_h(k)$ with $k \in 1, \ldots, N$.
for $k \in 1, \ldots, N$ **do**
  for $h \in 1, \ldots, H$ **do**
    Compute $\varphi_h(\mathbf{x}(k))$.
  end
  Set $\widetilde{\lambda}_h(k) \stackrel{\text{def}}{=} \dfrac{\varphi_h(\mathbf{x}(k))}{\sum_{j=1}^{H} \varphi_j(\mathbf{x}(k))}$.
end
for $k \in 1, \ldots, N$ **do**
  for $h \in 1, \ldots, H$ **do**
    Set $\lambda_h(k) \stackrel{\text{def}}{=} \dfrac{\widetilde{\lambda}_h(k)}{\sum_{l=1}^{N} \widetilde{\lambda}_h(l)}$.
  end
end

Algorithm 4: Computation of weights $\lambda_h(k)$.

Input: Rule set $\mathcal{H}$, data points $\mathbf{x}(k)$, weights $\lambda_h(k)$ with $k \in 1, \ldots, N$.
Output: KS-statistic $\eta_{\mathcal{F},\mathcal{H}}$.
for $h \in 1, \ldots, H$ **do**
  Compute a KS-statistic $\eta_{\mathcal{F},h}$ using Algorithms 2 or 3.
end
Set $\eta_{\mathcal{F},\mathcal{H}} \stackrel{\text{def}}{=} \max_{h \in 1,\ldots,H} \eta_{\mathcal{F},h}$.

Algorithm 5: KS-statistic $\eta_{\mathcal{F},\mathcal{H}}$ for a complete rule set $\mathcal{H}$.

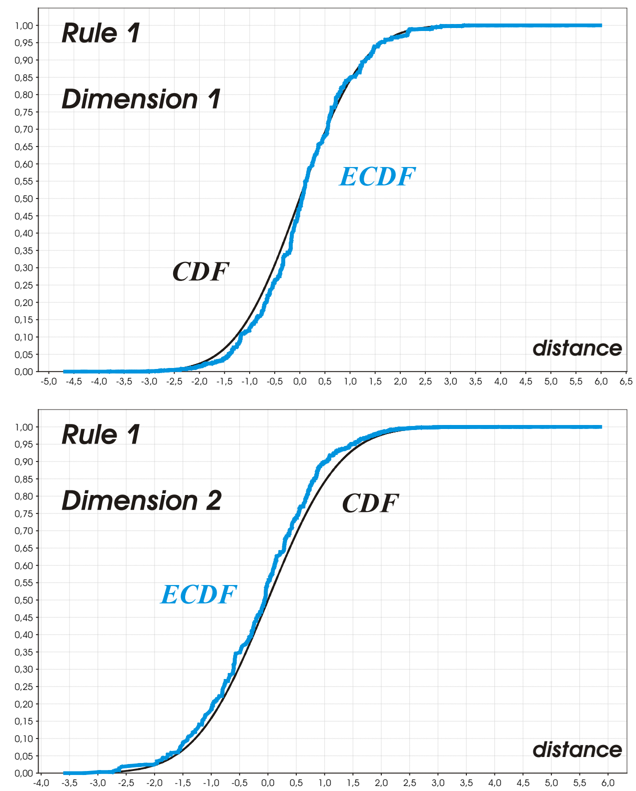

Fig. 9. Example 4: Robust decomposition of test for a mixture model into tests for single rules.

IV. Conclusion and Outlook

In this article, we have shown how the GoF of multivariate Gaussian mixture models can be rated. This allows the assessment of certain Mamdani-type fuzzy classifiers which also can be regarded as radial basis function neural networks which constitute a suitable way to achieve interpretable rules. We also introduced techniques that can be used to assess their "fitness" in some more detail (single rules or membership functions). This way, we are equipped with techniques that enable us to measure the probably most important property regarding the understandability of rules.

In our future work we want to increase the information gain of single parameters. E.g., it is possible to recognize properties of single parameters (center and radii) of a rule premise by evaluating the characteristics of the ECDF. Applying methods known from regression tests this could be done automatically. Furthermore, alternative evaluations using the ECDF could base upon correlation coefficients, for example. As the proposed approach allows the application of arbitrary distribution functions, it is not restricted to the assessment of Gaussian mixture models. Other kinds of fuzzy systems, e.g., fuzzy systems that are able to process categorical data, could be considered easily. This way, many more well-known classifier paradigms could be assessed.

In principle, the techniques introduced in this article can be utilized in any classification application. An important application area where we need those techniques is the field of Organic Computing [24]: We will enable intelligent distributed systems ("individuals") – e.g., teams of robots, software agents, animats, or smart nodes of sensor networks – to learn from each other by exchanging knowledge in form of rules [17], [25]. We assume that the individuals' environment is dynamic. That is, rules must be adapted (learned) on-line. To learn from each other, these individuals must – besides other tasks – assess their own knowledge by means of self-awareness mechanisms and it must be assured that single rules are potentially useful for other individuals. "Potentially" means that they actually are useful if the other individuals must classify similar input data (i.e., with the same underlying distribution). Both is possible if the classifiers actually model the data and this can be verified by means of the measures presented in this article. As we must adapt the rules on-line, we need techniques that enforce a good fit during on-line training of a fuzzy classifier.

Acknowledgements

This work is supported by the German Research Foundation (DFG), grant SI 674/3-1.

References

[1] A. Piegat, *Fuzzy Modeling and Control*. Physica-Verlag, Heidelberg, New York, 2001.

[2] A. A. Freitas, *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer Verlag, Berlin, Heidelberg, New York, 2002.

[3] D. Nauck and R. Kruse, "NEFCLASS – a neuro-fuzzy approach for the classification of data," in *Proceedings of the 1995 ACM Symposium on Applied Computing (SAC 1995),* Nashville, 1995, pp. 461–465.

[4] M. Riedmiller and H. Braun, "A direct method for faster backpropagation learning: The RPROP algorithm," in *Proceedings of the 1993 IEEE International Conference on Neural Networks (ICNN '93),* San Francisco, vol. 1, 1993, pp. 586 – 591.

[5] M. F. Moller, "A scaled conjugate algorithm for fast supervised learning," *Neural Networks*, vol. 6, no. 1, pp. 525 – 533, 1993.

[6] J. Kahlert, *Fuzzy Control für Ingenieure – Analyse, Synthese und Optimierung von Fuzzy-Regelungssystemen*. Friedrich Vieweg & Sohn Verlagsgesellschaft, Braunschweig, Wiesbaden, 1995.

[7] K. G. Manton, M. A. Woodbury, and H. D. Tolley, *Statistical Applications Using Fuzzy Sets*, ser. Probability and Mathematical Statistics. Wiley-Interscience, New York, 1994, vol. 90, no. 431.

[8] X. Fu, C. Ong, S. Keerthi, G. G. Hung, and L. Goh, "Extracting the knowledge embedded in support vector machines," in *IEEE-INNS-ENNS International Joint Conference on Neural Networks,* Budapest, vol. 1, 2004, pp. 291 – 296.

[9] J.-H. Chiang and P.-Y. Hao, "Support vector learning mechanism for fuzzy rule-based modeling: A new approach," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 1, pp. 1 – 12, 2004.

[10] C. Weihs and U. M. Sondhauss, "Combining mental fit and data fit for classification rule selection," in *Exploratory Data Analysis in Empirical Research – Proceedings of the 25th Annual Conference of the Gesellschaft für Klassifikation e.V. 2001,* Munich, ser. Studies in Classification, Data Analysis, and Knowledge Organization, M. Schwaiger and O. Opitz, Eds. Springer Verlag, Berlin, Heidelberg, New York, 2002, pp. 188 – 203.

[11] F. Schwenker, H. Kestler, and G. Palm, "Three learning phases for radial-basis-function networks," *Neural Networks*, vol. 14, no. 4 – 5, pp. 439 – 458, 2001.

[12] O. Buchtala, P. Neumann, and B. Sick, "A strategy for an efficient training of radial basis function networks for classification applications," in *Proceedings of the IEEE-INNS International Joint Conference on Neural Networks,* Portland, vol. 2, 2003, pp. 1025 – 1030.

[13] O. Buchtala, A. Hofmann, and B. Sick, "Fast and efficient training of RBF networks," in *Artificial Neural Networks and Neural Information Processing – ICANN/ICONIP 2003 (Joint International Conference),* Istanbul, ser. LNCS, O. Kaynak, E. Alpaydin, E. Oja, and L. Xu, Eds., no. 2714. Springer Verlag, Berlin, Heidelberg, New York, 2003, pp. 43 – 51.

[14] J.-S. Roger Jang and C.-T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference systems," *IEEE Transactions on Neural Networks*, vol. 4, no. 1, pp. 156 – 159, 1993.

[15] H. C. Andersen, A. Lotfi, and L. C. Westphal, "Comments on "Functional equivalence between radial basis function networks and fuzzy inference systems"," *IEEE Transactions on Neural Networks*, vol. 9, no. 6, pp. 1529 – 1531, 1998.

[16] J.-S. Roger Jang, "Author's reply," *IEEE Transactions on Neural Networks*, vol. 9, no. 6, pp. 1531 – 1532, 1998.

[17] O. Buchtala and B. Sick, "Techniques for the fusion of symbolic rules in distributed organic systems," in *Proceedings of the IEEE Mountain Workshop on Adaptive and Learning Systems (SMCals/06),* Logan, 2006, pp. 85 – 90.

[18] T. Poggio and F. Girosi, "A theory of networks for approximation and learning," Massachusetts Institute of Technology – Artificial Intelligence Laboratory & Center for Biological Information Procesing – Whitaker College," A.I. Memo No. 1140, C.B.I.P. Paper No. 31, 1989.

[19] G. W. Snedecor and W. G. Cochran, *Statistical Methods*, 8th ed. Iowa State University Press, Ames, Iowa, 1989.

[20] Y. I. Ingster and I. A. Suslina, *Nonparametric Goodness-of-Fit Testing Under Gaussian Models*, ser. Lecture Notes in Statistics. Springer Verlag, New York, 2003, vol. 169.

[21] L. Sachs, *Angewandte Statistik: Anwendung statistischer Methoden*, 10th ed. Springer Verlag, Berlin, Heidelberg, New York, 2002.

[22] M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions*, 2nd ed. John Wiley & Sons, New York, 1993.

[23] F. Höppner, F. Klawonn, and R. Kruse, *Fuzzy-Clusteranalyse*. Friedr. Vieweg & Sohn, Braunschweig/Wiesbaden, 1997.

[24] R. Allrutz, C. Cap, S. Eilers, D. Fey, H. Haase, C. Hochberger, W. Karl, B. Kolpatzik, J. Krebs, F. Langhammer, P. Lukowicz, E. Maehle, J. Maas, C. Müller-Schloer, R. Riedl, B. Schallenberger, V. Schanz, H. Schmeck, D. Schmid, W. Schröder-Preikschat, T. Ungerer, H.-O. Veiser, and L. Wolf, "Organic Computing – Computer- und Systemarchitektur im Jahr 2010," VDE/ITG/GI-Positionspapier, 2003.

[25] O. Buchtala and B. Sick, "Functional knowledge exchange within an intelligent distributed system," in *Proceedings of the 20th International Conference on Architecture of Computing Systems – System Aspects in Pervasive and Organic Computing (ARCS 2007),* Zurich, 2007, (accepted – to appear).