Generalized Hidden-Mapping Minimax Probability Machine for the Training and Reliability Learning of Several Classical Intelligent Models

Zhaohong Deng[1,2,3 *], Junyong Chen[1], Te Zhang[1], Longbing Cao[4], Shitong Wang[1]

1 School of Digital Media, Jiangnan University, Wuxi 214122, China
2 Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, Minjiang University, Fuzhou 350108, China
3 Jiangsu Key Laboratory of Digital Design and Software Technology, Wuxi 214122, China
4 Faculty of Engineering and Information Technology, University of Technology Sydney, New South Wales, Australia
* Corresponding author: dengzhaohong@jiangnan.edu.cn

***Abstract*:** Minimax Probability Machine (MPM) is a binary classifier that optimizes the upper bound of the misclassification probability. This upper bound of the misclassification probability can be used as an explicit indicator to characterize the reliability of the classification model and thus makes the classification model more transparent. However, the existing related work is constrained to linear models or the corresponding nonlinear models by applying the kernel trick. To relax such constraints, we propose the generalized hidden-mapping minimax probability machine (GHM-MPM). GHM-MPM is a generalized MPM. It is capable of training many classical intelligent models, such as feedforward neural networks, fuzzy logic systems, and linear and kernelized linear models for classification tasks, and realizing the reliability learning of these models simultaneously. Since the GHM-MPM, similarly to the classical MPM, was originally developed only for binary classification, it is further extended to multi-class classification by using the obtained reliability indices of the binary classifiers of two arbitrary classes. The experimental results show that GHM-MPM makes the trained models more transparent and reliable than those trained by classical methods.

Keywords: Classification, fuzzy logical systems, kernel tricks, minimax probability, neural networks, reliability learning.

1. Introduction

When constructing a classifier, high classification accuracy is often desired. Meanwhile, the reliability of classification models is also of concern in many practical applications, such as medical diagnosis. This is because the reliability index of a model makes it more transparent to users and makes the corresponding

recognition results more readily accepted. Hence, the reliability learning of classification models is critical.

Researchers have performed some interesting work on the reliability learning of the intelligent models. A classical outcome is the Minimax Probability Machine (MPM) [20, 22], which can be regarded as a typical reliability learning method of classification. MPM's optimization objective is to minimize the upper bound of the misclassification probability of learning the model parameters. The upper bound of the misclassification probability can be used as an explicit indicator to evaluate the reliability of classification models. In addition, the kernelized version of MPM was proposed in [20, 22] for nonlinear classification tasks. Moreover, several improved MPM algorithms have been presented from different viewpoints [8, 15-17, 32]. In [15-17], it was indicated that, in some cases, the misclassification probabilities of two classes should be distinguished because one class may be more important than the other. In [32], MPM was extended for regression. In [8], MPM was introduced to train a TSK fuzzy classifier towards a more transparent and interpretable classification model. In addition to MPMs, the reliability learning of intelligent models has been addressed from other viewpoints. For example, the concepts of "conflict" and "ignorance" were introduced to indicate the reliability of classification models in [24, 25].

In the existing work, the minimax probability decision-based methods have shown distinct advantages. First, the probability decision has a solid foundation of mathematical theory. Second, it is easy to understand and interpret due to its conciseness. However, at present, the minimax probability decision is applicable to very few intelligent models for realizing reliability learning. An important question is how to extend this method to additional intelligent models. In addition, existing work mainly focuses on binary classification and lacks in-depth study of multi-class problems. In summary, although the minimax probability decision has demonstrated high potential, its applicability is still very limited.

In response to the abovementioned challenges, a Generalized Hidden-Mapping Minimax Probability Machine (GHM-MPM) is proposed in this paper. First, the relation between the hidden-mapping model and several classical intelligent models, such as fuzzy logical systems and feedforward neural networks, is discussed. Then, these methods are incorporated into a unified hidden-mapping framework. With this framework, GHM-MPM is applied for the model training and reliability learning. Furthermore, GHM-MPM is extended to solve multi-classification problems by the "One-Against-One" scheme. In particular, the reliability indices that are obtained based on the "One-Against-One" scheme for two arbitrary classes have been used to characterize the separation degree of the corresponding classes.

The main contributions of this work are summarized as follows:

1) The proposed GHM-MPM is a universal method, which is applicable to many classical intelligent models for model training and reliability learning.

2) By the "One-versus-One" scheme, GHM-MPM can be used for multi-class classification. The reliability index of every binary sub-classifier describes the separation degree of the corresponding two classes, thereby

allowing users to better understand the complexity of the classification task.

The rest of this paper is organized as follows. Section 2 introduces the basic concepts of the classical minimax probability machine and its kernelized version. In Section 3 to Section 6, the generalized hidden-mapping MPM is proposed. Experimental results and analyses on synthetic and real-world datasets are reported in Section 7. Finally, this paper is concluded in Section 8.

2. Minimax probability decision technique

In this section, the MPM and its kernelized version [22] are reviewed briefly.

2.1 Minimax Probability Machine

The minimax probability machine [22] is a binary classification model for minimizing the upper bound of the misclassification probability. Given a dataset that contains two classes that are sampled from $\boldsymbol{x} \sim (\boldsymbol{u}_+, \boldsymbol{\Sigma}_+)$ and $\boldsymbol{x} \sim (\boldsymbol{u}_-, \boldsymbol{\Sigma}_-)$ randomly, where $\boldsymbol{u}_+, \boldsymbol{\Sigma}_+$ and $\boldsymbol{u}_-, \boldsymbol{\Sigma}_-$ represent the means and covariance matrices of the two classes, respectively, MPM defines the following optimization objective of finding a classification hyperplane $\boldsymbol{w}^T \boldsymbol{x} = b$:

$$\begin{aligned} &\max_{\alpha, \boldsymbol{w} \neq 0, b} \alpha \\ &s.t. \inf_{\boldsymbol{x} \sim (\boldsymbol{u}_+, \boldsymbol{\Sigma}_+)} pr(\boldsymbol{w}^T \boldsymbol{x} \geq b) \geq \alpha, \\ &\quad \inf_{\boldsymbol{x} \sim (\boldsymbol{u}_-, \boldsymbol{\Sigma}_-)} pr(\boldsymbol{w}^T \boldsymbol{x} \leq b) \geq \alpha \end{aligned} \tag{1}$$

where $\inf_{\boldsymbol{x} \sim (\boldsymbol{u}_+, \boldsymbol{\Sigma}_+)} pr(\boldsymbol{w}^T \boldsymbol{x} \geq b)$ denotes the infimum of the probability for the conditions $\boldsymbol{x} \sim (\boldsymbol{u}_+, \boldsymbol{\Sigma}_+)$ and $\boldsymbol{w}^T \boldsymbol{x} \geq b$.

Eqn. (1) implies that, for two-class data from $\boldsymbol{x} \sim (\boldsymbol{u}_+, \boldsymbol{\Sigma}_+)$ and $\boldsymbol{x} \sim (\boldsymbol{u}_-, \boldsymbol{\Sigma}_-)$, if $\boldsymbol{u}_+ = \boldsymbol{u}_-$, then Eqn. (1) does not have a meaningful solution and the optimal lower bound of the correct classification probability of future data $\alpha^* = 0$, namely, the optimal upper bound of the misclassification probability $1 - \alpha^* = 1$. Otherwise, an optimal classification hyperplane $(\boldsymbol{w}^*)^T \boldsymbol{x} = b^*$ exists and can be determined by solving the following convex optimization problem:

$$\begin{aligned} &\kappa^*(\alpha)^{-1} = \min_{\boldsymbol{w}} \sqrt{\boldsymbol{w}^T \boldsymbol{\Sigma}_+ \boldsymbol{w}} + \sqrt{\boldsymbol{w}^T \boldsymbol{\Sigma}_- \boldsymbol{w}} \\ &s.t.\ \boldsymbol{w}^T (\boldsymbol{u}_+ - \boldsymbol{u}_-) = 1 \end{aligned}, \tag{2.a}$$

where $\kappa^*(\alpha)^{-1} = \sqrt{\frac{1-\alpha^*}{\alpha^*}}$ (namely, $\kappa(\alpha) = \sqrt{\frac{\alpha}{1-\alpha}}$). Therefore, the optimal upper bound of the misclassification probability is obtained via

$$1 - \alpha^* = \frac{1}{1 + (\kappa(\alpha)^*)^2} = \frac{\left(\sqrt{(\boldsymbol{w}^*)^T \boldsymbol{\Sigma}_+ (\boldsymbol{w}^*)} + \sqrt{(\boldsymbol{w}^*)^T \boldsymbol{\Sigma}_- (\boldsymbol{w}^*)} \right)^2}{1 + \left(\sqrt{(\boldsymbol{w}^*)^T \boldsymbol{\Sigma}_+ (\boldsymbol{w}^*)} + \sqrt{(\boldsymbol{w}^*)^T \boldsymbol{\Sigma}_- (\boldsymbol{w}^*)} \right)^2}, \tag{2.b}$$

and the optimal b can be set to

$$b^* = (\boldsymbol{w}^*)^T \boldsymbol{u}_+ - \kappa(\alpha)^* \sqrt{(\boldsymbol{w}^*)^T \boldsymbol{\Sigma}_+ (\boldsymbol{w}^*)}. \tag{2.c}$$

If $\boldsymbol{\Sigma}_+$ and $\boldsymbol{\Sigma}_-$ are positive definite, the optimal hyperplane is unique.

Eqn. (2.a) is a second-order cone program (SOCP) optimization problem [21], which can be solved by toolkits such as SeDuMi [18]. A simple iterative least-square method was proposed in [22] for solving the problem.

2.2 Kernelized Minimax Probability Machine

The basic MPM (BMPM) was developed for linear models. When solving linearly inseparable problems, it is difficult to obtain a satisfactory result. The problem in the original feature space can be converted to a linearly separable problem in a high-dimensional space through the nonlinear mapping. The objective of the corresponding MPM is as follows [22]:

$$\begin{aligned} &\max_{\alpha,\boldsymbol{w}\neq 0,b} \alpha \\ &s.t. \inf_{\varphi(\boldsymbol{x})\sim\left(\boldsymbol{u}_{\varphi(+)},\boldsymbol{\Sigma}_{\varphi(+)}\right)} pr(\boldsymbol{w}^T\varphi(\boldsymbol{x})\geq b)\geq\alpha \ . \\ &\quad \inf_{\varphi(\boldsymbol{x})\sim\left(\boldsymbol{u}_{\varphi(-)},\boldsymbol{\Sigma}_{\varphi(-)}\right)} pr(\boldsymbol{w}^T\varphi(\boldsymbol{x})\leq b)\geq\alpha \end{aligned} \tag{3}$$

Here, $\varphi(\boldsymbol{x})$ is the vector that is mapped from $\boldsymbol{x}$ in the original space; the means and covariance matrices can be estimated from the mapped dataset $\boldsymbol{X}_\varphi=\{\varphi(\boldsymbol{x})_i\}$. Setting $\tilde{\boldsymbol{u}}_\varphi$ as the estimated mean vector of dataset $\boldsymbol{X}_\varphi$, the covariance matrices can be estimated via

$$\tilde{\boldsymbol{\Sigma}}_\varphi=(\boldsymbol{X}_\varphi-\tilde{\boldsymbol{u}}_\varphi\times\mathbf{1}_N^T)(\boldsymbol{X}_\varphi-\tilde{\boldsymbol{u}}_\varphi\times\mathbf{1}_N^T)^T\big/N \ , \tag{4}$$

where $\mathbf{1}_N$ denotes an N-dimensional column vector of ones.

The mapping function $\varphi(\cdot)$ is usually unknown, so it is difficult to obtain the dataset $\{\varphi(\boldsymbol{x}_i)\}$ and its means and covariance matrices in the mapped feature space. However, if the mapping meets the Mercer kernel condition, the kernel trick can be applied.

Set $\boldsymbol{X}=\{\boldsymbol{x}_i\}_{i=1}^N\in\boldsymbol{R}^{\boldsymbol{d}\times N}$ as a binary classification dataset, where the first N_+ columns and the last N_- columns of $\boldsymbol{X}$ are positive and negative classes, respectively:

$$\boldsymbol{X}=\left(\boldsymbol{X}_+,\boldsymbol{X}_-\right). \tag{5}$$

The Gram kernel matrix $\boldsymbol{K}$ of the training dataset can be defined as $\boldsymbol{K}_{ij}=\varphi(\boldsymbol{x}_i)^T\varphi(\boldsymbol{x}_j)=\boldsymbol{K}(\boldsymbol{x}_i,\boldsymbol{x}_j)$, $i,j=1,2,\cdots,N$. Denote the first N_+ columns of $\boldsymbol{K}$ as $\boldsymbol{K}_+$ and the last N_- columns as $\boldsymbol{K}_-$, namely:

$$\boldsymbol{K}=\left(\boldsymbol{K}_+,\boldsymbol{K}_-\right). \tag{6}$$

According to 20, 22], $\boldsymbol{w}$ in Eqn. (3) can be written as a linear combination of samples in the training dataset:

$$\boldsymbol{w}=\sum_{i=1}^{N_+}\lambda_i\varphi\left((\boldsymbol{x}_+)_i\right)+\sum_{j=1}^{N_-}\gamma_j\varphi\left((\boldsymbol{x}_-)_j\right). \tag{7}$$

Substituting Eqn. (7) with $\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\varphi}(+)}$ and $\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{\varphi}(-)}$ into Eqn. (2.a), the corresponding optimization objective is obtained:

$$\kappa(\alpha)^{-1}=\min_{\boldsymbol{m}}\left(\left\|\boldsymbol{m}^T\tilde{\boldsymbol{D}}_{K(+)}\right\|_2+\left\|\boldsymbol{m}^T\tilde{\boldsymbol{D}}_{K(-)}\right\|_2\right),\quad s.t.\ \boldsymbol{m}^T\left(\tilde{\boldsymbol{u}}_{K(+)}-\tilde{\boldsymbol{u}}_{K(-)}\right)=1 \tag{8}$$

where $\boldsymbol{m}=\left[\lambda_1\ \lambda_2\ \ldots\ \lambda_{N_+}\ \gamma_1\ \gamma_2\ \ldots\ \gamma_{N_-}\right]^T$,

$$\left(\tilde{\boldsymbol{u}}_{K(+)}\right)_i=\frac{1}{N_+}\sum_{j=1}^{N_+}\boldsymbol{K}\left(\boldsymbol{x}_i,(\boldsymbol{x}_+)_j\right), \tag{9.a}$$

$$\left(\tilde{\boldsymbol{u}}_{K(-)}\right)_i=\frac{1}{N_-}\sum_{j=1}^{N_-}\boldsymbol{K}\left(\boldsymbol{x}_i,(\boldsymbol{x}_-)_j\right), \tag{9.b}$$

$$\tilde{\boldsymbol{D}}_{K(+)}=\frac{1}{\sqrt{N_+}}\left(\boldsymbol{K}_+-\tilde{\boldsymbol{u}}_{K(+)}1_{N_+}^T\right), \tag{9.c}$$

$$\tilde{\boldsymbol{D}}_{K(-)}=\frac{1}{\sqrt{N_-}}\left(\boldsymbol{K}_--\tilde{\boldsymbol{u}}_{K(-)}1_{N_-}^T\right). \tag{9.d}$$

Eqn. (8) is also an SOCP problem, so it can be computed in the similar way to Eqn. (2.a).

Since $\varphi(\boldsymbol{x})$ is usually not known, we can compute $\boldsymbol{w}^T\varphi(\boldsymbol{x})$ in the decision function below.

$$\boldsymbol{w}^T\varphi(\boldsymbol{x})=\boldsymbol{m}^T\begin{bmatrix}K(\boldsymbol{x},\boldsymbol{x}_1)\\ \vdots\\ K(\boldsymbol{x},\boldsymbol{x}_N)\end{bmatrix}. \tag{10}$$

3 Generalized hidden-mapping minimax probability machine

To make the minimax probability decision method available for training additional intelligent models and realize the reliability learning of these models, the generalized hidden-mapping minimax probability machine (GHM-MPM) is proposed in this section.

The decision function of GHM-MPM can be written as

$$y=\text{sign}(\boldsymbol{w}^T\rho(\boldsymbol{x})-b), \tag{11}$$

and the corresponding optimization objective is

$$\max_{\alpha,\boldsymbol{w}\neq 0,b}\ \alpha\quad s.t.\ \inf_{\rho(\boldsymbol{x})\sim\left(\boldsymbol{u}_{\rho(+)},\boldsymbol{\Sigma}_{\rho(+)}\right)}pr(\boldsymbol{w}^T\rho(\boldsymbol{x})\geq b)\geq\alpha,\ \inf_{\rho(\boldsymbol{x})\sim\left(\boldsymbol{u}_{\rho(-)},\boldsymbol{\Sigma}_{\rho(-)}\right)}pr(\boldsymbol{w}^T\rho(\boldsymbol{x})\leq b)\geq\alpha \tag{12}$$

where $\rho(\boldsymbol{x})$ represents the feature vector in a new hidden space, which is mapped from $\boldsymbol{x}$ by hidden-mapping function $\rho(\cdot)$. The concepts of hidden mapping and hidden space are similar to the hidden layer and hidden neuron in neural networks.

Setting $\tilde{\boldsymbol{u}}_\rho$ as the estimated mean of dataset $\boldsymbol{X}_\rho=\{\rho(\boldsymbol{x})_i\}$, the covariance matrices can be estimated by

$$\tilde{\boldsymbol{\Sigma}}_\rho=\left(\boldsymbol{X}_\rho-\tilde{\boldsymbol{u}}_\rho\times\boldsymbol{1}_N^T\right)\left(\boldsymbol{X}_\rho-\tilde{\boldsymbol{u}}_\rho\times\boldsymbol{1}_N^T\right)^T\Big/N, \tag{13}$$

where $\boldsymbol{1}_N$ denotes an N-dimensional column vector of ones. Combining Eqn. (2.a) and Eqn. (13), the SOCP optimization objective function is obtained:

$$\kappa(\alpha)^{-1} = \min_{\boldsymbol{w}} \left(\left\| \boldsymbol{w}^T \tilde{\boldsymbol{D}}_{\rho(+)} \right\|_2 + \left\| \boldsymbol{w}^T \tilde{\boldsymbol{D}}_{\rho(-)} \right\|_2 \right), \quad \text{s.t. } \boldsymbol{w}^T \left(\tilde{\boldsymbol{u}}_{\rho(+)} - \tilde{\boldsymbol{u}}_{\rho(-)} \right) = 1 \tag{14}$$

where

$$\tilde{\boldsymbol{u}}_{\rho(+)} = \frac{1}{N_+} \sum_{i=1}^{N_+} \rho\left((\boldsymbol{x}_+)_i \right), \tag{15.a}$$

$$\tilde{\boldsymbol{u}}_{\rho(-)} = \frac{1}{N_-} \sum_{i=1}^{N_-} \rho\left((\boldsymbol{x}_-)_i \right), \tag{15.b}$$

$$\tilde{\boldsymbol{D}}_{\rho(+)} = \frac{1}{\sqrt{N_+}} \left(\boldsymbol{X}_{\rho(+)} - \tilde{\boldsymbol{u}}_{\rho(+)} 1_{N_+}^T \right), \tag{15.c}$$

$$\tilde{\boldsymbol{D}}_{\rho(-)} = \frac{1}{\sqrt{N_-}} \left(\boldsymbol{X}_{\rho(-)} - \tilde{\boldsymbol{u}}_{\rho(-)} 1_{N_-}^T \right). \tag{15.d}$$

Comparing Eqn. (14) with Eqn. (8), we observe that they share a unified form. Thus, they can be optimized in a similar way. Compared with Eqn. (2.a), it is not necessary to estimate the covariance matrices in Eqn. (14).

When $\rho(\cdot)$ is known, Eqn. (14) can be solved in the same way as Eqn. (2.a). When $\rho(\cdot)$ is unknown, Eqn. (14) can be solved in the same way as Eqn. (8) with the kernel trick. To unify these two cases, we set

$$\boldsymbol{q} = \begin{cases} \boldsymbol{w}, & \rho(\cdot) \text{ is known} \\ \boldsymbol{m}, & otherwise \end{cases}, \tag{16.a}$$

$$\beta(\boldsymbol{x}) = \begin{cases} \rho(\boldsymbol{x}), & \rho(\cdot) \text{ is known} \\ \left[\boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}_1) \quad \cdots \quad \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}_N) \right]^T, & \text{otherwise} \end{cases}. \tag{16.b}$$

Then, we obtain the unified SOCP form of the optimization objectives of these two cases:

$$\kappa(\alpha)^{-1} = \min_{\boldsymbol{q}} \left(\left\| \boldsymbol{q}^T \tilde{\boldsymbol{D}}_{\beta(+)} \right\|_2 + \left\| \boldsymbol{q}^T \tilde{\boldsymbol{D}}_{\beta(-)} \right\|_2 \right), \quad \text{s.t. } \boldsymbol{w}^T (\tilde{\boldsymbol{u}}_{\beta(+)} - \tilde{\boldsymbol{u}}_{\beta(-)}) = 1 \tag{17}$$

where

$$\tilde{\boldsymbol{u}}_{\beta(+)} = \frac{1}{N_+} \sum_{i=1}^{N_+} \beta\left((\boldsymbol{x}_+)_i \right), \tag{18.a}$$

$$\tilde{\boldsymbol{u}}_{\beta(-)} = \frac{1}{N_-} \sum_{i=1}^{N_-} \beta\left((\boldsymbol{x}_-)_i \right), \tag{18.b}$$

$$\tilde{\boldsymbol{D}}_{\beta(+)} = \frac{1}{\sqrt{N_+}} \left(\boldsymbol{X}_{\beta(+)} - \tilde{\boldsymbol{u}}_{\beta(+)} 1_{N_+}^T \right), \tag{18.c}$$

$$\tilde{\boldsymbol{D}}_{\beta(-)} = \frac{1}{\sqrt{N_-}} \left(\boldsymbol{X}_{\beta(-)} - \tilde{\boldsymbol{u}}_{\beta(-)} 1_{N_-}^T \right). \tag{18.d}$$

Now, the decision function of GHM-MPM can be rewritten as

$$y = \text{sign}(\boldsymbol{q}^T \beta(\boldsymbol{x}) - b). \tag{19}$$

We further use Fig. 1 to illustrate the decision function. The input vector $\boldsymbol{x}$ can be mapped to the hidden space through $\beta(\cdot)$ and substituted into MPM. If the output y is -1, then $\boldsymbol{x}$ is assigned to the negative class;

else, it is assigned to the positive class. Here, the outer $\beta(\cdot)$ is a replacement part, while the inner MPM is fixed. The only difference is in $\beta(\cdot)$ and the processing in MPM is same.

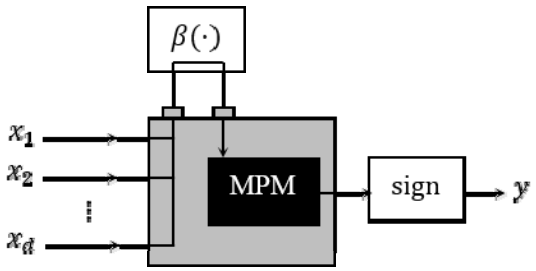


Fig. 1. GHM-MPM

4 GHM-MPM applications

Here, we show that GHM-MPM is applicable to many intelligent models, such as fuzzy logical systems and feedforward neural networks.

4.1 GHM-MPM and linear model

If $\beta(\boldsymbol{x}) = \rho(\boldsymbol{x}) = \boldsymbol{x}$, we obtain the basic MPM in [20, 22]. The difference between GHM-MPM and MPM is that the proposed algorithm of GHM-MPM needs not to compute the covariance matrices, which avoids the difficulty that is caused by the positive semi-definite covariance matrix and improves the stability of the algorithm.

4.2 GHM-MPM and kernel model

If $\rho(\cdot)$ is unknown, it is difficult to obtain the dataset $\boldsymbol{X}_\rho = \{\rho(\boldsymbol{x})_i\}$ in a hidden space. However, if $\rho(\cdot)$ meets the Mercer kernel condition, the kernel trick, which was reviewed in Section 2.2, can be applied to this problem. In this case, GHM-MPM is equivalent to the kernelized MPM in [20, 22].

4.3 GHM-MPM and Fuzzy Logic System

The Takagi-Sugeno-Kang Fuzzy Logic System (TSK FLS) [33], as a classic model of fuzzy logic systems, is widely applied due to its effectiveness. TSK FLS is a classical rule-based method [7, 27, 30]. The most commonly used fuzzy inference rule of TSK FLS, namely, R^k, is designed as follows:

$$\begin{aligned} &R^k: \\ &\text{If } x_1 \text{ is } A_1^k \wedge x_2 \text{ is } A_2^k \wedge \cdots \wedge x_d \text{ is } A_d^k, \\ &\text{Then } f_k(\boldsymbol{x}) = p_0^k + p_1^k x_1 + \cdots + p_d^k x_d \end{aligned} \tag{20.a}$$

where $k = 1, \cdots, K$, K is the number of fuzzy rules, $\wedge$ is a fuzzy conjunction operator, and A_i^k is a fuzzy subset subscribed by the input variable x_i for the kth rule. The output of TSK FLS can be formulated as

$$f_{TSK-FS}(\boldsymbol{x}) = \sum_{k=1}^{K} \frac{\mu_k(\boldsymbol{x})}{\sum_{k'=1}^{K} \mu_{k'}(\boldsymbol{x})} \cdot f_k(\boldsymbol{x}) = \sum_{k=1}^{K} \tilde{\mu}_k(\boldsymbol{x}) f_k(\boldsymbol{x}) \quad , \tag{20.b}$$

where $\boldsymbol{x} = [x_1, x_2, \cdots, x_d]^T$, $\mu_k(\boldsymbol{x})$ denotes the fuzzy membership function that is associated with the fuzzy set A^k, and $\tilde{\mu}_k(\boldsymbol{x})$ is the normalized $\mu_k(\boldsymbol{x})$. $\mu_k(\boldsymbol{x})$ can be calculated by

$$\mu_k(\boldsymbol{x}) = \prod_{i=1}^{d} \mu_{A_i^k}(x_i) . \tag{20.c}$$

Here, many space partitioning techniques, such as fuzzy c-means (FCM) clustering [2], can be used to estimate the parameters of the antecedents (the IF-part). Once the antecedents of TSK FLS are generated, we can express the following hidden-mapping form of the binary fuzzy classifier based on TSK FLS:

$$y = \text{sign}(\boldsymbol{w}^T \rho(\boldsymbol{x}) - b) \tag{21}$$

with

$$\rho(\boldsymbol{x}) = [\tilde{\boldsymbol{x}}_1^T, \tilde{\boldsymbol{x}}_2^T, \ldots, \tilde{\boldsymbol{x}}_K^T]^T \in \boldsymbol{R}^{K \times (d+1)}, \tag{22.a}$$

$$\tilde{\boldsymbol{x}}_k = \tilde{\mu}_k(\boldsymbol{x}) \boldsymbol{x}_e , \tag{22.b}$$

$$\boldsymbol{x}_e = (1, \boldsymbol{x}^T)^T , \tag{22.c}$$

$$\boldsymbol{w} = [\boldsymbol{p}_1^T, \boldsymbol{p}_2^T, \cdots, \boldsymbol{p}_K^T]^T , \tag{22.d}$$

$$\boldsymbol{p}_k = (p_0, p_1, \cdots, p_d)^T . \tag{22.e}$$

Hence, the TSK FLS-based binary classifier can be solved by GHM-MPM. More details about the MPM-based TSK FLS classifier can be found in [8].

4.4 GHM-MPM and neural network

The feedforward neural network [9-11, 28] can be used as a classification model. Fig. 2 shows a single-hidden-layer feedforward neural network (SHLFNN) classifier with multiple inputs and a single output.

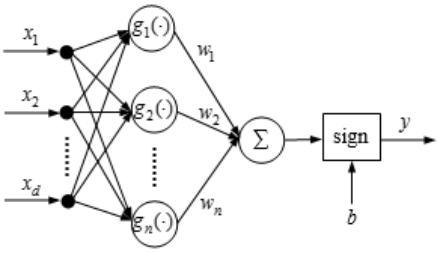


Fig. 2 A single-hidden-layer neural network classifier with a single output.

The output of a single-layer feedforward neural network can be formulated as follows:

$$y = \text{sign}\left(\sum_{i=1}^{n} g_i(\boldsymbol{x})^T w_i - b \right), \tag{23.a}$$

where $g_i(\cdot)$ is the activation function of the *i*th neuron in the hidden layer and *b* is a threshold. The hidden neurons can be determined in different ways. For example, if the Radial Basis Function (RBF) is used as the activation function, then the centres and widths of RBF can be calculated by FCM. In particular, according to [10, 11, 29], if $g_i(\cdot)$ is piecewise continuous, the hidden neurons can be randomly generated independent of the training data and the network retains its universal approximation capability.

Once the hidden layer is fixed, the SHLFNN training can be considered as a hidden-mapping linear model, where the hidden mapping function is

$$\rho(\boldsymbol{x}) = \begin{bmatrix} g_1(\boldsymbol{x}) & g_2(\boldsymbol{x}) & \cdots & g_n(\boldsymbol{x}) \end{bmatrix}^T. \tag{23.b}$$

Substituting Eqn. (23.b) into Eqn. (23.a), Eqn. (23) becomes Eqn. (11). Accordingly, the SHLFNN can be solved by GHM-MPM effectively. A multiple-hidden-layer feedforward neural network can also be reduced to a single-hidden-layer neural network with a more complicated hidden layer. Therefore, multiple-hidden-layer feedforward neural networks can also be trained by GHM-MPM.

5 Universal algorithm of GHM-MPM

According to the analysis in Section 3, the final optimization objective of GHM-MPM is a unified form of SOCP, regardless of whether $\rho(\cdot)$ is known or unknown. For this optimization problem, a universal algorithm is given here, which is a modified version of the iterative least-square algorithm in [22].

First, explanations about Eqn. (2.a) are given below. To satisfy the constraint $\boldsymbol{w}^T(\boldsymbol{u}_+ - \boldsymbol{u}_-) = 1$ in Eqn. (2.a), we set $\boldsymbol{w} = \boldsymbol{w}_0 + \boldsymbol{F}\boldsymbol{v}$, where $\boldsymbol{v} \in \boldsymbol{R}^{d-1}$, $\boldsymbol{w}_0 = (\boldsymbol{u}_+ - \boldsymbol{u}_-) \big/ \|\boldsymbol{u}_+ - \boldsymbol{u}_-\|_2^2$, and $\boldsymbol{F} \in \boldsymbol{R}^{d\times(d-1)}$ is an orthogonal matrix whose columns span the subspace of vectors that are orthogonal to $(\boldsymbol{u}_+ - \boldsymbol{u}_-)$. Hence, we have

$$(\boldsymbol{w}_0 + \boldsymbol{F}\boldsymbol{v})^T(\boldsymbol{u}_+ - \boldsymbol{u}_-) = \boldsymbol{w}_0^T(\boldsymbol{u}_+ - \boldsymbol{u}_-) = 1 \tag{24}$$

In detail, we set $\boldsymbol{u} = (\boldsymbol{u}_+ - \boldsymbol{u}_-) = (u_1, u_2, \cdots, u_d)^T$ and $u_m = \max(\boldsymbol{u})$, so that $\boldsymbol{F}$ can be built as follows:

$$\boldsymbol{F}(i,j) = \begin{cases} u_j / u_m, & i = m, j < m, \\ u_{j+1} / u_m, & i = m, j \geq m, \\ 1, & i = j < m, \\ 1, & i = j+1 > m, \\ 0, & else. \end{cases}. \tag{25}$$

Then, the constraint in Eqn. (2.a) is removed and a concise formula is obtained:

$$\kappa(\alpha)^{-1} = \min_{\boldsymbol{w}} \left(\sqrt{(\boldsymbol{w}_0 + \boldsymbol{F}\boldsymbol{v})^T \Sigma_+ (\boldsymbol{w}_0 + \boldsymbol{F}\boldsymbol{v})} + \sqrt{(\boldsymbol{w}_0 + \boldsymbol{F}\boldsymbol{v})^T \Sigma_- (\boldsymbol{w}_0 + \boldsymbol{F}\boldsymbol{v})} \right). \tag{26}$$

Furthermore, by eliminating the square roots in Eqn. (26), an equivalent form is obtained:

$$\inf_{\boldsymbol{v},\eta>0,\zeta>0} \eta + \frac{1}{\eta}(\boldsymbol{w}_0 + \boldsymbol{F}\boldsymbol{v})^T \Sigma_+ (\boldsymbol{w}_0 + \boldsymbol{F}\boldsymbol{v}) + \zeta + \frac{1}{\zeta}(\boldsymbol{w}_0 + \boldsymbol{F}\boldsymbol{v})^T \Sigma_- (\boldsymbol{w}_0 + \boldsymbol{F}\boldsymbol{v}) \tag{27}$$

Take the derivative of Eqn. (27) to be zero with respect to $\boldsymbol{v}$:

$$\frac{1}{\eta}\boldsymbol{F}^T\Sigma_+\left(\boldsymbol{w}_0+\boldsymbol{F}\boldsymbol{v}\right)+\frac{1}{\zeta}\boldsymbol{F}^T\Sigma_-\left(\boldsymbol{w}_0+\boldsymbol{F}\boldsymbol{v}\right)=0\,. \tag{28}$$

Meanwhile, let $\boldsymbol{u}_+=\tilde{\boldsymbol{u}}_{\beta(+)}$, $\boldsymbol{u}_-=\tilde{\boldsymbol{u}}_{\beta(-)}$, $\Sigma_+=\tilde{\boldsymbol{D}}_{\beta(+)}\tilde{\boldsymbol{D}}_{\beta(+)}^T$ and $\Sigma_-=\tilde{\boldsymbol{D}}_{\beta(-)}\tilde{\boldsymbol{D}}_{\beta(-)}^T$. Then, Eqn. (28) becomes

$$\begin{aligned}&\frac{1}{\eta}\left(\boldsymbol{F}^T\tilde{\boldsymbol{D}}_{\beta(+)}\right)\left(\boldsymbol{q}_0^T\tilde{\boldsymbol{D}}_{\beta(+)}\right)^T+\frac{1}{\zeta}\left(\boldsymbol{F}^T\tilde{\boldsymbol{D}}_{\beta(-)}\right)\left(\boldsymbol{q}_0^T\tilde{\boldsymbol{D}}_{\beta(-)}\right)^T\\&+\left(\frac{1}{\eta}\left\|\boldsymbol{F}^T\tilde{\boldsymbol{D}}_{\beta(+)}\right\|_2^2+\frac{1}{\zeta}\left\|\boldsymbol{F}^T\tilde{\boldsymbol{D}}_{\beta(-)}\right\|_2^2\right)\boldsymbol{v}=0\end{aligned}\,. \tag{29}$$

Now, we can approximate the optimal solution of Eqn. (29) by the iterative least-squares method in Algorithm 1.

Algorithm 1: PROCEDURE OF THE ITERATIVE LEAST-SQUARES METHOD FOR GHM-MPM

Algorithm of GHM-MPM	
Input:	$\{(\boldsymbol{x}_+)_i\}_{i=1}^{N_+}$, $\{(\boldsymbol{x}_-)_j\}_{j=1}^{N_-}$ and $\beta(\cdot)$.
	If $\rho(\cdot)$ is known, let $\beta(\boldsymbol{x})=\rho(\boldsymbol{x})$;
	If $\rho(\cdot)$ is unknown, let $\beta(\boldsymbol{x})=\left[\boldsymbol{K}(\boldsymbol{x},\boldsymbol{x}_1)\ \cdots\ \boldsymbol{K}(\boldsymbol{x},\boldsymbol{x}_N)\right]^T$;
Preprocess:	Calculate the following per Eqn. (18)
	$\tilde{\boldsymbol{u}}_{\beta(+)}$ and $\tilde{\boldsymbol{u}}_{\beta(-)}$,
	$\tilde{\boldsymbol{D}}_{\beta(+)}$ and $\tilde{\boldsymbol{D}}_{\beta(-)}$;
Initialize:	$\boldsymbol{q}_0\leftarrow\left(\tilde{\boldsymbol{u}}_{\beta(+)}-\tilde{\boldsymbol{u}}_{\beta(-)}\right)\Big/\left\|\tilde{\boldsymbol{u}}_{\beta(+)}-\tilde{\boldsymbol{u}}_{\beta(-)}\right\|_2^2$,
	Build $\boldsymbol{F}$ per Eqn. (25),
	$\boldsymbol{G}\leftarrow\left(\boldsymbol{F}^T\tilde{\boldsymbol{D}}_{\beta(+)}\right)\left(\boldsymbol{F}^T\tilde{\boldsymbol{D}}_{\beta(+)}\right)^T$,
	$\boldsymbol{H}\leftarrow\left(\boldsymbol{F}^T\tilde{\boldsymbol{D}}_{\beta(-)}\right)\left(\boldsymbol{F}^T\tilde{\boldsymbol{D}}_{\beta(-)}\right)^T$,
	$\boldsymbol{g}\leftarrow\left(\boldsymbol{F}^T\tilde{\boldsymbol{D}}_{\beta(+)}\right)\left(\boldsymbol{q}_0^T\tilde{\boldsymbol{D}}_{\beta(+)}\right)^T$,
	$\boldsymbol{h}\leftarrow\left(\boldsymbol{F}^T\tilde{\boldsymbol{D}}_{\beta(-)}\right)\left(\boldsymbol{q}_0^T\tilde{\boldsymbol{D}}_{\beta(-)}\right)^T$,
	$\eta_1\leftarrow 1$, $\zeta_1\leftarrow 1$, $i\leftarrow 1$;
Iterate:	$\boldsymbol{L}\leftarrow\frac{1}{\eta_i}\boldsymbol{G}+\frac{1}{\zeta_i}\boldsymbol{H}$,
	$\boldsymbol{s}\leftarrow-\left(\frac{1}{\eta_i}\boldsymbol{g}+\frac{1}{\zeta_i}\boldsymbol{h}\right)$,
	$\boldsymbol{L}\boldsymbol{v}_i=\boldsymbol{s}\Rightarrow\boldsymbol{v}_i=\boldsymbol{L}^{-1}\boldsymbol{s}$,
	$\boldsymbol{q}_i=\boldsymbol{q}_0+\boldsymbol{F}\boldsymbol{v}_i$,
	$\eta_{i+1}\leftarrow\sqrt{\left(\boldsymbol{q}_i^T\tilde{\boldsymbol{D}}_{\beta(+)}\right)\left(\boldsymbol{q}_i^T\tilde{\boldsymbol{D}}_{\beta(+)}\right)^T}$,
	$\zeta_{i+1}\leftarrow\sqrt{\left(\boldsymbol{q}_i^T\tilde{\boldsymbol{D}}_{\beta(-)}\right)\left(\boldsymbol{q}_i^T\tilde{\boldsymbol{D}}_{\beta(-)}\right)^T}$,
	$i\leftarrow i+1$,
	Until the stop criterion is satisfied;
Output:	$\boldsymbol{q}\leftarrow\boldsymbol{q}_i$,

$$\kappa \leftarrow \frac{1}{\eta_i + \zeta_i},$$

$$b \leftarrow \boldsymbol{q}^T \tilde{\boldsymbol{u}}_{\beta(+)} - \kappa \eta_i,$$

$$\alpha \leftarrow \frac{\kappa^2}{1+\kappa^2}.$$

In the input stage, we choose a mapping function $\beta(\cdot)$ and map the input vector to the hidden space. In the next two stages, multiple variables are calculated by the corresponding equations. Matrices $\boldsymbol{G}$, $\boldsymbol{H}$, g and h are subitems in Eqn. (29). According to the least-squares method, in the iteration stage, q approaches the optimal value. In the last output stage, optimal values of b and α are obtained.

6 Multi-classification and analysis of reliability indices

Like MPM, GHM-MPM is designed for binary classification. It can be easily extended to solve multi-classification problems by the "One-Against-One" (OAO) scheme and voting scheme [5, 25]. Indeed, if there are M classes, GHM-MPM will construct $M \times (M-1)/2$ binary classifiers. When a test sample arrives, it is predicted by each binary classifier and voted on by the M classes. If the most votes come from the mth class, the sample is classified into the mth class.

For binary classification problems, the α index, which is acquired from GHM-MPM directly, indicates the reliability of the model. For multi-classification problems, the reliability index of the whole model cannot be obtained. However, each binary classifier in the whole classification model that is obtained by GHM-MPM has an α index. The mean, median, minimum or weighted average of these α indices can be used to characterize the reliability of the whole model implicitly. For example, if we have 100 instances of class A, 100 instances of class B, and 200 instances of class C, then all the α indices can be merged into an overall index:

$$\begin{aligned}\alpha &= (100+100)/800 \times \alpha(A:B) \\ &\quad + (100+200)/800 \times \alpha(B:C) + (100+200)/800 \times \alpha(A:C) \\ &= 0.25 \times \alpha(A:B) + 0.375 \times \alpha(B:C) + 0.375 \times \alpha(A:C)\end{aligned} \tag{30}$$

In particular, the α of each binary classifier is related to the separation degree of the two corresponding classes. The larger the α index of a binary classifier, the bigger the separation degree of the two corresponding classes in the whole multi-class dataset. Therefore, it is very useful for the classification decision and analysis in multi-class classification. We can consider using an easier method to solve the binary classification problem, which has a much bigger α index. The obtained multiple α indices for a multi-class classification task will make the classification model more transparent. The above characteristics are very important in some applications, such as medical diagnosis.

7 Experiments

All the experiments are implemented with MATLAB on a 64-bit computer with 4 GB RAM and 3.4 GHz CPU. Four groups of experiments are designed to evaluate the performance of GHM-MPM and its applications.

Experiment 1 on a synthetic dataset is designed to evaluate the performance of GHM-MPM and the geometrical significance of the α index intuitively. Experiments 2 and 3 on real datasets from the UCI Machine Learning Repository [23] further evaluate the performance of GHM-MPM.

The proposed GHM-MPM is compared with several classic methods, including RBF-NN [18-21], ELM [13, 14], SVM [4, 5], KRR [6, 31], L2-TSK-FS[7] and ID3 [30]. The descriptions of these methods are listed in Table 1.

It is noted that many kernels can be used in the proposed GHM-MPM learning framework and each kernel has advantages. In this study, we employed only the commonly used linear and RBF kernels as examples. Several representative intelligent models, namely, neural networks and fuzzy systems, are also used as special cases of the proposed GHM-MPM. Of course, more models can be trained by the proposed GHM-MPM learning algorithm for different practical applications to achieve better performance when taking into account the characteristics of the modelling scenes.

Table 1 Descriptions of comparison methods and search grids of parameters.

Method	Description	Search Grid of Parameters
GHM-MPM: LINEAR	Linear version of GHM-MPM, $\rho(\boldsymbol{x})=\boldsymbol{x}$	None.
GHM-MPM: KERNEL	Kernel version of GHM-MPM, $\beta(\boldsymbol{x})=\left[\boldsymbol{K}(\boldsymbol{x},\boldsymbol{x}_1)\cdots\boldsymbol{K}(\boldsymbol{x},\boldsymbol{x}_N)\right]^T$.	RBF kernel width: $\sigma\in\{0.001,0.01,0.1,1,10,100,1000\}$.
GHM-MPM: TSKFLS	TSKFLS version of GHM-MPM, FCM determines the centers of Gaussian-type rules, mapping function is Eq. (22.a).	The number of fuzzy rules: $R\in\{4,9,16,25,49,64,81,100\}$, Scale parameter of width in Gaussian MF: $h\in\{0.001,0.01,0.1,1,10,100,1000\}$.
GHM-MPM: SHLFNN	SHLFNN version of GHM-MPM, FCM determines the centers of RBF-type neurons, mapping function is (23.b).	The number of neurons: $Q\in\{4,9,16,25,49,64,81,100\}$, Scale parameter of width in RBF: $h\in\{0.001,0.01,0.1,1,10,100,1000\}$.
RBF-NN	RBF neural network [18-21], from MATLAB toolboxes.	The maximum of neurons: $Q\in\{4,9,16,25,49,64,81,100\}$, Spread $\in\{1,5,10\}$.
ELM	Extreme Learning Machine [23-24], with RBF.	The number of neurons: $Q\in\{4,9,16,25,49,64,81,100\}$, Scale parameter of width in RBF: $h\in\{0.001,0.01,0.1,1,10,100,1000\}$.
SVM	Support Vector Machine [25], RBF kernel as default, from LIBSVM [26].	Regularization parameter: $C\in\{0.001,0.01,0.1,1,10,100,1000\}$, RBF kernel width: $\sigma\in\{0.001,0.01,0.1,1,10,100,1000\}$.
KRR	Kernel Ridge Regression, with RBF kernel, more in [27-28].	Parameter $\lambda\in\{0.001,0.01,0.1,1,10,100,1000\}$, RBF kernel width : $\sigma\in\{0.001,0.01,0.1,1,10,100,1000\}$.
L2-TSK-FS	L2-Norm Penalty-Based ε -Insensitive TSK FS[31]	The number of fuzzy rules: $R\in\{4,9,16,25,49,64,81,100\}$, Parameter $\lambda\in\{0.001,0.01,0.1,1,10,100,1000\}$.
ID3	ID3 decision tree [32]	None

For all the methods and all the datasets, by grouping the data sets for training and testing according to the 5-fold cross-validation (CV) strategy [1], the means and standard deviations (MEAN ± STD) of training

accuracies, testing accuracies and α indices are given. The search grids of the parameters that are adopted for CV are also listed in Table 1. Since RBF can realize nonlinear mapping, it is widely used in SVM and KRR. For fair comparison, we use the RBF kernel for the above kernel methods.

7.1 Experiment 1 on the Synthetic Dataset

7.1.1 Synthetic dataset

To show the geometrical significance of the α index and validate the classification performance of GHM-MPM, a 2-dimensional synthetic dataset is generated, as shown in Fig. 3, where "+", "*" and "x" represent classes A, B and C, respectively. Each class contains 100 instances. Instances of class A are generated with function $y = sin(x) + N(0,\sigma)$, where $x \in [0, 2\pi]$, $\sigma = 0.25$, and $N(0,\sigma)$ denotes Gaussian white noise with mean 0 and standard deviation σ. Instances of classes B and C are generated by 2-dimensional Gaussian white noise $N2(\boldsymbol{c},\boldsymbol{\sigma})$, with $[4.5,1.5]$ and $[1.5,-0.5]$ as the means $\boldsymbol{c}$, respectively, while $[0.5,0.5]$ is used as the standard deviation $\boldsymbol{\sigma}$.

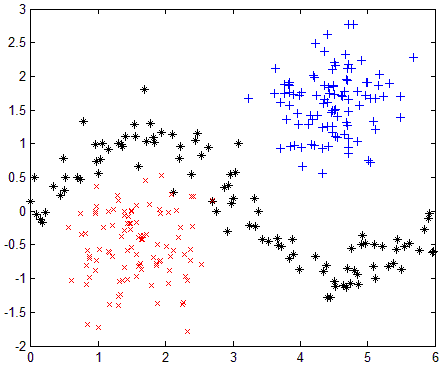


Fig. 3 Distribution of the 3-class Synthetic Dataset

7.1.2 **Experimental *Analysis***

By the 5-fold cross-validation strategy, the means and standard deviations of accuracies for training and testing of different methods are obtained, as shown in Table 2. In particular, for GHM-MPM, the mean and the standard deviation of the α indices have been presented for each classifier that is trained by it, as shown in Table 3.

When GHM-MPM is used to train the linear classification model (GHM-MPM: LINEAR) using the synthetic training dataset, the boundaries between pairs of arbitrary classes are illustrated in Fig. 4. Line(A:B) in Fig. 4 denotes the boundary between classes A and B, Line(B:C) denotes the boundary between classes B and C, and Line(A:C) denotes the boundary between classes A and C.

In addition, according to [22], the paired minimax ellipses can be drawn when using GHM-MPM: LINEAR. These ellipses are centred on the means of each class and shaped by the covariance matrices. The

Mahalanobis distances [26] from the centre to the points in the ellipse are always less than $\kappa(\alpha)$. When the smallest $\kappa(\alpha)$, namely, $\kappa^*(\alpha)$, is found, the paired ellipses will be tangent to each other and the intersection point will have the same Mahalanobis distance to the two classes. In other words, the maximum of the Mahalanobis distances to the arbitrary class will be minimized. This shows that MPM is designed to find the paired minimax "ellipsoids" for determining the decision hyperplane. Hence, GHM-MPM has the same geometric interpretation and transparency as MPM.

GHM-MPM is applied to several classical nonlinear intelligent models; the obtained boundary curves are shown in Fig. 5. Different types of curves denote the boundaries of different binary classifications.

In Fig. 4, classes A and C are linearly separable since they are far away from each other. Accordingly, α between these two classes is also large. Meanwhile, class A is close to class B and class C is close to class B, so these two binary classification problems are not linearly separable. This results in values of α for the corresponding two binary classifiers that are both small. From Table 3, the listed α indices indicate the difficulty of separating arbitrary two classes, which provides a clear understanding of the multi-class classification task.

The rows in Tables 2 and 3 show that for the linearly inseparable cases, the trained nonlinear models by GHM-MPM, such as GHM-MPM: TSKFLS and GHM-MPM: SHLFNN, obtain improved classification accuracies and reliability indices accordingly. The classification accuracies and α indices of nonlinear models that are trained by GHM-MPM are both better than those of the linear GHM-MPM: LINEAR. Although the α indices of different models that are trained by GHM-MPM are different, the tendency $\alpha(B:C) < \alpha(A:B) < \alpha(A:C)$ is consistent.

We also notice that the average α index, as a lower bound of the correct classification probability of future data, is smaller than the testing accuracy when GHM-MPM is used to train the model, except for GHM-MPM: KERNEL. This also occurred in the experiments in [22] due to tiny differences between distributions of training data and testing data in the mapped feature space. Thus, when GHM-MPM is applied with the kernel trick, the obtained α index should be used cautiously.

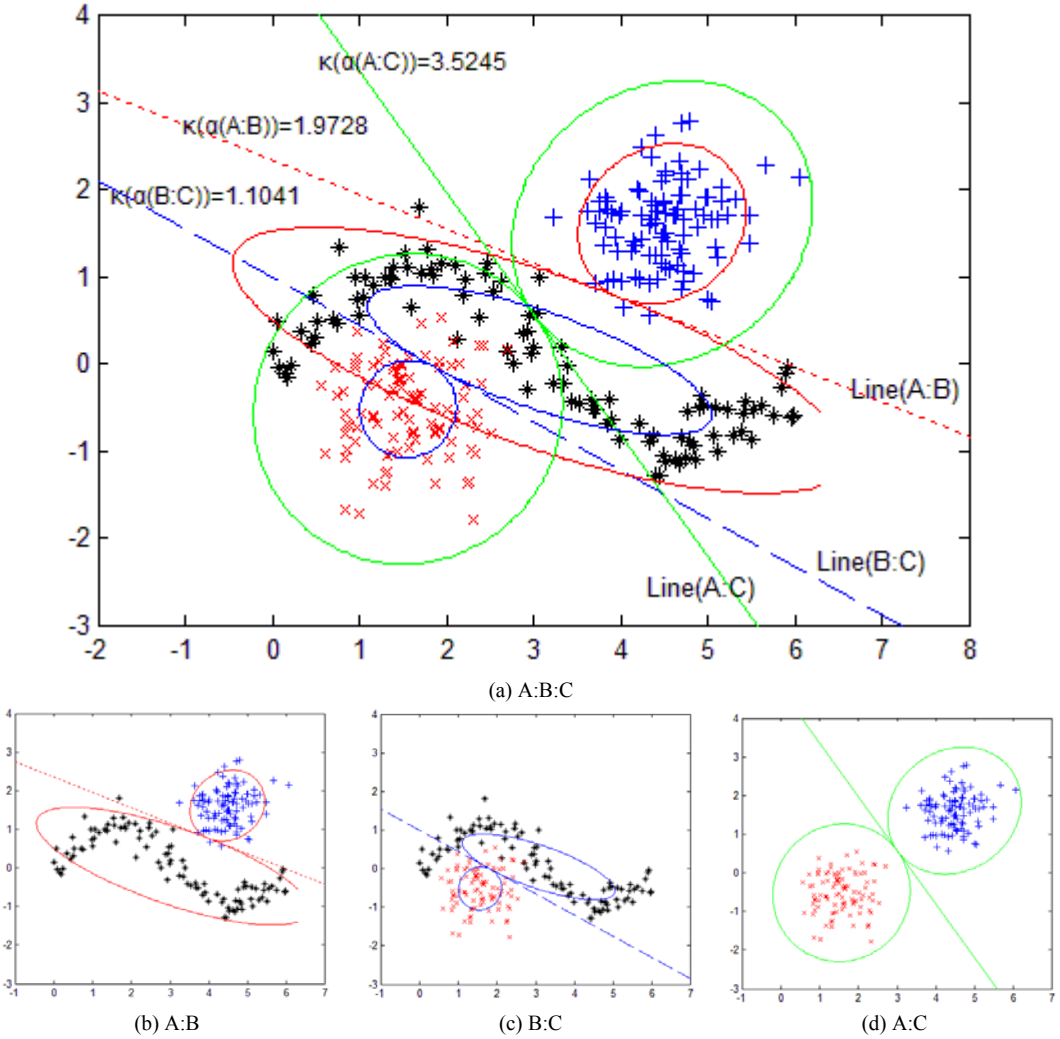
According to the above analyses, the classical classification models that are trained by GHM-MPM achieve competitive classification performance compared to those that are trained by the classical learning methods. However, the classical classification models that are trained by GHM-MPM demonstrate better transparency since GHM-MPM can provide α indices to describe the reliability of the trained binary classifiers and the separation degree of each pair of arbitrary classes in a multi-class classification task.

Table 2 Classification accuracies of different methods on synthetic dataset

ACC	GHM-MPM				RBF-NN	ELM	SVM	KRR	L2-TSK-FS	ID3
	LINEAR	KERNEL	TSKFLS	SHLFNN						
Testing	0.9100 ± 0.0224	0.9767 ± 0.0190	0.9900 ± 0.0091	**0.9933 ± 0.0091**	**0.9933 ± 0.0091**	0.9867 ± 0.0075	0.9900 ± 0.0091	0.9900 ± 0.0091	0.9733 ± 0.0091	0.9733 ±0.0190
Training	0.9033 ± 0.0099	0.9933 ± 0.0023	0.9908 ± 0.0035	0.9925 ± 0.0035	0.9900 ± 0.0037	0.9900 ± 0.0023	0.9875 ± 0.0059	0.9933 ± 0.0023	0.9775 ±0.0023	0.9867 ±0.0046

Table 3 The α Indices of different models trained by GHM-MPM on the synthetic dataset

α	GHM-MPM			
	LINEAR	KERNEL	TSKFLS	SHLFNN
$A:B$	0.7955 ± 0.0086	0.9977 ± 0.0008	0.9794 ± 0.0046	0.9786 ± 0.0037
$B:C$	0.5499 ± 0.0070	0.9390 ± 0.0079	0.8963 ± 0.0070	0.9010 ± 0.0053
$A:C$	0.9257 ± 0.0027	0.9999 ± 0.0000	0.9967 ± 0.0005	0.9970 ± 0.0006
Average	0.7570 ± 0.0061	0.9789 ± 0.0029	0.9575 ± 0.0040	0.9589 ± 0.0032

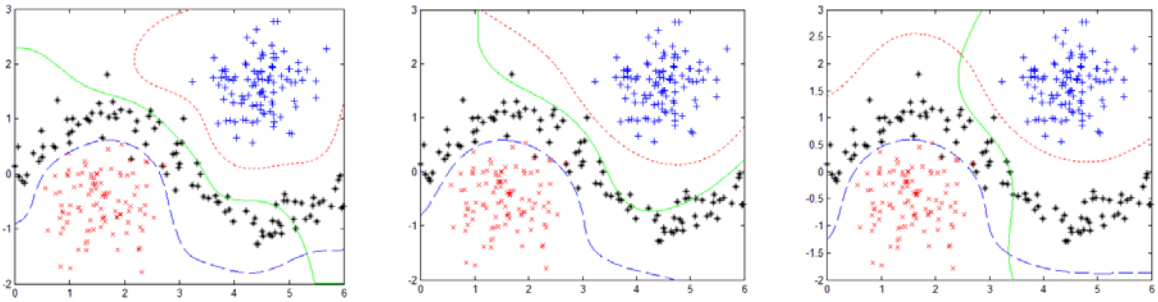


(a) A:B:C

(b) A:B (c) B:C (d) A:C

Fig. 4 The illustration of the performance of GHM-MPM: LINEAR on the synthetic dataset.

$(\alpha(\mathrm{A:B})=0.7956$, $\alpha(\mathrm{B:C})=0.5494$, $\alpha(\mathrm{A:C})=0.9255$; $\kappa(\alpha(\mathrm{A:B}))=1.9728$, $\kappa(\alpha(\mathrm{B:C}))=1.1041$, $\kappa(\alpha(\mathrm{A:C}))=3.5245)$.



(a) GHM-MPM: KERNEL (b) GHM-MPM: TSKFLS (c) GHM-MPM: SHLFNN

Fig. 5 The illustration of the performances of several nonlinear intelligent models that are trained by GHM-MPM on the synthetic dataset.

7.2 Experiment 2 on the Breast Dataset

In this experiment, the performance of GHM-MPM on medical diagnosis is evaluated. The adopted breast cancer dataset is from the UCI Machine Learning Repository, which contains 458 instances of the benign class and 241 instances of malignant class. Each instance is characterized by the following 9 attributes: clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli and mitoses. In this experiment, each attribute is normalized to the range [-1, 1].

7.2.1 Comparison of the Classification Accuracy

In this section, the proposed GHM-MPM is compared with related methods on the breast dataset. Experimental results are listed in Tables 4 and 5. According to Table 4, the classification accuracies of all methods are promising and the testing accuracy of GHM-MPM: SHLFNN is the highest. Because the separation degree of the benign class and the malignant class is large, even the linear model that is trained by GHM-MPM obtains high accuracy. In particular, GHM-MPM: KERNEL and GHM-MPM: SHLFNN obtain higher testing accuracies than SVM, which implies that the models that are trained by GHM-MPM have better generalization ability. Table 5 shows that α of GHM-MPM: LINEAR is 0.84, while nonlinear models that are trained by GHM-MPM obtain higher values. These results imply that nonlinear models that are trained by GHM-MPM achieve higher reliability indices than the linear model.

Table 4 Classification accuracies of different methods on the breast cancer dataset

ACC	GHM-MPM				RBF-NN	ELM	SVM	KRR	L2-TSK-FS	ID3
	LINEAR	KERNEL	TSKFLS	SHLFNN						
Testing	0.9685 ± 0.0130	0.9714 ± 0.0087	0.9671 ± 0.0109	**0.9728 ± 0.0117**	0.9642 ± 0.0101	0.9657 ± 0.0137	0.9671 ± 0.0112	0.9699 ± 0.0147	0.9671 ±0.0109	0.9356 ±0.0151
Training	0.9735 ± 0.0029	0.9732 ± 0.0022	0.9690 ± 0.0046	0.9732 ± 0.0031	0.9682 ± 0.0032	0.9725 ± 0.0035	0.9732 ± 0.0046	0.9700 ± 0.0037	0.9678 ±0.0022	0.9796 ±0.0027

Table 5 The α Indices of the models that are trained by GHM-MPM on the breast cancer dataset

α	GHM-MPM			
	LINEAR	KERNEL	TSKFLS	SHLFNN
Benign : Malignant	0.8395 ± 0.0036	0.8575 ± 0.0035	0.9230 ± 0.0159	0.8601 ± 0.0033

7.2.2 The Interpretability of GHM-MPM:TSKFLS on the Breast Dataset

Interpretability is very important for a medical diagnosis model. Here, the fuzzy-rule-based fuzzy system is adopted to show the interpretability of the model that was trained by the proposed GHM-MPM (TSKFLS).

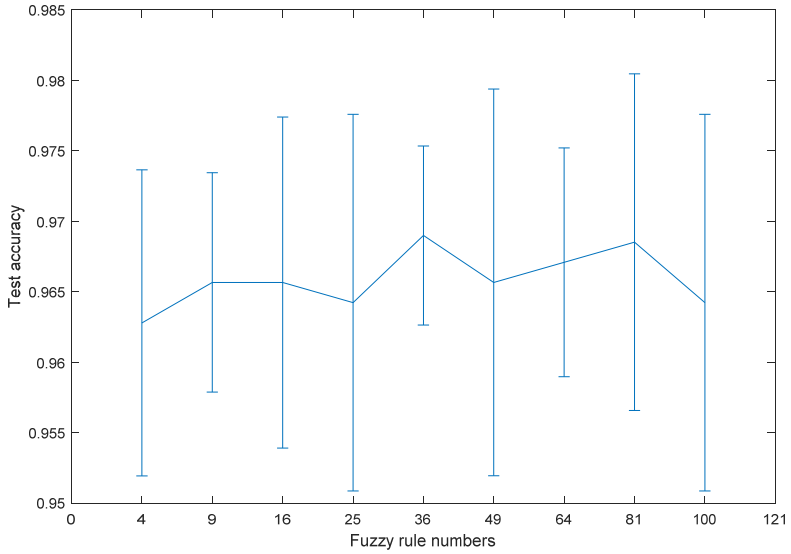


Fig. 6 Test accuracies of GHM-MPM:TSKFLS with different numbers of fuzzy rules.

Fig. 6 shows the test accuracies of GHM-MPM:TSKFLS with different numbers of fuzzy rules. According to Fig. 6, the classification performance of GHM-MPM:TSKFLS is influenced by the number of fuzzy rules to a certain extent, although the influence is not substantial on the breast dataset. To better show the interpretability of the fuzzy classifier that is trained by GHM-MPM:TSKFLS, we present a model with four rules. As shown in Table 6, the constructed fuzzy system contains three parts. We explain these parts as follows:

1) The first part is the fuzzy rule base in Part A of Table 6, which is used for fuzzy inference. With the fuzzy rule base, the fuzzy inference rules can be linguistically interpreted with expert knowledge.

2) The second part presents a decision threshold, which is introduced for the classification decision in GHM-MPM (TSKFLS). The decision threshold and the consequents of the TSK fuzzy system are learned based on the minimax probability decision principle. With the real output of the trained TSK fuzzy system and the decision threshold, the final decision can be given for the classification task.

3) The third part provides the reliability index of the trained classification model, where the reliability is characterized by the lower bound of the correct classification probability for the trained fuzzy classifier.

Table 6 Fuzzy systems generated by GHM-MPM:TSKFLS based on the breast dataset

Part A: Fuzzy rules base

TSK Fuzzy Rule R^k:

$IF\ x_1\ is\ A_1^k\left(c_1^k,\delta_1^k\right)\wedge x_1\ is\ A_1^k\left(c_1^k,\delta_1^k\right)\wedge\ldots\wedge x_1\ is\ A_1^k\left(c_1^k,\delta_1^k\right),\ THEN\ f_k\left(x\right)=p_{k0}+p_{k1}x_1+\ldots+p_{kd}x_d$

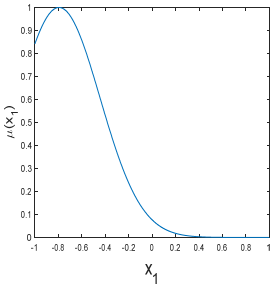
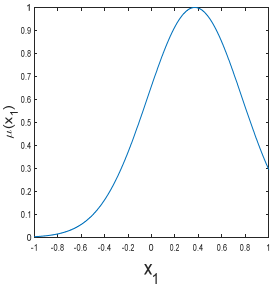
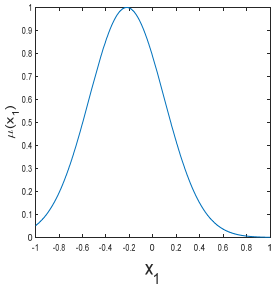
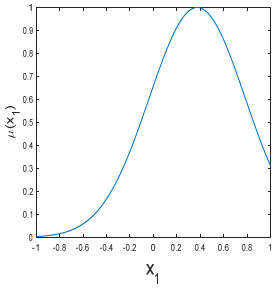
No. of rules	**Antecedent parameters** (Gaussian membership function parameters)	**Consequent parameters** (linear function parameters)
k	$\boldsymbol{c}^k=\left(c_1^k,\ldots,c_d^k\right)^T,\boldsymbol{\delta}^k=\left(\delta_1^k,\ldots,\delta_d^k\right)^T$	$\boldsymbol{p}_k=\left(p_{k0},p_{k1},\ldots,p_{kd}\right)^T$
1	$c^1=[-0.7940,-0.9571,-0.9400,-0.9507,-0.7798;$ $-0.7469,-0.7670,-0.9618,-0.9857]$ $\delta^1=[0.2454,0.1551,0.1599,0.1360,0.0915;$ $0.1979,0.1220,0.1573,0.0527]$	$\boldsymbol{p}_1=[0.7966,-0.7694,0.0157;$ $0.1192,-0.2716,0.0171;$ $0.2868,0.0938,0.2520,-0.2093]$
2	$c^2=[0.3689,0.4367,0.4209,0.1731,0.0520;$ $0.5565,0.2098,0.2838,-0.6005]$ $\delta^2=[0.3264,0.4293,0.3928,0.5302,0.2977;$ $0.4850,0.2967,0.5757,0.3456]$	$\boldsymbol{p}_2=[0.4147,0.0408,-0.0050;$ $0.0467,-0.0137,-0.0050;$ $0.0159,0.0369,-0.0339,0.0155]$
3	$c^3=[-0.2230,-0.8858,-0.8424,-0.8877,-0.7356,$ $-0.6980,-0.7177,-0.9044,-0.9714]$ $\delta^3=[0.2035,0.2123,0.2135,0.1864,0.1217;$ $0.2755,0.1548,0.2214,0.0746]$	$\boldsymbol{p}_3=[-1.0369,-0.4007,-0.0041;$ $-0.5028,-0.3005,0.1209;$ $-1.0408,-0.3258,-0.5742,-0.5802]$
4	$c^4=[0.3745,0.3130,0.3140,0.0555,-0.0121;$ $0.5578,0.1401,0.1665,-0.6497]$ $\delta^4=[0.3348,0.4182,0.3803,0.5149,0.2942;$ $0.4929,0.2888,0.5630,0.3198]$	$\boldsymbol{p}_4=[0.5046,-0.0653,-0.0062;$ $-0.0689,0.0107,0.0074;$ $-0.0435,-0.0598,0.0404,-0.0373]$

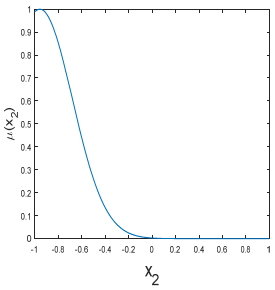
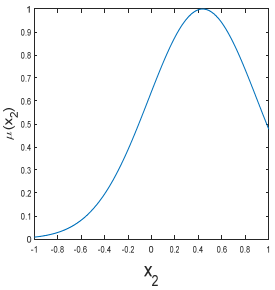
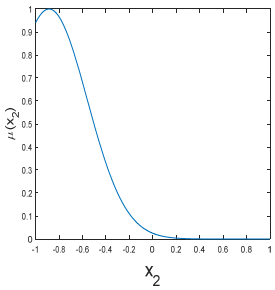
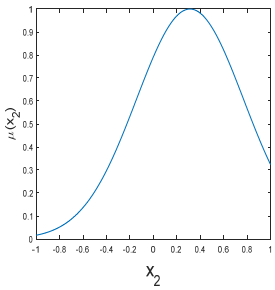
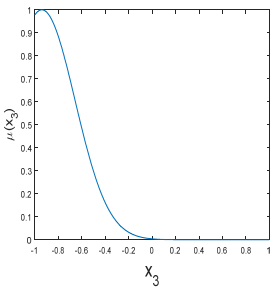
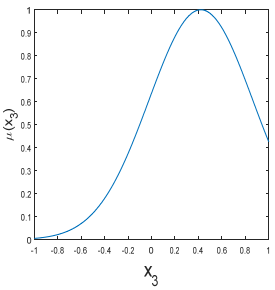
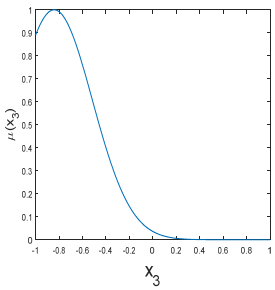
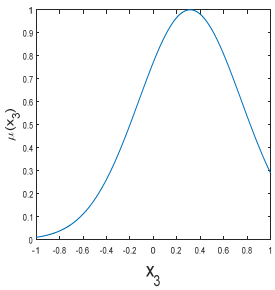
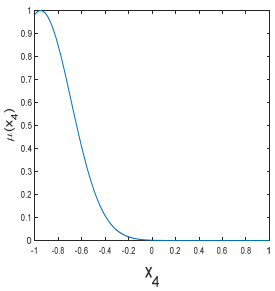
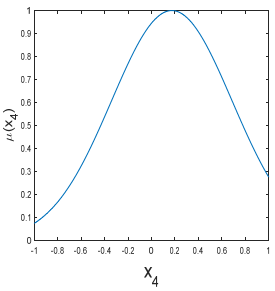
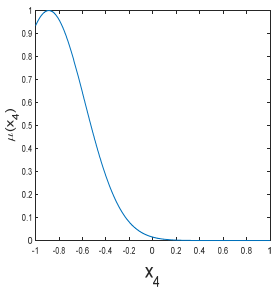
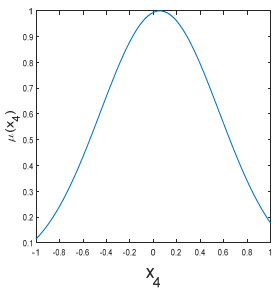
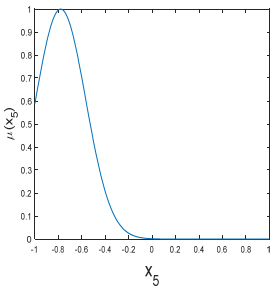
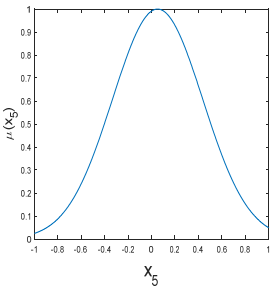
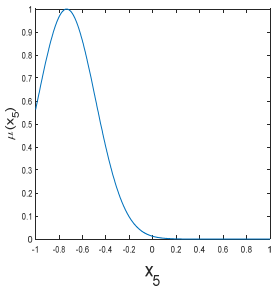
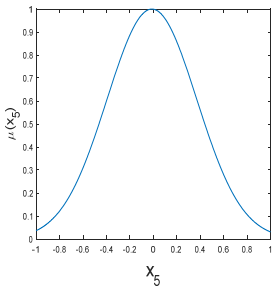
Part B: Decision threshold for classification

Decision threshold of GHM-MPM:TSKFLS: $b=0.6821$

Part C: Reliability index of the classification model

Lower bound of correct classification: $\alpha=0.9386$

Features	**Membership functions of fuzzy sets in Rules**			
	Fuzzy Rule 1	Fuzzy Rule 2	Fuzzy Rule 3	Fuzzy Rule 4
Clump thickness	 (**-0.79**,0.25)* Very low**	 (**0.36**,0.32) High	 (**-0.22**,0.20) Low	 (**0.37**,0.33) Very high

Uniformity of cell size	 (**-0.96**,0.16) Very low	 (**0.44**,0.43) Very high	 (**-0.89**,0.21) Low	 (**0.31**,0.42) High
Uniformity of cell shape	 (**-0.94**,0.16) Very low	 (**0.42**,0.39) Very high	 (**-0.84**,0.21) Low	 (**0.31**,038) High
Marginal adhesion	 (**-0.95**,0.14) Very low	 (**0.17**,0.53) Very high	 (**-0.89**,0.19) Low	 (**0.06**,0.51) High
Single epithelial cell size	 (**-0.78**,0.09) Very low	 (**0.05**,0.30) Very high	 (**-0.74**,0.12) Low	 (**-0.01**,0.29) High

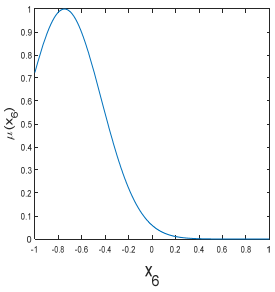
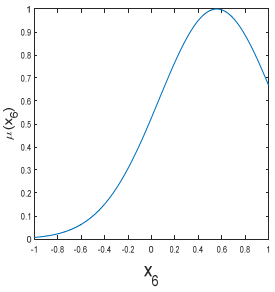
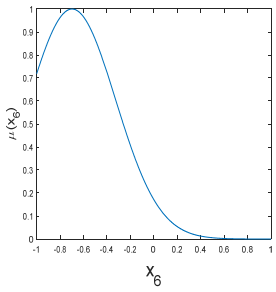
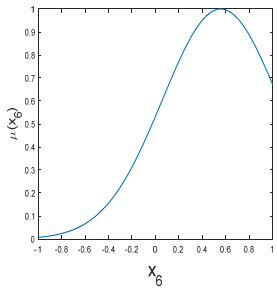
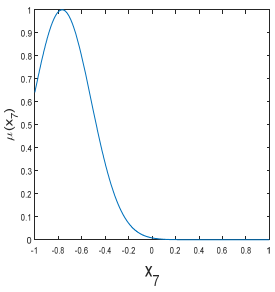
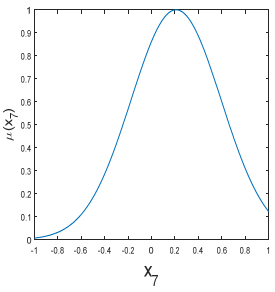
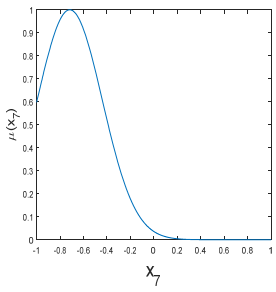
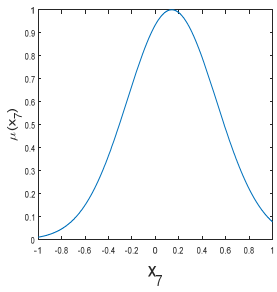
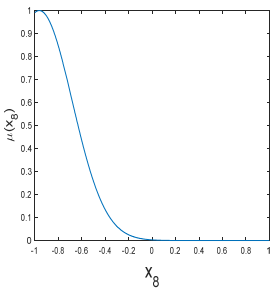
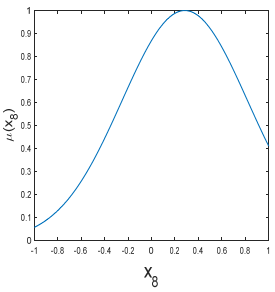
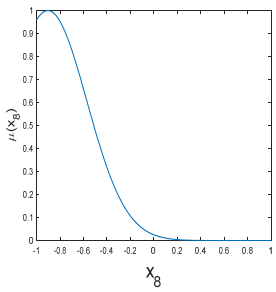
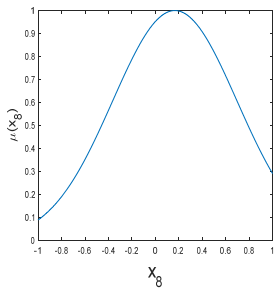
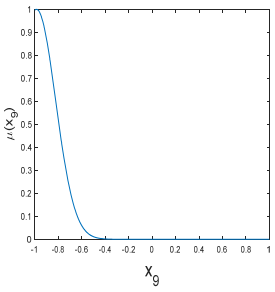
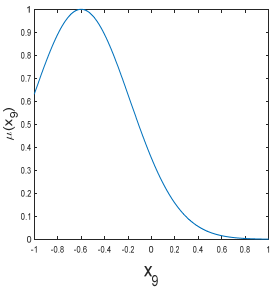
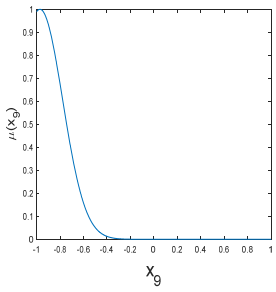
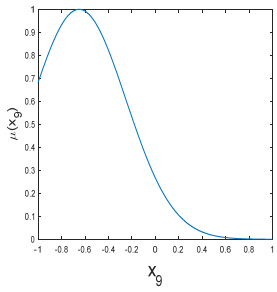
Bare nuclei				
	(**-0.75**,0.20)	(**0.55**,0.47)	(**-0.70**,0.28)	(**0.56**,0.49)
	Very low	High	Low	Very high
Bland chromatin				
	(**-0.77**,0.12)	(**0.21**,0.30)	(**-0.72**,0.15)	(**0.14**,0.29)
	Very low	Very high	Low	High
Normal nucleoli				
	(**-0.96**,0.16)	(**0.28**,0.58)	(**-0.90**,0.22)	(**0.17**,0.56)
	Very low	Very high	Low	High
Mitoses				
	(**-0.99**,0.05)	(**-0.60**,0.35)	(**-0.97**,0.07)	(**-0.65**,0.32)
	Very low	Very high	Low	High

Fig. 7 The membership functions and the possible linguistic explanation of each fuzzy subset in the antecedent of the fuzzy rules for the TSK fuzzy system that is obtained by GHM-MPM:TSKFLS based on the breast dataset.

***The parameter (c_1^1, δ_1^1) of the membership function of the fuzzy set that is associated with the clump thickness feature (the first dimension of the data) in the first rule.**

****A possible explanation for the obtained fuzzy set.**

In Table 6, each membership function of Part A corresponds to a fuzzy set, which can be explained by a medical expert in medical terms with medical knowledge. To provide further explanation, we present the corresponding membership functions (MFs) of each fuzzy set that are obtained for all the fuzzy rules in Fig. 7. Since each specialist may have his own understanding of a given fuzzy membership function, the explanations of the derived fuzzy rules from different specialists will vary. Thus, only a potential explanation for the derived fuzzy rules can be given. Consider the rules in the first row of Fig. 7 as an example. According to the antecedent parameters (centre c and variance δ) of feature "clump thickness" of the breast data in Fig. 7, i.e., (-0.79,0.25) for the 1st fuzzy rule, (0.36,0.32) for the 2nd fuzzy rule, (-0.22,0.20) for the 3rd fuzzy rule and (0.37,0.33) for the 4th fuzzy rule, four MFs can be generated to partition this feature space. Furthermore, these four MFs can be linguistically expressed as "high", "low", "very low" and "very high" in terms of the values of the centres. Similarly, the other features can also be divided into the corresponding four fuzzy subsets. Finally, with the linguistic expressions of the IF-part and the corresponding linear function of the THEN-part, the four fuzzy rules can be described linguistically as follows:

The first fuzzy rule:

If the clump thickness is very low, and if the uniformity of cell size is very low, and if the uniformity of cell shape is very low, and if the marginal adhesion is very low, and if the single epithelial cell size is very low, and if the bare nuclei is very low, and if the bland chromatin is very low, and if the normal nucleoli is very low, and mitoses is very low, THEN this rule gives the decision values of the two outputs with the following formula:

$$f^1(\mathbf{x}) = 0.7966 - 0.7694x_1 + 0.0157x_2 + 0.1192x_3 - 0.2716x_4 + 0.0171x_5 + 0.2868x_6 + 0.0938x_7 + 0.2520x_8 - 0.2093x_9 ;$$

The second fuzzy rule:

If the clump thickness is high, and if the uniformity of cell size is very high, and if the uniformity of cell shape is very high, and if the marginal adhesion is very high, and if the single epithelial cell size is very high, and if the bare nuclei is high, and if the bland chromatin is very high, and if the normal nucleoli is very high, and mitoses is very high, THEN this rule gives the decision values of the two outputs with the following formula:

$$f^2(\mathbf{x}) = 0.4147 + 0.0408x_1 - 0.0050x_2 + 0.0467x_3 - 0.0137x_4 - 0.0050x_5 + 0.0159x_6 + 0.0369x_7 - 0.0339x_8 + 0.0155x_9 ;$$

The third fuzzy rule:

If the clump thickness is low, and if the uniformity of cell size is low, and if the uniformity of cell shape is low, and if the marginal adhesion is low, and if the single epithelial cell size is low, and if the bare nuclei is low, and if the bland chromatin is low, and if the normal nucleoli is low, and mitoses is low, THEN this rule gives the decision values of the two outputs with the following formula:

$$f^3(\mathbf{x}) = -1.0369 - 0.4007x_1 - 0.0041x_2 - 0.5028x_3 - 0.3005x4 + 0.1209x_5 - 1.0408x_6 - 0.3258x_7 - 0.5742x_8 - 0.5802x_9 ;$$

The fourth fuzzy rule:

If the clump thickness is very high, and if the uniformity of cell size is high, and if the uniformity of cell shape is high, and if the marginal adhesion is high, and if the single epithelial cell size is high, and if the bare nuclei is very high, and if the bland chromatin is high, and if the normal nucleoli is high, and mitoses is high, THEN this

rule gives the decision values of the two outputs with the following formula:

$$f^4(\mathbf{x}) = 0.5046 - 0.0653x_1 - 0.0062x_2 - 0.0689x_3 + 0.0107x_4 + 0.0074x_5 - 0.0435x_6 - 0.0598x_7 + 0.0404x_8 - 0.0373x_9 .$$

In the above fuzzy base, all the rules are combined to produce the integrated output. Furthermore, from Table 6, two conclusions about the trained fuzzy classifier are obtained as follows:

1) The decision threshold of the classification model is 0.6821.
2) The lower bound of the correct classification probability, i.e., the model reliability that is obtained by GHM-MPM:TSKFLS, is 93.86%.

The above conclusions also enhance the interpretability of the obtained fuzzy classifier to some extent. While the lower bound of the correct classification probability ensures that users are more confident about the decision results, the decision threshold provides additional information for specialists to further analyse the diagnostic results. According to the above analysis, GHM-MPM:TSKFLS can infer understandable rules from data directly.

7.2.3 Comparing GHM-MPM:TSKFLS with the Hamming-Clustering-Based Rule Generation Method

In addition to the proposed rule-based GHM-MPM:TSKFLS with the good interpretability, there are many other methods that can infer understandable rules for classification tasks, such as the Hamming Clustering (HC)-based rule generation method [27]. The HC-based method has shown promising results for classification tasks. For example, its error rate on the breast dataset is below 3% [27], which is comparable to that obtained by the proposed GHM-MPM:TSKFLS. Another distinctive characteristic of the HC-based method is that it can reconstruct the deterministic AND-OR expression for the generated rules. Rules of this type are more concise than those generated by GHM-MPM:TSKFLS, in which the fuzzy AND expression is used. Therefore, it may be a useful strategy to improve the proposed GHM-MPM:TSKFLS by introducing the fuzzy AND-OR expression into the rules.

7.3 Experiment 3 on the Iris Dataset

In this experiment, we evaluate the performance of the proposed GHM-MPM on the Iris dataset in the UCI Machine Learning Repository, which contains 50 instances of Iris Setosa, 50 instances of Iris Versicolor and 50 instances of Iris Virginica. Each instance has 4 attributes (lengths and widths of sepal and petal), as shown in Fig. 8. In this experiment, each attribute is normalized to the range $[-1, 1]$. Experimental results are listed in Tables 7 and 8.

According to Table 7, the classification accuracies of the models that are trained by GHM-MPM are comparable to those trained by the classical methods, such as SVM.

According to Table 8 and Fig. 8, $\alpha(Versicolor:Virginica)$ is the smallest because Iris Versicolor is close to Iris Virginica. Since Iris Setosa is more isolated from the other two classes, $\alpha(Versicolor:Virginica) < \alpha(Setosa:Versicolor) < \alpha(Setosa:Virginica)$ always holds, which reflects the separation degrees

between different classes. Furthermore, the rows of Table 8 show that nonlinear models that are trained by GHM-MPM outperform the linear model in terms of reliability.

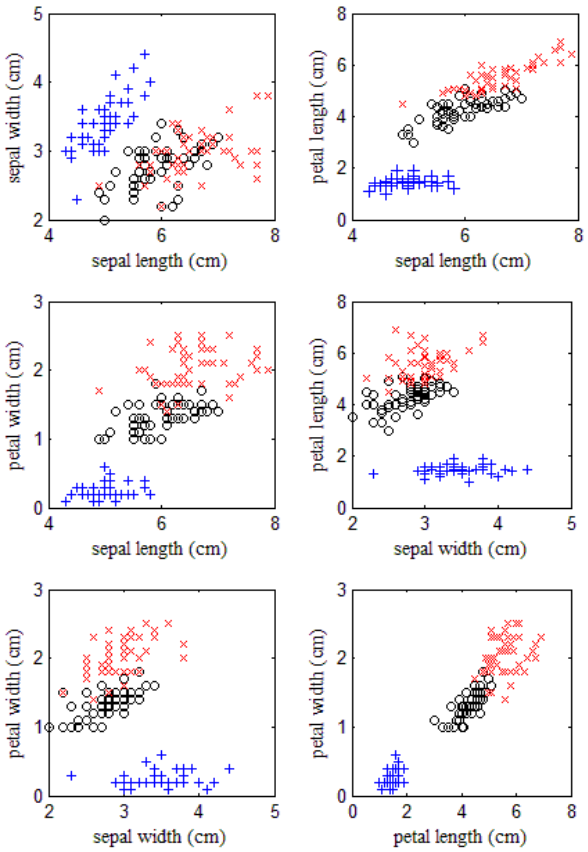


Fig. 8. Scatterplots of Iris Dataset
(Setosa = '+', Versicolor = 'o' and Virginica = 'x')

Table 7 Classification accuracies of different methods on the iris dataset

ACC	GHM-MPM				RBF-NN	ELM	SVM	KRR	L2-TSK-FS	ID3
	LINEAR	KERNEL	TSKFLS	SHLFNN						
Testing	0.9733 ± 0.0279	0.9733 ± 0.0279	0.9733 ± 0.0365	**0.9800 ± 0.0298**	**0.9800 ± 0.0298**	0.9733 ± 0.0279	0.9600 ± 0.0149	0.9600 ± 0.0279	0.9733 ±0.0149	0.9267 ±0.0435
Training	0.9800 ± 0.0075	0.9800 ± 0.0075	0.9867 ± 0.0075	0.9850 ± 0.0091	0.9867 ± 0.0075	0.9700 ± 0.0183	0.9650 ± 0.0070	0.9817 ± 0.0070	0.9717 ±0.0095	0.9750 ±0.0102

Table 8 The α indices of the models trained by GHM-MPM on the iris dataset

α	GHM-MPM			
	LINEAR	KERNEL	TSKFLS	SHLFNN
Setosa : Versicolor	0.9647 ± 0.0038	0.9713 ± 0.0037	0.9927 ± 0.0011	0.9893 ± 0.0008
Setosa : Virginica	0.9815 ± 0.0017	0.9900 ± 0.0009	0.9987 ± 0.0004	0.9975 ± 0.0004
Versicolor : Virginica	**0.7843 ± 0.0223**	**0.7858 ± 0.0221**	**0.8337 ± 0.0187**	**0.8319 ± 0.0195**
Average	0.9102 ± 0.0097	0.9157 ± 0.0089	0.9417 ± 0.0067	0.9396 ± 0.0069

7.4 Comparison of Running Times

The running time of GHM-MPM is analysed in this subsection. In Table 9, the running times of different methods on different datasets are presented. According to Table 9, GHM-MPM:LINEAR has the fastest training speed. GHM-MPM:SHLFNN and GHM-MPM:TSKFLS have lower training speeds than ELM, SVM and KRR on most datasets.

While the proposed GHM-MPM has shown different running times for several classical intelligent models, the models that are generated with GHM-MPM have different advantages and distinctive characteristics. For example, according to Tables 2, 4 and 7, GHM-MPM:SHLFNN has the highest testing accuracy. From subsection 7.2.2, GHM-MPM:TSKFLS can generate understandable rules from data directly and achieves competitive classification ability. Thus, the proposed GHM-MPM is suitable for different application scenes. For example, the interpretability and classification accuracy of a model are more important than the training speed in medical diagnosis. In this scenario, we select GHM-MPM:TSKFLS to generate a TSK fuzzy system for decision-making.

Table 9 Running times of different methods on different datasets (seconds)

Dataset	Time	GHM-MPM				RBF-NN	ELM	SVM	KRR	L2-TSK-FS	ID3
		LINEAR	KERNEL	TSKFLS	SHLFNN						
Synthetic Dataset	Training time	**7.50 E-04**	8.90 E-02	6.98 E-02	4.72 E-02	3.78 E-01	3.40 E-03	3.70 E-03	1.01 E-02	8.51 E-01	5.50 E-03
	Testing time	**7.30 E-05**	2.60 E-03	5.91 E-04	4.01 E-04	1.35 E-02	4.91 E-04	4.41 E-04	9.01 E-04	7.08 E-04	5.15 E-04
Breast	Training time	**5.10 E-04**	2.35 E-01	3.32 E-01	4.18 E-01	8.69 E-01	8.30 E-03	1.14 E-02	1.05 E-01	2.71 E+00	7.10 E-03
	Testing time	**5.98 E-05**	1.95 E-02	2.80 E-03	4.00 E-03	1.75 E-02	1.30 E-03	2.00 E-03	4.60 E-03	8.39 E-04	6.58 E-04
IRIS	Training time	**1.10 E-03**	1.76 E-02	1.40 E+00	8.66 E-02	2.88 E-01	2.70 E-03	**1.10 E-03**	2.70 E-03	2.47 E-01	5.50 E-03
	Testing time	**4.92 E-05**	8.91 E-04	1.50 E-03	1.10 E-03	1.38 E-02	4.86 E-04	1.31 E-04	4.27 E-04	8.95 E-04	7.13 E-04

8 Conclusions

In this paper, a generalized MPM method, i.e., GHM-MPM, has been proposed for training different

classical intelligent models and has realized the reliability learning of these models for binary classification. Our experiments show that the classical intelligent models that are trained by GHM-MPM achieve comparable classification performance to those trained by traditional methods. Furthermore, GHM-MPM has been extended to multi-class classification to analyse the corresponding task, which leads to a clearer understanding of the recognition task.

The training of models by GHM-MPM, such as TSK fuzzy logic systems and feedforward neural networks, demonstrates that many existing models can also be trained by the proposed method. Models that are trained by GHM-MPM are more transparent than those trained by traditional methods.

The proposed method also faces some challenges, which will be addressed in future work. For example, when the GHM-MPM method is used to train a classification model with the kernel trick, the testing accuracy may be lower than the reliability index. We are working on exploring such issues in depth.

Acknowledgement

This work was supported in part by the National Key Research Program of China under Grant 2016YFB0800803, in part by the National Natural Science Foundation of China under Grant 61772239 and Grant 61403247, in part by the Outstanding Youth Fund of Jiangsu Province under Grant BK20140001, in part by the Fundamental Research Funds for the Central Universities (JUSRP41704), in part by the Hong Kong Research Grants Council (PolyU 152040/16E), in part by the Hong Kong Polytechnic University (G-UA68, G-UA3W), and in part by the Open Fund Project of Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University) (MJUKF201725).

References

[1] S. An, W. Liu, S. Venkatesh, Fast cross-validation algorithms for least squares support vector machine and kernel ridge regression, Pattern Recognition, 40(8) (2007) 2154-2162.

[2] J.C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm, Computers & Geosciences, 10(84) (1984) 191-203.S. Chen, C.F.N. Cowan, P.M. Grant, Orthogonal least squares learning algorithm for radial basis function networks, IEEE Transactions on Neural Networks, 2(2) (1991) 302-309.

[3] S. Chen, C.F.N. Cowan, P.M. Grant, Orthogonal least squares learning algorithm for radial basis function networks, IEEE Transactions on Neural Networks, 2(2) (1991) 302-309.C.C. Chang, C.J. Lin, LIBSVM: A Library for Support Vector Machines, ACM Transactions on Intelligent Systems and Technology, 2(3) (2011) 1-27.

[4] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge University Press, 2000.Z. Deng, K. S. Choi, F. L. Chung, and S. Wang, Scalable tsk fuzzy modeling for very large datasets using minimal-enclosing-ball approximation, IEEE Transactions on Fuzzy Systems, 19(2) (2011) 210-226.

[5] C.C. Chang, C.J. Lin, LIBSVM: A Library for Support Vector Machines, ACM Transactions on Intelligent Systems and Technology, 2(3) (2011) 1-27.J.C. Bezdek, R. Ehrlich, W. Full, FCM: The fuzzy c-means clustering algorithm, Computers & Geosciences, 10(84) (1984) 191-203.

[6] Z. Deng, K.S. Choi, Y. Jiang, S. Wang, Generalized Hidden-Mapping Ridge Regression, Knowledge-Leveraged Inductive Transfer Learning for Neural Networks, Fuzzy Systems and Kernel Methods, IEEE Transactions on Cybernetics, 44(12) 2014 2585-2599.
[7] Z. Deng, K. S. Choi, F. L. Chung, and S. Wang, Scalable tsk fuzzy modeling for very large datasets using minimal-enclosing-ball approximation, IEEE Transactions on Fuzzy Systems, 19(2) (2011) 210-226.
[8] Z. Deng, L. Cao, Y. Jiang, S. Wang, Minimax Probability TSK Fuzzy System Classifier: A More Transparent and Highly Interpretable Classification Model, IEEE Transactions on Fuzzy Systems, 23(4) (2015) 813-826.
[9] S. Haykin, Neural Networks and Learning Machines, Pearson Prentice, Hall, 2009.
[10] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks, 2(89) (1989) 359-366.
[11] G. Huang, L. Chen, C.K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Transactions on Neural Networks, 17(4) (2006) 879-892.
[12] K.J. Hunt, R. Haas, R. Murray-Smith, Extending the functional equivalence of radial basis function networks and fuzzy inference systems, IEEE Transaction on Neural Networks, 7(3) (1996) 776-781.
[13] G. Huang, Q. Zhu, C.K. Siew, Extreme learning machine: theory and applications, Neurocomputing, 70(1) (2006) 489-501.
[14] S. Haykin, Neural Networks and Learning Machines, Pearson Prentice, Hall, 2009.
[15] K. Huang., H. Yang, I. King, et al., The Minimum Error Minimax Probability Machine, Journal of Machine Learning Research, 5(4) (2004) 1253-1286..
[16] K. Huang, H. Yang, I. King, M.R. Lyu, Learning classifiers from imbalanced data based on biased minimax probability machine, in: 2004 Computer Vision and Pattern Recognition (CVPR), 2004, pp. II-558-II-563.
[17] K. Huang, H. Yang, I. King, M.R. Lyu, Imbalanced learning with a biased minimax probability machine, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 36(4) (2006) 913-923.
[18] F. Jos. Sturm, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, Optimization Methods & Software, 11(1-4) (1999) 625-653.
[19] J.S.R. Jang, C.T. Sun, Functional equivalence between radial basis function networks and fuzzy inference systems, IEEE Transactions on Neural Networks, 4(1) (1993) 156-159.
[20] G.R.G Lanckriet., L.E. Ghaoui, C. Bhattacharyya, et al., Minimax probability machine, in: 2001 Neural Information Processing Systems (NIPS), 2001, pp. 801–807.
[21] M.S. Lobo, L. Vandenberghe, S. Boyd, et al., Applications of second-order cone programming, Linear Algebra & Applications, 284(98) (1998) 193-228.
[22] G.R.G. Lanckriet, L.E. Ghaoui, C. Bhattacharyya, M.I. Jordan, A robust minimax approach to classification, The Journal of Machine Learning Research, 3(3) (2003) 555-582.
[23] M. Lichman (2013), UCI machine learning repository [online]. Available: archive.ics.uci.edu/ml.
[24] E. Lughofer, Single-pass active learning with conflict and ignorance, Evolving Systems, 3(4) (2012) 251-271.
[25] E. Lughofer, O. Buchtala, Reliable All-Pairs Evolving Fuzzy Classifiers, IEEE Transactions on Fuzzy Systems, 21(4) (2013) 625-641.
[26] R.D. Maesschalck, D. Jouan-Rimbaud, D.L. Massart, The Mahalanobis distance, Chemometrics & Intelligent Laboratory Systems, 50(99) (2000) 1-18.
[27] M. Muselli, D. Liberati, Binary rule generation via Hamming clustering, IEEE Transactions on Knowledge and Data Engineering, 14(6) (2002) 1258-1268
[28] G. Ou, L.M. Yi, Multi-class pattern classification using neural networks, Pattern Recognition, 40(1) (2007) 4-18.
[29] J.H. Park, I.W. Sandberg, Universal approximation using radial-basis-function networks, Neural Computation, 3(2) (1991) 246-257.
[30] J.R. Quinlan, Induction of Decision Trees. Kluwer Academic Publishers, 1986.

[31] C. Saunders, A. Gammerman, V. Vovk, Ridge Regression Learning Algorithm in Dual Variables, in: 1998 International Conference on Machine Learning (ICML), 1998, pp. 515-521.
[32] T. Strohmann, G.Z. Grudic, A formulation for minimax probability machine regression, in: 2002 Neural Information Processing Systems (NIPS), 2002, pp. 769-776.
[33] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Transaction on Systems, Man and Cybernetics, SMC-15(1) (1985) 116-132.