Available online at www.sciencedirect.com

ScienceDirect

Fuzzy Sets and Systems ••• (••••) •••–•••

FUZZY sets and systems

www.elsevier.com/locate/fss

Incremental feature weighting for fuzzy feature selection

Ling Wang [a,b,*], Jianyao Meng [a,b], Ruixia Huang [a,b], Hui Zhu [a,b], Kaixiang Peng [a,b]

[a] *College of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China*
[b] *Key Laboratory of Knowledge Automation For Industrial Processes, University of Science and Technology Beijing, Ministry of Education, Beijing 100083, China*

Received 20 October 2017; received in revised form 19 October 2018; accepted 27 October 2018

Abstract

Feature selection presents many challenges and difficulties during online learning. In this study, we focus on fuzzy feature selection for fuzzy data stream. We present a novel incremental feature weighting method with two main phases comprising offline fuzzy feature selection and online fuzzy feature selection. A sliding window is used to divide the fuzzy data set. Each fuzzy input feature is assigned a weight from [0, 1] according to the mutual information shared between the input features and the output feature. These weights are employed to access the candidate fuzzy feature subsets in the current window and based on these subsets, the offline fuzzy features selection algorithm is applied to obtain the fuzzy feature subsets by combining the backward feature selection method with the fuzzy feature selection index in the first sliding window. The online feature selection algorithm is performed in each of the new sliding windows. The feature subset in the current window is updated by combining the fuzzy feature selection results from the previous sliding window with the current candidate fuzzy feature set according to the importance level of the fuzzy input feature. Finally, the evolving relationships of the fuzzy input features are found using the fuzzy feature weight between the sliding windows. Simulation results showed that the proposed algorithm obtains significantly improved adaptability and prediction accuracy compared with existing algorithms.

Keywords: Feature selection; Fuzzy; Incremental feature weighting; Mutual information

1. Introduction

The online monitoring of high-dimensional dynamical data has become increasingly important for observing systems in a wide range of advanced applications, such as telephone records, large sets of web pages, multimedia data, and sets of retail chain transactions [1]. Due to the complexity of dynamical systems, incremental learning is essential for knowledge discovery, which is an important aspect of human intelligence. Many applications need to process data in a sequence, which has led to the generation of a wide range of incremental learning methods [2], [3]. In particular, two types of situations must be addressed. In the first situation, the data set cannot be collected in one pass and batch

* Corresponding author.
E-mail address: lingwang@ustb.edu.cn (L. Wang).

https://doi.org/10.1016/j.fss.2018.10.021
0165-0114/

computing cannot be conducted, e.g., online applications [4] or interaction queries [5]. In the second situation, the data set is excessively large and it cannot be calculated in one pass due to limitations in terms of the computation capability and the memory size. Thus, the data set must be divided into segments and added successively. The relationships between the newly data added and the stored data structure must then be analyzed in order to acquire new knowledge. It is difficult to apply an incremental feature selection process to compensate for different operational modes because the importance level of features can change dynamically during the overall learning process. For example, specific features may be much more important initially than later in the process. Therefore, many incremental feature selection algorithms have been proposed for synchronously updating the classifiers [6–8,10]. In [6], the incrementally selected features for representing the information in the overall data set are evaluated using classifiers, which increases the computational cost of the feature selection process. In [7], a wrapper feature selection algorithm with an incremental Bayesian classifier (WFSIBC) was proposed where a greedy algorithm is used to find the optimal feature subset, which increases the time complexity and space complexity. The incremental feature neighborhood rough set (IFNRS) algorithm was presented by [8] to select the features incrementally but without considering constraints on discrete data. An incremental feature selection algorithm based on information granularity was proposed by [9] where the non-matrix structure improves the efficiency of the algorithm. Other incremental feature selection algorithms such as those presented by [10–13] can be employed for regression prediction. In the method described by [10], the useful features are selected incrementally by combining clustering algorithms with four new feature quality indices but the number of clusters needs to be predefined. A genetic algorithm and the K-means algorithm are used to determine the objective function and the optimal range, respectively, in the method proposed by [11] but the features subset is not globally optimal. An online unsupervised multi-view feature selection (OMVFS) algorithm was presented by [12], where feature selection is embedded directly in a clustering algorithm that retains the local information structure, but the parameters need to be set for the clustering algorithm, which leads to poor adaptability. The feature selection based on data streams (FSDS) algorithm was proposed by [13] for determining the importance level of features according to the regression results based on approximate data, but it has no independent evaluation index and it is mainly employed for updating the regression model.

It is difficult to apply an incremental feature selection process to a fuzzy feature space with fuzzy linguistic terms in order to improve the interpretability of the system. A significant problem when learning fuzzy features from data is the so-called curse of dimensionality, where the fuzzy feature space has higher dimensionality than the original feature space, which increases the complexity of the system. This problem was addressed by [14] where the fuzzy features were selected by evaluating the importance level of each fuzzy feature according to the minimum–maximum learning rule and the extended matrix built based on the fuzzy data, but this algorithm is unsuitable for dynamic systems. A local fuzzy feature selection strategy based on switching to a neighboring model was proposed by [15] for capturing the local dynamics, but is not possible to inactivate and reactivate the input variables in different stages. An online incremental feature weighting method was presented by [16] for updating the classifiers according to a separability criterion, where the weight was calculated for each feature and compared with all of the features. The most important feature received a weight of 1 and the others were weighted between 0 and 1. The weights of all the features were then used directly in the model for updating and inference, but this method may lead automatically to the curse of dimension reduction during rule evolution. Soft feature selection using a feature weighting method was presented by [17] based on a classifier, where the Fisher separability criterion was applied in the feature space to select suitable features, although it was more convenient to apply in the empirical feature space. An adaptive incremental fuzzy feature selection method using a neo-fuzzy neural network was presented by [18], which selects the model input and updates the network weight simultaneously, although this method leads to redundant features because the dependencies between the features are not analyzed. As described above, the inclusion of feature weights discriminates features that are more important and less important, thereby providing additional information to users and experts by giving insights into the feature selection process and its interpretability. However, in order to implement feature selection, these algorithms depend on a classifier or regression model, which increases the complexity of the model selection procedure.

2. Proposed approach

In this study, we propose an incremental feature weighting for fuzzy feature selection (IWFFS) algorithm with two main phases: offline fuzzy feature selection and online fuzzy feature selection. First, the fuzzy data are partitioned

Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021

by sliding windows. In the first sliding window, the offline fuzzy feature selection algorithm is applied, where the weight of each fuzzy input feature is calculated according to the mutual information between the fuzzy input features and the output feature, and they are then sorted in descending order according to their weights. In order to identify the differences between the fuzzy features according to their weights, the gradients of the weights of the adjacent fuzzy input features are calculated to find the optimal single fuzzy input feature to obtain the candidate fuzzy features subset, which narrows the search space. However, feature selection requires maintaining the whole optimal input feature subset. Thus, the fuzzy input feature selection index (FCI) is viewed as a measurement criterion for selecting incremental fuzzy features subsets, where it considers the mutual information shared between the input feature and output feature, as well as the mutual information between fuzzy input features. Next, online fuzzy feature selection is applied in each of the sliding windows. In order to adaptively track the changes in the fuzzy feature weights and obtain the optimal fuzzy features subset for the current sliding window, the candidate fuzzy features subset in the current sliding window and the optimal fuzzy features subset in the previous sliding window are used simultaneously to increase the interpretability of the model and to ensure the smoothness of the learning process. Regardless of whether the fuzzy data are categorical data or continuous data, our approach can obtain the optimal fuzzy features subset without using a classifier or regression model, or by predefining any parameter, which improves the adaptability of the algorithm.

3. Incremental feature weighting

3.1. Basic concept

The incremental feature weighting concept is an attempt to reduce the curse of dimensionality. This method is generally used for incremental feature selection by assigning continuous weights in the range of $[0, 1]$, which can be treated as the importance levels of the features in dynamic systems, instead of using crisp weights from $\{0, 1\}$. When the feature weights approach 0, their importance level is relatively low compared with others that have values near 1. The main advantage of feature weighting compared with feature selection is clear when applied in an incremental learning scenario. Incremental feature selection abruptly changes the weights from 0 to 1 or vice versa to follow the importance level of the features in a smooth and continuous manner. In the context of incremental feature selection, the feature weights must also be permanently and synchronously updated in order to select the most appropriate features with the minimum redundancy among the input features and the maximum dependency on the output feature. Before calculating the feature weights, we first fuzzify the original data, as explained in Section 3.2. In Section 3.3, we describe the detection of the distribution of the data by using the Hoeffding boundary to adaptively determine the sliding windows and to partition the fuzzy data. Finally, in Sections 3.4 and 3.5, we define the fuzzy information entropy and provide a method for calculating the fuzzy feature weights.

3.2. Fuzzification

The fuzzy approach is very simple and similar to the way humans think, so it is exploited widely in intelligent systems [14]. In addition, the fuzzy approach is helpful for abstracting at a high level rather than applying a simple hard division of the data. In general, it is necessary to establish a membership function based on experience or knowledge to achieve fuzzification. However, the available information and expert knowledge are difficult to obtain in many cases. In our method, the samples for each feature are clustered [19] to obtain the fuzzy data. We assume that the data set D_p in the pth ($p \geq 0$) sliding window contains n samples, L input features, and one output feature, which can be represented as follows.

$$D_p = \begin{bmatrix} x_p^{11} & \cdots & x_p^{1L} & y_p^1 \\ x_p^{12} & \cdots & x_p^{2L} & y_p^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_p^{1n} & \cdots & x_p^{nL} & y_p^n \end{bmatrix} \tag{1}$$

Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021

The input features vector is denoted as $\vec{X}_p = [X_p^1, X_p^2, \cdots, X_p^j, \cdots, X_p^L]$. The output feature vector is denoted as $Y_P = [y_p^1, y_p^2, \cdots, y_p^j, \cdots y_p^n]$. We assume that the input feature X_p^j is discretized after clustering [19] to form $\left|C_p^j\right|$ partitions, which can be fuzzified as a fuzzy features vector:

$$\tilde{X}_P^j = [\tilde{X}_p^{j1}, \tilde{X}_p^{j2}, \cdots, \tilde{X}_p^{jq}, \cdots, \tilde{X}_p^{j|C_p^j|}], \tag{2}$$

where $\tilde{X}_p^{jq}$ is the qth ($1 \le q \le \left|C_p^j\right|$) fuzzy feature of the jth ($1 \le j \le L$) feature in the pth sliding window, which can also be represented as the fuzzy equivalence class generated by $\tilde{X}_p^{jq}$ and all of the samples:

$$\tilde{X}_p^{jq} = \frac{\mu_{\tilde{X}_p^{jq}}(x_p^{1j})}{x_p^{1j}} + \frac{\mu_{\tilde{X}_p^{jq}}(x_p^{2j})}{x_p^{2j}} + \cdots + \frac{\mu_{\tilde{X}_p^{jq}}(x_p^{ij})}{x_p^{ij}} + \cdots + \frac{\mu_{\tilde{X}_p^{jq}}(x_p^{nj})}{x_p^{nj}}, \tag{3}$$

where $\frac{\mu_{\tilde{X}_p^{jq}}(x_p^{ij})}{x_p^{ij}}$ is a fuzzy partition of x_p^{ij} induced by fuzzy membership function $\mu_{\tilde{X}_p^{jq}}(x_p^{ij})$:

$$\mu_{\tilde{X}_p^{jq}}(x_p^{ij}) = \exp\left(-\frac{(x_p^{ij} - \bar{x}_p^j)^2}{2(\sigma_p^j)^2}\right),\ 1 \le i \le n,\ 1 \le j \le L,\ 1 \le q \le \left|C_p^j\right|, \tag{4}$$

where n is the number of samples, L is the number of input features, x_p^{ij} is the value of the jth input feature of the ith sample in the pth sliding window, and $\bar{x}_p^j$ and $(\sigma_p^j)^2$ are the mean and variance of the samples of the jth input feature in the pth sliding window, respectively.

3.3. Sliding window

Sliding windows are applied widely in dynamic conditions. Each sliding window reads the new data and discards the previous data. Based on the sliding window technique, we can test the latest data points in the current window to determine whether the characteristics of the data are similar to those in the last window or not. There are two main methods for applying sliding windows: using a fixed window size [20] or employing a decay factor to determine the abandoned data and retained data [21]. However, the sensitive parameters need to be predefined, where they can be determined according to the data distribution. For example, the Hoeffding bound [22] adjusts the window size according to the real data distribution and the range of features. During the incremental feature selection process, the aim of the sliding window is to detect the changes in features. The selected features in the adjacent sliding window are used to analyze the evolution of the weights for each input feature. Let ε be the Hoeffding bound. Suppose that we have n samples and the difference is $1 - \delta$ (δ is generally set to 0.05). For the feature with range R, the difference between the estimated mean value and the actual mean value does not exceed ε, which is computed as follows

$$\varepsilon = \sqrt{\frac{R \cdot R^T \ln(1/\delta)}{2n}} \tag{5}$$

The Hoeffding bound becomes smaller as the value of n increases, so the Hoeffding bound ε will approach 0 if n is sufficiently large. Thus, the estimated mean will fully approximate the true mean of the original samples.

The minimum window size N_H [23] can be obtained from the Hoeffding bound by:

$$N_H = \frac{R^2 \ln(1/\delta)}{2\varepsilon^2}. \tag{6}$$

In order to obtain ε, we assume that W_{p-1} and W_p are two sliding windows, and μ_{p-1} and μ_p are their means with probability $1 - \delta$, respectively. If $\left|\mu_{p-1} - \mu_p\right| \le 2\varepsilon$, then from Eq. (5) and Eq. (6), we have the following.

$$N_H = \frac{2R^2 \ln(1/\delta)}{(\mu_{p-1} - \mu_p)^2} \tag{7}$$

Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021

3.4. Fuzzy information entropy

The fuzzy information entropy [24] can be used as measure of the discernibility of a fuzzy feature. When the entropy value is greater, the discernibility is stronger and the fuzzy feature is more significant. When a novel fuzzy feature is added to the information system, the entropy value increases monotonously, which reflects how the newly added feature enhances the discernibility. Let f_p^s be the subset of fuzzy features in the pth sliding window. The fuzzy equivalence class generated by D_p^i and f_p^s is represented as $[D_p^i]_{f_p^s}$:

$$[D_p^i]_{f_p^s} = \frac{s(D_p^1, D_p^i)_{f_p^s}}{D_p^1} + \frac{s(D_p^2, D_p^i)_{f_p^s}}{D_p^2} + \cdots + \frac{s(D_p^k, D_p^i)_{f_p^s}}{D_p^1} + \cdots + \frac{s(D_p^n, D_p^i)_{f_p^s}}{D_p^n}, \tag{8}$$

where $s(D_p^k, D_p^i)_{f_p^s}$ is the relation value between D_p^k and D_p^i generated by f_p^s:

$$s(D_p^k, D_p^i)_{f_p^s} = \frac{1}{|f_p^s|} \sum_{j=1}^{L} \sum_{q=1}^{|C_p^j|} (1 - \left| \mu_{\tilde{X}_p^{jq}}(x_p^{ij}) - \mu_{\tilde{X}_p^{jq}}(x_p^{kj}) \right|), \ (\tilde{X}_p^{jq} \in f_p^s, k \neq i), \tag{9}$$

where $|f_p^s|$ is the number of fuzzy features contained in f_p^s, $\tilde{X}_p^{jq}$ is the qth fuzzy feature of the jth feature in the pth sliding window, $\mu_{\tilde{X}_p^{jq}}(x_p^{ij})$ is the membership value of the jth feature value of the ith sample in the pth sliding window corresponding to the fuzzy feature $\tilde{X}_p^{jq}$, $\mu_{\tilde{X}_p^{jq}}(x_p^{kj})$ is the membership value of the jth feature value of the kth sample in the pth sliding window corresponding to the fuzzy feature $\tilde{X}_p^{jq}$, and $B_{f_p^s}^i$ is the cardinality of subset f_p^s corresponding to the ith sample:

$$B_{f_p^s}^i = \left| [D_p^i]_{f_p^s} \right| = \sum_{k=1}^{n} s(D_p^k, D_p^i)_{f_p^s}. \tag{10}$$

The fuzzy information entropy of f_p^s is defined as follows.

$$H(f_p^s) = -\frac{1}{n} \sum_{i=1}^{n} \log \frac{B_{f_p^s}^i}{n} \tag{11}$$

The joint fuzzy information entropy [24] is also defined to measure the discernibility of a relation between the fuzzy features subset and fuzzy feature. When the entropy value is greater, the discernibility is stronger and the fuzzy feature is more closely linked to the fuzzy features subset. We assume that f_p^s is the subset of fuzzy features in the pth sliding window and $\tilde{X}_p^{jq}$ is the qth fuzzy feature of the jth feature in the pth sliding window. $[D_p^i]_{\tilde{X}_p^{jq}}$ is the fuzzy equivalence class generated by D_p^i and $\tilde{X}_p^{jq}$:

$$[D_p^i]_{\tilde{X}_p^{jq}} = \frac{s(D_p^1, D_p^i)_{\tilde{X}_p^{jq}}}{D_p^1} + \frac{s(D_p^2, D_p^i)_{\tilde{X}_p^{jq}}}{D_p^2} + \cdots + \frac{s(D_p^k, D_p^i)_{\tilde{X}_p^{jq}}}{D_p^1} + \cdots + \frac{s(D_p^n, D_p^i)_{\tilde{X}_p^{jq}}}{D_p^n}, \tag{12}$$

where $s(D_p^k, D_p^i)_{\tilde{X}_p^{jq}}$ is the relation value of D_p^k and D_p^i generated by $\tilde{X}_p^{jq}$, as follows.

$$s(D_p^k, D_p^i)_{\tilde{X}_p^{jq}} = 1 - \left| \mu_{\tilde{X}_p^{jq}}(x_p^{ij}) - \mu_{\tilde{X}_p^{jq}}(x_p^{kj}) \right| \tag{13}$$

The joint fuzzy information entropy between f_p^s and $\tilde{X}_p^{jq}$ is computed by:

$$H(\tilde{X}_p^{jq}, f_p^s) = -\frac{1}{n} \sum_{i=1}^{n} \log \frac{\left| [D_p^i]_{f_p^s} \cap [D_p^i]_{\tilde{X}_p^{jq}} \right|}{n}, \ (\tilde{X}_p^{jq} \notin f_p^s). \tag{14}$$

Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021

3.5. Fuzzy feature weight

A fuzzy feature weight is a continuous index in the range of [0, 1], which is used to measure the importance level of a fuzzy feature. A feature has a high importance level when its weight approaches 1 and a low importance level when its weight approaches 0. The fuzzy feature weight has is characterized by softness and smoothness. Softness means that features can be down-weighted instead of being discarded completely. The features with low weights still contain some information. Smoothness means that the fuzzy feature weight will change constantly, so a fuzzy feature with a low weight may become an important feature with a high weight in the subsequent learning process. The weight $w(\tilde{X}_p^{jq})$ of each fuzzy feature is calculated according to the mutual information:

$$w(\tilde{X}_p^{jq}) = \frac{I(\tilde{X}_p^{jq}, Y_p)}{\sum_{j=1}^{L}\sum_{q=1}^{|C_p^j|} I(\tilde{X}_p^{jq}, Y_p)}, \tag{15}$$

where $\tilde{X}_p^{jq}$ is the qth fuzzy feature of the jth feature in the pth sliding window, Y_p is the output feature in the pth slide window, and $I(\tilde{X}_p^{jq}, Y_p)$ is the mutual information between the input feature $\tilde{X}_p^{jq}$ and the output feature Y_p in the pth sliding window:

$$I(\tilde{X}_p^{jq}, Y_p) = H(\tilde{X}_p^{jq}) + H(Y_p) - H(\tilde{X}_p^{jq}, Y_p), \tag{16}$$

where $H(\tilde{X}_p^{jq})$, $H(Y_p)$ and $H(\tilde{X}_p^{jq}, Y_p)$ are calculated according to Eq. (11) and Eq. (14), respectively.

In particular, when Y_p is a class label, the entropy $H(Y_p)$ of Y_p is calculated by:

$$H(Y_p) = -\sum_{l=1}^{|C_p^Y|} \frac{n_l}{n} \log(\frac{n_l}{n}), \tag{17}$$

where $\left|C_p^Y\right|$ is the number of class labels and n_l is the number of samples with class label l.

The joint information entropy between $\tilde{X}_p^{jq}$ and Y_p is calculated as follows.

$$H(\tilde{X}_p^{jq}, Y_p) = -\sum_{l=1}^{|C_p^Y|} \frac{1}{n_l} \sum_{i=1}^{n_l} \log \frac{B^i_{\tilde{X}_p^{jq}}}{n_l} \tag{18}$$

However, if only the fuzzy feature weights are considered, the differences between fuzzy features might not be recognized when the weights between the fuzzy feature are very similar. Thus, we employ the fuzzy weight gradient to select the fuzzy features in order to avoid this problem. The fuzzy features are sorted according to the descending order of their weights and the gradient Δ_p^d of the adjacent fuzzy weights in the pth sliding window is calculated by:

$$\Delta_p^d = |w_p^d - w_p^{d+1}|,\ 1 \le d \le m-1 \tag{19}$$

$$m = \left|C_p^1\right| + \cdots + \left|C_p^j\right| + \cdots + \left|C_p^L\right|, \tag{20}$$

where w_p^{d+1} and w_p^d are the weights of the $(d+1)$th and the dth fuzzy input features sorted in order in the pth sliding window, respectively.

If the adjacent fuzzy weight gradients vary in a certain range, then the effect of the fuzzy input feature on the output feature remains at an almost steady level. If the adjacent weight gradients decrease greatly, then the effect of the fuzzy input feature on the output feature shrinks. In order to adaptively identify the cut-off point where the effect of the fuzzy input feature on output feature shrinks rapidly, the cut-off threshold is calculated for the dth weight gradient in the pth sliding window by using Eq. (23) according to the first $d-1$ weight gradients in the pth sliding window:

Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021

$$\rho_p^d = \frac{\sum_{d1=1}^{d-1} \Delta_p^{d1}}{d-1} + \sqrt{\sum_{d1=1}^{d-1} (\Delta_p^{d1} - \frac{\sum_{d2=1}^{d-1} \Delta_p^{d2}}{d-1})^2}\ ,\ d \geq 3. \tag{21}$$

When $\Delta_p^d > \rho_p^d$, the $(d+1)$th feature is the cut-off point after sorting in descending order.

3.6. Fuzzy input feature selection index

Regardless of whether the fuzzy feature weight or the gradient of the adjacent fuzzy feature weight is employed, they can only measure the quality of a single fuzzy feature. However, the aim of feature selection is to maintain the whole optimal input feature subset. Thus, FCI is employed as a measurement criterion for incremental fuzzy feature selection, where it considers the mutual information shared between the input feature and the output feature, as well as the mutual information between the fuzzy input features:

$$FCI(f_p^s) = \frac{|f_p^s| C\bar{F}_p}{\sqrt{|f_p^s| + |f_p^s|(|f_p^s|-1) F\bar{F}_p}}, \tag{22}$$

where $|f_p^s|$ is the number of fuzzy features contained by f_p^s in the pth sliding window, and $C\bar{F}_p$ is the mean value of the mutual information shared between the fuzzy input features and output feature in the pth sliding window:

$$C\bar{F}_p = \frac{\sum_{j=1}^{L} \sum_{q=1}^{|C_p^j|} I(\tilde{X}_p^{jq}, Y_p)}{|f_p^s|},\ (\tilde{X}_p^{jq} \in f_p^s). \tag{23}$$

$F\bar{F}_p$ is the mean value of the mutual information shared between the two fuzzy input features contained by f_p^s in the pth sliding window:

$$F\bar{F}_p = \frac{\sum_{j,u=1, j\neq u}^{L} \sum_{q,v=1, q\neq v}^{|C_p^{j,u}|} I(\tilde{X}_p^{jq}, \tilde{X}_p^{uv})}{|f_p^s|(|f_p^s|-1)},\ (\tilde{X}_p^{jq}, \tilde{X}_p^{uv} \in f_p^s). \tag{24}$$

The mutual information $I(\tilde{X}_p^{jq}, \tilde{X}_p^{uv})$ shared between the fuzzy feature $\tilde{X}_p^{jq}$ and the fuzzy feature $\tilde{X}_p^{uv}$ in the pth sliding window is calculated by:

$$I(\tilde{X}_p^{jq}, \tilde{X}_p^{uv}) = H(\tilde{X}_p^{jq}) + H(\tilde{X}_p^{uv}) - H(\tilde{X}_p^{jq}, \tilde{X}_p^{uv}), \tag{25}$$

where $H(\tilde{X}_p^{jq})$ and $H(\tilde{X}_p^{uv})$ are the entropies of the fuzzy input feature $\tilde{X}_p^{jq}$ and $\tilde{X}_p^{uv}$ in the pth sliding window, respectively, and $H(\tilde{X}_p^{jq}, \tilde{X}_p^{uv})$ is the joint information entropy between $\tilde{X}_p^{jq}$ and $\tilde{X}_p^{uv}$.

4. IWFFS

4.1. IWFFS framework

To enhance the effectiveness of incremental fuzzy feature selection, reduce the complexity, and identify the evolving relationships for each fuzzy input feature, the proposed framework for the IWFFS is shown in Fig. 1. First, the fuzzy data are partitioned using sliding windows, where offline fuzzy feature selection is used in the first sliding window and online fuzzy feature selection is applied in each of the sliding windows successively. During the offline fuzzy feature selection process, the weight of each fuzzy input feature is calculated based on the mutual information entropy between the input and output features, and the fuzzy features subset is then obtained according to their weights. The backward feature selection [25] and fuzzy input features selection indices are used to obtain the candidate fuzzy features subset, where backward feature selection indicates that the procedure starts with the full set of fuzzy features.

Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021

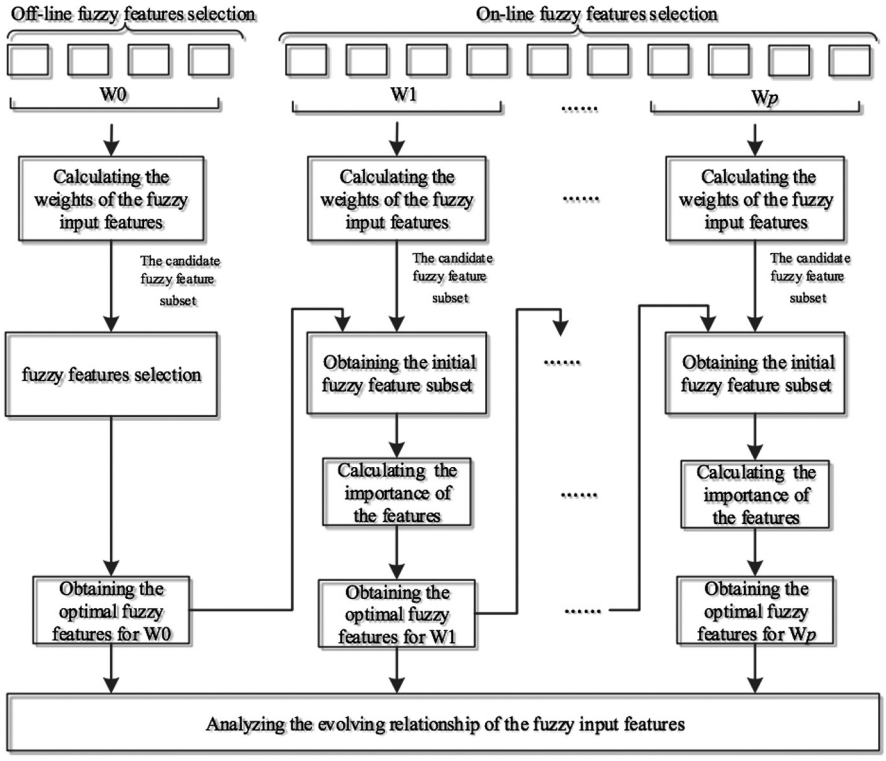


Fig. 1. Framework of the IWFFS.

In each step, the worst attribute remaining in the set is removed. The fuzzy feature with the lowest weight is then ignored to obtain the optimal fuzzy features subset. During the online fuzzy feature selection process, the candidate fuzzy features subset is obtained using the data in the current sliding window. The intersection of the optimal fuzzy features subset from the previous sliding window and the candidate fuzzy features subset in the current sliding window is treated as the initial optimal fuzzy features subset. The importance level is calculated between the initial optimal fuzzy features subset and the other input fuzzy features, and if the importance level is above 0, the input fuzzy feature is added to the optimal features subset. Finally, the evolving relationships among the input fuzzy features are analyzed according to their weights.

4.2. Offline fuzzy feature selection

The offline fuzzy features selection algorithm is summarized in Fig. 2. This algorithm is applied in the first sliding window and it comprises two main parts. In the first part, the weights of the fuzzy input features are calculated according to the mutual information shared between the fuzzy input features and the output feature, and the fuzzy features are then sorted in descending order of their weights to generate the candidate fuzzy features subset. In the second part, the fuzzy feature selection index is employed to select the fuzzy features, where the mutual information shared between the input features as well as the mutual information shared between the input and output features are all considered to obtain the global optimum fuzzy features subset according to the backward feature selection algorithm.

4.3. Online fuzzy feature selection algorithm

The online fuzzy feature selection algorithm is summarized in Fig. 3, which is applied in the subsequent sliding windows. First, the candidate fuzzy features set is generated using the data in the current sliding window according to the adjacent gradients of the weights. Next, the initial optimal fuzzy features subset in the current window is obtained according to the intersection between the optimal fuzzy features subset f_{p-1}^{s} in the $(p-1)$th sliding window and the

Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021

Proceduce:Extraction of the off-line fuzzy features selection

Input:Each sample dataset consists of m fuzzy features and a output feature:
$F\vec{X}_p^{in} = [\tilde{X}_p^{11}, \tilde{X}_p^{12}, ..., \tilde{X}_p^{1|C_p^1|}, \tilde{X}_p^{21}, \tilde{X}_p^{22}, ..., \tilde{X}_p^{2|C_p^2|}, ..., \tilde{X}_p^{jq}, ..., \tilde{X}_p^{L1}, \tilde{X}_p^{L2}, ..., \tilde{X}_p^{L|C_P^L|}]$
Output:The optimized fuzzy feature subset
$1 f_p^s = \varnothing, p = 0$
2 For each $\tilde{X}_p^{jq} \in \tilde{X}_p$
3 Caulculate the mutual information between the fuzzy input features and the output feature $I(\tilde{X}_p^{jq}, Y_p)$
4 End for
5 Calculate the weight of the fuzzy features,and sort the fuzzy input features in descending order according to the weight:
$\tilde{X}_p^{order} = \{\tilde{X}_p^{1,order}, \tilde{X}_p^{2,order}, ..., \tilde{X}_p^{d,order}, ..., \tilde{X}_p^{m,order}\}$
$6\ f_p^{cs} = \{\tilde{X}_p^{1,order}, \tilde{X}_p^{2,order}, \tilde{X}_p^{3,order}\}$
7 For each $\tilde{X}_p^{d,order} \in \tilde{X}_p^{order} - f_p^{cs}$
8 Calculate the dth weight $\triangle_p^d$ and choosing cut-off threshord ρ_p^d
9 If $\triangle_p^d > \rho_p^d (d \geq 3)$
$10\ f_{cs} = \{\tilde{X}_p^{1,order}, \tilde{X}_p^{2,order}, ..., \tilde{X}_p^{d,order}\}$
11 Break
12 End if
13 End for
14 Calculate $FCI(f_p^{cs})$
15 The label of the iteration:$label_{iter=1}$
16 While $(label_{iter})$
17 The label of updating:$label_{iter} = 1$
18 For $\tilde{X}_p^{d,order} \in f_p^{cs}$
$19 f_p^{cs}(X_p^{d,order}) = \{\tilde{X}_p^{1,order}, \tilde{X}_p^{2,order}, ..., \tilde{X}_p^{d-1,order}, \tilde{X}_p^{d+1,order}, ..., \tilde{X}_p^{|f_p^{cs}|,order}\}$
20 Calculate $FCI(f_p^{cs}(\overline{X_p^{d,order}}))$
21 If $FCI(f_p^{cs}(\overline{X_p^{d,order}})) > FCI(f_p^{cs})$
$22\ f_p^{cs} = f_p^{cs}(\overline{X_p^{d,order}})$
23 Update $FCI(f_p^{cs})$
$24\ label_{C_{fs}} = 1$
25 Break
26 End if
27 End for
28 if $label_{C_{fs}} = 0$
30 End if
31 End while
32 The optimized fuzzy feature subset$f_p^s = f_p^{cs}$

Fig. 2. Extraction of the offline fuzzy feature selections.

candidate fuzzy features set in the current sliding window. Finally, the optimal fuzzy features subset in the pth sliding window is obtained according to the importance levels of the features.

4.4. Computational complexity of the IWFFS algorithm

The time complexity of the IWFFS depends on the number of fuzzy features. We assume that m is the number of original fuzzy features. The IWFFS algorithm has two main phases comprising offline fuzzy feature selection and online fuzzy feature selection. For the offline fuzzy features selection algorithm, the cost of computing the mutual information is $O(m)$ and the complexity of calculating the weights of the fuzzy features based on their mutual information is $O(m)$. The fuzzy features are then sorted in descending order of their weights to calculate the gradients of the adjacent fuzzy weights $\Delta_p^d (d \geq 3)$. If Δ_p^d is higher than the cut-off threshold, then the d features sorted in descending order of their weights are treated as the candidate fuzzy features subset f_p^{cs}. In the worst case, the time complexity required is $O(m)$. In the best case, the time complexity required is $O(3)$, and thus the candidate fuzzy features subset f_p^{cs} is not updated until the fuzzy features selection index $FCI(f_p^{cs})$ does not increase. The worst case complexity for updating is $O(m-3)$. The total time complexity of the offline fuzzy features selection algorithm is $O(m+m+m+m)$, which is simplified as $O(m)$.

Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021

Procedure:Extraction of Extraction of the on-line fuzzy features selection

Input: The clustering centers $\mu_{p-1}^{q}(1 \le q \le |C_{p-1}|)$ in the $(p-1)$th sliding window, the dataset in the pth sliding window

Output: the optimized fuzzy feature subset f_p^q in the pth sliding window

1. $f_p^{cs} = \varnothing, p \ge 1$
2. For each $\tilde{X}_p^{jq} \in \tilde{X}_p$
3. Calculate the mutual information between the fuzzy input features and the output feature $I(\tilde{X}_p^{jq}, Y_p)$
4. End for
5. Calculate the weights of the fuzzy features, and sort the fuzzy input features in descending order according to the weight: $\tilde{X}_p^{order} = \{\tilde{X}_p^{1,order}, \tilde{X}_p^{2,order}, ..., \tilde{X}_p^{d,order}, ..., \tilde{X}_p^{m,order}\}$
6. $f_p^{cs} = \{\tilde{X}_p^{1,order}, \tilde{X}_p^{2,order}, \tilde{X}_p^{3,order}\}$
7. For each $\tilde{X}_p^{d,order} \in \tilde{X}_p^{order} - f_p^{cs}$
8. Calculate the dth weight $\triangle_p^d$ $(d \ge 3)$ and choosing the cut-off threshold ρ_p^d
9. If $\triangle_p^d > \rho_p^d$
10. $f_p^{cs} = \{\tilde{X}_p^{1,order}, \tilde{X}_p^{2,order}, ..., \tilde{X}_p^{d,order}\}$
11. Break
12. End if
13. End for
14. The initialization of the optimized fuzzy feature subset: $f_p^s = f_{p-1}^s \cap f_p^{cs}$
15. The label of the iteration: $label_{iter} = 1$
16. While($label_{iter}$)
17. Calculate the information entropy of the f_p^s
18. For each $\tilde{X}_p^{d,order} \in \tilde{X}_p^{order} - f_p^{cs}$
19. Calculate the information entropy of the $\tilde{X}_p^{d,order}$ and f_p^s
20. $SIG(\tilde{X}_p^{d,order}, f_p^s) = H(\tilde{X}_p^{d,order}, f_p^s) - H(f_p^s)$
21. End for
22. Choose the fuzzy feature with largest importance $H(\tilde{X}_p^{order}|f_p^s) = \max(SIG(\tilde{X}_p^{d,order}, f_p^s))$
23. If $H(\tilde{X}_p^{order}|f_p^s) > 0$
24. $f_p^s = f_p^s \cup \tilde{X}_p^{order}$
25. Else
26. $label_{iter} = 0$
27. End if
28. End while

Fig. 3. Extraction of the online fuzzy feature selections.

For the online fuzzy feature selection algorithm, the complexity is similar to that of the offline fuzzy feature selection for obtaining the candidate fuzzy feature subsets in the current window. The worst-case complexity is $O(m+m+m+m)$, where the optimal fuzzy feature subsets are obtained by calculating the importance levels of the fuzzy features retained in the candidate fuzzy feature subsets until the importance level is greater than 0. In the worst case, the time complexity for calculating the importance level is $O(m-3)$. The total time complexity of the online fuzzy feature selection is $O(m+m+m+m)$, the which is simplified as $O(m)$.

5. Experimental results and analysis

5.1. Experimental setup

Several experiments were conducted to test the proposed dynamic fuzzy features selection method. All of the data sets were downloaded from the UCI Machine Learning Repository [26]. The statistics for the data sets used in our experiments are shown in Table 1, which we employed to conduct all of our experiments and analysis. Table 1 shows that some of the data sets had continuous features but the classes were discrete, such as the classification data sets comprising SD, BCH, Glass, Liver, Wine, Yeast, SEA, Weather, and svmguide3. Others had continuous output features, such as the regression data sets comprising Airfoil Self-Noise, Concrete_Data, Energy efficiency, Red wine quality, White wine quality, Beijing PM2.5 data, and PPPTS. The sizes of the data sets varied from 214 to 50000 instances and the total number of original features in the data sets ranged up to 21.

Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021

Table 1
Descriptions of the data sets.

Data set	#No. of patterns	Dimensionality
SD	2000	2
BCH	1372	4
Glass	214	9
Liver	341	6
Wine	178	13
Yeast	1484	8
SEA	50000	3
Weather	18159	8
Symguide3	1243	21
Airfoil Self-Noise	1503	5
Concrete_Data	1030	8
Energy efficiency	768	8
Red wine quality	1599	11
White wine quality	4898	11
Beijing PM2.5 data	43800	6
PPPTS	45730	9

The performance of a classifier based on the feature selection results was measured using the classification accuracy (ACC) [27]:

$$\mathrm{ACC} = \frac{\sum_{k=1}^{|C|} a_k}{n}, \tag{26}$$

where a_k is the number of patterns with correct classification results corresponding to the correct class labels.

The performance of a regression model based on the feature selection results was measured using the mean absolute percentage error (MAPE) [28]:

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i - y_i|}{y_i}, \tag{27}$$

where y_i is the target value for the ith sample and $\hat{y}_i$ is the predicted value for the ith sample.

All of the experimental results were based on the averages over 20 runs. The hold-out test employed in this study was only used to estimate the performance based on the last sliding window, which is more appropriate for the concept drift situation under consideration and more efficient for computing the estimate.

5.2. *Effects of the sliding window size*

In order to evaluate the effects of different sliding window sizes on the IWFFS, the data sets were partitioned to form various sliding window, where 70 percent of the data were randomly selected for model training and the other 30 percent of the data were employed for testing the classification and prediction performance of the model. The classification based on multiple association rules (CMAR) [29] classifier and C4.5 classifier [23] were used to evaluate the classification performance based on the fuzzy features set. For CMAR, the support and confidence were set as 10% and 70%, respectively. A backpropagation (BP) neural network [30] and support vector regression (SVR) [31] were employed to evaluate the prediction performance based on the fuzzy features set. The hidden layer of the BP network had 10 nodes. The kernel function used for SVR was the radial basis function, where ε was 0.05 and C was 10. To ensure fair comparisons, all of the algorithms used the same experimental settings. In addition, all of the experiments were conducted over 20 times, where each used a random permutation of the data set.

As shown in Tables 2 and 3, the classification and prediction performance of the IWFFS method were better than that with a predefined window size because our proposed method can adaptively determine the window size according to the Hoeffding boundary, which improved the classification and regression prediction performance using the fuzzy data.

Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021

Table 2
Cumulative classification accuracy (ACC) with the 95% confidence interval for CMAR/C4.5 based on different window sizes.

Data set	Window size		
	5% (Percentage of data in the window)	10% (Percentage of data in the window)	Proposed method
SD	0.875±0.005/0.881±0.005	0.864±0.007/0.875±0.007	0.906±0.005/0.912±0.005
BCH	0.883±0.022/0.896±0.022	0.874±0.027/0.896±0.027	0.947±0.030/0.958±0.030
Glass	0.904±0.007/0.912±0.007	0.891±0.013/0.901±0.013	0.962±0.020/0.967±0.022
Liver	0.892±0.009/0.881±0.009	0.874±0.012/0.869±0.012	0.952±0.015/0.948±0.015
Wine	0.903±0.004/0.921±0.004	0.886±0.006/0.897±0.009	0.957±0.008/0.969±0.008
Yeast	0.911±0.011/0.916±0.012	0.895±0.014/0.903±0.015	0.960±0.015/0.974±0.015
SEA	0.921±0.021/0.934±0.018	0.904±0.018/0.912±0.019	0.983±0.022/0.989±0.023
Weather	0.878±0.036/0.896±0.039	0.858±0.035/0.869±0.036	0.948±0.036/0.957±0.036
svmguide3	0.918±0.033/0.926±0.034	0.920±0.034/0.917±0.036	0.934±0.033/0.941±0.033

Table 3
Cumulative prediction accuracy (MAPE) with the 95% confidence interval for BP/SVR based on different window sizes.

Data set	Window size		
	5% (Percentage of data in the window)	10% (Percentage of data in the window)	Proposed method
Airfoil Self-Noise	0.190±0.030/0.185±0.042	0.185±0.031/0.176±0.032	0.141±0.033/0.042±0.033
Concrete_Data	0.149±0.149/0.133±0.023	0.147±0.025/0.131±0.020	0.116±0.022/0.033±0.022
Energy efficiency	0.224±0.026/0.216±0.016	0.216±0.015/0.244±0.018	0.181±0.015/0.057±0.012
Red wine quality	0.174±0.023/0.161±0.020	0.168±0.020/0.159±0.019	0.124±0.013/0.052±0.013
White wine quality	0.153±0.017/0.142±0.015	0.144±0.013/0.136±0.015	0.104±0.011/0.031±0.012
Beijing PM2.5 data	0.254±0.057/0.243±0.055	0.241±0.052/0.226±0.043	0.173±0.046/0.026±0.052
PPPTS	0.254±0.052/0.249±0.031	0.236±0.053/0.219±0.054	0.194±0.045/0.057±0.034

5.3. Performance evaluation

In order to evaluate the performance of the IWFFS, we used four other feature selection algorithms comprising WFSIBC, IFNRS, OMVFS, and FSDS to compare the features obtained using the two classifiers (CMAR and C4.5) and two regression model (BP and SVR). As shown in Fig. 4, except for the SD data set, the number of features obtained using the IWFFS algorithm was much lower than the number of original features but also less than those obtained by the other feature selection algorithms. This is because the IWFFS algorithm considers the mutual information shared between the input fuzzy features and the output feature as well as the mutual information shared between the input fuzzy features to obtain the optimal fuzzy features subset.

Based on the features obtained, IWFFS was compared with WFSIBC and IFNRS according to the classification performance with the CMAR and C4.5 classifiers. Similarly, we compared IWFFS with OMVFS and FSDS according to the regression prediction performance with the BP neural network and SVR. As shown in Tables 4 and 5, except for the SD data set, the fuzzy features subsets obtained using IWFFS exhibited the best classification performance with the CMAR classifier, but also the best regression prediction performance with SVR. All of the experiments were conducted based on 20 random permutations and the results were obtained by averaging over all 20 runs.

In addition, the run times (s) for different algorithms are shown in Fig. 5. As expected, the time costs were less for the IWFFS algorithm than those for the other four feature selection algorithms because the weight gradients were employed by the IWFFS algorithm to reduce the search space for the optimal fuzzy features subset. Among all of the feature selection algorithms, the IWFFS algorithm had the longest run time for the SD data set but the shortest run time for the Wine data set. Thus, we conclude that the run time of the IWFFS algorithm is longer when the number of original fuzzy features is higher.

Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021

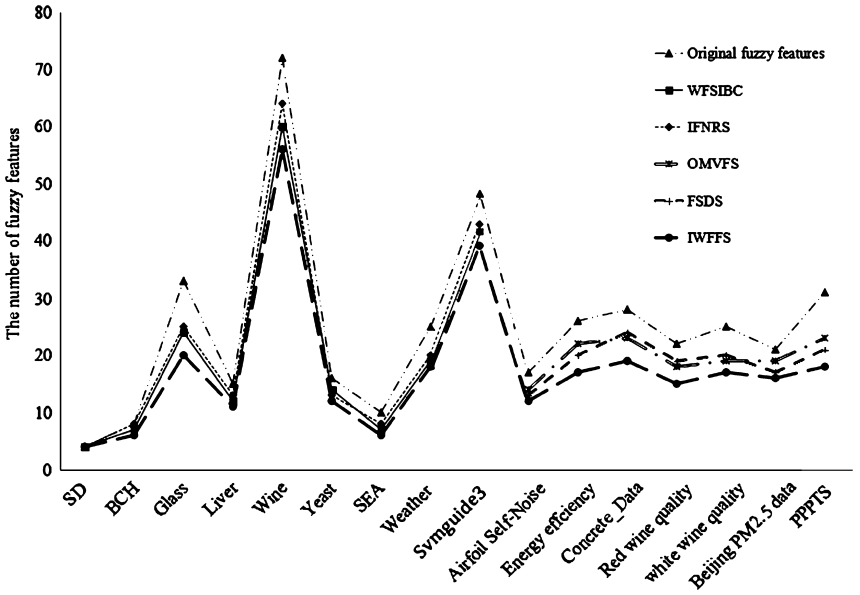


Fig. 4. Comparison of the fuzzy features obtained by different algorithms based on different data sets.

Table 4
Cumulative classification accuracy (ACC) with the 95% confidence interval for different algorithms.

Data set	Original features (CMAR/C4.5)	WFSIBC (CMAR/C4.5)	IFNRS (CMAR/C4.5)	IWFFS (CMAR/C4.5)
SD	0.906±0.005/0.912±0.005	0.906±0.005/0.912±0.005	0.906±0.054/0.912±0.005	0.906±0.005/0.912±0.005
BCH	0.921±0.022/0.925±0.048	0.940±0.037/0.945±0.039	0.936±0.035/0.938±0.037	0.947±0.030/0.958±0.030
Glass	0.932±0.027/0.906±0.026	0.951±0.026/0.9581±0.028	0.945±0.027/0.949±0.024	0.962±0.015/0.967±0.022
Liver	0.916±0.020/0.896±0.020	0.932±0.020/0.933±0.020	0.928±0.026/0.931±0.025	0.952±0.015/0.948±0.015
Wine	0.925±0.016/0.913±0.019	0.939±0.020/0.936±0.020	0.938±0.013/0.932±0.014	0.957±0.008/0.969±0.008
Yeast	0.915±0.023/0.924±0.024	0.943±0.020/0.946±0.020	0.928±0.018/0.935±0.018	0.960±0.015/0.974±0.015
SEA	0.923±0.025/0.909±0.025	0.956±0.020/0.961±0.020	0.944±0.026/0.925±0.023	0.983±0.022/0.989±0.023
Weather	0.919±0.037/0.918±0.039	0.945±0.030/0.943±0.030	0.936±0.038/0.928±0.034	0.948±0.036/0.957±0.036
svmguide3	0.934±0.033/0.941±0.033	0.942±0.032/0.944±0.034	0.957±0.035/0.965±0.036	0.975±0.033/0.982±0.033

Table 5
Cumulative prediction accuracy (MAPE) with the 95% confidence interval for different algorithms.

Data set	Original (BP/SVR)	OMVFS (BP/SVR)	FSDS (BP/SVR)	IWFFS (BP/SVR)
Airfoil Self-Noise	0.248±0.039/0.075±0.035	0.183±0.037/0.055±0.026	0.209±0.038/0.062±0.034	0.141±0.033/0.042±0.033
Concrete_Data	0.224±0.027/0.094±0.026	0.194±0.023/0.042±0.025	0.217±0.022/0.051±0.019	0.116±0.022/0.033±0.022
Energy efficiency	0.235±0.023/0.096±0.025	0.204±0.022/0.065±0.017	0.218±0.020/0.072±0.020	0.181±0.015/0.057±0.012
Red wine quality	0.232±0.020/0.104±0.027	0.181±0.022/0.067±0.028	0.205±0.022/0.069±0.019	0.124±0.013/0.052±0.013
White wine quality	0.206±0.020/0.087±0.020	0.192±0.024/0.064±0.018	0.186±0.022/0.068±0.029	0.104±0.011/0.031±0.012
Beijing PM2.5 data	0.311±0.057/0.076±0.055	0.254±0.055/0.054±0.057	0.264±0.053/0.065±0.029	0.173±0.046/0.026±0.052
PPPTS	0.341±0.047/0.106±0.048	0.272±0.046/0.074±0.034	0.295±0.052/0.086±0.039	0.194±0.045/0.057±0.034

5.4. Analysis of variable weights

We analyzed the effects of the variable weights for different features on the classifier. The SEA data set was partitioned using five windows where each window contained 10000 samples. As shown in Fig. 6, the weights of the fuzzy features $\tilde{X}_{11}$, $\tilde{X}_{22}$, and $\tilde{X}_{33}$ decreased continuously. In the 4th sliding window, the weights of the three fuzzy features were lowest and they had almost no effect on the output features, but the weights of fuzzy features $\tilde{X}_{13}$, $\tilde{X}_{21}$,

Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021

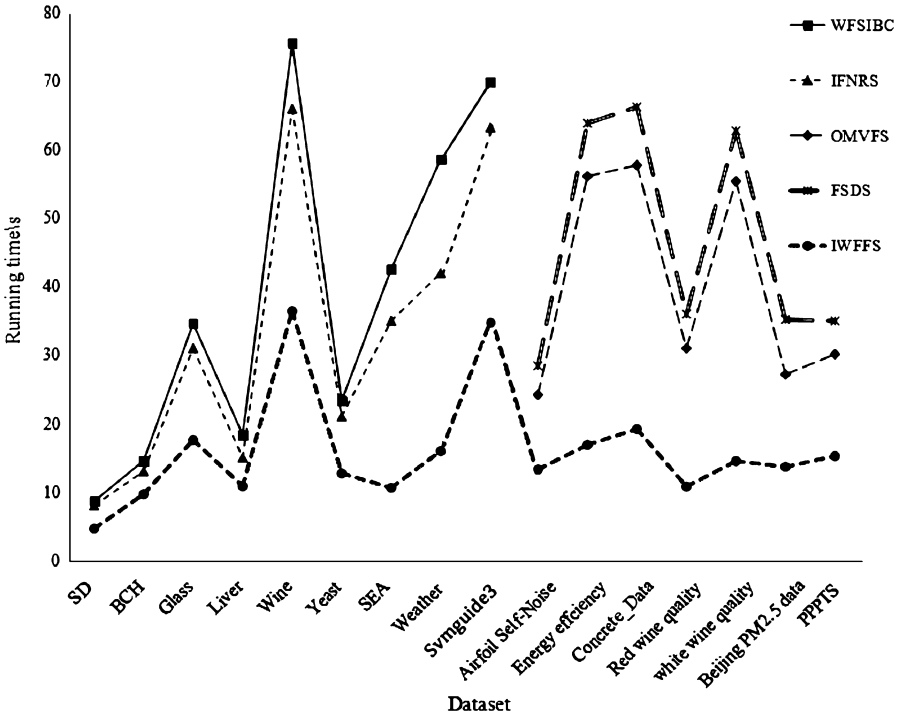


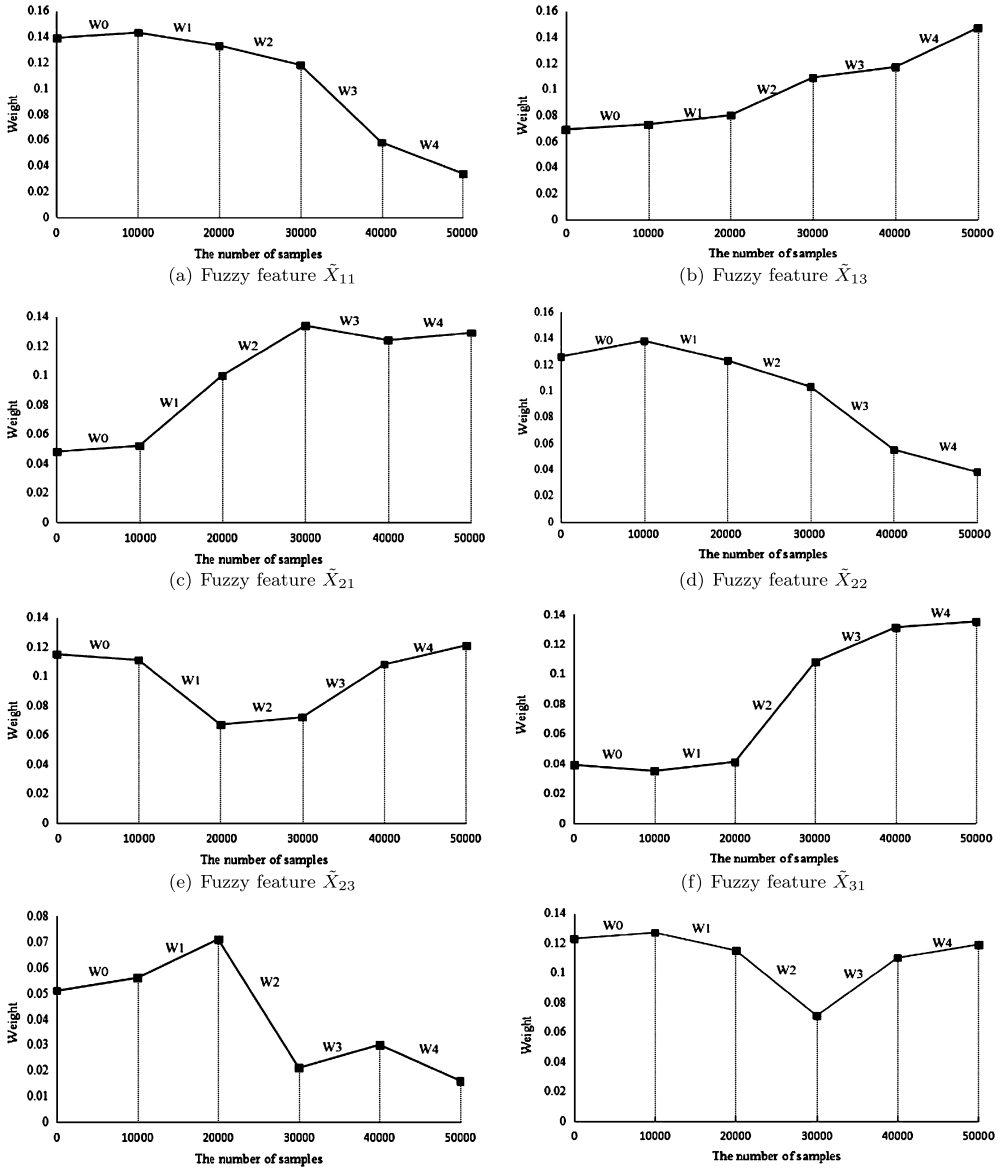
Fig. 5. Comparison of the run times with different algorithms.

and $\tilde{X}_{31}$ then increased continuously to reach their maximum values. For all of the windows, the weights of fuzzy feature $\tilde{X}_{23}$, and $\tilde{X}_{33}$ decreased initially and then increased gradually. In the 0th and 4th sliding windows, the weights for fuzzy features $\tilde{X}_{23}$ and $\tilde{X}_{34}$ changed little.

According to the order of $\tilde{X}_{11}$, $\tilde{X}_{13}$, $\tilde{X}_{21}$, $\tilde{X}_{22}$, $\tilde{X}_{23}$, $\tilde{X}_{31}$, $\tilde{X}_{33}$, and $\tilde{X}_{34}$, the fuzzy features were ignored in sequence from the initial fuzzy features set, before the classification performance was compared when each feature was removed and after the feature was restored, as shown in Fig. 7. Fig. 6 shows that the weights of fuzzy features $\tilde{X}_{11}$, $\tilde{X}_{22}$, and $\tilde{X}_{33}$ decreased continuously, and these features had increasingly weaker effects on the ACC. Therefore, in the 4th sliding window, the ACC with the feature removed was similar to that with the feature restored. By contrast, the weights of fuzzy features $\tilde{X}_{13}$, $\tilde{X}_{21}$, and $\tilde{X}_{31}$ increased continuously, so their effects on the ACC became increasingly important. In the 4th sliding window, the difference in ACC when the feature was removed and when the feature was restored was the largest according to Fig. 7. In addition, the weights of fuzzy features $\tilde{X}_{23}$ and $\tilde{X}_{34}$ decreased initially and then increased gradually, so their effect son the ACC increased at the beginning before decreasing gradually.

We also analyzed the effects of the variable weights of different features on the regression model. The PPPTS data set was partitioned using eight windows where each window contained 6000 samples. The weights of some of the fuzzy features in the PPPTS data set are shown in Fig. 8. The weights of fuzzy input features $\tilde{X}_{12}$, $\tilde{X}_{23}$, $\tilde{X}_{61}$, $\tilde{X}_{83}$, and $\tilde{X}_{91}$ tended to decrease, whereas the weights of fuzzy input features $\tilde{X}_{31}$, $\tilde{X}_{53}$, $\tilde{X}_{72}$, and $\tilde{X}_{74}$ tended to increase, while the weight of fuzzy input feature $\tilde{X}_{42}$ increased initially and then declined gradually. These fuzzy features were then each ignored among the initial fuzzy input features. The prediction performance was compared with each feature removed and when the feature was restored, as shown in Fig. 9. The weights of fuzzy features $\tilde{X}_{12}$, $\tilde{X}_{23}$, $\tilde{X}_{61}$, $\tilde{X}_{83}$, and $\tilde{X}_{91}$ decreased continuously, so these features had increasingly weaker effects on the prediction accuracy. In the 7th sliding window, the prediction accuracy when each feature was removed was similar to that when the feature was restored. By contrast, fuzzy input features $\tilde{X}_{31}$, $\tilde{X}_{53}$, $\tilde{X}_{72}$, and $\tilde{X}_{74}$ had increasingly important effects on the prediction accuracy because their weights increased so the difference in the prediction accuracy was larger when each feature was removed and restored. The weight of fuzzy input feature $\tilde{X}_{42}$ increased initially and then decreased gradually, so its effect on the prediction accuracy decreased at the beginning and then increased gradually.

Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021



(a) Fuzzy feature $\tilde{X}_{11}$ (b) Fuzzy feature $\tilde{X}_{13}$

(c) Fuzzy feature $\tilde{X}_{21}$ (d) Fuzzy feature $\tilde{X}_{22}$

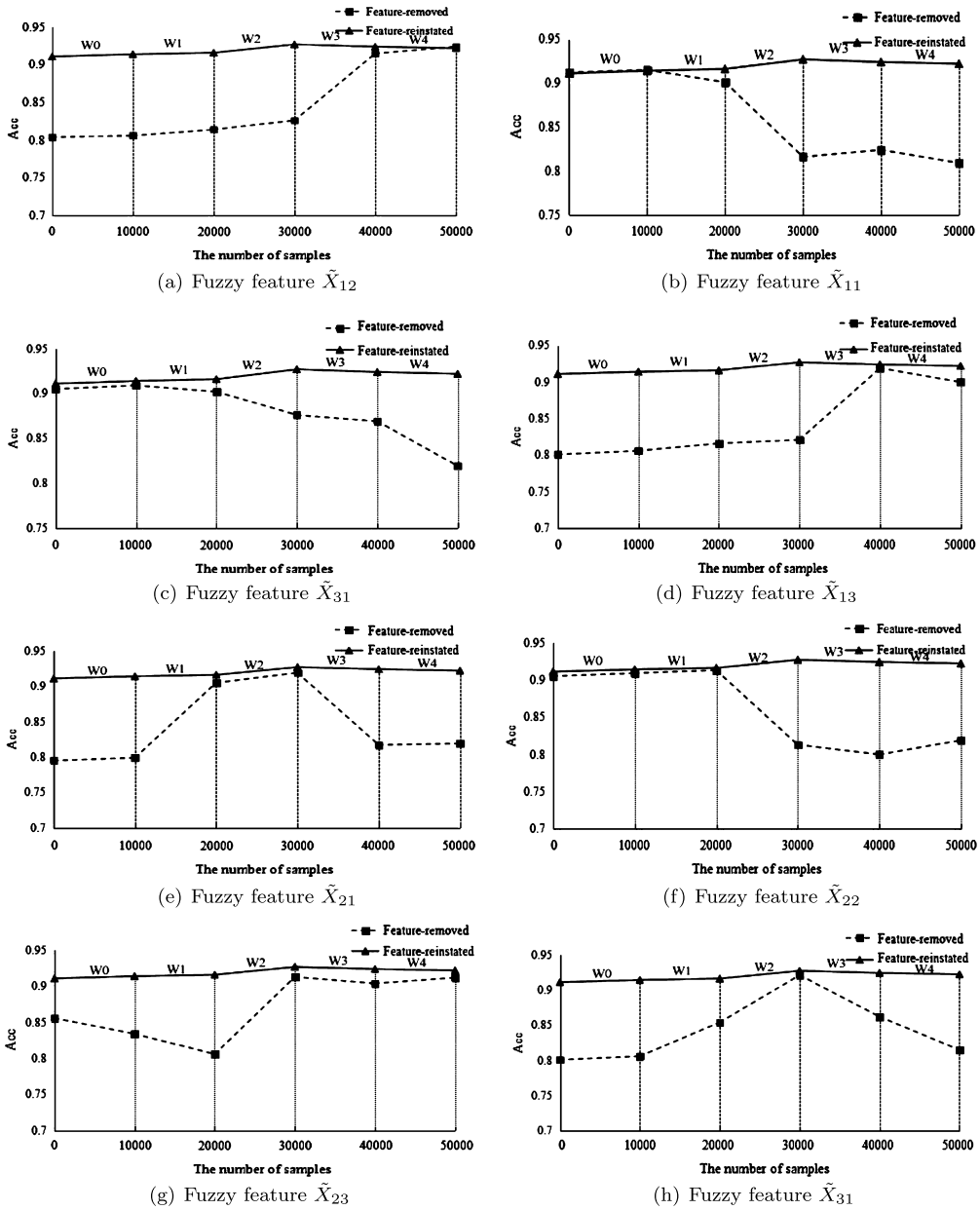
(e) Fuzzy feature $\tilde{X}_{23}$ (f) Fuzzy feature $\tilde{X}_{31}$

(g) Fuzzy feature $\tilde{X}_{33}$ (h) Fuzzy feature $\tilde{X}_{34}$

Fig. 6. Variable weights for some fuzzy features in the SEA data set.

Using the IWFFS algorithm, the optimal fuzzy features subset is obtained without requiring any parameters. The dependencies between the features are calculated using the mutual information irrespective of whether they are linear or nonlinear relationships, and thus the classification and prediction accuracy are improved during feature selection. In the offline fuzzy features selection algorithm, the weights of the fuzzy features are considered to obtain the candidate fuzzy features subset. The backward feature selection algorithm is employed to obtain the optimal fuzzy features subset, which reduces the run time and increases the efficiency. In the online fuzzy features selection algorithm, the intersection between the candidate fuzzy features subset in the current window and previous feature selection results is treated as the initial optimal fuzzy features subset for the current window, and the final optimal fuzzy features subset is obtained based on the importance levels of the fuzzy input features in the initial optimal fuzzy features subset. The evolving trends in the fuzzy input features are analyzed between different windows to identify the effects of the fuzzy features on the classification and regression prediction accuracy.

Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021



(a) Fuzzy feature $\tilde{X}_{12}$ (b) Fuzzy feature $\tilde{X}_{11}$

(c) Fuzzy feature $\tilde{X}_{31}$ (d) Fuzzy feature $\tilde{X}_{13}$

(e) Fuzzy feature $\tilde{X}_{21}$ (f) Fuzzy feature $\tilde{X}_{22}$

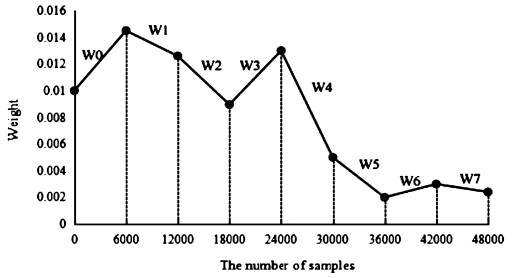
(g) Fuzzy feature $\tilde{X}_{23}$ (h) Fuzzy feature $\tilde{X}_{31}$

Fig. 7. Effects on the classification accuracy of ignoring some fuzzy features in the SEA data set.

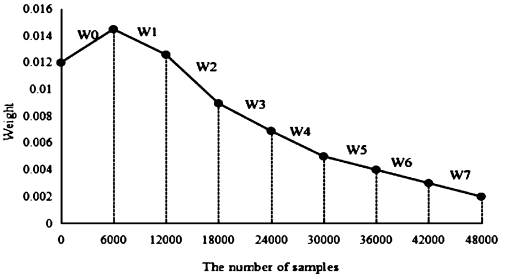
6. Conclusion

In this study, we proposed the IWFFS algorithm, which mainly evaluates the importance levels of the fuzzy features according to the weights of the fuzzy features in the current window. In the offline fuzzy features selection stage, the optimal fuzzy input features subset is obtained by the backward features selection algorithm and fuzzy features selection index. In the online fuzzy features selection stage, based on the candidate fuzzy features subset in the current sliding window and the optimal fuzzy feature subset in the previous sliding window, the optimal fuzzy features subset in the current sliding window is obtained according to the importance levels of the fuzzy features. The evolving trends in the fuzzy input features are analyzed between sliding windows to identify the effects of the fuzzy features on the ACC and prediction accuracy. Our simulation results showed that the proposed algorithm significantly improved the adaptability and prediction accuracy compared with existing methods.

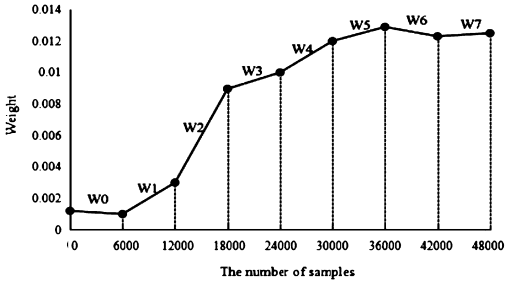
Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021



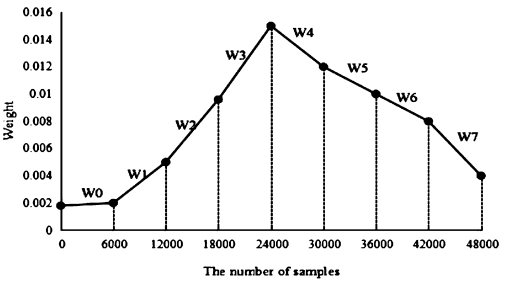
(a) Fuzzy feature $\tilde{X}_{12}$



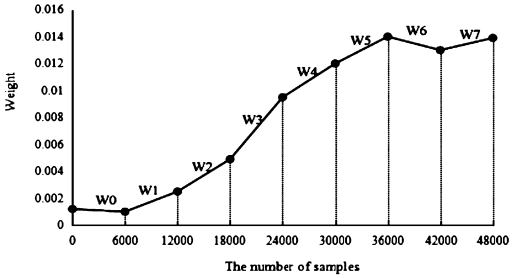
(b) Fuzzy feature $\tilde{X}_{23}$



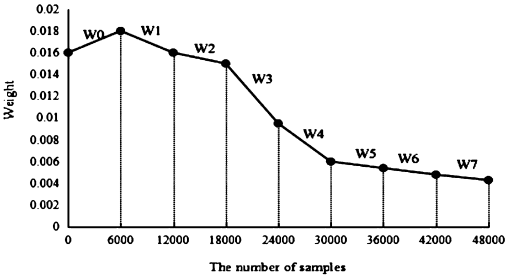
(c) Fuzzy feature $\tilde{X}_{31}$



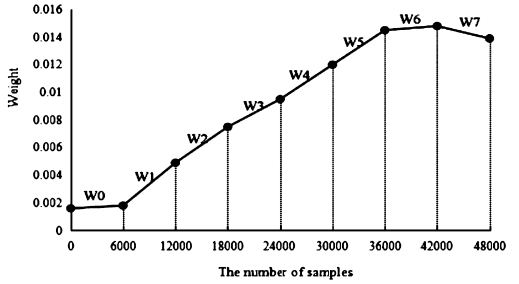
(d) Fuzzy feature $\tilde{X}_{42}$



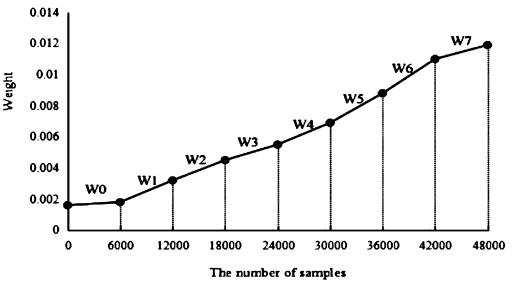
(e) Fuzzy feature $\tilde{X}_{53}$



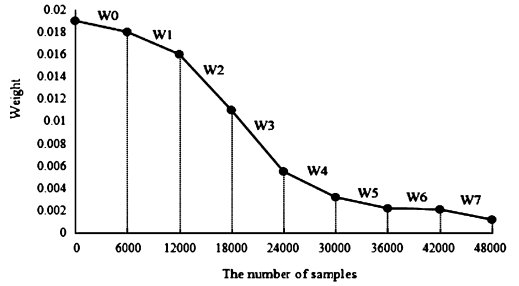
(f) Fuzzy feature $\tilde{X}_{61}$



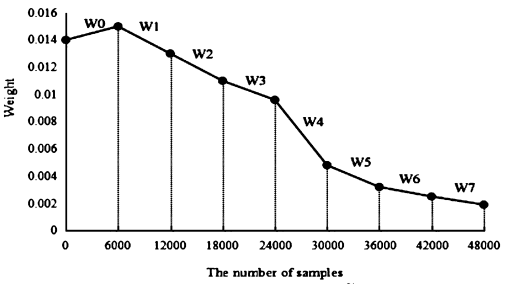
(g) Fuzzy feature $\tilde{X}_{72}$



(h) Fuzzy feature $\tilde{X}_{74}$



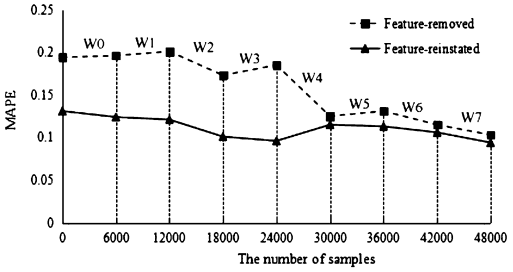
(i) Fuzzy feature $\tilde{X}_{83}$



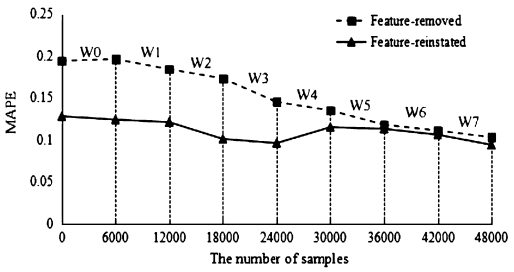
(j) Fuzzy feature $\tilde{X}_{91}$

Fig. 8. Effects on the prediction accuracy of ignoring some fuzzy features in the PPPTS data set.

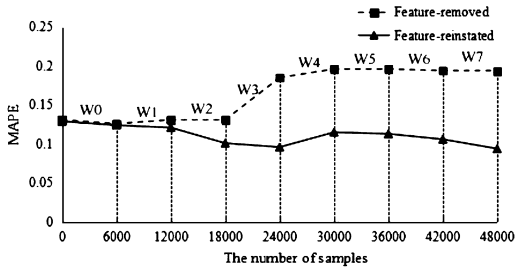
Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021



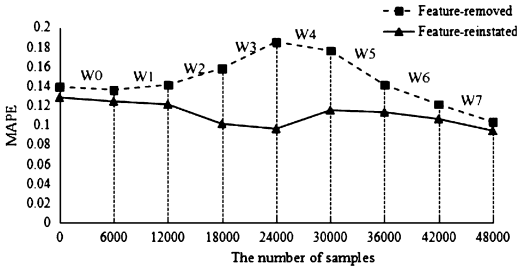
(a) Fuzzy feature $\tilde{X}_{12}$



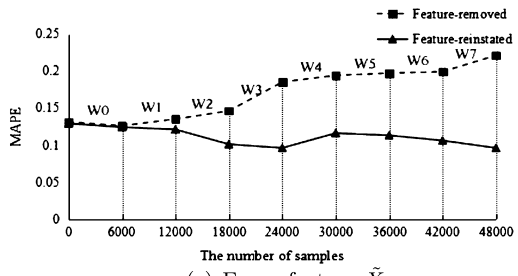
(b) Fuzzy feature $\tilde{X}_{23}$



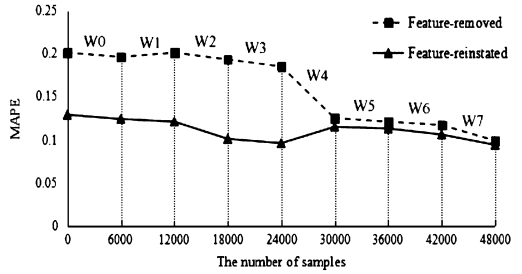
(c) Fuzzy feature $\tilde{X}_{31}$



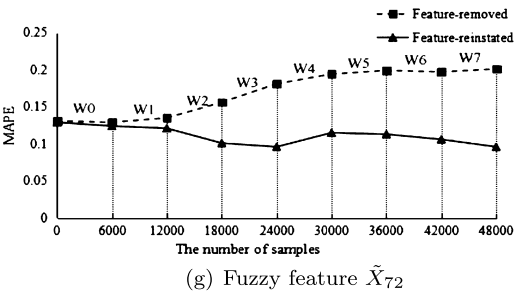
(d) Fuzzy feature $\tilde{X}_{42}$



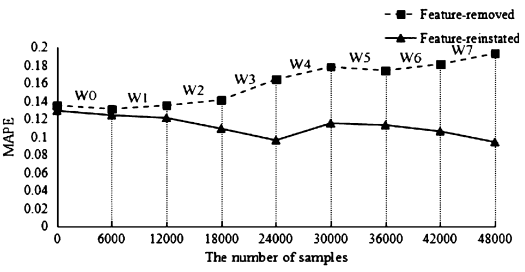
(e) Fuzzy feature $\tilde{X}_{53}$



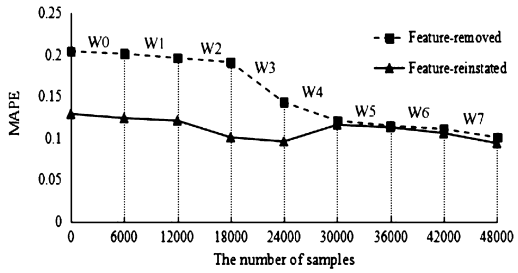
(f) Fuzzy feature $\tilde{X}_{61}$



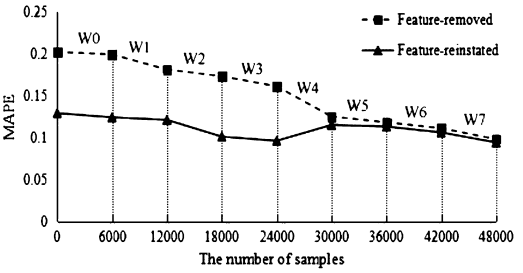
(g) Fuzzy feature $\tilde{X}_{72}$



(h) Fuzzy feature $\tilde{X}_{74}$



(i) Fuzzy feature X_{83}



(j) Fuzzy feature X_{91}

Fig. 9. Effects on the prediction accuracy of ignoring some fuzzy features in the PPPTS data set.

Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021

The main contribution of this study is our demonstration that fuzzy features can be selected from dynamic data to make more useful decisions. However, some problems still need to be addressed. First, the size of the sliding window that varies with the distribution of the data must be investigated. Second, we will apply our proposed algorithm to other domains to solve more complex problems.

Acknowledgements

This study was supported by the National Natural Science Foundation of China (Grant No. 61572073), National Key R&D Program of China (No. 2017YFB0306403).

References

[1] J. Gama, Knowledge discovery from data streams, Intell. Data Anal. 13 (3) (2009) 403–404.
[2] P. Angelov, D. Filev, N. Kasabov, Evolving Intelligent Systems Methodology and Applications, Wiley-IEEE Press, 2010.
[3] M. Sayed-Mouchaweh, E. Lughofer, Learning in Non-Stationary Environments: Methods and Applications, Springer, New York, 2012.
[4] A. Ghazikhani, R. Monsefi, H.S. Yazdi, Online neural network model for non-stationary and imbalanced data stream classification, Int. J. Mach. Learn. Cybern. 5 (1) (2014) 51–62.
[5] L. Bellatreche, A. Kerkad, Query interaction based approach for horizontal data partitioning, Int. J. Data Warehous. Min. 11 (2) (2015) 44–61.
[6] J. Wang, P. Zhao, S.C.H. Hoi, et al., Online feature selection and its applications, IEEE Trans. Knowl. Data Eng. 26 (3) (2006) 698–710.
[7] Y. Li, X. Gu, Feature selection for transient stability assessment based on improved maximal relevance and minimal redundancy criterion, Proc. CSEE 33 (34) (2013) 179–186.
[8] I.S. Bilal, P.D. Keshav, M.H. Alamgir, et al., Diversification of fuzzy association rules to improve prediction accuracy, in: Fuzzy Systems (Fuzz) in IEEE Explorer, 2010, pp. 1–8.
[9] Y. Jing, T. Li, C. Luo, et al., An incremental approach for attribute reduction based on knowledge granularity, J. Shandong Univ. 104 (2016) 24–38.
[10] S.S. Naqvi, W.N. Browne, C. Hollitt, Feature quality-based dynamic feature selection for improving salient object detection, IEEE Trans. Image Process. 25 (9) (2016) 4298–4313.
[11] W. Zhao, Y. Wang, D. Li, A dynamic feature selection method based on combination of GA with K-means, in: International Conference on Industrial Mechatronics and Automation, IEEE, 2010, pp. 271–274.
[12] W. Shao, L. He, C.T. Lu, et al., Online unsupervised multi-view feature selection, in: Data Mining (ICDM), 2016 IEEE 16th International Conference on, IEEE, 2016, pp. 1203–1208.
[13] H. Huang, S. Yoo, S.P. Kasiviswanathan, Unsupervised feature selection on data streams, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, ACM, 2015, pp. 1031–1040.
[14] Y. Li, Z.F. Wu, Fuzzy feature selection based on min–max learning rule and extension matrix, Pattern Recognit. 41 (1) (2008) 217–226.
[15] S. Alizadeh, A. Kalhor, H. Jamalabadi, et al., Online local input selection through evolving heterogeneous fuzzy inference system, IEEE Trans. Fuzzy Syst. 24 (6) (2016) 1364–1377.
[16] E. Lughofer, On-line incremental feature weighting in evolving fuzzy classifiers, Fuzzy Sets Syst. 163 (1) (2011) 1–23.
[17] M. Pratama, S.G. Anavatti, E. Lughofer, An incremental classifier from data streams, in: SETN, 2014, pp. 15–28.
[18] A.M. Silva, W. Caminhas, A. Lemos, et al., Adaptive input selection and evolving neural fuzzy networks modeling, Int. J. Comput. Intell. Syst. 8 (Supl. 1) (2015) 3–14.
[19] Ling Wang, Lulu Wu, Adaptive learning by using a new evolving clustering method, J. Comput. Inf. Syst. 21 (10) (2014) 9461–9468.
[20] L. Du, Q. Song, X. Jia, Detecting concept drift: an information entropy based method using an adaptive sliding window, Intell. Data Anal. 18 (3) (2014) 337–364.
[21] E. Cohen, M. Strauss, Maintaining time-decaying stream aggregates, in: Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, ACM, 2003, pp. 223–233.
[22] R.B. Kirkby, Improving Hoeffding Trees, The University of Waikato, 2007.
[23] Y. Mu, X. Liu, Z. Yang, et al., A parallel C4.5 decision tree algorithm based on MapReduce, Concurr. Comput., Pract. Exp. 29 (8) (2017).
[24] Z. Zhao, Y. Li, J. Yu, Optimal assembly tolerance design based on Fuzzy information entropy and seeker optimization algorithm, in: 3rd International Conference on Advanced Computer Theory and Engineering, vol. 5, 2010, pp. 610–613.
[25] W. Siedlecki, J. Sklansky, On automatic feature selection, Int. J. Pattern Recognit. Artif. Intell. 2 (1988) 197–220.
[26] A. Asuncion, D. Newman, UCI machine learning repository [DB/OL], [2017-6-20].
[27] Y. Zhang, B. Liu, X. Ji, et al., Classification of EEG signals based on autoregressive model and wavelet packet decomposition, Neural Process. Lett. 45 (2) (2017) 365–378.
[28] H. Wu, Y. Cai, Y. Wu, et al., Time series analysis of weekly influenza-like illness rate using a one-year period of factors in random forest regression, BioSci. Trends (2017), https://doi.org/10.5582/bst.2017.01035.
[29] W. Li, J. Han, J. Pei, CMAR: accurate and efficient classification based on multiple class-association rules, in: IEEE International Conference on Data Mining, IEEE, 2001, pp. 369–376.
[30] Y. Cui, X.L. Ma, Z. Liu, Application of improved BP neural network with correlation rules in network intrusion detection, Int. J. Secur. Appl. 10 (4) (2016) 423–430.
[31] L. Wang, Y. Guo, G. Xin, An asymmetric weighted SVR for construction final accounting prediction, J. Inf. Comput. Sci. 11 (5) (2014) 1387–1394.

Please cite this article in press as: L. Wang et al., Incremental feature weighting for fuzzy feature selection, Fuzzy Sets Syst. (2018), https://doi.org/10.1016/j.fss.2018.10.021