Neurocomputing 291 (2018) 226–236

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Learning from small datasets containing nominal attributes

Der-Chiang Li*, Hung-Yu Chen, Qi-Shi Shi

Department of Industrial and Information Management, National Cheng Kung University, No. 1, University Road, Tainan City 70101, Taiwan (R.O.C.)

ARTICLE INFO

Article history:
Received 30 June 2017
Revised 28 December 2017
Accepted 15 February 2018
Available online 27 February 2018

Communicated by Dr Xin Luo

Keyword:
Small data
Nominal input
Continuous output
Virtual sample

ABSTRACT

In many small-data-learning problems, owing to the incomplete data structure, explicit information for decision makers is limited. Although machine learning algorithms are extensively applied to extract knowledge, most of them are developed without considering whether the training sets can fully represent the population properties. Focusing on small data which contains nominal inputs and continuous outputs, this paper develops an effective sample generating procedure based on fuzzy theories to tackle the learning issue by data preprocessing. According to the derived fuzzy relations between categories and continuous outputs, the possibilities of the combinations of categories (virtual samples) can be aggregated when continuous outputs are given. Proper virtual samples are further selected by using fuzzy alpha-cut on the possibility distributions, and these are added to the training sets to form new ones. In the experiment, sixteen datasets taken from the UC Irvine Machine Learning Repository are examined with back-propagation neural networks and support vector regressions. The results reveal that the forecasting accuracies of the two models are significantly improved when they are built with the proposed new training sets. Moreover, the results also indicate the proposed method outperforms bootstrap aggregating and the synthetic minority over-sampling technique-Nominal-Continuous with the greatest amount of statistical support.

1. Introduction

The issues of learning with small data have been known about for over one hundred years, since they were first revealed in the *t*-distribution [1] in 1908. These problems have not yet been overcome because they are still being discovered as many new learning domains are developed [2–5]. Although machine learning algorithms have been extensively applied to extract knowledge from data over the past few decades, most of them are developed based on the assumption that each value has enough representatives in the training sets [6]. Once the population information contained in training sets is insufficient, the knowledge extracted by algorithms is thus limited or less precise [7].

Gathering more samples until algorithms can carry out sufficient learning could be an intuitive approach to the issue of small-data-learning problems. However, achieving this is sometimes difficult and/or costly in certain domains, such as patient profiles for rare diseases [2,3], image pixels in pattern recognition [4,5], and minority class instances in imbalanced data [8,9]. Therefore, the better utilization of on hand data in order to obtain more precise population information is worthy of further research [7].

Previous studies [10,11] have examined training sizes in order to build more meaningful learning models. In fact, algorithms can still sometimes make precise models with a few samples, though the information they contain may be limited. To further explain this, Fig. 1 displays two possible distributions of two small datasets relative to their populations. In Fig. 1(a), the samples are evenly distributed in a population, and most learning algorithms can thus build precise models. Conversely, in Fig. 1(b), the samples are concentrated in a part of the population, and all algorithms thus make biased models regardless of sample sizes.

Another issue that makes it difficult for algorithms to learn more information from small data is the gaps between pairs of two close samples, as shown in Fig. 2. The gaps (referred to as information gaps) should be filled with samples in a complete dataset; however, these samples are not available in a small dataset. The unavailable samples may contain important properties, such as borderlines for tree-based algorithms to correctly identify sample class values. In addition, since tree-based algorithms (e.g. C4.5 [12]) need to partition continuous data into discrete intervals with an equal width before evaluating the classification purity, the expected size of intervals is usually unavailable in small data, since some intervals that contain no samples are integrated with their nearest intervals. If an insufficient number of candidate positions exist for the purity evaluation, then the trees that are built and the resulting hierarchy of the classification rules will be small.

* Corresponding author.
E-mail address: lidc@mail.ncku.edu.tw (D.-C. Li).

https://doi.org/10.1016/j.neucom.2018.02.069
0925-2312/

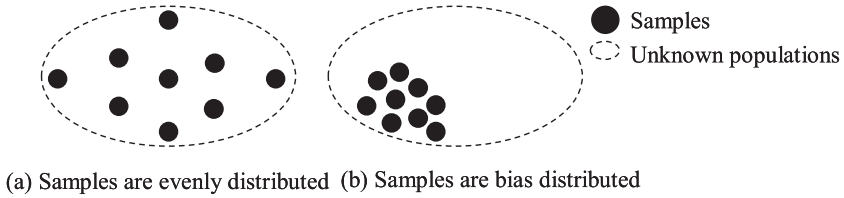

Fig. 1. Two situations in which small data may be distributed relative to the populations.

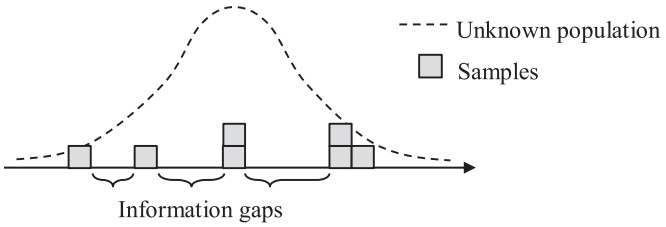

Fig. 2. Information gaps between pairs of two close samples in a small dataset.

In the past few decades there have been numerous approaches developed to tackle the issue of learning with insufficient samples [13–17]. Among these approaches, the virtual sample generation (VSG) methods are considered as effective based on their use of data preprocessing [17]. The most well-known VSG method is the bootstrap procedure (BP) [18], which draws samples from original datasets with replacement to form bootstrapping sets. Algorithms can thus train a sample at least twice to gradually revise the identified patterns, which enables them to represent the behaviors of the actual data. Based on BP, ensemble learning methods, such as bootstrap aggregating (bagging) [19], were developed to reduce the influence of the overfitting issue in training sets. Moreover, bagging is also considered to be an effective approach to handle small datasets [20].

Some VSG approaches have also been proposed to create samples for the minority class when working with imbalanced data [8,9]. Among these, the Synthetic Minority Over-Sampling Technique-Nominal (SMOTE-N) [8], developed based on the *k*-nearest neighbor (K-NN) algorithm, may be the most extensively applied method. In SMOTE-N, the numerical values of synthetic data are created along continuous vectors between the minority class's instances and their NNs, while the category values of synthetic data are determined by the most frequently appearing ones in the NNs. It is reasonable that SMOTE-N adopts this way to determine synthetic category values, since it employs the Value Difference Metric (VDM) [21] to pre-define the relations between class values and category values for distance computation, where the distance determines the *k* NNs. In addition to SMOTE-N, the Minority Cloning Technique (MCT) [9] has also shown notable effectiveness within this learning domain.

With regard to the issue of the information gaps in small data, BP cannot overcome it and further enlarge the loss of information, since some observations may not be selected to form the bootstrapping sets based on a certain probability [7]. Moreover, although SMOTE-N can make numerous synthetic values to fill the gaps, these synthetic values are essentially distributed within the value domains of the minority class. If the training samples are concentrated within a part of the minority class's population, as shown in Fig. 1(b), then this would decrease the effectiveness of SMOTE-N. To overcome the issues represented in Figs. 1(b) and 2, the principle of information diffusion [22], as developed based on the fuzzy theory [23], was proposed to extend value domains on the both sides of an individual sample. Later, based on [22], some studies [24–26] proposed further filling the information gaps with virtual values.

However, the aforementioned fuzzy-based methods can only create samples with numerical inputs. Although bagging can handle nominal inputs, it creates the same samples as the original sample. Moreover, since SMOTE-N employs VDM to pre-define the relations (distances) between class values and category inputs for NN searching, the pre-definition restricts this approach to classification problems. Though other studies [27–30] have presented methods to deal with nominal inputs, they are not considered suitable for sample generation.

Focusing on small data which contains nominal inputs and continuous outputs, this study proposes a VSG procedure to overcome the learning issue by means of data preprocessing. The procedure contains three main processes, fuzzy relation extraction (FRE), sample generation (SG), and sample filtering (SF). The FRE process, developed based on the concept of the data preprocessing in the M5' model tree [31], employs fuzzy techniques to extract the fuzzy relations between the continuous outputs and nominal inputs. The SG process, developed based on possibility theory [32], aggregates the possibility values for all possible combinations of category values by a fuzzy operation when output values are given. The SF process filters out improper combinations by employing fuzzy α-cut on the distributions of the possibility values. Finally, the remaining combinations are treated as certified virtual samples and added to the original training sets to form new training sets. Note that the proposed VSG procedure can be integrated with some other VSG methods to handle various types of small data, as the outputs are continuous numbers.

In order to evaluate the effectiveness of the proposed procedure, ten datasets taken from the UC Irvine Machine Learning Repository (UCI) are examined with the back-propagation neural network (BPN) and support vector regression (SVR). To deal with the numerical attributes in the datasets, a fuzzy-based VSG method [25] is used as an example. Moreover, bagging and SMOTE-Nominal Continuous (SMOTE-NC) are also implemented as benchmarks for further comparisons of effectiveness.

This paper is organized as follows: Section 2 mainly reviews the theoretical bases of this study; Section 3 introduces the proposed procedure; Section 4 addresses the experimental environment and settings; Section 5 discusses the experimental results and findings. Finally, the conclusions and suggestions for further studies are discussed in Section 6.

2. Related studies

This section introduces related studies the work of which is employed in our method, including research on nominal input preprocessing, the mega trend diffusion technique (MTD) [26], possibility theory [32], and the possibility assessment mechanism (PAM) [7].

2.1. The nominal input preprocessing in the M5'

In the M5', a data preprocess is proposed to make the *k* possible category values in a nominal input represented by $k-1$ synthetic binary variables according to their numerical outputs. To

Table 1
An example with two nominal inputs and one continuous output.

No.	$\mathbf{X}_1$	$\mathbf{X}_2$	Output	No.	$\mathbf{X}_1$	$\mathbf{X}_2$	Output
1	C	E	0.308	6	C	F	0.212
2	D	F	0.699	7	A	E	−0.597
3	A	E	−0.937	8	B	E	−0.777
4	B	E	−0.335	9	D	F	0.459
5	C	F	0.565	10	C	F	0.552

Table 2
Categories are sorted according to the averaged output values.

Order	Possibly category values	Output average
1	A	−0.767
2	B	−0.556
3	C	0.409
4	D	0.579

Table 3
Categories are denoted by the synthetic binary values.

Category values	B, C, D	C, D	D
A	0	0	0
B	1	0	0
C	1	1	0
D	1	1	1

further explain the procedure, an example is given in Table 1. There are ten samples, one numerical output, and two nominal inputs $\mathbf{X}_1$ and $\mathbf{X}_2$ whose possible category values are {A, B, C, D} and {E, F}, respectively. Here, taking $\mathbf{X}_1$ for instance, the procedure first computes the averaged output values for {A, B, C, D} and then sorts the category values in the ascending order of their averaged output values, as shown in Table 2. The three synthetic variables are determined as {B, C, D}, {C, D}, and {D}, and the four category values are then denoted as the binary values in Table 3.

2.2. The mega-trend-diffusion technique

In 2007, Li et al. [26] proposed the MTD technique to estimate sample domains for small data and represent the possibility distributions by fuzzy membership functions (MFs). Suppose that a small sample set $X = \{x_1, x_2, \ldots, x_n\}$ has n observations and the minimum and maximum values are min and max, respectively, the lower bound (L) and upper bound (U) are estimated by

$$U = u_{set} + \text{skew}_U \times \sqrt{-2 \times \hat{s}_x^2 / N_U \times \ln\left(10^{-20}\right)}, \tag{1}$$

$$L = u_{set} - \text{skew}_L \times \sqrt{-2 \times \hat{s}_x^2 / N_L \times \ln\left(10^{-20}\right)}, \tag{2}$$

respectively, where u_{set} is the distribution center and is obtained by

$$u_{set} = (\min + \max)/2 \tag{3}$$

skew_L defined by$N_L/(N_L + N_U)$ and skew_U defined by $N_U/(N_L + N_U)$ are left and right skewness of the distribution, respectively; N_L and N_U are the numbers of values that are smaller and greater than u_{set}, respectively; and $\hat{s}_x^2$ is the sample variance of X. Based on L, u_{set}, and U, a fuzzy triangular MF can be drawn as shown Fig. 3 and formulated as

$$\mathrm{F}_X(x) = \begin{cases} (x - L)/(C - L), & L \le x \le C \\ (U - x)/(U - C), & C < x \le U \\ 0, & otherwise \end{cases} \tag{4}$$

to denote the possibility distribution of X.

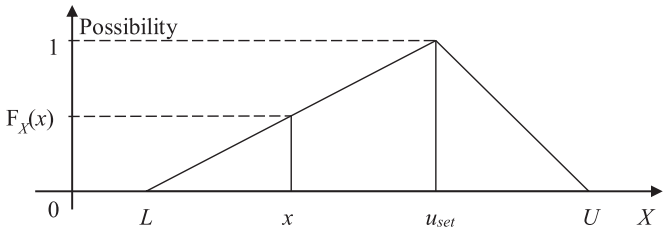

Fig. 3. Representing the possibility distribution of X by a triangular MF.

2.3. The possibility theory

In 1978, Zadeh [32] proposed the possibility theory, in which a fuzzy MF is regarded as a possibility distribution when a value may be given from a variable. Numerous later studies [33–35] then examined possibility/ probability transformations. Zadeh [32] indicated that there is a heuristic connection between possibilities and probabilities, where the connection is that the lower the possibility of an event then the lower its probability. Moreover, Zadeh also treated possibility as fuzzy restriction and defined the possibility of $X = u$ as postulated to be equal to its MF $F_X(u)$. Additionally, if X contains n attributes X_i which take values in U_i, $i = 1, \ldots, n$, the n-array possibility distribution function is given by fuzzy relation in the Cartesian product $U = U_1 \times \cdots \times U_n$, i.e.,

$$\mathrm{F}_{X_1,\cdots X_2}(u_1 \cdots u_n) \triangleq \mathrm{F}_{X_1}(u_1) \wedge \cdots \wedge \mathrm{F}_{X_n}(u_n), u_i \in U_i, i = 1, \ldots, n, \tag{5}$$

where $\wedge$ denotes a t-norm operation.

2.4. The possibility assessment mechanism (PAM)

Based on a heuristic concept, PAM was proposed to create virtual values for a given MF. The steps in PAM to create a virtual value vx of X are briefly summarized as follows, with Fig. 3 as an example:

Step 1. Randomly draw a vx from $[L, U]$ and compute its MF value $\mathrm{F}_X(vx)$.
Step 2. Randomly take a random seed (rs) from [0, 1] as a threshold value.
Step 3. If $\mathrm{F}_X(vx) > rs$, then vx can be treated as a certified virtual value; otherwise, vx is discarded.

Repeat Steps 1–3 until a certified vx is obtained.

2.5. SMOTE-NC

SMOTE was developed to overcome the classification problems with imbalanced classes. In addition to SMOTE, two extensive versions, SMOTE-NC and SMOTE-N, were also proposed in [8]. SMOTE-NC and SMOTE-N were designed when the data contains nominal inputs but with different pre-definitions of nominal distances. SMOTE-NC takes Med to denote a constant distance when the category values of NNs are different from those of an examining instance, where Med is the median of the standard deviations of numerical attributes of the minority class. SMOTE-N employs VDM to compute the distances between category values in a nominal input based on the class values. Obviously, SMOTE-NC can be applicable to numeric forecasting while SMOTE-N cannot. Therefore, this study adopts SMOTE-NC to carry out an effectiveness comparison in the experiments.

Table 4
The symbols defined in a small data set (**SDS**).

Instance No.	Nominal inputs (X)					Numerical output
	X_1	...	X_b	...	X_q	Y
1	$x_{1,1}$	...	$x_{1,b}$	...	$x_{1,q}$	y_1
...	...	...	...	...	...	...
i	$x_{i,1}$	...	$x_{i,b}$	...	$x_{i,q}$	y_i
...	...	...	...	...	...	...
n	$x_{n,1}$	...	$x_{n,b}$	...	$x_{n,q}$	y_n

Table 5
The subsets of the example in Table 1.

No.	X_1	Y	No.	X_2	Y
3	A	−0.937	1	E	0.308
7	A	−0.597	3	E	−0.937
4	B	−0.335	4	E	−0.335
8	B	−0.777	7	E	−0.597
1	C	0.308	8	E	−0.777
5	C	0.565	2	F	0.699
6	C	0.212	5	F	0.565
10	C	0.552	6	F	0.212
2	D	0.699	9	F	0.459
9	D	0.459	10	F	0.552

3. The proposed method

In this section, we will first define the notations which are used in the whole section, then introduce the proposed procedure in three subsections, and finally outline the implementation steps.

3.1. Definition of notations

Assume that a small data set (**SDS**) having n instances, q nominal inputs (**X**), and one numerical output (**Y**), as listed in Table 4. Suppose that a jagged array $\mathbf{C} = \{ c_{b,j} | b = 1, 2, \ldots, q,\ j = 1, 2, \ldots, k_b \}$ denotes the array of possible category values in the q nominal inputs, and k_1,..., k_b,..., k_q are the sizes of possible category values in $\mathbf{X}_1$,..., $\mathbf{X}_b$,..., $\mathbf{X}_q$, respectively. Moreover, we take m as the initial expected size of the virtual samples.

3.2. The fuzzy relation extraction

As explained in Section 2.1, the transformation procedure in the M5' gives us a hint to define the relations between the possible category values and their continuous output values, as SMOTE-N employs VDM to pre-define the relations (distances) between category inputs and class values. In contrast to the M5' which transforms a nominal input into synthetic binary variables, this study derives the fuzzy relations between each possible category value and its fuzzy MF of output values. The detailed procedure is described as follows, steps with the example in Section 2.1.

Step 1. For each of a nominal input $\mathbf{X}_b$, separate **SDS** into k_b subsets according to its k_b possible category values, where $b = 1, 2, \ldots, q$. Suppose the inputs in Table 1 are $\mathbf{X}_1$ and $\mathbf{X}_2$, and Table 1 is thus divided into four and two subsets according to their four and two possible category values, respectively, as summarized in Table 5. Note that this step is to make the process easier to understand, but is not necessary when implementing this method.

Step 2. For each of the k_b categories $\{c_{b,1}, c_{b,2}, \ldots, c_{b,k_b}\}$ in $\mathbf{X}_b$, employ MTD mentioned in Section 2.2 to estimate the value bounds of its output values, and then build its triangular MF in the form shown in Eq. (4). In the example, the four categories in $\mathbf{X}_1$ are {A, B, C, D}, and their estimated lower bounds (L), upper bounds (U), and distribution centers (u_{set}) are summarized in Table 6.

Table 6
The estimated value bounds (L, U) and center (u_{set}) of the example.

Inputs	Category values	L	u_{set}	U
$\mathbf{X}_1$	A	−1.921	−0.767	0.387
	B	−2.056	−0.556	0.944
	C	−0.211	0.388	0.988
	D	−0.235	0.579	1.393
$\mathbf{X}_2$	E	−2.187	−0.315	0.622
	F	0.108	0.455	1.151

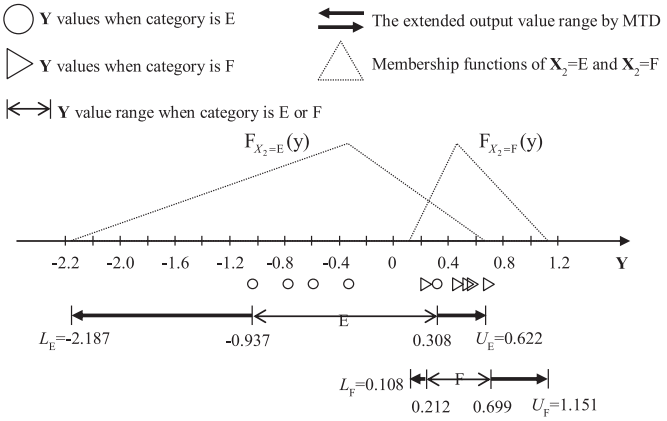

Fig. 4. The principle of the fuzzy relations between output values and categories.

Repeat Steps 1 to 2 until all the MFs of the possible categories **C** are obtained. Accordingly, there are totally $\sum_{b=1}^{q} k_b$ MFs. In the example, there are $(4 + 2) = 6$ MFs. Here, we take Fig. 4 to explain the extracted fuzzy relations between output values and $\mathbf{X}_2$ in the example. From Table 5, when $\mathbf{X}_2$ is E, the value range of **Y** is found to be [−0.937, 0.308]. Conversely, this means the value of $\mathbf{X}_2$ is possible to be E only when **Y** is located within [−0.937, 0.308] from the observation in the ten samples. Similarly, $\mathbf{X}_2$ is possible to be F only when **Y** is located within [0.212, 0.699]. However, it is possible that **Y** is located out of [−0.937, 0.308] when $\mathbf{X}_2$ is E, since only ten samples cannot stand for the whole population. Accordingly, this paper adopts MTD to extend the value domains of **Y** when $\mathbf{X}_2$ is E and F to increase the possibility that $\mathbf{X}_2$ can be E or F when **Y** is out of the current value ranges. The use of MFs is to control the possibility of **Y** values when $\mathbf{X}_2$ is E or F. For example, when $\mathbf{X}_2$ is E and **Y** is closer to the lower bound of E (L_E), the possibility values would be closer to zero; on the other hand, if **Y** is closer to current observations, such as −0.4, the condition $\{ \mathbf{X}_2 = \mathrm{E} | \mathbf{Y} = -0.4 \}$ would have a very high possibility of occurring. Moreover, conditions such as $\{ \mathbf{X}_2 = \mathrm{E} | \mathbf{Y} = 0.8 \}$ and $\{ \mathbf{X}_2 = \mathrm{F} | \mathbf{Y} = 0 \}$ are impossible and thus cannot occur.

3.3. The sample generation

When the fuzzy relations are derived, the following steps are used to create a template for further sample generation.

Step 1. Arrange all possible combinations of the possible category values in the q nominal inputs to obtain the $K = \prod_{b=1}^{q} k_b$ combinations, which are denoted by

$$\mathbf{COB} = \begin{bmatrix} \mathbf{COB}_1 \\ \mathbf{COB}_2 \\ \vdots \\ \mathbf{COB}_K \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{2,1} & \cdots & c_{q,1} \\ c_{1,2} & c_{2,1} & \cdots & c_{q,1} \\ \vdots & \vdots & \vdots & \vdots \\ c_{1,k_1} & c_{2,k_2} & \cdots & c_{q,k_q} \end{bmatrix}. \quad (6)$$

Table 7
The MF values of the example when the output value is 0.8.

Inputs	Category values	L	u_{set}	U	MF values
1	A	−1.921	−0.767	0.387	**0.000**
	B	−2.056	−0.556	0.944	0.096
	C	−0.211	0.388	0.988	0.314
	D	−0.235	0.579	1.393	0.729
2	E	−2.187	−0.315	0.622	**0.000**
	F	0.108	0.455	1.151	0.504

Table 8
The possibility values of the eight combinations when the output value is 0.8.

COB No.	X_1	X_2	Possibility	COB No.	X_1	X_2	Possibility
1	A	E	**0.000**	5	A	F	**0.000**
2	B	E	**0.000**	6	B	F	0.220
3	C	E	**0.000**	7	C	F	0.398
4	D	E	**0.000**	8	D	F	0.606

Note that this process increases the time complexity to $O(\prod_{b=1}^{q} k_b)$.

Step 2. Based on the possibility theory mentioned in Section 2.3, use Eq. (5) with setting the t-norm operation as the product to aggregate the possibility value of the ith combination in **COB** by

$$\mathrm{P}(\mathrm{COB}_i|\mathbf{Y}) = \prod_{b=1}^{q} \mathrm{F}_{\mathrm{X}_b=c_{b,\bullet}}(\mathbf{Y}), \tag{7}$$

where $i = 1, 2, \ldots, K$, and $c_{b,\bullet}$ denotes a category value of the bth input in the ith combination. However, when the sizes of nominal inputs are large, the possibility would be a very small value and difficult to handle. Eq. (7) is thus suggested to be redefined as

$$\mathrm{P}(\mathrm{COB}_i|\mathbf{Y}) = \sqrt[q]{\prod_{b=1}^{q} \mathrm{F}_{\mathrm{X}_b=c_{b,\bullet}}(\mathbf{Y})}, \tag{8}$$

and it is conceptually regarded as the geometric mean of the possibility of category values.

From now on, **COB** becomes a template. When an output value is given, we can obtain the possibility values of all the combinations in **COB** by using Eqs. (4) and (8). Here, the results of the example are listed in Tables 7 and 8 when an output value is 0.8, where the results of Tables 7 and 8 are obtained by using Eqs. (4) and (8), respectively. In Table 7, the MF values (i.e. the possibility values) are zero when $\mathbf{X}_1$ is A and when $\mathbf{X}_2$ is E, since the output value 0.8 exceeds their estimated domains. In Table 8, only **COB** 6, 7, and 8 are not zero. If we further compare Table 8 with Table 1, **COB** 7 and 8 are real samples with larger output values in Table 1, where **COB** 7 is samples No. 5, 6, and 10, and **COB** 8 is samples No. 7 and 9. Moreover, the averaged output value of samples No. 5, 6, and 10 is $(0.565 + 0.212 + 0.552) = 0.443$, and that of samples No. 2 and 9 is $(0.699 + 0.459) = 0.579$. When the output value is 0.8, 0.579 is closer to it than 0.443 is, and the possibility value of **COB** 8 is larger than that of **COB** 7. Certainly, this is a simple example, and the situation could be more complicated and not easy to explain when input sizes are large. In addition, it is worth noting that **COB** 6 is not a real sample but appears to have non-zero possibility. Nevertheless, the possibility value of **COB** 6 is very small, and this means the combination has a low possibility of happening based on our learning results for the ten real samples. On the other hand, the proposed method extends the possibility to create combinations that do not appear in the few observations but probably exist in the real populations.

When a small dataset contains only nominal inputs, the following steps are used to complete the sample generation.

Step 1. Utilitize MTD mentioned in Section 2.2 to estimate the lower bound (L), upper bound (U), and center (u_{set}) of **Y** by Eq. (1), Eq. (2), and Eq. (3), respectively, and then formulate the MF of **Y** in the form shown in Eq. (4).

Step 2. Employ PAM mentioned in Section 2.4 to create a set of m virtual output values.

Step 3. Combine each of the m virtual output values with **COB** to form $K \times m$ virtual samples.

Step 4. Input the m virtual output values in **COB** to obtain $K \times m$ possibility values.

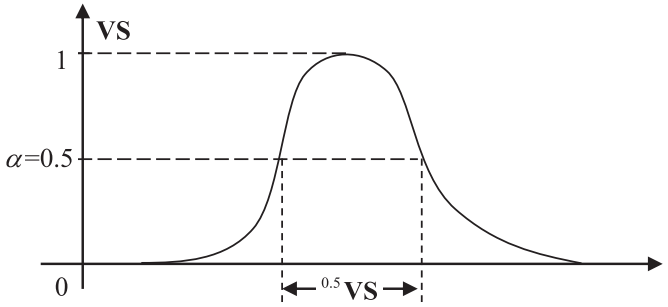

Fig. 5. Employ α-cut to select virtual samples.

When adopting some learning algorithms in which input coefficients are meaningless, such as BPN, we suggest extending the possibility values as an additional input (i.e. a virtual attribute) to facilitate the learning. The possibility values of training and testing sets can be obtained by inputting their output values in **COB**. Once testing sets have some category values which do not exist in training sets, the possibility values of the unseen combinations in **COB** are suggested to set as the max possibility value as 1.

When a small dataset has both numerical and nominal inputs, the following steps are used to complete the sample generation.

Step 1. Employ some other VSG approaches, such as that in [25] or SMOTE-NC, to create the numerical input and output values of m virtual samples.

Step 2. Combine each of the m virtual samples with **COB** to form $K \times m$ virtual samples.

Step 3. Input the m virtual output values in **COB** to obtain $K \times m$ possibility values.

3.4. The sample filtering

When the $K \times m$ virtual samples are created, some combinations in **COB** have zero or very small possibility values, as shown in Table 8. These combinations are not considered suitable to be qualified virtual samples. Based on fuzzy techniques, this study adopts α-cut ($\alpha \in [0, 1]$) to filter the created samples, as shown in Fig. 5, and the selected virtual samples ($^{\alpha}\mathbf{VS}$) are defined as

$$^{\alpha}\mathbf{VS} = \{\mathrm{COB}_i |\ \mathrm{P}(\mathrm{COB}_i|\mathbf{Y} = y_l) > \alpha, i = 1, 2, \ldots, K; l = 1, 2, \ldots, m\}, \tag{9}$$

where q is the number of nominal inputs. Note that we take a normal and convex set as an illustration to denote the shape of **VS** in Fig. 5, and the real shape would depend on the properties of the learning data.

As seen in Fig. 5, if α-cut is larger, the virtual samples that are located closer to the distribution centers are then selected; however, the selected virtual samples may be very similar to the original samples, which would lead to the issue of overfitting. On the other hand, if α-cut is smaller, more virtual samples are then selected; nevertheless, certain parts of the selected virtual samples, having non-existent combinations in the real populations, thus become noise that affects the forecasting accuracy of the learning algorithms. It should be noted that the sizes of the selected virtual

Table 9
The profiles of the eleven UCI datasets.

Datasets	Instances	Input Attributes		Predicting targets
		Numerical	Nominal	
Abalone	4177	7	1	The number of rings
Automobile	159	11	14	The price of cars
Auto MPG	392	3	4	The city-cycle fuel consumption
Bike[1]	731	4	5	The count of causal users by day
Bike[2]	731	4	5	The count of causal users by day
Coil[1]	167	8	3	The seven algae population distributions
Coil[2]	167	8	3	The seven algae population distributions
Coil[3]	167	8	3	The seven algae population distributions
Coil[4]	167	8	3	The seven algae population distributions
Coil[5]	167	8	3	The seven algae population distributions
Coil[6]	167	8	3	The seven algae population distributions
Coil[7]	167	8	3	The seven algae population distributions
Forest fires [36]	517	8	4	The burned area of the forest
Housing	506	11	2	The house values in Boson
Servo	167	2	2	The rise time of a servomechanism
Servo*	167	0	4	The rise time of a servomechanism

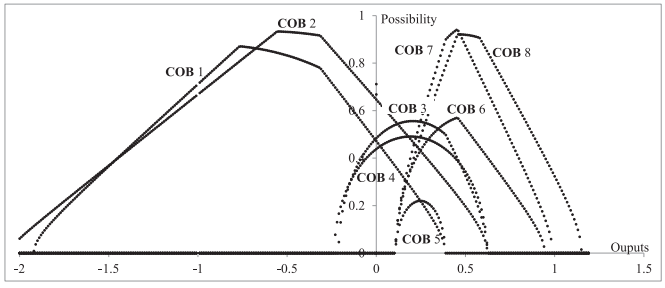

Fig. 6. The possibility distributions of the eight combinations.

samples cannot be confirmed before the generation procedure is done, although we set the initial expected size as m, and the maximum size of the virtual sample pool (**VS**) can be known as $K \times m$.

Here, we take the example in Table 8 for further explanation. As shown in Fig. 6, we plot the possibility distributions of the eight combinations when continuous outputs are given. From Fig. 6, we can observe that the combinations **COB** 3, 4, 5, and 6 can be filtered out when α is larger than 0.6, where **COB** 4, 5, and 6 do not exist in Table 1. On the other hand, although a smaller α value makes **COB** 4, 5, and 6 possible to be certified virtual samples and be added in the training sets, there is a risk that these combinations are actually noise and would lower the precision of the models.

3.5. The implementation outline

The outline of implementing the proposed method is as follows:

Step 1. Fuzzy relation extraction:

1 Identify the possible category values and their corresponding output values for the q nominal inputs in **SDS**.
2 Employ MTD (Section 2.2) to formulate the MFs of the output values for each category value.

Step 2. Construct the template combinations (**COB**):

1 Arrange all possible combinations with category values.
2 Based on possibility theory (Section 2.3), formulate possibility computing functions for each combination.

Step 3. Sample generation:

1 When data has only nominal inputs:
1.1 Employ PAM (Section 2.4) to create virtual output values.
1.2 Combine the virtual output values with **COB** to form virtual samples.
1.3 Input virtual output values in **COB** to obtain possibility values.
2 When data has both nominal and numerical inputs:
2.1 Employ some other VSG approaches (e.g. [8,25]), to create numerical inputs and outputs of virtual samples.
2.2 Combine each virtual sample with **COB** to form virtual samples.
2.3 Input virtual output values in **COB** to obtain possibility values.

Step 4. Employ α-cut to filter virtual samples.
Step 5. Add the filtered virtual samples to **SDS** to form new training sets.

4. The experimental environment

This section addresses the experimental environment of this study.

4.1. The datasets examined

To meet the capabilities of the proposed method, which handles nominal inputs and continuous outputs, eight datasets are taken from UCI in order to assess the effectiveness of the approach, where the datasets are extended as sixteen datasets, as summarized in Table 9. The instances which have missing or unidentified values are removed, and the revisions made in the datasets are described as follows:

1. Bike: This is the "Bike Sharing" data in UCI. The inputs of instant, dteday, mnth, and holiday are removed because these are unique or are denoted by other inputs. Moreover, we further regard the data as two datasets, "Bike[1]" and "Bike[2]", by taking the counts of casual and registered users as the outputs, respectively.
2. Coil: This is the "Coil 1999 Competition" data in UCI. We only take the training set because the testing set does not have outputs for validation. Since this data has seven outputs, we make it as seven sets, each of which has one output.
3. Forest fires: The input "month" and the output "area" are transformed into seasonal labels and logarithmic

Table 10
The results of the 10-fold cross-validation when the model is BPN.

Data sets	VS (%)	SDS	Bagging	SMOTE-NC	PM
Abalone	100	18.64 ± 003.24*	16.36 ± 001.93*	18.07 ± 006.50*	9.71 ± 003.65
	200	18.64 ± 003.24*	15.93 ± 001.88*	16.97 ± 001.89*	9.68 ± 003.64
	300	18.64 ± 003.24*	15.52 ± 001.83*	16.81 ± 003.73*	9.46 ± 003.56
	400	18.64 ± 003.24*	15.50 ± 001.83*	15.80 ± 002.24*	9.22 ± 003.47
	500	18.64 ± 003.24*	15.62 ± 001.84*	16.67 ± 001.85*	9.23 ± 003.47
Automobile	100	58.83 ± 028.80*	35.80 ± 015.81*	32.70 ± 011.83*	20.35 ± 004.18
	200	58.83 ± 028.80*	35.56 ± 015.70*	28.25 ± 010.31*	20.30 ± 002.81
	300	58.83 ± 028.80*	35.74 ± 015.78*	23.35 ± 007.57*	19.22 ± 003.73
	400	58.83 ± 028.80*	34.01 ± 015.02*	22.39 ± 008.04*	20.69 ± 007.19
	500	58.83 ± 028.80*	34.85 ± 015.39*	27.74 ± 011.56*	22.42 ± 006.40
AutoMPG	100	19.00 ± 007.71*	15.52 ± 003.32*	15.63 ± 003.21*	13.42 ± 002.37
	200	19.00 ± 007.71*	15.08 ± 003.23*	15.11 ± 002.93*	13.89 ± 002.89
	300	19.00 ± 007.71*	15.22 ± 003.26*	14.75 ± 002.16*	13.10 ± 002.37
	400	19.00 ± 007.71*	14.87 ± 003.18*	14.17 ± 003.13*	13.16 ± 002.29
	500	19.00 ± 007.71*	15.01 ± 003.21*	16.82 ± 002.90*	13.29 ± 002.31
Bike[1]	100	136.97 ± 056.43*	122.16 ± 47.05*	144.72 ± 063.35*	112.19 ± 050.74
	200	136.97 ± 056.43*	119.54 ± 55.86*	126.29 ± 049.95*	112.28 ± 052.78
	300	136.97 ± 056.43*	119.05 ± 50.63*	130.56 ± 057.23*	112.15 ± 050.67
	400	136.97 ± 056.43*	116.01 ± 52.89*	129.77 ± 051.69*	110.57 ± 048.00
	500	136.97 ± 056.43*	119.41 ± 55.80*	133.90 ± 058.54*	111.85 ± 051.13
Bike[2]	100	55.89 ± 016.96*	52.86 ± 014.39*	54.11 ± 017.43*	50.49 ± 016.76
	200	55.89 ± 016.96*	52.96 ± 014.42*	53.27 ± 017.16*	50.16 ± 016.65
	300	55.89 ± 016.96*	52.72 ± 014.36*	54.62 ± 017.59*	49.94 ± 016.58
	400	55.89 ± 016.96*	53.22 ± 014.49*	54.54 ± 017.57*	50.22 ± 016.67
	500	55.89 ± 016.96*	53.49 ± 014.56*	54.70 ± 017.62*	50.70 ± 016.83
Coil[1]	100	589.95 ± 286.81*	569.98 ± 306.65*	**553.24 ± 307.90***	554.36 ± 321.93
	200	589.95 ± 286.81*	560.76 ± 301.69*	**512.30 ± 285.12***	541.17 ± 314.27
	300	589.95 ± 286.81*	560.22 ± 301.40*	581.37 ± 323.56*	541.77 ± 314.62
	400	589.95 ± 286.81*	564.83 ± 303.88*	**525.86 ± 292.66***	527.16 ± 306.14
	500	589.95 ± 286.81*	568.97 ± 306.11*	**513.50 ± 285.79***	529.29 ± 307.38
Coil[2]	100	317.78 ± 139.29*	312.39 ± 145.59*	305.13 ± 154.20*	264.55 ± 122.82
	200	317.78 ± 139.29*	311.82 ± 145.32*	295.14 ± 149.15*	260.27 ± 120.83
	300	317.78 ± 139.29*	311.66 ± 145.24*	286.55 ± 144.80*	260.97 ± 121.16
	400	317.78 ± 139.29*	312.16 ± 145.48*	293.58 ± 148.36*	260.94 ± 121.14
	500	317.78 ± 139.29*	311.78 ± 145.30*	294.67 ± 148.91*	265.99 ± 123.49
Coil[3]	100	282.99 ± 147.80*	272.23 ± 138.86*	283.53 ± 146.95*	252.09 ± 149.60
	200	282.99 ± 147.80*	273.63 ± 139.57*	260.03 ± 134.77*	248.39 ± 146.26
	300	282.99 ± 147.80*	273.07 ± 139.29*	268.08 ± 138.94*	245.54 ± 145.71
	400	282.99 ± 147.80*	274.33 ± 139.93*	269.62 ± 139.74*	247.54 ± 146.90
	500	282.99 ± 147.80*	275.02 ± 140.28*	280.29 ± 145.27*	247.74 ± 147.01
Coil[4]	100	118.03 ± 040.83*	112.71 ± 038.02*	116.54 ± 042.19*	109.12 ± 037.70
	200	118.03 ± 040.83*	113.08 ± 038.14*	118.49 ± 042.90*	108.65 ± 037.54
	300	118.03 ± 040.83*	112.95 ± 038.10*	114.57 ± 041.48*	106.25 ± 036.71
	400	118.03 ± 040.83*	112.47 ± 037.93*	117.47 ± 042.53*	107.53 ± 037.15
	500	118.03 ± 040.83*	112.55 ± 037.96*	115.71 ± 041.89*	107.22 ± 037.04
Coil[5]	100	257.80 ± 097.16*	256.54 ± 096.13*	255.52 ± 098.81*	245.49 ± 094.87
	200	257.80 ± 097.16*	255.55 ± 095.76*	254.36 ± 098.36*	241.60 ± 093.21
	300	257.80 ± 097.16*	256.63 ± 096.16*	260.47 ± 099.72*	244.97 ± 094.65
	400	257.80 ± 097.16*	255.51 ± 095.74*	258.30 ± 099.88*	241.59 ± 094.21
	500	257.80 ± 097.16*	255.10 ± 095.59*	257.54 ± 099.59*	241.53 ± 093.18
Coil[6]	100	377.27 ± 214.42*	346.58 ± 185.61*	374.43 ± 211.68*	337.15 ± 176.44
	200	377.27 ± 214.42*	345.70 ± 185.14*	375.21 ± 212.12*	337.96 ± 176.86
	300	377.27 ± 214.42*	345.75 ± 185.16*	362.84 ± 205.13*	333.45 ± 174.50
	400	377.27 ± 214.42*	343.76 ± 184.10*	367.09 ± 207.53*	319.07 ± 166.97
	500	377.27 ± 214.42*	345.59 ± 185.08*	374.29 ± 211.59*	335.07 ± 175.35
Coil[7]	100	197.34 ± 108.34*	199.06 ± 109.86*	191.99 ± 102.93*	154.85 ± 069.61
	200	197.34 ± 108.34*	196.36 ± 108.37*	181.58 ± 097.35*	149.26 ± 067.09
	300	197.34 ± 108.34*	192.88 ± 106.46*	197.52 ± 105.89*	147.02 ± 066.09
	400	197.34 ± 108.34*	191.20 ± 105.53*	183.58 ± 098.42*	139.04 ± 062.50
	500	197.34 ± 108.34*	191.78 ± 105.85*	197.45 ± 105.86*	140.00 ± 062.93
Forest fires	100	77.12 ± 027.39*	68.19 ± 026.02*	66.21 ± 024.15*	65.40 ± 023.58
	200	77.12 ± 027.39*	66.87 ± 025.52*	67.64 ± 024.68*	61.55 ± 022.19
	300	77.12 ± 027.39*	66.20 ± 025.26*	69.18 ± 025.24*	62.17 ± 022.41
	400	77.12 ± 027.39*	66.61 ± 025.42*	65.11 ± 023.75*	60.78 ± 021.91
	500	77.12 ± 027.39*	66.35 ± 025.32*	67.50 ± 024.63*	60.78 ± 021.91
Housing	100	22.74 ± 005.80*	17.62 ± 002.67*	17.05 ± 002.94*	15.41 ± 002.46
	200	22.74 ± 005.80*	17.16 ± 002.42*	16.29 ± 002.49*	15.61 ± 002.40
	300	22.74 ± 005.80*	17.42 ± 002.59*	16.60 ± 003.68*	15.33 ± 002.45
	400	22.74 ± 005.80*	17.87 ± 002.80*	16.35 ± 002.94*	15.46 ± 002.47
	500	22.74 ± 005.80*	17.47 ± 002.86*	16.50 ± 003.13*	15.57 ± 002.48

(*continued on next page*)

Table 10 (*continued*)

Data sets	VS (%)	SDS	Bagging	SMOTE-NC	PM
Servo	100	109.03 ± 065.69*	88.56 ± 054.36*	97.08 ± 053.87*	68.72 ± 047.00
	200	109.03 ± 065.69*	89.96 ± 055.22*	84.63 ± 062.63*	69.05 ± 047.45
	300	109.03 ± 065.69*	89.80 ± 055.12*	88.32 ± 055.96*	66.69 ± 045.43
	400	109.03 ± 065.69*	91.01 ± 055.86*	90.63 ± 056.02*	66.93 ± 047.41
	500	109.03 ± 065.69*	89.40 ± 054.88*	92.53 ± 056.75*	61.88 ± 041.77
Servo*	100	88.58 ± 030.00*	81.99 ± 023.91*	93.46 ± 034.41*	71.12 ± 024.25
	200	88.58 ± 030.00*	81.60 ± 023.80*	91.72 ± 037.51*	69.46 ± 020.57
	300	88.58 ± 030.00*	81.07 ± 023.64*	76.91 ± 028.08*	64.83 ± 026.57
	400	88.58 ± 030.00*	81.41 ± 023.74*	78.26 ± 027.08*	65.51 ± 024.54
	500	88.58 ± 030.00*	81.55 ± 023.78*	89.04 ± 031.35*	65.80 ± 018.00
Averages	100	170.50 (14.06%)	160.53 (08.73%)	163.71 (10.50%)	146.53
	200	170.50 (15.35%)	159.47 (09.50%)	156.08 (07.53%)	144.33
	300	170.50 (15.95%)	159.12 (09.94%)	160.16 (10.52%)	143.30
	400	170.50 (17.32%)	159.05 (11.37%)	156.41 (09.87%)	140.96
	500	170.50 (16.48%)	159.62 (10.79%)	159.30 (10.61%)	142.40

values according to the climate in Portugal and the suggestion made by the data provider [36], respectively.

4. Servo*: This data is formed by taking the two discrete numerical inputs in "Servo" as being nominal inputs.

4.2. The designs of the experiments

In this study, we adopt 10-fold cross-validation to carry out the experiments. However, one fold is taken as the training set (named SDS hereafter) to simulate a small data set and the other nine folds are regarded as testing sets. The error indicator employed in this study is the mean absolute percentage error (MAPE), which is computed by

$$\mathrm{MAPE}_j = \frac{1}{N-n}\sum_{i=1}^{N-n}\left|\frac{y_i-\hat{y}_i}{y_i}\right| \times 100\%, \tag{10}$$

where y_i is the ith output value in the testing set, $\hat{y}_i$ is the prediction of y_i, N is the instance number, n is the size of a training set, $N-n$ is the size of the testing set, and j refers to the jth cross-validation.

To identify whether significant differences exist between the proposed method and the other approaches with statistical support, paired t-tests with a two-tailed test are performed. The null hypothesis (H_0) and alternative hypothesis (H_a) are formulated as

$$\begin{cases} H_0: \mu_d = 0 \\ H_a: \mu_d \neq 0 \end{cases}, \tag{11}$$

where μ_d is the average of $\{ d_j | \ j = 1, 2, \ldots, k \}$, and d_j is the error deviation between the proposed method and the other approaches in the jth cross-validation run. The test level α is set as 0.05.

The VSG approaches adopted in this study for effectiveness comparison are bagging (using BP) and the SMOTE-NC. Moreover, this study integrates the fuzzy-based VSG method [25] with the proposed procedure when data contains numerical inputs. The parameters of the three VSG approaches are as follows: the sizes of virtual samples are set as 100%, 200%, 300%, 400%, and 500% relative to the sizes of training sets; the α value to filter samples is set as 0.7.

The modeling software adopted in the experiments is Weka 3.8.1. The learning algorithms employed in this work are BPN and SVR, which can be found in Weka as labeled as "MultilayerPerceptron" and "SMOreg", respectively. The parameter settings of the learning algorithms in Weka take the default values. Moreover, bagging can also be found in Weka labeled as "Bagging", and the parameters of BPN and SVR in this are also the defaults. Note that the objective of this study is not making a precise model, but is examining the differences when different types of virtual samples are added to training sets. Moreover, controlling this factor is another objective for paired t-tests.

The experimental platform is Windows 10 (64bit) with CPU i5-4200M (2.5 GHz) and 8 GB RAM. The language employed to code the proposed method and SMOTE-NC is Java, where SMOTE-NC is an overridden version of SMOTE downloaded from the package manager in WEKA.

5. Experimental results and discussion

This section discusses the experimental results and findings.

5.1. Results of the experiments

The results of BPN and SVR are summarized in Tables 10 and 11, in which "PM" denotes the proposed method; "VS (%)" means the sizes of virtual samples relative to the sizes of the training sets; the symbol * stands for the significant difference of the paired t-test results; the bold values are smaller (better) than the error values of "PM". Moreover, the averaged MAPEs (marked as "Averages") are summarized at the bottom of the tables, where the numbers in percent are the relative rates of the error deviations between "SDS", "Bagging", and "SMOTE-NC" and the proposed method. Take the "Averages" of "SDS" in Table 10 for example, the percentages mean "PM" can help improve the errors of "SDS" by up to 14.06%, 15.35%, 15.95%, 17.32%, and 16.48% on average when "VS (%)" are 100, 200, 300, 400, and 500, respectively. From the results compared in Tables 10 and 11, we can observe that "PM" achieves a substantial reduction in the averaged MAPEs of "SDS"; moreover, "PM" results in smaller averaged MAPEs than the averaged MAPEs of "Bagging" and "SMOTE-NC" with the greatest amount of statistical support. This result further indicates that "PM" could be considered as an effective method to help learning algorithms, such as BPN and SVR, improve the forecasting errors when learning with small datasets.

5.2. Findings from the experimental results

Some findings from the experimental results listed in Tables 10 and 11, are summarized as follows:

1 The samples created by SMOTE-NC made the SVR output better predictions than the BPN, since SMOTE-NC generates samples using a linear mechanism and pairs of real samples. From Table 10, we can find that the MAPES are less stable than the MAPEs in Table 11 as "VS (%)" increases in the nine datasets, Bike[1], Bike[2], Coil[1], Coil[3], Coil[4], Coil[5], Coil[7], Forest fires, and Servo*.

Table 11
The results of the 10-fold cross-validation when the model is SVR.

Data sets	VS (%)	SDS	Bagging	SMOTE-NC	PM
Abalone	100	15.08 ± 000.34*	11.94 ± 000.25*	13.35 ± 000.34*	5.99 ± 002.07
	200	15.08 ± 000.34*	11.65 ± 000.24*	13.24 ± 000.41*	5.82 ± 002.01
	300	15.08 ± 000.34*	11.50 ± 000.24*	13.26 ± 000.31*	5.73 ± 001.98
	400	15.08 ± 000.34*	11.49 ± 000.24*	12.81 ± 000.26*	5.71 ± 001.97
	500	15.08 ± 000.34*	11.17 ± 000.23*	12.86 ± 000.32*	5.79 ± 002.00
Automobile	100	**18.18 ± 003.17***	19.06 ± 004.44*	18.30 ± 004.17*	18.21 ± 004.12
	200	18.18 ± 003.17*	18.94 ± 004.41*	17.97 ± 004.11*	17.89 ± 004.03
	300	18.18 ± 003.17*	18.79 ± 004.37*	**17.28 ± 004.75***	17.91 ± 004.03
	400	18.18 ± 003.17*	18.87 ± 004.39*	17.09 ± 002.80*	16.67 ± 002.99
	500	18.18 ± 003.17*	18.91 ± 004.40*	**16.72 ± 003.65***	17.56 ± 003.52
AutoMPG	100	14.70 ± 002.56*	13.80 ± 001.41*	12.91 ± 002.30*	12.54 ± 001.83
	200	14.70 ± 002.56*	13.58 ± 001.38*	**12.13 ± 001.69***	12.34 ± 001.80
	300	14.70 ± 002.56*	13.09 ± 001.33*	**11.71 ± 001.30***	12.24 ± 001.45
	400	14.70 ± 002.56*	13.09 ± 001.33*	**11.66 ± 001.67***	12.03 ± 001.63
	500	14.70 ± 002.56*	13.15 ± 001.35*	12.57 ± 0v2.54*	11.32 ± 001.29
Bike1	100	112.76 ± 031.90*	104.47 ± 043.80*	**93.43 ± 040.53***	100.80 ± 045.58
	200	112.76 ± 031.90*	102.75 ± 043.04*	**82.43 ± 027.13***	99.55 ± 045.31
	300	112.76 ± 031.90*	99.86 ± 041.77*	**80.32 ± 031.34***	98.10 ± 044.92
	400	112.76 ± 031.90*	99.81 ± 041.75*	**82.74 ± 031.01***	98.10 ± 043.32
	500	112.76 ± 031.90*	99.88 ± 041.78*	**77.85 ± 028.19***	98.21 ± 044.60
Bike2	100	51.97 ± 015.29*	**48.42 ± 013.69***	**48.61 ± 016.32***	49.71 ± 013.67
	200	51.97 ± 015.29*	**48.70 ± 013.77***	**49.03 ± 016.45***	49.96 ± 013.73
	300	51.97 ± 015.29*	**48.58 ± 013.74***	**47.58 ± 015.97***	48.87 ± 013.44
	400	51.97 ± 015.29*	**48.59 ± 013.74***	**48.36 ± 016.23***	49.14 ± 013.51
	500	51.97 ± 015.29*	**48.95 ± 013.84***	**48.89 ± 016.41***	49.05 ± 013.48
Coil[1]	100	**215.28 ± 086.85***	**210.40 ± 082.53***	219.59 ± 092.81*	216.78 ± 088.32
	200	215.28 ± 086.85*	**211.07 ± 082.79***	215.98 ± 091.28*	211.43 ± 086.14
	300	215.28 ± 086.85*	211.64 ± 083.02*	214.41 ± 090.62*	208.23 ± 084.83
	400	215.28 ± 086.85*	210.02 ± 082.38*	213.98 ± 090.44*	204.97 ± 083.51
	500	215.28 ± 086.85*	211.46 ± 082.95*	214.33 ± 090.59*	205.55 ± 083.74
Coil[2]	100	130.86 ± 044.14*	127.69 ± 046.29*	135.12 ± 054.70*	127.36 ± 046.38
	200	130.86 ± 044.14*	**127.79 ± 046.33***	134.12 ± 054.29*	128.13 ± 046.66
	300	130.86 ± 044.14*	**127.11 ± 046.08***	132.33 ± 051.27*	127.31 ± 046.36
	400	130.86 ± 044.14*	127.55 ± 046.24*	130.37 ± 052.77*	123.82 ± 045.09
	500	130.86 ± 044.14*	127.85 ± 046.35*	130.32 ± 049.25*	127.36 ± 046.38
Coil[3]	100	134.83 ± 060.92*	134.35 ± 060.55*	135.58 ± 061.32*	113.95 ± 045.02
	200	134.83 ± 060.92*	134.71 ± 060.72*	136.96 ± 061.94*	113.64 ± 044.90
	300	134.83 ± 060.92*	132.01 ± 059.50*	128.30 ± 058.03*	109.48 ± 043.26
	400	134.83 ± 060.92*	132.57 ± 059.75*	135.36 ± 061.22*	106.38 ± 042.03
	500	134.83 ± 060.92*	132.89 ± 059.90*	136.27 ± 061.63*	106.43 ± 042.05
Coil[4]	100	54.35 ± 017.23*	54.39 ± 017.23*	54.86 ± 017.01*	52.38 ± 019.10
	200	54.35 ± 017.23*	54.71 ± 017.33*	54.62 ± 016.94*	51.46 ± 018.76
	300	54.35 ± 017.23*	54.80 ± 017.36*	53.42 ± 016.57*	48.18 ± 017.56
	400	54.35 ± 017.23*	55.02 ± 017.43*	54.09 ± 016.78*	47.48 ± 017.31
	500	54.35 ± 017.23*	54.51 ± 017.27*	54.73 ± 016.97*	47.55 ± 017.33
Coil[5]	100	138.98 ± 046.52*	140.47 ± 047.52*	141.83 ± 051.78*	120.58 ± 047.98
	200	138.98 ± 046.52*	140.43 ± 047.51*	143.95 ± 052.55*	121.81 ± 048.47
	300	138.98 ± 046.52*	140.30 ± 047.46*	136.80 ± 049.94*	120.58 ± 047.98
	400	138.98 ± 046.52*	141.35 ± 047.82*	136.82 ± 050.94*	117.29 ± 046.67
	500	138.98 ± 046.52*	142.15 ± 048.09*	140.04 ± 051.13*	122.04 ± 048.56
Coil[6]	100	144.70 ± 082.72*	146.60 ± 086.62*	152.99 ± 089.37*	114.25 ± 067.82
	200	144.70 ± 082.72*	145.83 ± 086.17*	150.46 ± 087.89*	116.62 ± 069.23
	300	144.70 ± 082.72*	143.94 ± 085.05*	152.53 ± 089.10*	115.89 ± 068.80
	400	144.70 ± 082.72*	141.61 ± 083.68*	160.26 ± 093.62*	113.50 ± 067.38
	500	144.70 ± 082.72*	141.30 ± 083.49*	156.10 ± 091.19*	109.25 ± 064.86
Coil[7]	100	73.20 ± 018.12*	72.49 ± 017.54*	72.08 ± 017.09*	63.44 ± 022.47
	200	73.20 ± 018.12*	72.07 ± 017.43*	72.63 ± 017.22*	59.54 ± 021.09
	300	73.20 ± 018.12*	72.30 ± 017.49*	66.96 ± 015.88*	60.16 ± 021.31
	400	73.20 ± 018.12*	71.57 ± 017.31*	70.18 ± 016.64*	59.56 ± 021.10
	500	73.20 ± 018.12*	71.67 ± 017.34*	71.33 ± 016.92*	59.36 ± 021.03
Forest fires	100	54.00 ± 011.25*	57.12 ± 010.02*	58.02 ± 010.97*	53.94 ± 011.25
	200	54.00 ± 011.25*	56.94 ± 009.98*	55.18 ± 010.43*	53.24 ± 011.11
	300	54.00 ± 011.25*	57.07 ± 010.01*	58.85 ± 011.13*	51.81 ± 010.81
	400	54.00 ± 011.25*	57.80 ± 010.13*	58.15 ± 010.99*	52.56 ± 010.96
	500	54.00 ± 011.25*	57.88 ± 010.15*	57.91 ± 010.95*	53.15 ± 011.09
Housing	100	22.44 ± 003.46*	21.15 ± 003.73*	19.64 ± 002.93*	18.49 ± 002.39
	200	22.44 ± 003.46*	21.21 ± 003.75*	19.02 ± 002.70*	18.60 ± 002.40
	300	22.44 ± 003.46*	19.93 ± 002.81*	19.57 ± 002.59*	18.73 ± 002.42
	400	22.44 ± 003.46*	20.30 ± 003.58*	18.87 ± 002.37*	17.99 ± 002.32
	500	22.44 ± 003.46*	20.43 ± 003.61*	18.95 ± 002.68*	18.13 ± 002.21

(*continued on next page*)

Table 11 (*continued*)

Data sets	VS (%)	SDS	Bagging	SMOTE-NC	PM
Servo	100	71.07 ± 021.98*	65.86 ± 025.20*	67.43 ± 025.27*	64.45 ± 026.51
	200	71.07 ± 021.98*	64.41 ± 022.20*	68.61 ± 025.41*	53.96 ± 015.04
	300	71.07 ± 021.98*	65.55 ± 025.04*	66.59 ± 021.96*	54.50 ± 014.32
	400	71.07 ± 021.98*	68.07 ± 022.78*	67.05 ± 021.81*	60.64 ± 017.48
	500	71.07 ± 021.98*	66.43 ± 022.55*	68.02 ± 022.32*	64.21 ± 021.03
Servo*	100	88.58 ± 030.00*	81.99 ± 023.91*	93.46 ± 034.41*	71.12 ± 024.25
	200	88.58 ± 030.00*	81.60 ± 023.80*	91.72 ± 037.51*	69.46 ± 020.57
	300	88.58 ± 030.00*	81.07 ± 023.64*	76.91 ± 028.08*	64.83 ± 026.57
	400	88.58 ± 030.00*	81.41 ± 023.74*	78.26 ± 027.08*	65.51 ± 024.54
	500	88.58 ± 030.00*	81.55 ± 023.78*	89.04 ± 031.35*	65.80 ± 018.00
Averages	100	83.93 (10.35%)	81.65 (07.84%)	83.35 (09.72%)	75.25
	200	83.93 (12.02%)	81.40 (09.28%)	82.19 (10.15%)	73.85
	300	83.93 (13.27%)	80.88 (09.99%)	80.33 (09.37%)	72.80
	400	83.93 (14.13%)	80.97 (10.98%)	81.50 (11.56%)	72.08
	500	83.93 (13.39%)	81.03 (10.29%)	81.48 (10.77%)	72.70

2 On the whole, the MAPEs of "Bagging" are quite stable as "VS (%)" increases, regardless of the learning algorithm used. Although BP would create different distributions of bootstrapping sets when training sizes are small, the learning procedure in bagging that takes several bags (the default value in WEKA is ten) for prediction aggregation succeeds in making stable predictors.
3 Although "PM" achieves better predictions than "Bagging" and "SMOTE-NC" in the averaged MAPEs, it is notable that "PM" performs worse than "Bagging" in Bike[2], Coil[1], and Coil[2] when models are SVRs, and than "SMOTE-NC" in Bike[1] when models are BPNs, and in Automobile, AutoMPG, Bike[1], and Bike[2] when models are SVRs. This further implies that there is no method which can succeed in performing better than the other approaches when learning with various kinds of data.
4 From the "Averages" shown in Tables 10 and 11, we can find that the smallest averaged MAPEs of "Bagging," "SMOTE-NC," and "PM" appear to "VS (%)" range from 200 to 400. This indicates that there is a limit to the usefulness of virtual samples. Once more samples are added to the training sets, some of the samples would become noises that lowers the preciseness of the identified patterns.

6. Conclusions

The small dataset learning problem has been known about for many years, and presents a challenge to the development of effective forecasting methods since most learning algorithms are built on the assumption that each value has enough representatives in the training sets. In the last decade, numerous VSG algorithms have been proposed for data pre-processing to help with knowledge discovery in databases; however, most of the fuzzy-based VSG approaches are dedicated to dealing with small datasets that only have numerical attributes. In real world situations, nominal inputs such, as gender and color, are commonly contained in most data, and these are hard to deal with using fuzzy-based VSG approaches.

Accordingly, this study proposes a VSG procedure to handle small datasets that contain nominal inputs. Based on the fuzzified relations between each category and continuous output values, new estimated data are derived by the proposed method, where the possibility of the occurrence of combinations can be obtained. Samples created with the proposed method are selected according to their possibility of occurrence to form new training sets. In the experiments, there were sixteen datasets taken from the UCI machine learning repository examined with BPN and SVR. The results show that the forecasting accuracies of the two models are improved with the greatest amount of statistical support when they are built with the proposed training sets. This indicates that the new training sets make the learning models output more precise predictions. Moreover, the results also imply the proposed method outperforms bagging and SMOTE-NC with the greatest amount of statistical support when learning most of the sixteen datasets.

However, further applying the proposed method to classification problems could be difficult, since the fuzzy relations of category values are extracted based on the distributions of continuous outputs. Although nominal outputs can be denoted by numbers, the distributions of the outputs are still limited in distinct intervals, and this would lower the preciseness of the fuzzified relations. One possible way of dealing with nominal outputs is to apply similar operations proposed in this paper to the VDM values as SMOTE-N employs to compute distances, where the distances are the appearance frequencies of category values relative to the class values. Transforming the VDM values into fuzzy relations for extracting additional information, such as the possibility of a combination, could help improve the accurate rates of classifiers, and is thus an interesting direction for future studies.

References

[1] W.S. Gosset, The probable error of a mean, Biometrika 6 (1908) 1–25.
[2] G.Y. Chao, T.I. Tsai, T.J. Lu, H.C. Hsu, B.Y. Bao, W.Y. Wu, M.T. Lin, T.L. Lu, A new approach to prediction of radiotherapy of bladder cancer cells in small dataset analysis, Expert Syst. Appl. 38 (2011) 7963–7969.
[3] C.-J. Huang, H.-F. Wang, H.-J. Chiu, T.-H. Lan, T.-M. Hu, E.-W. Loh, Prediction of the period of psychotic episode in individual schizophrenics by simulation–data construction approach, J. Med. Syst. 34 (2010) 799–808.
[4] D. Tang, N. Zhu, F. Yu, W. Chen, T. Tang, A novel sparse representation method based on virtual samples for face recognition, Neural Comput. Appl. 24 (2014) 513–519.
[5] C. Shao, X. Song, X. Yang, X. Wu, Extended minimum-squared error algorithm for robust face recognition via auxiliary mirror samples, Soft Comput. 20 (2016) 3177–3187.
[6] E. Byon, A.K. Shrivastava, Y. Ding, A classification procedure for highly imbalanced class sizes, IIE Trans. 42 (2010) 288–303.
[7] D.-C. Li, W.-K. Lin, C.-C. Chen, H.-Y. Chen, L.-S. Lin, Rebuilding sample distributions for small dataset learning, Decis. Support Syst. 105 (2018) 66–76.
[8] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Int. Res. 16 (2002) 321–357.
[9] L. Jiang, C. Qiu, C. Li, A novel minority cloning technique for cost-sensitive learning, Int. J. Pattern Recognit. Artif. Intell. 29 (2015) 1551004.
[10] R.V. Krejcie, D.W. Morgan, Determining sample size for research activities, Educ. Psychol. Meas. 30 (1970) 607–610.
[11] S.J. Raudys, A.K. Jain, Small sample size effects in statistical pattern recognition: recommendations for practitioners, IEEE Trans. Pattern Anal. Mach. Intell. 13 (1991) 252–264.
[12] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1986) 81–106.
[13] A. Mirzaei, Y. Mohsenzadeh, H. Sheikhzadeh, Variational relevant sample-feature machine: a fully Bayesian approach for embedded feature selection, Neurocomputing 241 (2017) 181–190.
[14] M.J. Fard, P. Wang, S. Chawla, C.K. Reddy, A Bayesian perspective on early stage event prediction in longitudinal data, IEEE Trans. Knowl. Data Eng. 28 (2016) 3126–3139.

[15] C.-J. Chang, W.-L. Dai, C.-C. Chen, A novel procedure for multimodel development using the grey silhouette coefficient for small-data-set forecasting, J. Oper. Res. Soc. 66 (2015) 1887–1894.
[16] E.A. Sezer, H.A. Nefeslioglu, C. Gokceoglu, An assessment on producing synthetic samples by fuzzy C-means for limited number of data in prediction models, Appl. Soft Comput. 24 (2014) 126–134.
[17] P. Niyogi, F. Girosi, T. Poggio, Incorporating prior information in machine learning by creating virtual examples, Proc. IEEE 86 (1998) 2196–2209.
[18] B. Efron, Computers and the theory of statistics: thinking the unthinkable, SIAM Rev. 21 (1979) 460–480.
[19] J. Błaszczyński, J. Stefanowski, Neighbourhood sampling in bagging for imbalanced data, Neurocomputing 150 (2015) 529–542.
[20] M. Skurichina, R.P. Duin, Bagging, boosting and the random subspace method for linear classifiers, Pattern Anal. Appl. 5 (2002) 121–135.
[21] S. Cost, S. Salzberg, A weighted nearest neighbor algorithm for learning with symbolic features, Mach. Learn. 10 (1993) 57–78.
[22] C. Huang, Principle of information diffusion, Fuzzy Sets Syst. 91 (1997) 69–90.
[23] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.
[24] C.F. Huang, C. Moraga, A diffusion-neural-network for learning from small samples, Int. J. Approx. Reason. 35 (2004) 137–161.
[25] D.-C. Li, W.-K. Lin, L.-S. Lin, C.-C. Chen, W.-T. Huang, The attribute-trend-similarity method to improve learning performance for small datasets, Int. J. Prod. Res. 55 (2017) 1898–1913.
[26] D.C. Li, C.S. Wu, T.I. Tsai, Y.S. Lina, Using mega-trend-diffusion and artificial samples in small data set learning for early flexible manufacturing system scheduling knowledge, Comput. Oper. Res. 34 (2007) 966–982.
[27] J. Domingo-Ferrer, A. Solanas, A measure of variance for hierarchical nominal attributes, Inf. Sci. 178 (2008) 4644–4655.
[28] R. Chi-Wing, J. Pei, A.W.C. Fu, K. Wang, Online skyline analysis with dynamic preferences on nominal attributes, IEEE Trans. Knowl. Data Eng. 21 (2009) 35–49.
[29] D. Coppersmith, S.J. Hong, J.R. Hosking, Partitioning nominal attributes in decision trees, Data Min. Knowl. Discov. 3 (1999) 197–217.
[30] L. Jiang, C. Li, S. Wang, Cost-sensitive Bayesian network classifiers, Pattern Recognit. Lett. 45 (2014) 211–216.
[31] Y. Wang, I. Witten, Inducing model trees for continuous classes, in: M. van Someren, G. Widmer (Eds.), Proceedings of the Poster Papers of the European Conference on Machine Learning, Prague, Czech Republic, 1997, pp. 128–137.
[32] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets Syst. 1 (1978) 3–28.
[33] D. Dubois, H. Prade, S. Sandri, On possibility/probability transformations, in: R. Lowen, M. Roubens (Eds.), Fuzzy Logic. Theory and Decision Library (Series D: System Theory, Knowledge Engineering and Problem Solving), vol 12, Springer, Dordrecht, 1993, pp. 103–112.
[34] D. Dubois, L. Foulloy, G. Mauris, H. Prade, Probability-possibility transformations, triangular fuzzy sets, and probabilistic inequalities, Reliab. Comput. 10 (2004) 273–297.
[35] R. Flage, P. Baraldi, E. Zio, T. Aven, Probability and possibility-based representations of uncertainty in fault tree analysis, Risk Anal. 33 (2013) 121–133.
[36] P. Cortez, A. Morais, A data mining approach to predict forest fires using meteorological data, in: Proceedings of the 13th EPIA 2007 – Portuguese Conference on Artificial Intelligence, Guimaraes, Portugal, 2007, pp. 512–523.

Der-Chiang Li is a Distinguished Professor at the Department of Industrial and Information Management, National Cheng Kung University, Taiwan. He received his Ph.D. degree in the Department of Industrial Engineering at Lamar University Beaumont, Texas, USA, in 1985. As a research professor, his current interests focus on machine learning with small data sets. His articles have appeared in *Decision Support Systems, Omega, Information Sciences, European Journal of Operational Research, Computers & Operations Research, International Journal of Production Research, Neurocomputing*, and other publications.

Hung-Yu Chen is a Ph.D. candidate at the Institute of Information Management, National Cheng Kung University, Taiwan. His current research interests are focused on the learning issue of small datasets. His articles have appeared in *The International Journal of Advanced Manufacturing Technology*.

Qi-Shi Shi is a Ph.D. student in the Department of Industrial and Information Management, National Cheng Kung University, Taiwan. His current research interests are focused on the small dataset learning.