Knowledge-Based Systems 24 (2011) 740–748

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A novel virtual sample generation method based on Gaussian distribution

Jing Yang [a,*], Xu Yu [a], Zhi-Qiang Xie [a,b], Jian-Pei Zhang [a]

[a] College of Computer Science and Technology, Harbin Engineering University, Harbin, China
[b] College of Computer Science and Technology, Harbin University of Science and Technology, Harbin, China

ARTICLE INFO

Article history:
Received 15 January 2010
Received in revised form 20 December 2010
Accepted 25 December 2010
Available online 31 December 2010

Keywords:
Virtual sample
Regularization theory
Cost-sensitive learning
Gaussian distribution
Prior knowledge

ABSTRACT

Traditional machine learning algorithms are not with satisfying generalization ability on noisy, imbalanced, and small sample training set. In this work, a novel virtual sample generation (VSG) method based on Gaussian distribution is proposed. Firstly, the method determines the mean and the standard error of Gaussian distribution. Then, virtual samples can be generated by such Gaussian distribution. Finally, a new training set is constructed by adding the virtual samples to the original training set. This work has shown that training on the new training set is equivalent to a form of regularization regarding small sample problems, or cost-sensitive learning regarding imbalanced sample problems. Experiments show that given a suitable number of virtual sample replicates, the generalization ability of the classifiers on the new training sets can be better than that on the original training sets.

1. Introduction

Classification is one of the most active fields of data mining. For many years, researchers in areas including machine learning, pattern recognition, and statistics are contributing to this field, and have proposed many classification methods, such as neural network [2,7,25], support vector machine (SVM) [17,19,20], and decision tree [27]. Practice shows that these technologies are all with good generalization ability if the training samples are sufficient. But the classification problem is far from being solved, because traditional classification technologies are not able to process the noisy, imbalanced, and small sample data set effectively.

Since noise is inevitable, how to learn and improve the performance of the classifiers under noise is an important problem. If the training set is large enough, the influence caused by noise is little, and the current learning algorithms can find a good classification rule to avoid over-fitting the data. However, in many real-world problems, not enough samples can be used, and moreover the sample set is always imbalanced, so over-fitting and generalization problem arise easily. Thus it is important and urgent to improve the learning ability of classifiers on a noisy, imbalanced, and small sample data set.

The rest of this paper is organized as follows. Section 2 introduces the research status of learning algorithms on noisy, imbalanced, and small sample data set. Section 3 presents a novel virtual sample generation (VSG) method based on Gaussian distribution. Section 4 explores the reasons why it can work. Section 5 reports on experiments. Section 6 summaries the main contribution of this paper and discusses the issues related to the proposed method.

2. Research status

It is a general perception that more data usually provide more information to a training system and can make the classifiers reach higher learning accuracy. However, researchers have tried to find effective ways to acquire knowledge from noisy, imbalanced, and small sample data set when a large data set is unavailable. Some of the related studies are introduced in the following.

2.1. Previous VSG methods

The concept of virtual samples was first introduced by Poggio and Vetter [14], and has been applied in many fields. Usually, virtual sample can be seen as additional training samples created from the current set of examples by utilizing specific knowledge about the task at hand. The idea of virtual samples is a possible way of incorporating prior information in classification learning problems.

From the introduction of virtual sample, many VSG methods have been proposed in different machine learning research fields. Broadly speaking, VSG methods can be classified into two categories by their generation thought. One is to generate virtual samples by extracting the nontrivial prior knowledge hidden in the

[*] Corresponding author. Address: M-2-1503, Culture Home, 258 NanTong Street, NanGang District, Harbin City, 150001 Heilongjiang Province, China. Tel.: +86 13624508906; fax: +86 0451 82519602.
E-mail address: yangjing@hrbeu.edu.cn (J. Yang).

0950-7051/$ - see front matter © 2011 Published by Elsevier B.V.
doi:10.1016/j.knosys.2010.12.010

question being solved. The other is to generate virtual samples by the idea of perturbing the original samples. Detailed descriptions are given in the following.

2.1.1. Generating virtual samples by extracting the nontrivial prior knowledge

Examples of this category first came from Poggio and Vetter. They created virtual samples by the application of prior knowledge to improve recognition ability in the field of pattern recognition. The method is, given a 3D view of an object, to create new images from any other angles through mathematical transformations. The new images generated are called virtual samples. In most real-world pattern recognition fields, extracting the prior knowledge and creating virtual samples are highly nontrivial. Wen et al. [23] proposed another VSG method of this category, which generated virtual samples using prototype faces. Besides the application in image recognition, VSG methods of this category have also been applied in many other fields, such as handwritten number recognition [15], text recognition [16,22], and noise source recognition [24].

2.1.2. Generating virtual samples by the idea of perturbing the original samples

Several noise replication methods can be considered as this category. Among them, Lee [10] proposed a method, which generated virtual samples by adding small normal noise to the original samples. But he did not give an approach to determine the parameters of Gaussian distribution. Moreover, he did not give any theoretical analysis to prove the effectiveness of the method. Li and Fang [11] proposed a non-linear VSG method, which combined a unique group discovery technique with a VSG method. Wang and Yang [21] proposed a perturbance-based VSG method, which added a small constant to every dimension of the p-dimensional training sample. Thus every training sample can generate p virtual samples. Zhang and Chen [26] introduced another VSG method, which firstly divided the training samples of the rare class into p groups by k-nearest-neighbor algorithm, then generated virtual samples by averaging every two samples of each group, and finally kept the labels unchanged.

As is mentioned above, the first category is to generate virtual samples by extracting the nontrivial prior knowledge, so the rationality can be assured, but the adaptability is very low. The second category is to generate virtual samples by perturbing, so the adaptability can be assured, but the rationality is very low. Thus the previous VSG method is either with a low rationality or with a low adaptability. Detailed definitions on rationality and adaptability are given in Section 3.

2.2. The addition of noise to the input data

The addition of noise to the input data can lead to improvements in generalization performance. Bishop [3] and An [1] have proven that training with noise is equivalent to Tikhonov regularization [18] if the standard deviation of the noise is little. Bishop [3] also found that the coefficient of the regularization was related with the standard deviation of the noise. This method can improve the generalization ability of the learning methods to some degree, but it cannot expand the sample set effectively. Thus the generalization ability improved by this method is limited.

2.3. Regularization theory

Regularization theory is another approach to solve over-fitting and generalization problem on small sample problems. One of the central issues in classification is to determine the optimal degree of complexity for the model. A model which is too limited will not capture enough of the structure in the data, while one which is too complex will model the noise on the data (the phenomenon of over-fitting). In either case the performance on new data, that is the ability of the classification to generalize, will be poor [3].

One technique that is often used to control the over-fitting phenomenon in such cases is that of regularization, which involves adding a penalty term to the error function. The technique of regularization makes use of a relatively flexible model, and then controls the variance by modifying the error function by the addition of a penalty term $\Omega(y)$. Thus the total error function becomes

$$E_t = E + \lambda\Omega(y) \tag{1}$$

where λ is a positive number that is usually called the regularization coefficient and $\Omega(y)$ is a cost function that constrains the space of possible solutions according to some form of prior knowledge. In effect λ now controls the effective complexity of the model and hence determines the degree of over-fitting. Although the introduction of regularization terms can control over-fitting for models with many parameters, this raises the question of how to determine a suitable value for the regularization coefficient λ.

2.4. Cost-sensitive learning

Cost-sensitive learning is an effective approach to solve over-fitting and generalization problem on imbalanced sample problems. The class imbalanced datasets occur in many real-world applications where the class distributions of data are highly imbalanced. For imbalanced sample problems, traditional classifiers usually lead to a poor learning accuracy on the rare class. One fundamental assumption in the traditional classifiers is that the goal of the classifiers is to maximize the accuracy. Under this assumption, in the case of imbalanced sample problems, predicting everything as the prevalent class is often the right thing to do. But this always leads to a poor effect for the rare class. Thus a natural thought to solve this question is to raise the misclassification cost of the rare class. For example, the misclassification cost of the rare class can be set to 10 times that of the prevalent class. Cost-sensitive learning is a type of learning in data mining that takes the misclassification costs into consideration. Thus, it is an effective approach to solve the imbalanced sample problems.

The goal of this type of learning is to minimize the total cost. The key difference between cost-sensitive learning and cost-insensitive learning is that cost-sensitive learning treats the different misclassifications differently. Cost-insensitive learning does not take the misclassification costs into consideration. The goal of this type of learning is to pursue a high accuracy of classifying examples into a set of known classes. Detailed descriptions on cost-sensitive learning can be found in Kukar and Kononenko [9] and Ling and Sheng [13].

3. VSG method based on Gaussian distribution

As is mentioned above, the previous VSG methods are hard to take both rationality and adaptability into consideration. In this section, a novel VSG method based on Gaussian distribution is proposed, generating virtual samples by utilizing the most common prior knowledge, smoothness. Theoretical justification for this method will be provided in Section 4.

3.1. Relevant definitions and theories

As Poggio and Vetter did not give an explicit definition on virtual sample, this paper made a summary of this concept and gave an explicit definition by the following. Along with the emergence of virtual sample, many VSG methods were proposed. But so far no indexes have been given to evaluate the performance of the generation methods, so this paper also introduces two indexes qualitatively.

3.1.1. Virtual samples

Let $e = (x,f(x))$ be a random training sample, where $x \in R_n$, $f(x) \in \{-1,1\}$. By application of prior knowledge K, a transform (T,y_T) can be defined, and then a new sample $(Tx,y_Tf(x))$ of the original sample e can be generated. The relation of y_T to T depends upon the prior knowledge of the problem and may be quite complex. The new samples generated are called virtual samples. So for a given training data set $D = \{(x_1,y_1), \ldots, (x_n,y_n)\}$, a virtual sample data set, $D' = \{(x'_1, y'_1), \ldots, (x'_n, y'_n)\}$, can be generated by a right transform T, where $x'_i = Tx_i$, $y'_i = y_T(y_i)$.

3.1.2. Rationality

Rationality represents the likelihood that the virtual samples generated by the VSG method belong to the real feature space. That is to say if most of the virtual samples generated by a VSG method are real samples, then the rationality of the method is high, otherwise the rationality is low.

3.1.3. Adaptability

Adaptability is another performance index, which represents the universality of the research fields, to which the VSG method can be applied. That is to say, if a VSG method is applicable to many different fields, then the adaptability of the method is high, otherwise the adaptability is low.

3.1.4. Gaussian distribution and its relevant property

Gaussian distribution is one of the most important distributions of probability and statistics, introduced by Gauss on his study of error theory. Gauss once used it to depict the error. Experience has shown that many random variables, such as measurement errors, the height of adult males, and reaction time in psychological experiments, can be considered to follow Gaussian distribution. Further theoretical studies have shown that a random variable affected by a large number of tiny, independent random factors will follow Gaussian distribution. The density function of the Gaussian random variable is

$$F(x) = \frac{1}{\sqrt{2\pi}\eta} \int_{-\infty}^{x} e^{-\frac{(y-\mu)^2}{2\eta^2}} dy, \quad -\infty < x < +\infty \tag{2}$$

It can also be written as $N(\mu,\eta^2)$ for short, where μ is the mean, and η is the standard error. According to probability theory, a Gaussian random variable, x, must obey the following formula:

$$P(-3\eta \leqslant x - \mu \leqslant 3\eta) \approx 0.997 \tag{3}$$

This is the 3η principle of Gaussian distribution, which means that the probability of a Gaussian random variable lying within 3η of the mean is close to 1.

3.1.5. Error analysis

As is mentioned above, the measurement error follows Gaussian distribution. Thus the measurement value can be considered to follow Gaussian distribution $N(\mu,\eta^2)$, where μ is the true value of the continuous attribute, η is the standard error associated with the precision of the measurement. For example, if an animal expert finds a new kind dragonfly, and he sends several students to measure the length of the wings. Since error is inevitable, so the measurement results must be different. Suppose that only observation error exists and the precision of the measurement tool is millimeter. So the measurement results can be accurate to millimeter and recorded to the next digit of millimeter by observing. Although the measurement results are different, it is obvious that all of them must be labeled as dragonfly.

3.2. The generation thought of VSG method based on Gaussian distribution

As is mentioned in Section 2.1, it is hard for the previous VSG method to take both rationality and adaptability into consideration. A good way to balance the trade-off between adaptability and rationality is to extract smoothness, which ensures that if two inputs are close, the two corresponding outputs are also close. According to smoothness, this paper proposed a novel VSG method based on Gaussian distribution (VSGGD). Our basic thought is to generate several Gaussian random numbers around a certain original sample, and keep the label unchanged.

3.3. Parameter estimation

So far, two questions following remain unsolved if we want to generate virtual samples by Gaussian distribution,

(1) How to determine the mean of Gaussian distribution, μ.
(2) How to determine the standard error of Gaussian distribution, η.

As described before, our basic thought is to generate Gaussian random numbers around an original sample, so how to determine the mean and the variance is important. The size of the two parameters is quite related to the authenticity of the virtual samples. In this paper, we proposed a method based on error theory to determine the mean and the standard error. As is illustrated by the example in Section 3.1.5, the measurement value is actually a random number following Gaussian distribution $N(\mu_1, \eta_1^2)$, where μ_1 represents the true value of the attribute, η_1^2 is related with the precision of the measurement. So if we generate virtual sample from these random numbers, the authenticity can be assured with a large probability. That is to say, we can set $\mu = \mu_1$, $\eta = \eta_1$.

However, we do not know the true value μ_1, as there must be a certain amount of error during the measurement process. Thus the attribute value x_i can be considered approximately as the true value. That is to say, we can set $\mu = x_i$. The standard error can be determined by the concrete process of data acquisition. Let ε denote the error limit of a sample, which is corrected to a decimal places. According to the error theory, as to the rounding error, we can obtain that $\varepsilon = \frac{1}{2}10^{-a}$. Thus if we generate virtual samples within ε of x_i, the labels will keep unchanged with a large probability. According to the 3η principle of Gaussian distribution, almost all the random numbers generated by Gaussian distribution lie within 3η of the mean. So 3η can be considered to be equal to ε, that is to say, η is equal to one-third of ε. If the attribute value is obtained by observing and measuring, the error will be mainly affected by the instrument precision and personal habits. Likewise, η can be determined.

3.4. Generation process of the proposed VSG method

Let $\{x_1,\ldots,x_n,x_{n+1},\ldots,x_k\}$ be a k-dimensional sample R in the training data set, the first n attributes be continuous, and the last $k - n$ attributes be discrete. The detailed procedure to generate m virtual samples can be depicted as follows:

(1) With respect to the first n continuous attributes, m random numbers for each attribute can be generated by Gaussian distribution $N(\mu,\eta^2)$, where μ and η can be determined by the method described before.

(2) With respect to the discrete attribute, keep the values unchanged. In order to be consistent with the process of continuous attributes, we can also generate m random numbers by Gaussian distribution $N(\mu, \eta^2)$, where η^2 is zero.

In fact, we take another equivalent way to generate the random numbers. That is to generate random numbers by Gaussian distribution $N(0, \eta^2)$ firstly, and then add the random numbers to μ. Thus the error function in Section 4 will be expressed more easily. So for each attribute value x_i,m random numbers can be obtained by computing ($x_i + \xi_{ij}$, $i = 1,\ldots,k$; $j = 1,\ldots,m$), where ξ_{ij} is a small noise following the distribution of $N(0, \eta_i^2)$. Finally the m k-dimensional virtual samples can be generated easily by combining the random numbers of each attribute.

3.5. VSGGD algorithm and the corresponding classification algorithm

Firstly, we propose two VSG algorithms, VSGGDS and VSGGDI, denoting the VSG method based on Gaussian distribution on small sample problems and on imbalance sample problems, respectively. Then we propose the corresponding classification algorithm CVSGGD. The pseudo-codes of VSGGDS, VSGGDI and CVSGGD are shown in Tables 1–3, respectively. In order to express it briefly, suppose that the question to be solved contains only two classes, Class 0 and Class 1. For imbalanced sample problems, suppose Class 0 is the rare class. The multi-class case is similar to the two-class case.

Table 1
The VSGGDS algorithm.

```
Input: the number of virtual sample replicates $m$. The original training set,
    $T = \{(x_i,0), i = 1, \ldots, n_0\} \bigcup \{(x_i,1), i = 1, \ldots, n_1\}$
Output: the generated virtual sample data set $T_v$
Process:
$q = 1$; $R = 0$;      /*$R$ is initialized by a $m * k$ zero matrix*/
for $i = 1$ to $n_0 + n_1${
  for $j = 1$ to $k${
    $r_j = Random(0, \eta_j^2, m) + u_j * Ones(m, 1)$;        /*generate a
          $m$-dimensional random column vector*/
    $R = Adding(R,j,r_j)$;        /*add $r_j$ to the $j$th column of $R$*/
  }
  for $p = 1$ to $m${
    $T_v[q] = Row(R,p)$;        /*take the $p$th row as a virtual sample*/
    $q++$;
  }
}
```

Table 2
The VSGGDI algorithm.

```
Input: the number of virtual sample replicates $m$. The original training set,
    $T = \{(x_i,0), i = 1,\ldots, n_0\} \bigcup \{(x_i,1), i = 1, \ldots, n_1\}$
Output: the generated virtual sample data set $T_v$
Process:
$q = 1$; $R = 0$;        /*$R$ is initialized by a $m * k$ zero matrix*/
for $i = 1$ to $n_0 + n_1${
  if ($y_i = 0$){
    for $j = 1$ to $k${
      $r_j = Random(0, \eta_j^2, m) + u_j * Ones(m, 1)$;      /*generate a
          $m$-dimensional random column vector*/
      $R = Adding(R,j,r_j)$;      /*add $r_j$ to the $j$th column of $R$*/
    }
    for $p = 1$ to $m${
      $T_v[q] = Row(R,p)$;      /*take the $p$th row as a virtual sample*/
      $q++$;
    }
  }
}
```

Table 3
The CVSGGD algorithm.

```
Input: the number of virtual sample replicates $m$. The original training set,
    $T = \{(x_i,0), i = 1, \ldots, n_0\} \bigcup \{(x_i,1), i = 1, \ldots, n_1\}$
Output: classification function $f$
Process:
$T_v = VSGGD(T)$;        /*generate virtual samples from the original training
    set*/
$T^* = T \bigcup T_v$;        /*generate the new training set by the addition of virtual
    samples*/
$f = F(T^*)$;        /*$f$ denotes the function implemented by a classification
    model $F$ trained on the new training set $T^*$*/
```

4. The theoretical proof on why CVSGGD works

As is discussed in Section 3, the VSG method proposed is to generate virtual samples by utilizing smoothness. Since smoothness is the most common form of prior knowledge, so the adaptability of the method is strengthened. Moreover, this method can generate replicas around the original sample, so according to smoothness, the rationality can also be improved. In this section, we will explore why CVSGGD works even in some classification fields where smoothness does not hold.

4.1. Small sample problem

Let $y(x, w)$ denote a certain kind of classification model. Suppose we choose the sum-of-squares error as an error function, then the expected risk can be expressed as

$$E = \frac{1}{2}\int\int \|y(x,w) - t\|^2 p(x,t)dx\,dt$$
$$= \frac{1}{2}\sum_k \int\int \{y_k(x,w) - t_k\}^2 p(t_k|x)p(x)dx\,dt_k \tag{4}$$

where $\|\cdots\|$ denotes the Euclidean distance, and k denotes the number of the class. The function $p(x,t)$ represents the probability density of the data in the joint input-target space, $p(t_k|x)$ denotes the conditional density for t_k given the value of x, and $p(x)$ denotes the unconditional density of x.

4.1.1. Training on the original data set

If we train directly on the original training set, we need to minimize the empirical risk

$$E = \frac{1}{2n}\sum_{i=1}^{n} \|y(x_i,w) - t\|^2 = \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{k}\{y_k(x_i,w) - t_k\}^2 \tag{5}$$

where n denotes the number of the original training samples. As n is too small, the direct training can lead to over-fitting easily. So the learning accuracy on small sample problems is usually poor.

4.1.2. Training on the new data set

Firstly, suppose we can generate m virtual samples for every sample in the original feature space by VSGGDS algorithm. Then train on the new data set composed by the original data set and m virtual sample data sets. In this case, the error function is

$$E_t^1 = E + mE_v \tag{6}$$

where

$$E_v = \frac{1}{2}\sum_k \int\int\int \{y_k(x+\xi,w) - t_k\}^2 p(t_k|x)p(x)p(\xi)dx\,dt_k\,d\xi \tag{7}$$

E_t^1 denotes the error function on the new data set, E denotes the error function on the original data set, and E_v denotes the error function on one virtual sample data set. In formula (7), ξ is a random vector with a same dimension size as x, each dimension of which is a random variable following $N(0,\eta_i^2)$, and $p(\xi)$ denotes the probability density function of ξ.

In order to represent E_t^1 by E, we expand the classification function as a Taylor series,

$$y_k(x+\xi) = y_k(x) + \sum_i \xi_i \frac{\partial y_k}{\partial x_i} + \frac{1}{2}\sum_i \sum_j \xi_i \xi_j \frac{\partial^2 y_k}{\partial x_i \partial x_j} + O(\xi^3) \tag{8}$$

where ξ_i is the ith dimension random variable of ξ. As is discussed in Section 3.3, the variance of ξ_i is small enough, so it is valid to neglect the higher order terms in the Taylor expansion.

As the random vector ξ is independent between different dimensions, and ξ_i follows Gaussian distribution $N(0,\eta_i^2)$. Thus we have

$$\begin{aligned} &\int \xi_i p(\xi) d\xi = 0 \quad \int \xi_i \xi_j p(\xi) d\xi = \eta_i^2 \delta_{ij} \\ &\int \xi_i^3 p(\xi) d\xi = 0 \quad \int \xi_i^4 p(\xi) d\xi = 3\eta_i^4 \end{aligned} \tag{9}$$

where δ_{ij} denotes the usual Kronecker delta symbol, in other words, $\delta_{ij} = 1$ if $i = j$, and $\delta_{ij} = 0$ otherwise.

Substituting the Taylor series expansion into the error function, and integrating over ξ, we obtain

$$E_v = E + E^R \tag{10}$$

where the extra term E^R is given by

$$\begin{aligned} E^R = &\frac{1}{2}\sum_k \int\int \sum_i \eta_i^2 \left\{ \left(\frac{\partial y_k}{\partial x_i}\right)^2 + \{y_k(x,w) - t_k\} \frac{\partial^2 y_k}{\partial x_i^2} \right\} p(t_k|x)p(x)dx\,dt_k \\ &+ \frac{3}{8}\sum_k \int\int \sum_i \eta_i^4 \left(\frac{\partial^2 y_k}{\partial x_i^2}\right)^2 p(t_k|x)p(x)dx\,dt_k \\ &+ \frac{1}{4}\sum_k \int\int \sum_i \sum_{j>i} \eta_i^2 \eta_j^2 \frac{\partial^2 y_k}{\partial x_i^2} \frac{\partial^2 y_k}{\partial x_j^2} p(t_k|x)p(x)dx\,dt_k \\ &+ \frac{1}{2}\sum_k \int\int \sum_i \sum_{j>i} \eta_i^2 \eta_j^2 \left(\frac{\partial^2 y_k}{\partial x_i \partial x_j}\right)^2 p(t_k|x)p(x)dx\,dt_k \end{aligned} \tag{11}$$

So the error function can be written in the following form:

$$E_t^1 = (m+1)E + mE^R \tag{12}$$

As m is a positive number, so minimizing E_t^1 is equivalent to minimizing $\frac{E_t^1}{m+1}$ (i.e., $E + \frac{m}{m+1}E^R$).This has the form of a regularization term added to the usual sum-of-squares error, with the coefficient of the regularizer determined by $\frac{m}{m+1}$. Thus we have proven that training with virtual samples generated by VSGGDS is mathematically equivalent to a form of regularization.

As is mentioned in Section 2.3, a suitable regularization coefficient can avoid over-fitting effectively. Thus if we select a suitable number m, VSGGDS algorithm can solve the over-fitting and generalization problem effectively. So the addition of virtual samples generated by VSGGDS leads to the same result that one gets from the more principled and sophisticated approach of regularization theory.

4.2. Imbalanced sample problem

In order to express it briefly, suppose the question to be solved contains only two classes, Class 0 and Class 1, and Class 0 is the rare class. The multi-class case is similar to the two-class case. In this case, the expected risk can be expressed as

$$\begin{aligned} E &= \frac{1}{2}\int\int \|y(x,w) - t\|^2 p(x,t)dx\,dt \\ &= \frac{1}{2}\sum_{k=0}^{1} \int\int \{y_k(x,w) - t_k\}^2 p(t_k|x)p(x)dx\,dt_k \end{aligned} \tag{13}$$

4.2.1. Training on the original data set

Let $\{(x_i,0),\ i = 1,\ldots,n_0\} \bigcup \{(x_i,1),\ i = 1,\ldots,n_1\}$ be the original training set, T. Let $n_1 \gg n_0$, that is to say, the size of Class 1 is much larger than the size of Class 0. Similar to the case of small sample problem, if we train directly on the original data set, we need to minimize the empirical risk

$$E = \frac{1}{2n}\sum_{i=1}^{n} \|y(x_i,w) - t\|^2 = \frac{1}{2n}\sum_{i=1}^{n}\sum_{k=0}^{1}\{y_k(x_i,w) - t_k\}^2 \tag{14}$$

where $n = n_0 + n_1$. As the size of the rare class is too small, over-fitting will arise easily in estimating the label of future observation in the rare class. So the learning accuracy on the rare class cannot be assured.

4.2.2. Training on the new data set

Suppose we can generate m virtual samples for every sample in the rare class by VSGGDI algorithm and then train on the new data set. In this case, the error function is

$$E_t^2 = E + mE_v^0 \tag{15}$$

where

$$E_v^0 = \frac{1}{2}\int\int\int \{y_0(x+\xi,w) - t_0\}^2 p(t_0|x)p(x)p(\xi)dx\,dt_0\,d\xi \tag{16}$$

denoting the error function on the rare class of one virtual sample data set. Let E_0 and E_1 denote the error function on the rare class and on the prevalent class, respectively. Thus we have

$$E = E_0 + E_1 \tag{17}$$

According to formula (7), we can obtain

$$E_v = E_v^0 + E_v^1 \tag{18}$$

where E_v^0 is defined in formula (16), and E_v^1 is given by

$$E_v^1 = \frac{1}{2}\int\int\int \{y_1(x+\xi,w) - t_1\}^2 p(t_1|x)p(x)p(\xi)dx\,dt_1\,d\xi \tag{19}$$

denoting the error function on the prevalent class of one virtual sample data set. According to formula (11), we have

$$E^R = E_0^R + E_1^R \tag{20}$$

where

$$\begin{aligned} E_0^R = &\frac{1}{2}\int\int \sum_i \eta_i^2 \left\{ \left(\frac{\partial y_0}{\partial x_i}\right)^2 + \{y_0(x,w) - t_0\} \frac{\partial^2 y_0}{\partial x_i^2} \right\} p(t_0|x)p(x)dx\,dt_0 \\ &+ \frac{3}{8}\int\int \sum_i \eta_i^4 \left(\frac{\partial^2 y_0}{\partial x_i^2}\right)^2 p(t_0|x)p(x)dx\,dt_0 \\ &+ \frac{1}{4}\int\int \sum_i \sum_{j>i} \eta_i^2 \eta_j^2 \frac{\partial^2 y_0}{\partial x_i^2} \frac{\partial^2 y_0}{\partial x_j^2} p(t_0|x)p(x)dx\,dt_0 \\ &+ \frac{1}{2}\int\int \sum_i \sum_{j>i} \eta_i^2 \eta_j^2 \left(\frac{\partial^2 y_0}{\partial x_i \partial x_j}\right)^2 p(t_0|x)p(x)dx\,dt_0 \end{aligned} \tag{21}$$

$$\begin{aligned} E_1^R = &\frac{1}{2}\int\int \sum_i \eta_i^2 \left\{ \left(\frac{\partial y_1}{\partial x_i}\right)^2 + \{y_1(x,w) - t_1\} \frac{\partial^2 y_1}{\partial x_i^2} \right\} p(t_1|x)p(x)dx\,dt_1 \\ &+ \frac{3}{8}\int\int \sum_i \eta_i^4 \left(\frac{\partial^2 y_1}{\partial x_i^2}\right)^2 p(t_1|x)p(x)dx\,dt_1 \\ &+ \frac{1}{4}\int\int \sum_i \sum_{j>i} \eta_i^2 \eta_j^2 \frac{\partial^2 y_1}{\partial x_i^2} \frac{\partial^2 y_1}{\partial x_j^2} p(t_1|x)p(x)dx\,dt_1 \\ &+ \frac{1}{2}\int\int \sum_i \sum_{j>i} \eta_i^2 \eta_j^2 \left(\frac{\partial^2 y_1}{\partial x_i \partial x_j}\right)^2 p(t_1|x)p(x)dx\,dt_1 \end{aligned} \tag{22}$$

According to formulas (10), (17), (18), (20), we obtain that

$$E_v^0 = E_0 + E_0^R \tag{23}$$

According to formulas (15), (17), (23), the error function can be written in the following form:

$$E_t^2 = (m+1)E_0 + E_1 + mE_0^R \tag{24}$$

As we have discussed in Section 3.3, η_i^2 is usually small enough, so the neglect of the third term in formula (24) is valid. Thus the error function can be approximately written in the following form

$$E_t^2 \approx (m+1)E_0 + E_1 \tag{25}$$

Since the misclassification cost of the rare class is $m + 1$ times that of the prevalent class, the learning accuracy on the rare class can be improved. Thus we have proven that training with virtual samples generated by VSGGDI is mathematically equivalent to cost-sensitive learning. As is discussed in Section 2.4, a suitable number m can avoid over-fitting on the rare class effectively. According to the practical requirements, we can select a suitable number m, which can control the trade-off of learning accuracy between the rare class and the prevalent class.

5. Experiments

In this section, four experiments are performed on three data sets to demonstrate that the generalization ability of the classifiers can be improved on a training set preprocessed by VSGGD algorithm. The three data sets are iris data set and sonar data set from the UCI [4] Machine Learning Repository, and intrusion detection data set from KDD CUP 99.

5.1. Classification on iris data set

Iris data set is perhaps the best known data set to be found in the pattern recognition field. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. Each sample consists of four attributes, namely, sepal length, sepal width, petal length, and petal width. The training set and testing set in this experiment are chosen and preprocessed as follows:

(1) Select 10 samples randomly as training set from each class of iris data set, so the total number of training samples is 30. Select 60 samples randomly as testing set from the remaining 120 samples. This data set is labeled as data set I.
(2) With respect to data set I, reduce the sample number of each class from 10 to 2, so the total number of training samples is 6. Keep the testing set unchanged. This data set is labeled as data set II.
(3) Generate virtual samples by VSGGDS algorithm from data set II, and set $m = 4$. So the total number of the training samples also increases to 30, and each class contains 10 samples, including 8 virtual samples. The testing set remains unchanged. This data set is labeled as data set III.

We select C-SVC as the classification algorithm [5], where C is a penalty factor. Since radial basis function (RBF) has a good adaptability on non-linear, and high dimensional data set [4], we select RBF as kernel function:

$$K(x,y) = exp\left(\frac{-\|x-y\|^2}{2\sigma^2}\right) \tag{26}$$

where σ is a width parameter, x and y are n-dimensional vectors in the original feature space.

As this experiment is a multi-classification problem, we select one-against-all (1–v–r) approach, which transforms a c-class problem into c two-class problems. In this experiment, the best σ and C are obtained by 10-fold cross-validation [8]. The classification results are compared on data set I, II, III. Five runs of 10-fold cross-validation are performed on each data set, and the average results are reported in Table 4.

As shown in Table 4, when the sample number of each category is 10, the learning accuracy is comparatively high, but if the number is reduced to 2, the classification effect is not very satisfactory. The main reason is that the size of the training set is so small that the generalization ability is affected greatly. On data set III, by the addition of virtual samples, the classification effect has been improved obviously as compared with data set II. So the generalization ability of the classifiers can be improved effectively on a training set preprocessed by VSGGDS algorithm.

5.2. Classification on intrusion detection data set

5.2.1. Experimental data

The intrusion detection data set in this experiment is kddcup_10_per from KDD CUP 99, containing 494021 records. Each record consists of 41 feature attributes, among which 34 attributes are continuous, and the remaining 7 attributes are discrete. The data set contains 23 classes, among which “Normal” is normal network behavior, and the other 22 classes (such as “Back”, “Neptune”, “Smurf”, etc.) are intrusion behaviors. In this experiment, we map the 23 classes to 5 types, namely, Normal, Dos, R2L, U2R and Probing [6]. The distribution of different types is shown in Table 5.

5.2.2. Data set construction

The training set and testing set in this experiment are chosen and preprocessed as follows:

(1) Select 11,500 samples from kddcup_10_per. As the sample number of U2R in kddcup_10_per is only 52, so the selected data set includes all the records of U2R, among which 30 records are selected randomly as training samples. For R2L, we select 698 records randomly, among which 300 records are training samples. For Probing, we select 1400 records randomly, among which 600 records are training samples. For the remaining two classes, we set the ratio of the record number selected from Normal and Dos equal to the ratio of their total record number. Thus we select 1860 records of Normal randomly, among which 900 records are training samples, and we select 7490 records of Dos randomly, among which 3700 records are training samples. Thus all

Table 4
The precision of the SVM algorithm on three different data sets.

Training set	Parameters		Training set size	Testing set size	Precision (%)
	C	σ			
I	32768	31.6	30	60	96.7
II	512	8.0	6	60	73.3
III	32	1.0	30	60	91.7

Table 5
The distribution of different types.

Type	Size	Percentage (%)
Normal	97278	19.69
Dos	391458	79.24
Probing	4107	0.83
R2L	1126	0.23
U2R	52	0.01

Table 6
The distribution of data set i.

Type	Training set size	Testing set size
Normal	900	960
Dos	3700	3790
Probing	600	800
R2L	300	398
U2R	30	22

the 11500 records are divided into 5530 training samples and 5970 testing samples. This data set is labeled as data set i. The distribution is shown in Table 6.

(2) With respect to data set i, for U2R and R2L of the training set, firstly generate a certain amount of virtual samples by VSGGDI algorithm. Then add the virtual samples into the original training set to generate a new training set. The detailed procedure is that we set $m = 10$ for U2R, so the total number of training samples for U2R is 330, including 300 virtual samples. We set $m = 3$ for R2L, so the total number of training samples for R2L is 1200, including 900 virtual samples. So the number of the training set increases to 6730. Keep the testing set unchanged. This data set is labeled as data set ii.

(3) With respect to data set i, for U2R and R2L of the training set, firstly generate virtual samples by the method of Zhang and Chen [26]. Then add the virtual samples into the original training set. The testing set remains unchanged. This data set is labeled as data set iii.

5.2.3. Conversion from character attributes to numerical attributes

The data set consists of four character attributes. Since SVM algorithm only accepts numerical vectors, the character attribute values need to be converted into numerical attribute values. The detailed method can refer to Dong [6].

5.2.4. Data normalization

Since the value range of each attribute in the original data set is different, the data need to be normalized. We would like to map the continuous attribute value to the range [0.0, 1.0] by computing

$$V = \frac{v - min(f_i)}{max(f_i) - min(f_i)} \tag{27}$$

where V is the attribute value after normalization, v is the attribute value of original data, and $min(f_i)$, $max(f_i)$ are the minimum and maximum values of an attribute, f_i, respectively.

5.2.5. The experimental results

Two performance indexes referred in this experiment are shown by the following.

$$\begin{aligned} DR &= d_1/D_1 \\ FR &= d_2/D_2 \end{aligned} \tag{28}$$

where DR is detection rate, and FR is false alarm rate. $D1$ is the total number of abnormal samples, $d1$ is the number of abnormal samples detected, $D2$ is the total number of normal samples, and $d2$ is the number of the misclassification normal samples.

For data set i, we select FMSVM algorithm [12] as the classification algorithm. For data set ii, iii, we select C-SVC as the classification algorithm. As this experiment is a multi-classification problem, so we also select one-against-all (1–v–r) approach. The experiment also selects RBF (defined in formula (26)) as a kernel function. In this experiment, the best σ and C are also obtained by 10-fold cross-validation. The classification results are compared on data set i, ii, iii. Five runs of 10-fold cross validation are performed on each data set, and the average results are reported in Table 7.

Table 7
The experimental results on three different data sets.

Sample set	Experimental results (%)			
	Dos	Probe	U2R	R2L

i	DR = 93 FR = 8	DR = 90
		FR = 7
			DR = 33
			FR = 4
				DR = 34
				FR = 4

ii	DR = 93 FR = 9	DR = 82
		FR = 7
			DR = 50
			FR = 4
				DR = 42
				FR = 3

iii	DR = 92
	FR = 10
		DR = 64
		FR = 6
			DR = 35
			FR = 5
				DR = 37
				FR = 4

As is shown in Table 7, for the type of DOS, the classification precision of our method does not have an obvious advantage. For the type of Probe, the classification precision is even worse than the precision of FMSVM algorithm. But for the types of U2R and R2L, our method has an obvious advantage. This shows that, for imbalanced sample problems, although the learning accuracy on the prevalent class may be reduced, the learning accuracy on the rare class can be improved effectively if the training set is preprocessed by VSGGDS algorithm. This can be explained by the truth, proven in Section 4, that training with virtual samples generated by VSGGDI algorithm is mathematically equivalent to cost-sensitive learning. Although a certain amount of virtual samples are also generated by the method of Zhang and Chen [26], the effect of the classification is not improved. This shows that a VSG method based on perturbing is not always effective.

5.3. Sensitive analysis

As is proven in Section 4, training with virtual samples generated by VSGGDS or VSGGDI is mathematically equivalent to a form of regularization or cost-sensitive learning. So the learning accuracy may be influenced largely by the number of virtual sample replicates, m. In this section, we will conduct another 2 experiments on sonar data set to illustrate that the determination of m is important to the finally classification effect. The sonar data set is a stand data set from the UCI Machine Learning Repository, including two classes. Class "mine" contains 111 patterns obtained by bouncing sonar signals off a metal cylinder at various angles and under various conditions, and Class "rock" contains 97 patterns obtained from rocks under similar conditions. Each pattern is a set of 60 numbers in the range 0.0 to 1.0.

5.3.1. Sensitive analysis on VSGGDS algorithm

We select 10 samples randomly from each class as training set, and 30 samples randomly from each class as testing set. So the selected training data set can be considered as a small sample data

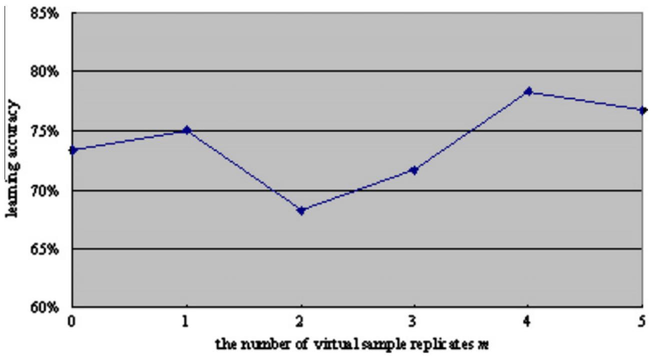

Fig. 1. Classification on the original small data set and on the 5 new data sets.

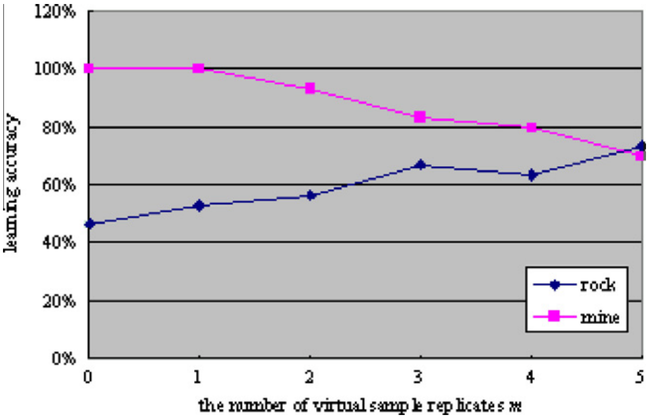

Fig. 2. Classification on the original imbalanced data set and on the 5 new data sets.

set. We generate virtual samples with VSGGDS algorithm from the above data set. Set $m = 1, 2, 3, 4, 5$, and we obtained 5 new data sets, which contain 40, 60, 80, 10, 120 samples, respectively. The experimental method is similar to that introduced in Section 5.1. The results on the original data set and the 5 new data sets are depicted in Fig. 1.

Fig. 1 shows that the learning accuracy may be influenced largely by m. A suitable value m can improve the generalization ability effective, while an inappropriate value m may lead to a poor classification result. In real-world problems, a suitable value m can be obtained by cross-validation, which is an effective way for model selection [8].

5.3.2. Sensitive analysis on VSGGDI algorithm

We select 80 and 10 samples randomly as training data set from "mine" and "rock", respectively and 30 samples randomly from the rest samples of each class as testing set. So the selected training data set can be considered as an imbalanced sample data set, and "rock" is the rare class. We generate virtual samples with VSGGDI algorithm from the above data set. Set $m = 1, 2, 3, 4, 5$, and we obtain 5 new data sets, in which "rock" contains 20, 30, 40, 50, 60 samples, respectively, and the size of "mine" is unchanged. The experimental method is similar to that introduced in Section 5.1. The results on the original data set and the 5 new data sets are depicted in Fig. 2.

Fig. 2 shows that the learning accuracy on different classes is influenced largely by m. The learning accuracy on the rare class is improved with the increase of m, while the learning accuracy on the prevalent class is reduced with the increase of m. It is easy to understand this phenomenon. As is demonstrated in formula (25), m is related with the misclassification cost on imbalanced sample problems. Thus a larger value m means a larger misclassification cost on the rare class. So in order to make the total cost minimal, if m is set to a larger value, the learning accuracy on the rare class will be improved, while the learning accuracy on the prevalent class will be reduced correspondingly.

If we assume that m can be set to infinite, in this case, the learning accuracy on the rare class will be 100%, while the learning accuracy on the rare class will be zero. That is to say, the classifiers will just predict everything as the rare class. In real-world problems, we need to consider both learning accuracy on the two classes, so m should be determined by the practical classification requirements.

6. Conclusion and discussion

In this paper, a novel VSG method based on Gaussian distribution was proposed. Such a method can work well because the original training set may be very small, imbalanced, full of noise and may not capture the whole target distribution. Since the classifiers' learning results can be more accurate on the new training set, the proposed method provides a good choice for classification tasks on noisy, imbalanced, and small sample training set.

Previous VSG methods are hard to take both rationality and adaptability into consideration. Different to them, the proposed method generates virtual samples by utilizing smoothness, so the adaptability is strengthened. Moreover, this method can generate replicas around the original sample. So according to smoothness, in most classification research fields, it can generate relatively correct samples. Thus the rationality can be improved. More importantly, even in research fields where smoothness assumption did not hold, training with virtual samples generated by the proposed method was effective. It had been proven to be mathematically equivalent to a form of regularization regarding small sample problems, or cost-sensitive learning regarding imbalanced sample problems. Therefore this method exhibits a new way to generate virtual samples.

The experiments reported in this paper showed that given a suitable number of virtual sample replicates, the generalization ability of the classifiers on the new training set can be better than that on the original training set. Although in this paper, the mean and the standard error of Gaussian distribution, μ, σ, had been determined, how to determine them for a better effect remains an important problem deserving further study.

Acknowledgments

This project is supported by the National Natural Science Foundation of China (Nos. 60873037, 61073043), the Natural Science Foundation of Heilongjiang Province of China (No. F200901), the China Postdoctoral Science Foundation funded (No. 20090460880), Heilongjiang Province Postdoctoral Science Foundation (No. LBH-Z09214) and Harbin Outstanding Academic Leader Foundation of Heilongjiang Province of China (No. 2010RFXXG054).

References

[1] G. An, The effects of adding noise during backpropagation training on a generalization performance, Neural Computation (1996) 643–674.
[2] R. Andrews, J. Diederich, A.B. Tickle, Survey and critique of techniques for extracting rules from trained artificial neural networks, Knowledge-Based Systems 8 (6) (1995) 373–389.
[3] C.M. Bishop, Training with noise is equivalent to Tikhonov regularization, Neural Computation 7 (1) (1995) 108–116.
[4] C.J.C. Burges, A tutorial on support vector machine for pattern recognition, Data Mining and Knowledge Discovery 2 (2) (1998) 121–167.
[5] C. Cortes, V.N. Vapnik, Support-vector network, Machine Learning 20 (1995) 273–297.
[6] C.Y. Dong, Study of support vector machines and its application in intrusion detection systems, Ph.D.Thesis, Xidian University, China, 2004.
[7] T.S. Hu, X.N. Guo, X. Fu, Y.B. Lv, A neural networks approach for solving linear bilevel programming problem, Knowledge-Based Systems 23 (3) (2010) 239–242.
[8] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: Proceedings of the 14th Joint International Conference on Artificial Intelligence, vol. 2, Montreal, Canada, 1995, pp. 1137–1143.
[9] M. Kukar, I. Kononenko, Cost-sensitive learning with neural networks, in: Proceedings of the 13th European Conference on Artificial Intelligence, Brighton, UK, 1998, pp. 445–449.
[10] S.S. Lee, Noisy replication in skewed binary classification, Computational Statistics & Data Analysis 34 (2) (2000) 165–191.
[11] D.C. Li, Y.H. Fang, A non-linearly virtual sample generation technique using group discovery and parametric equations of hypersphere, Expert Systems with Applications 36 (1) (2009) 844–851.
[12] K.L. Li, H.K. Huang, S.F. Tian, Fuzzy multi-class support vector machine and application in intrusion detection, Chinese Journal of Computers 28 (2) (2005) 274–280.
[13] C.X. Ling, V.S. Sheng, A comparative study of cost-sensitive classifiers, Chinese Journal of Computers 30 (8) (2007) 1203–1211.

[14] T. Poggio, T. Vetter, Recognition and structure from one 2d model view: observations on prototypes, object classes, and symmetries, A. I. Memo 1347, Artificial Intell. Lab., MIT, Cambridge, MA, 1992.

[15] B. Schölkopf, P. Simard, A. Smola, V.N. Vapnik, Prior knowledge in support vector kernels, in: Advances in Neural Information Processing Systems 10, MIT Press, Cambridge, MA, 1998, pp. 640–646.

[16] J.S. Su, B.F. Zhang, X. Xu, Advances in machine learning based text categorization, Journal of Software 17 (9) (2006) 1848–1859.

[17] X.L. Tang, L. Zhuang, J. Cai, C.B. Li, Multi-fault classification based on support vector machine trained by chaos particle swarm optimization, Knowledge-Based Systems 23 (5) (2010) 486–490.

[18] A.N. Tikhonov, V.Y. Arsenin, Solution of ill-posed problems, W.H. Winston, 1977

[19] V.N. Vapnik, An overview of statistical learning theory, IEEE Transactions on Neural Network 10 (5) (1999) 988–999.

[20] S. Wan, T.C. Lei, A knowledge-based decison support system to analyze the debris-flow problems at Chen-Yu-Lan river, Taiwan, Knowledge-Based Systems 22 (8) (2009) 580–588.

[21] W.D. Wang, J.Y. Yang, Quadratic discriminant analysis method based on virtual training samples, Acta Automatica Sinica 34 (4) (2008) 400–407.

[22] X.D. Wang, L. Guo, J. Fang, Research on ontology-driven text virtual sample constructing, Computer Science 35 (3) (2008) 142–145.

[23] J.W. Wen, S.W. Luo, J.L. Zhao, H. Huang, A small sample face recognition statistical learning method based on virtual samples, Journal of Computer Research and Development 39 (7) (2002) 814–818.

[24] R.W. Xu, L. He, L.K. Zhang, Z.Y. Tang, S. Tu, Research on virtual sample based identification of noise source in ribbed cylindrical double-shells, Journal of Vibration and Shock 27 (5) (2008) 32–35.

[25] B. Yu, D.H. Zhu, Combining neural networks and semantic feature space for email classification, Knowledge-Based Systems 22 (5) (2009) 376–381.

[26] L. Zhang, G.H. Chen, Method for constructing training data set in intrusion detection system, Computer Engineering and Applications 42 (28) (2006) 145–146, 180.

[27] Z.H. Zhou, Z.Q. Chen, Hybrid decision tree, Knowledge-Based Systems 15 (8) (2002) 515–528.