Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-020-02665-w

ORIGINAL RESEARCH

Semi-supervised active learning algorithm for SVMs based on QBC and tri-training

Hailong Xu[1] · Longyue Li[1] · Pengsong Guo[1]

Received: 4 August 2020 / Accepted: 3 November 2020

Abstract

For the problem that large-scale labeled samples are not easy to acquire in the course of Support Vector Machines (SVMs) training, a Semi-Supervised Active Learning Algorithm for SVMs (QTB-ASVM) is proposed in the paper, which efficiently combines the semi–supervised learning based on Tri-Training and active learning based on Query By Committee (QBC) with SVMs. With this method, QBC active learning is used to select the samples which are the most valuable to current SVM classifier, and Tri-Training is used to exploit useful information that remains in the unlabeled samples. The experimental results show that the proposed approach can considerably reduce the labeled samples and costs compared to the SVMs which is either not applied with semi-supervised learning or active learning or applied with only one of them, and at the same time it can ensure that the accurate classification performance is kept as the passive SVM, while improving generalization performance and also expediting the SVM training.

Keywords Active learning · Support vector machines · Semi-supervised learning · Query by committee · Tri-training

1 Introduction

Support vector machines (SVMs) (Brefeld et al. 2004; Wang et al. 2017) is a state-of-the-art statistical learning model. This machine learning algorithm has strong theoretical foundations and excellent empirical successes.

Due to its good performance, SVMs has been used successfully in a variety of tasks such as text classification (Mohamed et al. 2018), handwritten digit recognition (Wang et al. 2016), as well as object recognition. However, the classical training algorithms of SVMs are supervised learning approaches which train the classifier from a large number of labeled samples. When supervised machine learning techniques are applied to realistic applications, large amounts of data are available but correctly labeling it to create a training set is costly, tedious, time-consuming and exceptional difficult, whereas acquiring large quantity of unlabeled data is readily available. Meanwhile, these unlabeled data contain a wealth of information which help train the learning machine, and the ignorance of these data will create enormous waste to the data resources. Thus, finding ways to minimize the number of labeled instances and use these unlabeled samples effectively is undoubtedly beneficial while still maintaining desired accuracy of SVMs.

There are two main learning methods, semi-supervised and active learning methods respectively reduce markers in the study sample size from different angles in order to reduce the cost of learning.

The most frequent semi-supervised learning algorithms include Expectation Maximization algorithm (EM), Self-training algorithm, Co-training algorithm, Co-EM algorithm and transductive method and etc. Since Bennett et al. put forward the semi-supervised support vector machine in 1998 (Brefeld Ulf 2004; Xu et al. 2020), scholar's recently focus their research more on its increase, and the scholars had done a lot of researches on it, such as concave semi-supervised support vector machine (CS^3VM) proposed by Glenn Fung et al., transductive Support Vector Machine (TSVM) proposed by Joachims T (Mohamed Goudjil 2018), Co-EM SVM algorithm (Brefeld Ulf 2004) proposed by Brefeld and Scheffer et al., and so on. Although semi-supervised learning algorithms adopt only a small number of labeled samples, their learning results are still not as good as the supervised learning algorithm which based on large amount of data,

✉ Longyue Li
20132143202@mail.sdufe.edu.cn

1 Air and Missile Defense College, Air Force Engineering University, Shaanxi 710051, Xi'an, People's Republic of China

Published online: 25 November 2020

such as the TSVM algorithm in which the distribution ratio of positive and negative samples and the number of positive samples in advance are estimated, so once the incorrect estimates or incorrect sample classification information are used in some steps of algorithm, this kind of error will accumulate gradually with the increase of the number of iterations, and finally affect the accuracy of the model inevitably (Mohamed Goudjil 2018; Wang 2016).

Active learning ideas of machine learning filed can be applied to SVM training learning based on semi-supervised learning for reference. In the process of learning, according to the learning process, the learning machine can actively choose the best samples which include the unlabeled samples for training classifier, and add these samples to the existing training sample set, these newly added samples can considerably improve the soft-classification accuracy of semi-supervised SVM for unlabeled samples, so that it can reduce the incorrect classification information dramatically, and improve the classification accuracy of SVM. However, when the frequent active learning algorithms, such as Error-reduction Sampling, Uncertainty Sampling, and Query-By-Committee, etc. are used in SVM training, the classification accuracy can be improved. However, too many redundant samples are selected in the meantime, and the entire sample space should be searched before each sample selection to determine which sample can be selected. Thus, the learning time are long and computational complexity are high, therefore it's not feasible for SVM training learning which adopts a large number of unlabeled samples.

Experiments and various practical applications prove that the SVM training algorithm which use semi-supervised learning or active learning separately have greatly improved the performance comparing with the classical SVM. So the combination between two learning approaches and SVM training could make full use of their respective advantages for SVM training learning, therefore greatly improve the generalization capability and dramatically reduce the cost of tagging training samples (Gokhan Tur et al. 2005a, b; Seung et al. 1992; Xu et al. 2010; Zhao 2012).

2 SVM semi-supervised active learning

At present, there are few researches on combining semi-supervised learning with active learning to training SVM. There is some existing exploratory study, for example McCallum et al. (1998) combine the improved QBC approach with EM algorithm to achieve the text classification, Muslea et al. (2002) expand the approach of literature (McCallum et al. 1998) and the expanded approach which combines the Co-EM algorithm with active learning, similar to QBC, which is called Co-Testing algorithm or Co-EMT algorithm, the approaches in literature (Wang et al. 2017; Jie et al. 2004) which combine the QBC algorithm with EM algorithm are applied to SVM training learning. Beyond that, there are some approaches which combine the uncertainty- based sampling active learning (or UBS for short) with semi-supervised learning method and these approaches are applied to the speech recognition, the spoken language understanding, and telephone classification, etc. (Gokhan Tur et al.2005a, b; Zhang et al. 2013).

Some algorithms which combine the QBC-based active learning with EM algorithm are proposed in literature (Gokhan Tur et al.2005a, b; Muslea et al.2002; Wang et al. 2017; Jie et al. 2004; Zhang et al. 2013), while reducing the number of tagging samples, such active learning methods based on QBC or UBS are all easy for the selection of singular point samples which are of higher uncertainty, and when sample singularity to the training sample set is added, the classifier error will increase, thus causing error propagation problems. Meanwhile, the Co-EM approach applies the probability label gained from one view to the classifier design of the other view just from a technical point of view, and its rationality is merely a hypothesis too. In essence, Co–EM just applies the EM approach in every view. On the whole, The EM framework is not met and there are no guarantees for the global convergence of algorithm theoretically. The Co-training approach and Co-EM approach based on Co-training set need every view to satisfy both the compatible condition and irrelevant condition, however, these two conditions are difficult to be met in most practical applications, or it is even impossible to give the natural segmentation of feature set. In order to solve this problem, Goldman and Zhou proposed a Co-training algorithm that does not need fully redundant views. However, in order to estimate confidence degree of annotation, in the process of choosing unlabeled samples to be tagged and the classifiers to forecast the label for the unlabeled samples, tenfold-cross-validation are used frequently, which makes it very time-consuming and difficult to estimate the confidence degree stably.

For the above problems of Co-training and its improved algorithms, Zhou et al. (2005) propose the Tri-training algorithm which neither need fully redundant views nor require the use of different types of classifiers. Three classifiers are used in the Tri-training algorithm, while the algorithm is no binding on the set of properties and learning algorithms used by the three classifiers, and tenfold-cross-validation is not used any more. So this approach can not only easily handle the problems of estimating confidence degree of tagging sample and forecasting the label for unlabeled samples, but also can use the ensemble learning to improve the generalization capability. Therefore, the Tri-training algorithm has more extensively applicable scope and is more efficient, and are more suitable for solving the classification problem of sparse labeled sample set (Hoi et al. 2005; Xu et al. 2016).

Based on the above analysis, and the literature (Hoi et al. 2005; McCallum et al. 1998; Wang et al. 2017; Jie et al. 2004; Xu et al. 2016), this paper improves the Tri- training-based semi-supervised learning and QBC-based active learning, and presents a semi-supervised active learning algorithm for SVM which is based on the improved QBC, Tri-training and Co-EM SVM (QBC and Tri-training-Based Active SVM, or QTB-ASVM for short). Section 3 describes the algorithm ideas. Section 4 describes our semi-supervised active learning algorithm for SVM in detail and Sect. 5 presents experiments that demonstrate its applicability and effectiveness, as well as the discussions on results. Finally, Sect. 6 provides suggestions on future work.

3 QTB-ASVM algorithm ideas

A semi-supervised active learning for SVM is an approach in which the semi-supervised learning and active learning are used simultaneously, so the key issue is how to combine the two learning approaches with SVM training effectively.

3.1 QTB-ASVM algorithm flow

SVM active learning generally chooses the samples which are the most uncertain and whose confidence degree of tag is lower for current learning machine to tag and train, and these samples which are relatively certain or fully representative will not be used in training learning. These samples can be used in the semi-supervised learning method, so that it can make full use of the beneficial information of these unlabeled samples to classifier training, this can avoid the error propagation caused by uncertainty of initial classifier in the process of active learning, thus it can improve the performance of the SVM active learning. Based on this, in order to combine these two methods with SVM training effectively, the following Fig. 1 shows the flow diagram of QTB-ASVM which describes how to combine SVM with the two learning approaches in training process.

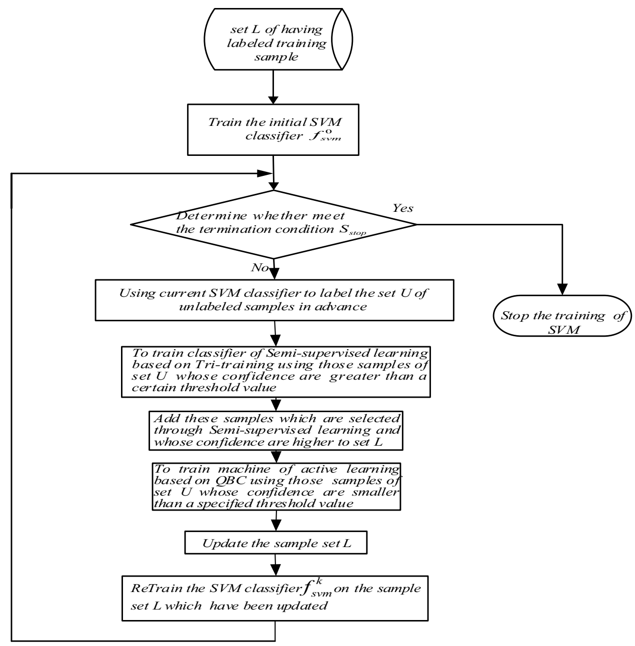

Fig. 1 Flow diagram of QTB-ASVM algorithm

To combine SVM training with the two approaches effectively, as shown in Fig. 1, in the process of QTB-ASVM algorithm, these samples whose confidence degree of tag are higher will be used in the improved Tri–training-based semi-supervised learning, and these samples whose confidence degree are lower, i.e., these samples which are the most "uncertain" for current SVM classifier will be used in the improved QBC-based active learning, while our algorithm will use the three SVM classifiers trained by Tri-training approach as the members of QBC committee. So first of all, it need to be ensured that SVM active learning has good initial classifiers, thus in the iterative process of active learning, it can avoid the error propagation phenomenon caused by the weaker initial classifier and the lower confidence degree of forecaster label comparing with the past active learning algorithms. Meanwhile, through the SVM active learning, the most beneficial and informative samples which are selected in the process of SVM active learning will provide abundant labeled samples for semi-supervised learning of next round.

3.2 SVM model used in training learning

In the initial stage of QTB-ASVM algorithm, there are some problems, such as less labeled samples and uneven distribution of samples, etc. So if the standard SVM model in the training is used, the optimal separating hyperplane will shift to the direction in which some sort of sample of margin region is fewer, this deviation will lead to sampling the repetitive, similar, meaningless and isolated samples in the SVM training learning. In order to solve this problem, when training the initial classifier f_{svm}^{0} of QTB-ASVM algorithm, an improved weight SVM which adopts different weight coefficient for different category or different samples will be used, so as to improve the classification performance.

The original optimization problem of weight SVM is described as following:

$$\begin{cases} min\ \frac{1}{2}\|\omega\|^2 + \mathrm{C}\sum_{\mathrm{i=1}}^{1} \lambda_i \mathrm{s_i} \xi_\mathrm{i} \\ \mathrm{s.t.}\ \mathrm{y_i}(\omega^\mathrm{T}\phi(\mathrm{x_i}) + \mathrm{b}) \geq 1 - \xi_\mathrm{i}, \xi_\mathrm{i} \geq 0,\ i = 1, \cdots, n \end{cases} \tag{1}$$

where s_i is the weight factor of different sample x_i, λ_i is weight factor of sample x_i belonging to class y_i, the other symbols are the same to the parameters of objective function of generalized SVM optimal classification hyperplane.

With the process of semi-supervised active learning for SVM, in order to reflect the contribution of unlabeled samples selected by semi-supervised to SVM training, inspired by Co- EM SVM algorithm (Brefeld Ulf et al. 2004), in the recursive learning, the base classifier of QTB-ASVM algorithm adopts the following improved weight SVM, and the original optimal problem can be described as following:

$$min\ :\ \frac{1}{2}\|\omega\|^2 + C\left(\sum_{i=1}^{n}\lambda_i s_i \xi_i + C_s\sum_{j=n+1}^{n+k} c_j^* \xi_j^*\right) \tag{2}$$

$$s.t:\ \ \forall_{i=1}^{n} y_i\left[(\omega \cdot x_i) + b\right] \geq 1 - \xi_i \tag{3}$$

$$\forall_{j=n+1}^{n+k} \hat{y}_j\left[(\omega \cdot x_j^*) + b\right] \geq 1 - \xi_j^* \tag{4}$$

$$\forall_{i=1}^{n}\xi_i > 0,\ \ \forall_{j=n+1}^{n+k}\xi_j^* > 0 \tag{5}$$

where s_i,c_j^* denotes the different sample weight, λ_i is same to the parameter in Eq. (1),C_s denote contribution degree of unlabeled samples which are selected by semi–supervised learning to the train the classifier. $\hat{y}_j$ is the class label of sample x_j^*, which is tagged in the process of semi-supervised learning, and the rest of the parameters have the same semantics as the formula 1and Co-EM SVM algorithm (Brefeld Ulf et al. 2004).

In the meantime, the weight coefficient of weighted SVM in formula 2 could adopt the following approach to calculate:

Firstly, in the initial stages, make parametric value of class weight λ_i to be inversely proportional to ratio of the positive and negative sample, and weight parameter of each sample to be identical i.e.$s_i = 1/n$, then to construct the initial classifier f_{svm}^0.

Secondly, on the basis of the f_{svm}^0, and within the learning process of QTB-ASVM algorithm and the kth sample process, the classifier f_{svm}^{k-1} will be used to select m samples whose sample distance d is closest to the hyperplane in the sample set U, where the distance d can be easily calculated as follows:

$$d = \frac{f(x)}{\|\omega\|} \tag{6}$$

where in Eq. (6), provided x is a sample, and the $f(x)$ is decision function of SVM, so according to the SVMs theory, it is easy to know that the vertical distance d can be a criterion of class probability to measure x belong to which class, i.e. $f(x)$ can measure the probability of uncertainty of instance.

According to Eq. (6), these samples which are most likely to be support vector samples will be selected as the incremental sample set X_I^k, and after these samples are labeled and added to training sample set. In the new training sample set, those positive and negative samples in the margin of classification using classifier f_{svm}^{k-1} will be selected, and let the weight of these selected samples are greater than those samples which are distributed outside the margin of classifier. Meanwhile, let parametric value of class weight be inversely proportional to ratio of the positive and negative sample in the margin of classification, then the algorithm continues the machine learning procedure of training SVMs, and trains the classifier f_{svm}^k.

In Eq. (2), the value of parameter C_s can be determined in this way. Firstly, the initial value of C_s can be set to a very small number, then let the initial value double in every iterative training procedure, until C_s is equal to 1. The parametric value of s_i and c_j^* can be determined similarly through this method in the Co-EM SVM algorithm, that is, provided class probability of sample x_i predicted respectively by the three SVM basic classifiers of Tri-Training algorithm, which are $P_{svm_1}(y|x_i)$, $P_{svm_2}(y|x_i)$ and $P_{svm_3}(y|x_i)$, then the weight of sample x_i can be set to the difference between the largest and least of the three class probability values, i.e. $\underset{j=1,2,3}{max}\{P_{svm_j}(y|x_i)\} - \underset{j=1,2,3}{min}\{P_{svm_j}(y|x_i)\}$, then all sample's weight is normalized.

3.3 Method of SVM probability output

In our algorithm, the confidence degree of tagging the samples are needed to be calculated, but the decision function of classical SVM is "hard output ", that is, it doesn't yield probability output. Scholars have done the related researches about the probability output for SVM. In order to simplify the calculation and bring no affects to the measurement result at the same time, Eq. (6) can be used (Xu et al. 2010), i.e. the distance between sample and optimal classification hyperplane as the measurement of different class probability, so inspired by the literature (Anshu Singla et al. 2018; Hoi et al. 2005), our algorithm adopts the following Eq. (7) to measure the confidence degree of tag:

$$conf(x_i) = \frac{1}{1 + \exp(-|f(x_i)|/\|\omega\|)} \tag{7}$$

In Eq. (7), the meaning of formula $|f(x_i)|/\|\omega\|$ is the same as the Eq. (6). According to Eq. (7), it is easy to know that $0.5 \leq conf(x_i) \leq 1$, and as the $f(x_i)$ changes, the changing trend of $conf(x_i)$ is shown in Fig. 2.

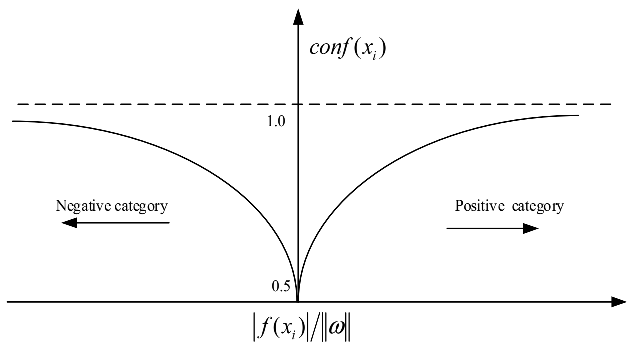

Fig. 2 Changing trend of confidence with the changing of distance d (defined in the Eq. (6))

As shown in Fig. 2, according to the SVMs theory and active learning mentioned previously, we know that the closer the value of $cong(x_i)$ is to 0.5, the more uncertain of labeling sample x_i can be, and the more informative sample is going to be. Meanwhile the tag confidence of sample x_i is correspondingly lower. On the contrary, the closer the value of $cong(x_i)$ is to 1, the more certain the label of sample x_i is positive or negative, i.e. the confidence of sample which belongs to a certain category is higher.

3.4 Semi-supervised learning method based on Tri-training

In the Tri-training algorithm (Zhou et al. 2005), firstly, three training sample assemblies which are labeled can be obtained by Bootstrap Sampling on the labeled sample set, then it can train one classifier on each training sample set. In the process of training, the new labeled samples for one of three classifiers is provided collaboratively by the other two classifiers. Specifically, first through the training on the initial labeled training sample set L, three different initial classifier h_1 、h_2 and h_3 can be obtained. If two of the three classifiers have forecasted the same label for the same sample x, then the sample is considered to have higher tag confidence, and the labeled sample x will be added to the labeled training samples set of the the third classifier. For example, provided h_2 and h_3 are the classifiers and the classification results for unlabeled sample x of sample set U are $h_2(x)$, $h_3(x)$, if $h_2(x)$ is consistent with $h_3(x)$, then the sample x will be marked as $h_2(x)$ or $h_3(x)$ and meanwhile the sample will be added to the training sample set of h_1. By this, it will construct the new training sample set S_1' for h_1, that is $S_1' = L \cup \{x | x \in U \wedge h_2(x) = h_3(x)\}$, similarly, the training sample set for h_2 and h_3 will also be extended respectively. Then on the extended sample set, the three classifiers will be retrained. So repeat the process above till the performance of the three classifiers have no more changes.

When predicting labels for samples, Tri-training algorithm no longer choose one classifier to classify as usual algorithm, but the vote approach which is always used in the ensemble learning is used, i.e. the three classifiers are integrated to predict labels for unknown samples.

In the iterative training process of Tri-training algorithm, since the initial classifier may be "weak", and unlabeled samples may be wrongly tagged, noises might be introduced to the third classifier. For instance, provided that the sample x is unanimously marked as $h_2(x)$ or $h_3(x)$ by classifier h_2 and h_3, and if tag accuracy of the two classifiers are higher enough, the training result of h_1 could be optimized by adding the labeled sample x to training samples set of h_1, otherwise, it will influence the training effect and form the noise of training sample set to which due to the introducing of sample x, "error accumulation" phenomenon in the iterative learning of Tri-training might also be caused along with the learning procedure, which lead to unstable performance and big error of classification. In the meantime, error estimation function of iterative learning algorithm will use a large number of labeled samples, and in the case of lacking labeled samples and labeling samples costly, it will be difficult to provide a large number of tagged samples as test sample set.

To solve the problems mentioned above, combining with a semi-supervised learning theory and the idea of active learning, in the QTB-ASVM algorithm, some improvements based on Tri-training algorithm, are given as follows:

(1) Firstly, according to Eq. (7), the algorithm selects those samples whose confidence are greater or greater than a certain threshold value T_{th}, that is, just select some samples whose current label confidence is higher to add to the training sample set, instead of adding all unlabeled samples to the training sample set of Tri-training, by this way, it can reduce the influence of "error accumulation" due to incorrect classification for some samples, and meanwhile it can accelerate the iterative training process of Tri-training.

(2) When judging whether the results of two classifiers to tag unlabeled samples are consistent, Eq. (8) can be used to calculate the similarity degree between sample x and labeled sample set L, then judge whether these selected unlabeled samples whose similarity degree is greater than a certain threshold value Td_{th}, so that these selected unlabeled samples not only have higher classification confidence for the labeled samples, but also will be the most "representative" for the unlabeled samples. So, the way mentioned above can ensure the accuracy of SVM classifier for unlabeled samples to some extend and reduce the influence of noise or outliers in the iterative training.

The measuring method of difference between the sample x_i and current training sample set L can use the following equation:

$$d_{\cos}(x_i, L) = \frac{1}{n}\sum_{x_j \in L} d_{\cos}(x_i, x_j) \tag{8}$$

In the Eq. (8), n denotes the sample size of set L, $d_{cos}(x_i, x_j)$ denotes the similarity degree between sample x_i and x_j which is defined as follows:

$$d_{\cos}(x_i, x_j) = \frac{\left|\Phi(x_i)\cdot\Phi(x_j)\right|}{\left\|\Phi(x_i)\right\|\left\|\Phi(x_j)\right\|} = \frac{\left|K(x_i, x_j)\right|}{\sqrt{K(x_i, x_i)K(x_j, x_j)}} \tag{9}$$

where Φ(.) denotes non-linear mapping function and K(.,.) denotes the SVMs kernel function, so $\Phi(x_i)$ and $\Phi(x_j)$ are the corresponding coordinates of sample x_i and x_j which are projected to a higher feature space H by Φ(.).

In the SVM active learning, considering both the uncertainty of candidate sample to separating hyperplane measured by Eq. (6) and the diversity between candidate unlabeled samples and labeled samples, the Eq. (8) is revised as follows:

$$d(x_i, L) = \lambda_{\cos} d + (1 - \lambda_{\cos}) d_{\cos}(x_i, L) \tag{10}$$

In the Eq. (10), the variable d defined in Eq. (6) denotes the distance from the sample x_i to the separating hyperplane,$\lambda_{\cos}$ is the balance factor (0.5 if considering balance, that is, the parameter can balance between the distance d and the diversity $d_{\cos}(x_i, L)$ of sample x_i, so that the SVM classifier can be trained on these most "representative" samples which are not only nearest to the separating hyperplane, but also have huge difference between the candidate samples and current training sample set.

(3) Using the improved weighted SVM proposed in Sect. 3.2 to train the basic classifier, it can reduce the influence of unlabeled samples which are selected by the Tri-training algorithm and are not the most beneficial to the training, meanwhile it can improve the effectiveness of semi-supervised learning in our proposed algorithm.

(4) The output of Tri-training algorithm uses the ensemble of three base classifiers, and the weight of each classifier is calculated by calculation method of AdaBoost.

(5) In the active learning process of our proposed algorithm, these samples which are relative "certain" and the unlabeled samples which are "uncertain" and are tagged correctly by domain expert will be test sample set, so by this way, additional samples are not much needed as test sample set, and meanwhile, the test samples will also increase along with the procedure of training learning, this will ensure the scale and representative of test sample set.

3.5 Active learning method based on QBC used in the algorithm of QTB-ASVM

According to the discussion of Sect. 2, in the process of active learning, the QTB-ASVM algorithm will employ the active learning approach based on QBC [8] (Seung et al. 1992), and the three weighted SVM classifiers trained by the improved Tri-training algorithm of part D will be committee members of QBC, then the committee will be used to vote for predicting the sample label, and the most inconsistent samples of vote will be chosen as the candidate samples for tagging category. According to the different approaches of measuring the most inconsistency of vote, currently, there are two main types of QBC approach: one way (Kullback Leibler- divergence, or KL-d for short) proposed by McCallum and Nigam (McCallum et al. 1998) use the relative entropy to measure the difference of committee vote, and the other way proposed use vote entropy (or VE for short) to measure the inconsistency of committee vote.

Table 1 The vote results of QBC committee for samples

Committee members	Samples		
	X_1	X_2	X_3
h_1	$0.53(C_1)$	$0.44(C_2)$	$0.74(C_1)$
h_2	$0.54(C_1)$	$0.45(C_2)$	$0.81(C_2)$
h_3	$0.61(C_1)$	$0.54(C_1)$	$0.76(C_2)$

Table 2 Vote entropy and relative entropy of sample label

Samples	Measurement method	
	Vote entropy (VE)	Relative entropy (KL-d)
X_1	0.0	0.059
X_2	1.0	0.068
X_3	0.76	0.046

In both QBC approaches, if using the KL-d method, then the greater relative entropy is, the greater the difference of committee vote is, but this measure way will leak some samples whose classification results of committee members are inconsistent, and the leaked samples are just what the QBC algorithm needs to select. Similarly, if using the VE method, then the greater the VE value is, the greater the difference of committee vote is, although the samples of inconsistent vote are selected, but it doesn't consider the class conditional probability of sample, namely $P_j(C|x_i)$. And this will also leak some samples whose information are informative and helpful for training the learning machine. For instance, the results of QBC committee samples based on two types problems (c_1 and c_2) are presented in Tables 1 and 2, where the QBC committee consists of three SVM classifiers, that is h_1, h_2 and h_3, the variable x_1, x_2 and x_3 represent three samples respectively.

As shown in Table 2, the KL-d values of sample x_1 and x_2 are close to equal, so according to measurement rules of relative entropy, the two samples will be selected as a candidate samples, but the sample x_1 whose VE value is equal to zero (vote consistently) will be omitted by using the measurement of VE. Meanwhile, if using the measurement rules of vote entropy, the sample x_3 will be selected to be candidate sample, but the sample will be omitted if using the relative entropy to measure.

Thus it can be found that there are some committee members which are more uncertain to the classification of sample x_1 and x_3, and according to the principles and ideas of active learning, these samples which classifiers are uncertain to classify include rich information for training classifier (Anshu et al. 2018; Ion et al. 2002; Wang Handing et al. 2017; Xu et al. 2015). So these samples should be selected for the candidate, and it will be beneficial to tag the selected samples for training classifier.

For the problem of QBC active learning discussed above and in the literature (Fang et al. 2016), where it uses complementary method of vote entropy and relative entropy to solve such problem. However, in actual application, QBC algorithm is easy to be influenced by the unbalanced distribution of the sample, such as some outliers which has higher classification uncertainty and easier to be selected as the candidate samples. However, if these selected samples are added to the training sample set, it will greatly influence performance of the classifier and lead to error propagation, thus it will impact the performance of active learning.

Therefore, in order to solve the problems of QBC method, by combining with these approaches discussed in Sects. 2, 3.3 and 3.4, the QTB-ASVM algorithm will make the following improvements:

(1) The committee members of QBC will make use of three base classifiers trained by the improved Tri-training algorithm proposed in Sect. 3.4, therefore making the initial classifier of QBC in iterative training have strong robust, and reduce the influence of error propagation.

(2) After the unlabeled sample set U are tagged predictively, some samples whose classification confidence are smaller or smaller than a certain threshold Q_{th} are selected by Eq. (7), and these selected "uncertain" samples will be the unlabeled samples of QBC training learning. By this way, the iterative process of active learning can be speeded up, and the performance of the QBC active learning can be improved. At the same time, in order to guarantee the accuracy that the current classifier forecast the label for the unlabeled samples, the decision function $f(x_i)$ in Eq. (7) will use the ensemble output of three base classifiers, that is $f(x_i) = \alpha_1^t h_1(x_i) + \alpha_2^t h_2(x_i) + \alpha_3^t h_3(x_i)$, where variable a_i^t is the corresponding weight of each classifier in nth iteration of training learning, and the weight of each classifier is calculated by calculation method of AdaBoost. Therefore, the weight of classifier with high performance is bigger correspondingly, but smaller on the contrary.

(3) When using the QBC method to select the candidate samples, the following method can be used:

Step 1 According to Eq. (11), calculate the vote entropy of sample x_i.

$$D(x_i) = -\frac{1}{\ln \min(K, |C|)} \sum_{k=1}^{|C|} \frac{V(c_k, x_i)}{K} \ln \frac{V(c_k, x_i)}{K} \tag{11}$$

In the Eq. (11), $V(c_k,x_i)$ denotes the vote number of sample x_i which is tagged as category c_k by committee members.

Step 2 Select these samples whose VE value are greater or greater than a certain threshold value VE_{th}, that is to select the most inconsistent samples of vote. And for those samples whose VE value are less than a certain threshold value VE_{th} or equal to zero, that is to select the most consistent samples of vote, and then calculate their KL-d value according to Eq. (12) to.

$$D(x_i) = \frac{1}{K} \sum_{j=1}^{K} D[P_j(C|x_i) \| P_{avg}(C|x_i)] \tag{12}$$

In Eq. (12), variable C is a category collection which may include sample x_i, that is $C = \{c_k\}$, in which K is number of committee members, and $P_{avg}(C|x_i)$ is defined as a mean of class conditional probability of sample x_i, i.e. $P_{avg}(C|x_i) = \frac{1}{K} \sum_{j=1}^{K} P_j(C|x_i)$, meanwhile, $D[\cdot||\cdot]$ denotes the information measurement of two conditional probabilistic distribution, for $P_1(C)$ and $P_2(C)$, then $D[P_1(C)||P_2(C)] = \sum_{k=1}^{|C|} P_1(c_k) \ln \frac{P_1(c_k)}{P_2(c_k)}$.

Step 3 According to Eq. (10), calculate similarity degree $d_{cos}(x_i, L)$ of these samples selected in Step 2, if the the $d_{cos}(x_i, L)$ value of these samples are less than a threshold value Qd_{th} and the KL-d value of these samples are greater than a certain threshold value KL-d_{th}, then these samples should be added to the candidate samples set.

By using the above way, these samples which are the most inconsistent by vote and either do not exist or of a small number in the training sample set will be selected as candidate samples. So to some extent, the leaking of some samples which are the most informative for training can be avoided because of the using of the VE or KL-d separately, meanwhile the training sample set can be enriched and kept varied, so that it can decrease the probability of selecting the outliers and accelerate the iterative process of active learning.

3.6 Termination strategies of active learning for algorithm of QTB-ASVM

The purpose of SVM active learning is to reduce the cost of tagging samples, so that it can use as little labeled samples as possible to train SVM classifier as high performance. According to the discussion and analysis in Sect. 2 and our active learning procedure proposed in Sect. 3.1, it is easy to conclude that, in addition to the key sampling function Q (Xu et al. 2010), there are some other factors which influence the performance of SVM active learning, such as termination strategies of active learning $S_{stop,}$ a strategy which needs to estimate the time when the SVM active learning can reach to the best level, so that the algorithm can stop tag the unlabeled samples and terminate the process of active learning. This can avoid a lot spending in the process of selecting samples and tagging these selected samples and achieve less performance improvement.

Figure 3 shows the performance comparison between the active learning and random sampling. As shown in Fig. 3, when the active learning algorithm uses just about 10% of training data on average, 93% classification accuracy have been achieved, while during random sampling it is achieved using 30% of the data.

And We also can find in curves of Fig. 3 that the performance ceases to improve approximately when 94% accuracy is achieved, if the train learning continues, performance boosts is not clear when 35% of training data is used. That is, 15% of samples are tagged costly, but less improvement is achieved. And according to the ideas of the active learning, this kind of phenomenon should be avoided, especially when the learning is more focused on reducing the cost of tagging samples and tiny performance boost have less influence on classifier.

Based on the above analysis, for the SVM active learning which uses the Gaussian kernel function, in our algorithm, one integrated termination strategy of active learning is given as follows:

Step 1 In the process of SVM active learning, the kth time when using the sampling function Q to select candidate samples, in addition to tagged the selected samples by experts, meanwhile, using the previous SVM classifier f_{svm}^{k-1} to calculate their classification accuracy called η_{k-1}.

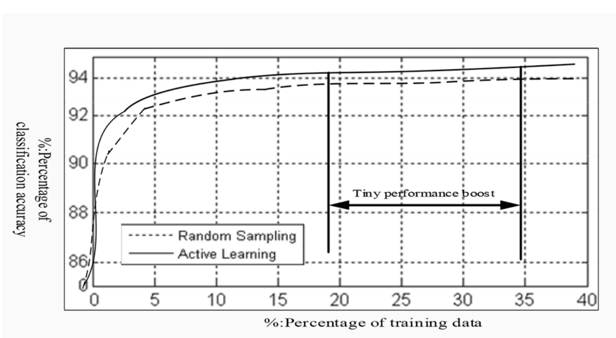

Fig. 3 Classification accuracy comparison between the active learning and random sampling

Step 2 Record the "most uncertain "samples selected by SVM active learning in kth time, and recorded as S_k.

Step 3 When the value η_k recorded in Step 1 is approaching a threshold and the performance curve of the classifier exhibits a "rise-peak-drop" pattern during training learning, that is, along with the SVM active learning process, when the η_k is equal to a certain value at a certain time (which can be calculated by few consecutive sampling), and the performance has no obvious boost and the set size of S_k recorded in step 2 is in a decreasing trend (which can be judged practically by the change rate of η_k), then the active learning process should be terminated.

4 QTB-ASVM Algorithm description

4.1 Symbols definition

h_1^t,h_2^t and h_3^t denote the three SVM base classifiers trained in the tth training process respectively.

X_o^t denotes the original training sample set in the tth training process.

X_I^t denotes candidate unlabeled sample set selected by QBC active learning in the tth training process.

$\hat{X}_I^t$ denotes the samples which are tagged correctly by domain expert in X_I^t.

f_{svm}^t denotes the ensemble SVM classifier trained in the tth training process.

α_i^t denotes the corresponding weight of each base classifier of f_{svm}^t.

$S\alpha_i^t$, $Q\alpha_i^t$ denote the weight of each base classifier of ensemble classifier after the semi-supervised learning and active learning based on QBC respectively.

T_{th} denotes the classification confidence threshold of semi-supervised learning based on Tri-training.

Q_{th} denotes the classification confidence threshold of active learning based on QBC.

Td_{th} denotes the similarity threshold in semi –supervised learning.

Qd_{th} denotes the similarity threshold in active learning.

UT_{th}^t denotes the set of these samples which are selected from the unlabeled sample set U and whose classification confidence are all greater than a certain threshold T_{th} in the tth semi-supervised learning process.

UQ_{th}^t denotes the set of these samples which are selected from the unlabeled sample set U and whose classification confidence are all less than a certain threshold Q_{th} in the tth active learning process.

VE_{th} denotes the threshold of vote entropy.

$KL{-}d_{th}$ denotes the threshold of relative entropy.

4.2 Algorithm procedure

4.2.1 Input

seed labeled data L, unlabeled data U, sample number m of sampling per time, SVM training algorithm *Learn* and Incremental SVM training algorithm *IncLearn*, threshold value T_{th}, Q_{th}, Td_{th}, Qd_{th}, VE_{th} and $KL\!-\!d_{th}$, termination condition S_{stop}.

4.2.2 Output

Classifier f_{svm}, tagging unlabeled samples.

4.2.3 Initialization

{

$X_o^0 = L$;

for $i \in \{1, 2, 3\}$

$S_i = BootstrapSamle(L); h_i^0 = Learn(S_i);$

$e_i^{'} = 0.5; l_i^{'} = 0;$

end for

$f_{svm}^0 = 1/3(h_1^0 + h_2^0 + h_3^0)$;

$t = 1$;

}

4.2.4 SVM active learning Loop

Step 1 While training learning in the tth(t = 1,2,…) round, verify whether the SVM classifier f_{svm}^{t-1} satisfy the stopping criterion S_{stop}, if f_{svm}^{t-1} satisfies S_{stop}, stop the learning loop and the output should be $f_{svm} = f_{svm}^{t-1}$, otherwise go to Step 2.

Step 2 Determine whether the set U is empty, if U is empty, then terminate the training learning and the output result is $f_{svm} = f_{svm}^{t-1}$. Otherwise, use the ensemble classifier f_{svm}^{t-1} to forecast the category for the unlabeled samples in set U, then according to Eq. (7), let UT_{th}^t be a set which contains the first m samples whose classification confidence is greater than the threshold T_{th}.

Step 3 On the labeled set X_o^{t-1} and sample set UT_{th}^t, repeat *Semi_Training* subprocedure until the performance of h_1^{t-1}, h_2^{t-1} and h_3^{t-1} have no more improvements.

Step 4 Determine whether the ensemble classifier f_{svm}^t satisfy the stopping criterion S_{stop}, where $f_{svm}^t = S\alpha_1^t h_1^{t-1} + S\alpha_2^t h_2^{t-1} + S\alpha_3^t h_3^{t-1}$. If f_{svm}^{t-1} satisfies S_{stop}, then out $f_{svm} = f_{svm}^t$ and terminate the training learning, otherwise go to Step 5.

Step 5 Use the ensemble classifier f_{svm}^t to forecast the category for the unlabeled samples, after according to Eq. (7), let UQ_{th}^t be a set which contains the first m samples whose classification confidence is less than the threshold Q_{th}^t. Then on the sample set X_o^{t-1} and sample set UQ_{th}^t, proceed to do *Active_Training* subprocedure.

Step 6 verify whether the SVM classifier f_{svm}^t satisfy the stopping criterion S_{stop}, where the classifier f_{svm}^t is a ensemble classifier which is defined as $f_{svm}^t = Q\alpha_1^t h_1^{t-1} + Q\alpha_2^t h_2^{t-1} + Q\alpha_3^t h_3^{t-1}$, if f_{svm}^t satisfies S_{stop}, stop the learning loop and out $f_{svm} = f_{svm}^t$, otherwise let $t = t + 1$ and go to Step 2.

In the above main procedure, *Semi_Training* subprocedure denotes the improved semi-supervised learning process based on Tri-Training which is discribed as following:

Semi_Training(X_o^{t-1} , UT_{th}^t , h_1^{t-1} , h_2^{t-1} , h_3^{t-1})

{

for *i*=1 to 3 do

L_i is empty set ;

$update_i = False$;

$e_i = MeasureError(h_j^{t-1} \& h_k^{t-1})\ (j,k \neq i)$;

If($e_i < e_i^{'}$) then

for each $x \in UT_{th}^t$ do

If ($h_j^{t-1}(x) = h_j^{t-1}(x)\ (j,k \neq i)$

and $d_{\cos}(x,L) > Td_{th}$)%*According to* Eq.(8)

then $L_i = L_i \cup \{(x, h_j(x))\}$;

end for

if($l_i^{'} = 0$) then $l_i^{'} = \left\lfloor \frac{e_i}{e_i^{'} - e_i} + 1 \right\rfloor$;

if($l_i^{'} < |L_i|$)

then if ($e_i |L_i| < e_i^{'} l_i^{'}$)

then $update_i = True$;

else if($l_i^{'} > \frac{e_i}{e_i^{'} - e_i}$)

then

$L_i = subSample(L_i, \left\lceil \frac{e_i^{'} l_i^{'}}{e_i} - 1 \right\rceil)$;

$update_i = True$;

end for

for *i*=1 to 3 do

if($update_i = True$)

then

$h_i^{t-1} = incLearn(h_i^{t-1}, X_o^{t-1}, L_i)$; $e_i^{'} = e_i$

; $l_i^{'} = |L_i|$;

end for

}

for *i*=1 to 3 do

% *estimate the training error of* h_i^{t-1}

$\varepsilon_i = MeasureError(h_i^{t-1})$

$S\alpha_i^t = \frac{1}{2}\ln(\frac{1-\varepsilon_i}{\varepsilon_i})$;

%*Use the calculation method of AdaBoost*

end for

Where in the avbove main procedure, *Active_Training* subprocedure denotes the active learning procedure -based QBC, which is discribed as following:

Active_Training(X_o^{t-1} , UQ_{th}^t , h_1^{t-1} , h_2^{t-1} , h_3^{t-1})

{

$X_I^t = 0$;

for each $x \in UQ_{th}^t$ do

% h_1^{t-1} , h_2^{t-1} and h_3^{t-1} are the committee members of QBC

% $VE(x)$ denotes the vote entropy of sample x calculated by Eq. (11)

if($VE(x) > VE_{th}$)

$X_I^t = X_I^t \cup x$;

else if($VE(x) \leq VE_{th}$ *or* $VE(x) = 0$)

then if($KL - d(x) > \mathrm{KL} - d_{th}$

and $d_{\cos}(x,L) < Qd_{th}$)

% $d_{\cos}(x,L)$ is calculated by Eq.(10)

% KL-d(x) is calculated by Eq.(12)

$X_I^t = X_I^t \cup x$;

end for

On set ,let $\hat{X}_I^t$ be a sample set whose samples are tagged correctly by domain expert E_{user}

for *i*=1 to 3 do

$h_i^{t-1} = incLearn(h_i^{t-1}, X_o^{t-1}, \hat{X}_I^t)$;

%On the sample set X_o^{t-1} and incremental sample set $\hat{X}_I^t$,to train the classifier h_i^{t-1} using the incremental training algorithm

$\varepsilon_i = MeasureError(h_i^{t-1})$;

%Estimate the training error of classifier h_i^{t-1}

$Q\alpha_i^t = \frac{1}{2}\ln(\frac{1-\varepsilon_i}{\varepsilon_i})$;

%Adopt the calculation method in AdaBoost

end for

$U = U - X_I^t$;%Delete the labeled sample set X_I^t from sample set U

$X_0^{t-1} = X_0^{t-1} \cup \hat{X}_I^t$;%Add the selected sample set % $\hat{X}_I^t$ to the labeled sample set

}

Table 3 Summary of datasets used in experiments: the number of instances and number of classes

Dataset	Number of instances	Number of classes (percentage of positive class, percentage of negative class)
Breast-cancer-wisconsin	699	2 (65.5%, 34.5%)
Ionosphere	351	2 (35.9%, 64.1%)
House-votes-84	435	2 (45.2%, 54.8%)
hepatitis	155	2 (20.6%, 79.4%)
credit-screening	690	2 (44.5%, 55.5%)
Glass	214	2 (76.2%, 23.8%)

5 Experiment and analysis

In this section, we present SVMs active learning experiments in which we examine the applicability of the method suggested in previous section.

5.1 Datasets and experimental settings

To verify our method, we use the standard dataset from UCI repository of machine learning database and the summary of datasets used in our experiments is shown in Table 3.

In our experiments, each dataset is divided into two parts. The first part is initial training set composed of 2% of all instances selected using the *Bootstrap Sample* method, the second is unlabeled training set composed of 98% of all instance and their labels were discarded factitiously for verifying the active learning approach proposed in this paper.

Experimental data using the UCI database of breast cancer—Wisconsin, ionosphere, house votes—84, hepatitis, credit approval and glass data set. In algorithm experiments, the sampled randomly choose 2% of the sample as the initial sample, the SVM training to remove the remaining 98% of the sample category tag as the candidate did not mark sample set.

For the convenience of comparison algorithm, in the experiment, the SVM model used in all semi-supervised learning adopts the classical weight SVM, and the other learning process use the weight SVM proposed in Sect. 3.2, where the kernel function use the Gaussian *RBF* kernel function whose parametric setting way given in the literature (Xu et al. 2010), and the parameter C was set to 100. The termination strategy of all active learning algorithm uses the integrated strategy proposed in Sect. 3.5. The initial bitch size m which restrict the numbers of samples selected to annotate was set to 8, and along with performance change of classifier, the parameter m should be changed adaptively, so that it can decrease the sample number of each sampling. And for convenience, in the semi-supervised learning, the sampling number of m is set to a fixed value.

In order to reduce the influence of error propagation on semi- supervised learning, the samples selected to train semi-supervised machine should have higher classification

Table 4 Comparison between the numbers of samples needed to be tagged and classification accuracy of corresponding algorithm

Datasets	Algorithms					
	Number of selected samples ± std					
	Classification accuracy (%) ± std					
	Random	Semi_Training	Active_Training	Active(KL-d)	Active(VE)	QTB-ASVM
Breast-cancer-wisconsin	458.2 ± 5.1	440 ± 3.8	98.3 ± 2.4	65.2 ± 2.1	68.3 ± 1.8	**60.2** ± 1.8
	91.3 ± 2.4	91.3 ± 2.2	91.3 ± 1.6	91.3 ± 1.4	91.3 ± 1.3	91.3 ± 1.2
Ionosphere	80.5 ± 3.9	92.3 ± 2.9	26.5 ± 1.5	24.5 ± 1.7	25.3 ± 1.2	**23.5** ± 1.4
	92.5 ± 2.1	92.5 ± 2.1	92.5 ± 1.3	92.5 ± 2.1	92.5 ± 1.3	92.5 ± 1.1
House-votes-84	186.3 ± 2.8	156.5 ± 2.3	79.3 ± 1.8	62.4 ± 2.5	63.5 ± 2.1	**59.5** ± 1.8
	93.2 ± 1.9	93.2 ± 1.6	93.2 ± 1.2	93.2 ± 1.9	93.2 ± 1.8	93.2 ± 0.9
Hepatitis	138.5 ± 2.9	86.2 ± 2.1	24.3 ± 2.1	25.6 ± 1.5	24.5 ± 1.7	**22.3** ± 1.5
	89.4 ± 2.5	89.4 ± 1.8	89.4 ± 1.8	89.4 ± 1.3	89.4 ± 1.6	89.4 ± 0.8
Credit-approval	463.2 ± 7.3	410.2 ± 4.6	223.5 ± 3.1	183.2 ± 2.2	178.5 ± 1.6	**168.3** ± 1.4
	93.6 ± 3.8	93.6 ± 2.4	93.6 ± 1.9	93.6 ± 2.1	93.6 ± 1.8	93.6 ± 1.5
Glass	163.4 ± 4.6	95.2 ± 2.7	35.3 ± 1.7	28.2 ± 1.8	28.4 ± 1.5	**27.2** ± 1.4
	88.2 ± 3.3	88.2 ± 1.7	88.2 ± 1.5	88.2 ± 1.9	88.2 ± 1.4	88.2 ± 1.2

confidence degree, so in the algorithm, the parameter T_{th} which range from 0.5 to 0.7 is more suitable, here T_{th} is set to 0.85 with compromise. Meanwhile, in order to ensure the selected samplers will be the most "representative" for a large number of unlabeled samples, the parameter Td_{th} *is* set to higher value as 0.85. In the active learning, in order to ensure the selected samples to be the "most uncertain" and avoid the " most informative" samples to be leaked, the parameter Q_{th} is set to 0.6, VE_{th} is set to 0.55, $KL\text{-}d_{th}$ is set to 0.045. Similarly, in order to avoid selecting the similar samples and ensure the "representative", the parameter Qd_{th} which is less than 0.5 is more suitable, which is set to 0.45 in our experiment.

5.2 Results and discussion

To verify the effectiveness and efficiency of SVMs active learning proposed in Sect. 4, we compared our SVMs active learning method (QTB-ASVM) with other methods, for example Random sampling, Semi_Training, Active_Training, Active(VE) and Active(KL-d), where the Semi_Training is the method proposed in Step 3 of QTB-ASVM algorithm, and the Active_Training is the method presented in Step 5 of our algorithm, meanwhile, Active(VE) is similar to the QTB-ASVM algorithm except the Active(VE) method which only uses the vote entropy to measure in Step 5 of QTB-ASVM, similarly, Active(KL-d) is also similar to the QTB-ASVM algorithm except the Active(KL-d) method, which only uses the relative entropy to measure in Step 5 of QTB-ASVM. For the convenience of comparing each algorithm, apart from the above differences, the other processes of each algorithm are identical with the QTB-ASVM algorithm.

In our experiments, to demonstrate the effectiveness of our method, six group experiments are performed and a tenfold cross-validation was also applied. In each group

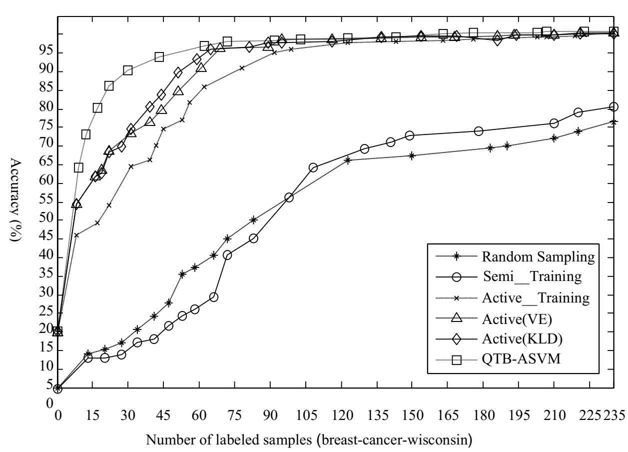

Fig. 4 QTB-ASVM(breast-cancer-wisconsin)

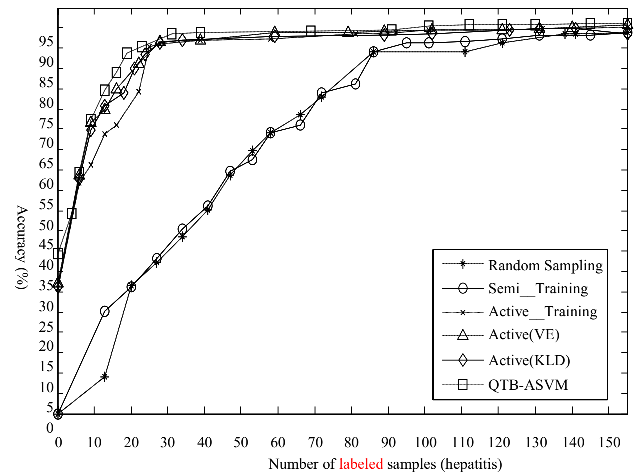

Fig. 5 QTB-ASVM (hepatitis)

experiment, we repeated five times and the result value is the average value of the five times. The experimental results of our approach compared with other active learning sampling algorithms are presented in Table 4, in which the symbol *std* represent the standard deviation of corresponding data results, the first row of each group experimental result (that is, the background color of the row is gray) denotes the number of samples selected to annotated and the second row denotes the corresponding accuracy of classification. There is one caveat, as shown in Table 4, the first row of corresponding Semi_Training algorithm denotes the number of unlabeled samples which are selected by semi-supervised learning, which are tagged by the ensemble classifier of iterative learning, and the result value of other algorithms is the number of candidate samples which are selected by corresponding algorithms and tagged by domain experts. As shown in the Table 4, we use the number of samples selected to annotate the accuracy of classifier as the metrics, which is to show which approach will minimize the number of labeled instances to minimize human annotation effort while still maintaining desired accuracy. It can be seen from Table 4, comparing the QTB—ASVM, Semi_Training, Active_Training, Active (VE), and the Active (KL-d) with random sampling algorithm, the labeled samples used by the front five algorithms are obviously less than the random sampling while achieving the same SVM classification accuracy. Comparing with Active_Training algorithm which only adopts the QBC active learning strategy and doesn't use the semi-supervised learning strategy, the labeled sampling used by the Active (VE), and the Active (KL—d) algorithm are also less than Active_Training algorithm while achieving the same classification accuracy. For example, as shown in Table 3, comparing with Active-Training, the labeled samples used by the Active(VE) and Active(KL-d) are reduced about 20.3% and 20.1% respectively while achieving 91.4%

average accuracy approximately on six datasets. This illustrates that the integrated method which combines with both semi-supervised learning and active learning will reduce the cost of tagging samples and improve the generalization performance of SVM compared to the adoption of only one of the two learning approaches. Meanwhile, the Active (VE) or Active (KL-d) algorithm may need about 7.6% or 7.8% more samples to be tagged respectively compared to QTB-ASVM while achieving the same classification accuracy approximately, this can also be observed obviously from Figs. 4 and 5, Table 3 shows that the improved active learning strategies based on QBC given in part B of Sect. 3 is effective.

In order to compare the classification accuracy of each algorithm with the changing of labeled samples number more obviously, the changing trend of performance of each algorithm on the "breast-cancer-wisconsin" dataset and "hepatitis" dataset are shown in Figs. 4 and 5. As shown in Figs. 4 and 5, the graphs indicate that when the labeled training samples are fewer, the partial performance curves of Semi-Train are even lower than the curves of random sampling algorithm especially on the initial training stage. That's because the training samples are fewer on the initial stage, and these selected training samples are all " more certain" samples for learning machine, but according to the SVM theory and active learning ideas, the possibilities that the most "uncertain" samples are support vectors seem very great. So if only the semi-supervised method is used in the training learning, these " more certain" samples can assist the training learning in improving the SVM performance, so that it needs to repeat training learning for many times while achieving the higher classification accuracy of SVM, and this undoubtedly will increase the time complexity of the algorithm.

From the Table 2, Figs. 4 and 5, we can easily draw a conclusion that whatever active learning approach will provide more benefit for the learning and cost approximately 30% label effort while keeping a good performance for SVMs classifier. But between the active learning algorithms, the classification accuracy curves of QTB-ASVM, Acitve(VE) and Active(KL-d) algorithm are all higher than the curves of Active_Training algorithm as a whole, which is, the classification accuracy of the previous three approaches are higher than that of the Active_Training while tagging the same number of samples. This phenomenon is more apparent especially in the initial stage of training learning. Meanwhile, comparing to the DRB-ASVM algorithm given in the literature(Xu et al. 2010), the number of labeled samples used by QTB-ASVM, Active(VE) and Active(KL-d) algorithm are all fewer than the DRB-ASVM algorithm, the reason of which, on the initial stage of training, the semi-supervised learning of the three algorithms select the " more certain" samples for classifier to assist the learning in training the SVM, this will make the training learning to have a strong initial SVM base classifier in the iterative stage of active learning, and these most "uncertain" unlabeled samples which are selected by the active learning process and which are most likely to be the support vector samples are tagged and added to the sample set of training, through this, it will further improve the generalization the SVM classifier. Both the curves of Figs. 4 and 5 indicate that the performance curves of these active learning algorithms are all higher than the curve of Semi_Training algorithms obviously.

Thus through the experiment and above analysis, the method which combines with both semi-supervised learning and active learning efficiently will reduce the cost of tagging samples greatly and improve the generalization performance of SVM obviously.

6 Conclusions and future work

We have presented a SVM training approach which combines the semi-supervised learning based on Tri-Training with the active learning based on QBC efficiently. During the SVMs learning process, in order to reduce annotation effort while maintaining the SVMs classification performance, we firstly employ the improved approach based on Tri-Training to select these samples which not only have higher classification confidence for the labeled samples, but also will be the most "representative" for the unlabeled samples. Then in the training process, we further employ the improved active learning based on QBC to select these labeled samples which are the most "inconsistent" by vote and are not in the training sample set or fewer for keeping the training sample set varied. To further reduce the number of unlabeled instances to be annotated and improve the generalization performance of SVM, some other measures also have been applied in the training learning, for example, the improved weighted SVM, the ensemble classifier, incremental SVM training algorithm and so on. Several standard datasets from UCI repository of machine learning database are used in our experiments to verify the effectiveness and efficiency. The experimental results show that our proposed semi-supervised active learning approach is effective and can dramatically reduce the cost of tagging samples, while without degrading the performance or with slight loss of performance.

There are several directions for further study. One direction is to set the thresholds used in our proposed method automatically according to the training sample set. In our current approach, we set the value manually and it will affect the SVMs classification performance in some way. Another direction is to further study the definition of a termination criterion for SVMs active learning, because an appropriate stopping criterion will also have a great impact on saving the annotation effort and keeping SVMs performance.

Acknowledgement The authors wish to express their gratitude to the referees for their helpful comments and kind suggestions in revising this paper. This work is substantially supported by grants from the Natural Science Foundation of China (Nos. 71701209, 72071209), and the Natural Science Foundation of Shaanxi Province of China (Nos. 2019JQ-250), and the China Postdoctoral Science Foundation funded project (2017M613415, 2019M653962).

References

Brefeld , Tobias S (2004) Co-EM support vector learning. In: Proceedings of the 21th International Conference on Machine Learning. ACM New York, NY, USA, pp 16–24

Fang Y, Chunyang H, Zhang LL et al (2016) Breakout prediction classifier for continuous casting based on active learning GA-SVM. Chin Mech Eng 27(12):1609–1914

Gokhan T, Hakkani-Tűr D, Schapire RE (2005) Combining active and semi-supervised learning for spoken language understanding. Speech Commun 45(2):171–186

Goudjil M, Koudil M, Bedda M et al (2018) A novel active learning method using SVM for text classification. Int J Autom Comput 15(3):290–298

Handing W, Yaochu J, John D (2017) Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems. IEEE Trans Cybern 47(9):2664f–22677

Hoi SCH, Lyu MR (2005) A semi-supervised active learning frame work for image retrieval. In: Proceedings of 2005 IEEE computer society conference on computer vision and pattern recognition, Los Angels, IEEE Computer Society pp 302–309

Ion M, Steven M, Knoblock CA (2002) Active + semi-supervised learning = robust multi-view learning. In: Proceedings of the nineteenth international conference on machine learning, pp 435–442

Jie XU, Pf SHI (2004) Active learning with labeled and unlabeled samples for content-based image retrieval. J Shang Hai Jiao Tong Univ 38(12):2068–2072

Kothari R, Jain V (2003) Learning from labeled and unlabeled data using a minimal number of queries. IEEE Trans Neural Netw 14(6):1496–1505

Li LY, Liu L FXGZ et al (2016) Modified particle swarm optimization for BMDS interceptor resource planning. Appl Intell 44(3):471–488

Li LY, Liu FX, L GZ et al (2016) Performance analysis and optimal allocation of layered defense M/M/N queueing systems. Math Probl Eng pp 1–32

McCallum AK, Nigam K (1998) Employing EM and pool-based active learning for text classification. In: Proceeding of the 15th international conference on machine learning, Morgan Kaufmann Publishers, USA, pp 350–358

Seung HS, Opper M, Sompolinsky H (1992) Query by committee. In: Proceedings of the 15th annual ACM conference on computational learning theory, Pittsburgh, Morgan Kauf mann, pp 287~294

Singla A, Patra S (2018) A fast partition-based batch-mode active learning technique using SVM classifier. Soft Comput 22(14):4627–4637

Tur G, Hakkani-Tűr D, Schapire RE (2005) Combining active and semi-supervised learning for spoken language understanding. Speech Commun 45(2):171–186

Wang XG, Liu LX (2017) Research on building bank anti-fraud model based on tri-training semi supervised learning and fuzzy SVM active learning. In: 2017 3rd Workshop on Advanced Research and Technology in Industry Applications (WARTIA 2017), Atlantis Press, pp 383–387

Wang XJ, Luo GC, Ke Q, Aiguo C et al (2016) Image retrieval method based on SVM and active learning. Appl Res Comput 33(12):3836–3838

Xu HL, Wang XD, Liao Y et al (2010a) New approach for optimizing model of RBF-SVM based on PSO. Control Decis 25(3):367–370

Xu HL, Wang XD, Liao Y et al (2010b) Incremental training algorithm of SVM based on active learning. Control and Decision 25(2):282–286

Xu HL, Bie XF, Feng H et al (2015) Active learning algorithm for SVM based on QBC. Syst Eng Electron 37(12):2865–2871

Xu HL, Long GZ, Bie XF et al (2016) Active learning algorithm of SVM combining tri-training semi-supervised learning and convex-hull vector. Pattern Recognit Artif Intell 29(1):39–46

Xu Z, Cheng C, Sugumaran V (2020) Big data analytics of crime prevention and control based on image processing upon cloud computing. J Surveill Secur Saf 1:16–33

Yan-fei P, Yong-gang S, De-jian W (2014) A novel retrieval method based on SVM and active learning. Comput Eng Sci 36(7):1371–1376

Zhang JM, Sun CM, Yan T (2013) Video Annotation for semi-supervised active learning based on adaptive SVM. Comput Eng 39(8):190–195

Zhao WZ, Ma HF, Li ZQ et al (2012) Efficient active learning for semi-supervised document clustering. J Softw 23(6):1486–1499

Zhou ZH, Li M (2005) Tri-training: exploiting unlabeled data using three classifiers. IEEE Trans Knowl Data Eng 17(11):1529–1541