Neurocomputing 172 (2016) 198–206

Contents lists available at ScienceDirect

Neurocomputing

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

Boosted Near-miss Under-sampling on SVM ensembles for concept detection in large-scale imbalanced datasets

Lei Bao [a,b], Cao Juan [a], Jintao Li [a], Yongdong Zhang [a,*]

[a] Laboratory for Advanced Computing Technology Research, ICT, CAS, Beijing 100190, China
[b] Graduate University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:
Received 1 December 2013
Received in revised form
21 April 2014
Accepted 26 May 2014
Available online 27 June 2015

Keywords:
Concept learning
Large-scale
Imbalance
Ensmeble learnning
Support Vector Machine
Boosted
Near-miss
Under-sampling

ABSTRACT

Considering the challenges of using SVM to learn concepts from large-scale imbalanced datasets, we proposed a new method: Boosted Near-miss Under-sampling on SVM ensembles (BNU-SVMs). The BNU-SVMs is under the framework of under-sampling ensemble method, where a sequence of SVMs is trained and the training dataset for each base SVM is selected by a Boosted Near-miss Under-sampling technique. More specifically, by adaptively updating weights over negative examples, the most near-miss negative examples in output space are selected in each iteration. Since the training dataset is balanced and reduced by under-sampling and the performance of classifier is improved by ensembles, the BNU-SVMs is a promising solution for large-scale and imbalance problem. Moreover, the negative examples selected by BNU-SVMs not only contain the most representative ones from data distribution perspective, but also cover the easily misclassified ones from data accuracy perspective. Therefore, the outperformance of the BNU-SVMs is expected. In addition, considering the computation cost caused by high-dimensional visual features, we proposed a kernel-distance pre-computation technique to further improve the efficiency of the BNU-SVMs. Experiments on TRECVID benchmark datasets show that the BNU-SVMs outperforms the previous methods significantly, which demonstrates that the BNU-SVMs is a both effective and efficient solution to concept detection in large-scale imbalanced datasets.

1. Introduction

Concept detection, as a foundation of video analysis and retrieval, has been a popular research topic for years [1]. It is commonly viewed as a supervised machine learning problem, which aims to learn the mapping function between low-level visual features and high-level semantic concepts based on the annotated training data. Support Vector Machine (SVM) [37], with its solid theoretical foundations and empirical success in practices, has been one of the most popular learning methods for concept detection [1].

However, as the explosive growth of video data and the development of web collaborative annotations, a large amount of annotated video data is available [2–4], which poses two challenges to SVM based concept detection. The first challenge comes from the large scale of annotated datasets. Taking the benchmark datasets for concept detection in TRECVID [5,6] as an example, from 2005 to 2012 the size of video collection is increased from 100 h to 800 h as shown in Fig. 1(a). 800 h videos roughly contain 400,000 shots, in which one shot is a training example. Since the standard SVM optimization has $O(l^3)$ computing complexity and $O(l^2)$ space complexity, where l is the number of training example [37], the computing and space cost of SVM becomes extremely expensive for large-scale datasets. The second challenge is the imbalance of the datasets. In most case, only a small portion of the data belongs to a specific concept, while the others are not (In our paper, the minority class always refers to the positive class.). As shown in Fig. 1(b), among 346 concepts in the TRECVID 2011 dataset [7], there are 224 (64.7%) concepts in which the imbalance ratio (#negative class vs. #positive class) of their training dataset is over 50:1, and only 46 (12.3%) concepts have a reasonable balanced dataset, in which the ratios is between 10:1 and 1:10. As stated in [9], given an imbalanced dataset, SVM's decision boundary is likely to be skewed to minority class, which will undermine the performance of the classifier. These two challenges motivate us to explore the traditional SVM and seek a solution for concept detection in large-scale imbalanced datasets.

In recent years, learning from imbalanced data has been one of the challenges in data mining communities [11], and some techniques have been developed to address this problem [29,30]. Among them, the under-sampling ensemble method tends to be a promising solution for concept detection in large-scale datasets. Firstly, under-sampling is a data preprocessing technique that balances the class distribution by removing examples from the majority class (negative class). For the large-scale imbalanced

* Corresponding author.

http://dx.doi.org/10.1016/j.neucom.2014.05.096
0925-2312/

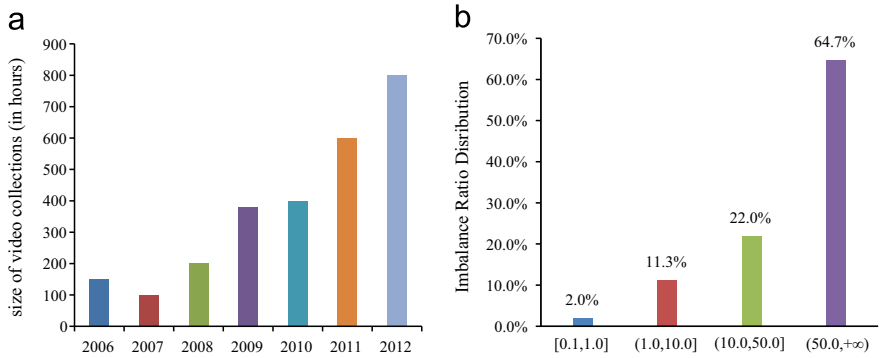

Fig. 1. (a) The size of video collections in TRECVID; (b) the imbalance ratios distributions of 346 concepts in TRECVID 2011 datasets.

datasets, it not only balances the class distribution, but also significantly reduces the time and space cost during training process, as the new rebalanced datasets is much smaller than the original one. However, one disadvantage is the potential loss of useful information contained in abandoned examples, which will hurt the performance of the classifier. Secondly, ensemble learning is a well-known method for improving the performance of a single classifier by combining several of them [12,13]. Nevertheless, ensembles technique itself cannot be directly applied on the large-scale imbalanced datasets, because its base classifier suffers the same two problems as we discussed above. Fortunately, these two techniques under-sampling and ensembles can be easily integrated together: under the ensemble framework the training dataset for each base classifier is sampled from the original dataset by under-sampling, which is so called under-sampling ensemble method. This combination method can alleviate the problems caused by the large-scale imbalanced datasets because the datasets used in each base classifier is balanced and reduced by under-sampling. Also compared with under-sampling, its performance is more stable because ensemble scheme lowers the variation of each individual classifier and reduces the possibility of losing useful information as more examples from the majority class are used. Another benefit of the under-sampling ensemble method is its independence of the underlying classifiers, which implies that SVM potentially can be integrated as the base classifier and the advantage of SVM can be inherited for concept detection. Taking into account of the all above advantages of under-sampling ensemble method, a lot of works have been developed under this framework [29]. However, only quit few works applied it on concept detection [20] and this becomes the main topic we will study in this paper.

In this paper, we proposed the Boosted Near-miss Under-sampling on SVM Ensemble (BNU-SVMs) for concept detection in large-scale imbalanced datasets. In BNU-SVMs, a sequence of SVMs is trained and the training examples for base SVM are selected by Boosted Near-miss Under-sampling technique. More specifically, by adaptively updating weights over negative examples, in each iteration the most near-miss negative examples in output space are selected. From the perspective of data distribution, since the data distribution in the output space is a reflection of its original distribution in feature space, BNU-SVMs will select the negative examples which are not only close to the positive ones but also still have not been separated from positive ones by previous classifiers. From the perspective of data accuracy, the easily misclassified negative examples will also be covered by BNU-SVMS, as those examples are obviously close to the correctly classified positive examples in output space. By considering both data-distribution and data accuracy, we can expect that the BNU-SVMs will outperform than the previous under-sampling ensemble methods.

In addition, considering the high computation cost caused by high-dimensional visual features, we proposed a kernel-distance pre-computation technique to further improve the efficiency of the BNU-SVMs. The time-consuming kernel distance computation part is separated from SVM optimization so that the distances can be computed in parallel by computer cluster and the repetitive distance computation during training process can be avoided. Moreover, those pre-computed distance values are stored in amount of binary files and indexed by hashing for fast access. All of these strategies further improve the efficiency of our proposed method.

Experimental results on TRECVID benchmark datasets show that the proposed BNU-SVMs outperforms the previous methods significantly, which demonstrates that the BNU-SVMs is a both effective and efficient solution to concept detection with large-scale imbalanced datasets.

In this paper, we firstly review the related work in Section 2 and the details of the proposed BNU-SVMs are presented in Section 3. In Section 4, we analyze the experimental results on TRECVID dataset. Finally, we draw the conclusions in Section 5.

2. Related work

2.1. Support Vector Machines

The essential idea of SVM is to find the optimal hyperplane with maximum margin between two classes [37]. Given a training datasets S with l training examples $S=\{(\mathbf{x}_i,y_i)|\,\mathbf{x}_i\in\Re^d,y_i\in\{1,-1\},i=1,\cdots,l\}$, where $\mathbf{x}_i$ represents the d-dimensional feature vector of the ith example and y_i is its class label. For a soft margin method, the optimal hyperplane can be obtained by solving the following optimization problem:

$$\begin{aligned}\min_{\mathbf{w},b,\xi}\quad & \tfrac{1}{2}\|\mathbf{w}\|^2+C\sum_{i=1}^{l}\xi_i\\ \text{subject to}\quad & y_i\left(\mathbf{w}^T\phi(\mathbf{x}_i)+b\right)\geq 1-\xi_i,\\ & \xi_i\geq 0,i=1,\cdots,l,\end{aligned}\tag{1}$$

where $\phi(\mathbf{x}_i)$ maps $\mathbf{x}_i$ into a higher-dimensional space, C is a tradeoff between a large margin and a small effort penalty and ξ_i are slack variables, which measures the degree of misclassification of $\mathbf{x}_i$. Due to the possible high dimensionality of $\mathbf{w}$, the problem in Eq. (1) is usually solved by its following dual problem:

$$\begin{aligned}\min_{\boldsymbol{\alpha}}\quad & \tfrac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j)-\sum_{i=1}^{l}\alpha_i\\ \text{subject to}\quad & \sum_{i=1}^{l}\alpha_i y_i=0,\end{aligned}$$

$$0 \le \alpha_i \le C, \tag{2}$$

where $\boldsymbol{\alpha}$ is the Lagrangian parameter, and $K(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function. After the problem in Eq. (2) is solved, the following decision function is used to predict labels of testing data:

$$h(\mathbf{x}) = sign(f(\mathbf{x})) = sign\left(\sum_{i=1}^{l} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b\right). \tag{3}$$

Besides labeling$\mathbf{x}$ by the sign of the decision value $f(\mathbf{x})$, $|f(\mathbf{x})|$ also represents the distance from $\mathbf{x}$ to the decision hyperplane and indicates the corresponding prediction confidence. As stated in [16], the distance can be converted to a probability estimate by the sigmoid model as below:

$$p(\mathbf{x}) = \frac{1}{1+\exp(\alpha f(\mathbf{x}) + \beta)}, \tag{4}$$

where the parameters $\alpha$and β are maximum likelihood estimates based on the training set.

Due to its theoretical and practical advantages, SVM has been very popular in machine learning and data mining, and has become the default choice in most concept detection schemes. However, as the dataset is becoming large-scale and imbalanced, SVMs have been challenged.

Firstly, it has been identified that SVMs are sensitive to imbalanced datasets [14,15]. As shown in Eq. (1), the slack variables ξ_i are assigned by the same cost C for both the positive and negative examples. In order to reduce the total penalty term (the second term in Eq. (1)), the solution to this optimization problem would overfit the negative examples. That will cause the skew of decision boundary and lower the performance of classifier.

Secondly, for large-scale datasets, the SVM training is time-consuming. As a Quadratic Programming (QP) problem shown in Eq. (2), SVM optimization needs $O(l^3)$ time and $O(l^2)$space complexities. In practical this problem is often scaled down by decomposition methods. By using Sequential Minimal Optimization (SMO) [17], LIBSVM [8] reduces the space and time complexities to $O(l)$ and #iteration $\times$ $O(l)$, respectively. Empirically, the number of iterations may be higher order than linear to the number of training data. We simply assume the number of iterations as l, and then the training time cost is $O(l^2)$. In addition, because all elements of $K(\mathbf{x}_i, \mathbf{x}_j)$ are too large to store in computer memory, $K(\mathbf{x}_i, \mathbf{x}_j)$ is usually calculated with optimization process. Assume each kernel calculation costs $O(d)$, the cost of SVM training becomes $O(dl^2)$. Moreover, because parameters of SVM (C and γ in kernel function $K(\bullet)$) significantly impact the performance of concept detection, a k-fold cross-validation is usually applied for parameter selection. Given p parameters, the time cost of parameter selection is $pkO(d(l/k)^2)$. The total time cost becomes $pkO(d(l/k)^2)+O(dl^2)$, which is considerable for large-scale datasets. Especially, besides the large scale of dataset, the high dimension of visual features used in concept detection is also an important factor which affects the efficiency of SVM. Take the Bag-of-Words (BoW) representation of local keypoint features as an example, to achieve a satisfied performance, its dimension is usually over thousands [18,19]. Such a high-dimension feature will extremely slow down the SVM training process as the kernel calculations are embedded. Since that, we should also consider the efficiency problems introduced by high-dimension visual features.

Those problems inspires us to seek a solution to handle the challenges caused by large-scale imbalanced datasets while still inherit the advantages of SVM.

2.2. Concept detection in large-scale and imbalanced datasets

Considering the challenges of using SVM to learn concepts from large-scale imbalanced datasets, the researchers in multimedia community have made some improvements [14,10,20].

One popular method, which has been widely applied in concept detection [1] to deal with the imbalance problem, is the Different Error Costs (DEC) method [14]. In DEC, the objective function (Eq. (1)) of SVM is modified to have two different misclassification costs C^+ and C^- for positive examples and negative examples, respectively. For reasonable good classification results, C^+/C^- is usually equals to the majority to minority class ratio. That will make the modified SVM less biased to negative examples as the positive examples are assigned with a higher cost. However, this method still suffers the problem from large-scale datasets. To handle the both large-scale and imbalanced dataset, [10] proposed a Support Cluster Machine (SCM). It samples the support vectors (as informative examples) and the centers of Kernel Means (as representative examples) to rebalance the training datasets, and speeds up the training procedure. As an under-sampling method, the experiment results in [10] show that the time cost of SCM is fewer than the baseline SVM by using the whole examples, but it performance is slightly worse because the under-sampling technique ignores the potentially useful information in abandoned examples. Furthermore, AdaOUBoost is proposed in [20], which ensembles SVMs by utilizing over-sampling and under-sampling techniques to rebalance the training datasets and avoids losing the potentially useful information. With the over-sampling technique, their experimental results [20] show that its performance is better than the baseline SVM but its time cost is also about 8 times higher.

The reviews on the previous works on concept detection show that it is very necessary to seek a both effective and efficient solution for concept learning in large-scale imbalanced datasets.

2.3. The under-sampling ensembles

In general, from the under-sampling technique's point of view, the previous under-sampling ensemble methods can be classified into two categories: Random Under-Sampling Ensembles (RUSE) and Boosted Under-Sampling Ensembles (BUSE).

Random Under-Sampling Ensembles (RUSE): RUSE is an imbalanced version of Bagging [21]. The base classifiers are trained in parallel and the training dataset for each base classifier is built by random under-sampling, where the number of positive examples is fixed (by the usage of all of them or resampling them) and an equal size of negative examples are selected by random sampling with or without replacement from the original negative class. Note that, as reported in [22], there are no significant changes in results between sampling with and without replacement. In this paper, we choose sampling with replacement. The previous works in this category include UnderBagging [29], Asymmetric Bagging [24], EUS SVM [25] and EasyEnsemble [26].

Boosted Under-Sampling Ensembles (BUSE): BUSE is an imbalanced version of Boosting (Adaboost) [23]. By adaptively updating weights over examples (misclassified examples gain weight and the others lose weight), the BUSE selects the easily misclassified negative examples in each iteration. The previous implements of BUSE include Asymmetric AdaBoost [27], RUSboost [28] and BalanceCascade [26]. The pseudo code for Boosted Under-Sampling Ensembles is shown in Algorithm 1.

Algorithm 1. Boosted Under-Sampling Ensembles (BUSE)

Input: Training dataset: $S = \{\mathbf{x}_i, y_i\}, i = 1, \cdots, l$; Positive class: $S^+ = \{\mathbf{x}_i^+, y_i^+\}, i = 1, \cdots, l^+$; Negative class : $S^- = \{\mathbf{x}_i^-, y_i^-\}, i = 1, \cdots, l^-$; $S = S^+ \cup S^-$ and $l = l^+ + l^-$; Number of iterations: T; Base classifier: I

1: $D(i)=1/l \text{for} i=1,\cdots,l$
2: **for** $t=1$ to T **do**
3: draw positive set S_t^+ with size l^+ from S^+ by random replacement sampling
4: draw negative set S_t^- with size l^+ from S^- by weighted replacement sampling with D_t
5: train base classifier: $h_t=I(S_t^+,S_t^-)$
6: $\varepsilon_t=\sum_{i=1}^{l} D_t(i)\left[h_t(\mathbf{x}_i)\neq y_i\right]$
7: $\alpha_t=\frac{1}{2}\ln\left(\frac{\varepsilon_t}{1-\varepsilon_t}\right)$
8: $D_{t+1}(i)=\frac{D_t(i)}{Z_t}\times\exp(-\alpha_t y_i h_t(\mathbf{x}_i))$ for $i=1,\cdots,l^-$, the sum of D_{t+1} is normalized to 1 byZ_t.
9: **end for**

Output: classifier $H(\mathbf{x})=sign\left(\sum_{t=1}^{T}\alpha_t h_t(\mathbf{x})\right)$

Besides the above two under-sampling techniques, there is another type of under-sampling technique, which samples the most representative examples based on the data distribution [31,32]. Zang and Mani [31] took the distances between negative examples and positive examples as a metric to select the most representative examples, where the best performer NearMiss-2 selects those negative examples whose average distances to three farthest positive examples are smallest. Show-Jane and Lee [32] proposed a cluster-based under-sampling which clusters all the original dataset into some clusters and selects the most representative negative examples from each cluster. Even though, the effectiveness of distribution-based methods has been proved, they have not been integrated with ensembles yet. One possible reason is that, as stated in [38], the diversity of the base classifier is an essential factor for a successful ensemble. However, the distribution-based methods will select the training subsets on relatively stable data distribution, which could not achieve diverse classifiers. This becomes the motivation of our work to develop a new distribution-based under-sampling ensembles method.

3. The proposed

3.1. Boosted Near-miss Under-sampling on SVM ensembles

To develop a new distribution-based under-sampling ensemble method, we further analyze the output of SVM. As we discussed in Section 2.1, the sign of $f(\mathbf{x})$ in Eq. (3) indicates **x**'s class label and the absolute value $|f(\mathbf{x})|$ indicates the distance from **x** to the decision hyperplane. Moreover, the decision function$f(\bullet)$also can be considered as a mapping function, which maps data **x** from its original input space to the output space. The sign of its value in output space depends on which side of the hyperplane **x** falls on in the original space, and its absolute value in output space depends on the distance from **x** to the hyperplane in original space. Therefore, the data distribution in the output spaces can be considered as a reflection of its distribution in original space. Especially, if the two examples are very close in input space, there are also likely close in the output space, unless they are in different sides of the hyperplane. This implies that, we can use the data distribution in output space from classifier's perspective to under-sample the most representative examples. In addition, since data distribution in output space also depends on base classifiers, it changes when more base classifiers are added in ensemble. That could help us achieve the diversity of training subsets and make a perfect combination of the distribution-base under-sampling and ensemble method. All of these observations motivate us to propose the Boosted Near-miss Under-sampling on SVM ensembles (BNU-SVMs).

As shown in Algorithm 2, in BNU-SVMs, a sequence of base SVM classifiers is trained and the training dataset for each base classifier is balanced by under-sampling, where l^+ positive examples are sampled by random replacement sampling and the same size of negative examples are sampled by weighted replacement sampling. The weights of negative examples are adaptively updated, where the negative examples that are close to positive examples in current output space gain more weight, the others gain less weight.

The details of weights updating are shown in step 6–11 in Algorithm 2. Firstly, instead of taking the decision value as example's value in output space, we transfer decision values to probability estimates. As the probability estimate is ranged from 0 to 1, which normalizes the output values from different base classifiers to the same scale. Moreover, in order to make the full use of every positive example, the near-miss weight $w(i)$ of the i-th negative example is accumulated by its weight from an individual positive example as shown in step 9. More specifically, the near-miss weight is calculated by the radial basis function. As a common choice, the parameter σ_p for the p-th positive example is set as its average distance to all negative examples as shown in step 8. Note that, for the i-th negative example, the more far from the p-th positive example, the less weight it can get from the p-th positive example. Finally, the final weight for next weighted resampling is updated by multiplying previous weight $D_t^-(i)$ with current near-miss weight $w(i)$.

As shown in Algorithms 1 and 2, the proposed BNU-SVMs is similar to the BUSE. Their main difference is that the weights in proposed BNU-SVMs are not updated based on current accuracy of examples (as BUSE does) but on the data distribution in output space. As the output value also includes the information of data accuracy (the sign of $f(\mathbf{x})$ indicates **x**'s class label), we can expect the near-miss examples selected by proposed BNU-SVMs contains the information from both data distribution and data accuracy perspectives.

From the perspective of data distribution, the BNU-SVMs is designed to select the near miss examples in output space rather than the original input feature space as previous NearMiss method did [31]. The near-miss weights in BNU-SVMs are measured in output space, which not only depend on data distribution in original input space but also depend on the base classifiers. Therefore, the BNU-SVMs will focus on the negative examples which are not only close to the positive ones but also still have not been separated from positive ones by previous base classifiers. Meanwhile, the data distribution in output space changes with the classifiers, which ensures the diversity of training subsets. Since that, the BNU-SVMs can benefit from the ensembles method and its performance is expected to be better than the previous distribution-based methods.

From the perspective of examples' accuracy, it is obvious that the easily misclassified negative examples are also close to the correctly classified positive examples in output space, because they are in the same side of the current hyperplane. Since that, examples selected by BNU-SVMs will cover those examples selected by BUSE. Furthermore, BNU-SVMs will also select the negative examples which are close to misclassified positive examples. Those examples are also informative because they are ambiguous with positive examples and they are the main reason for misclassifying those positive examples. Therefore the proposed BNU-SVMs is also expected to outperform BUSE.

As the proposed BNU-SVMs considers the information from both data distribution and data accuracy, it could be an effective method for imbalance dataset. In additional, within the under-sampling ensemble framework, the proposed BNU-SVMs also shows its efficiency for large-scale dataset as we will discuss in the following sub sections.

Algorithm 2. Boosted Near-miss Under-sampling on SVM ensemble

Input: Training dataset: $S=\{\mathbf{x}_i,y_i\}, i=1,\cdots,l$; Positive class: $S^+=\{\mathbf{x}_i^+,y_i^+\}, i=1,\cdots,l^+$; Negative class: $S^-=\{\mathbf{x}_i^-,y_i^-\}, i=1,\cdots,l^-$; $S=S^+\cup S^-$ and $l=l^++l^-$; Number of iterations: T; SVM classifier: I

1: $D_1^+(i)=1/l^+$ for $i=1,\cdots,l^+$, $D_1^-(i)=1/l^-$ for $i=1,\cdots,l^-$
2: **for** $t=1$ to T **do**
3: draw positive set S_t^+ with size l^+ from S^+ by random replacement sampling
4: draw negative set S_t^- with size l^+ from S^- by weighted replacement sampling with D_t^-
5: train SVM classifier: $(h_t,p_t)=I(S_t^+,S_t^-)$
6: $w(i)=0$ for $i=1,\cdots,l^-$
7: **for** $p=1$ to l^+ **do**
8: $\sigma_p=\frac{1}{l^-}\sum_{i=1}^{l^-}\left|p_t(\mathbf{x}_i^-)-p_t\left(\mathbf{x}_p^+\right)\right|$
9: $w(i)=w(i)+\exp\left(-\frac{1}{2\sigma_p{}^2}\left(p_t(\mathbf{x}_i^-)-p_t\left(\mathbf{x}_p^+\right)\right)^2\right)$ for $i=1,\cdots,l^-$
10: **end for**
11: $D_{t+1}^-(i)=\frac{D_t^-(i)}{Z_t}\times w(i)$ for $i=1,\cdots,l^-$, the sum of D_{t+1}^- is normalized to 1 by Z_t.
12: **end for**

Output: classifier $H(\mathbf{x})=sign\left(\sum_{t=1}^{T}h_t(\mathbf{x})\right)$, $P(\mathbf{x})=\frac{1}{T}\sum_{t=1}^{T}p_t(\mathbf{x})$

3.2. Kernel distance pre-computation

As we discussed in Section 2.1, for high-dimension of visual features, the kernel calculation will be very time-consuming. However, in SVM implement, the kernel calculation is often embedded in SVM optimization, because the $l\times l$ kernel matrix is too large to be stored in computer memory [8]. That slows down the SVM training process. In our proposed BNU-SVMs, the size of the training subset is reduced to $2l^+$ by under-sampling. It is possible to store the $2l^+\times 2l^+$ kernel matrix in memory, and we can consider separating the kernel evaluation from the SVM optimization. Such that this time-consuming part can be computed in parallel by computer cluster and the repetitive kernel calculation in optimization can be avoid.

More specifically, we pre-computed the kernel distances between all training examples instead of kernel values, which further avoid the repetitively kernel distance computation during parameter selection and different concepts learning in the same dataset. Taking the exponential chi-square kernel as an example (which performs the best for BoW features empirically), its formula is given by

$$K(\mathbf{x}_i,\mathbf{x}_j)=\exp\left(-\gamma\chi^2(\mathbf{x}_i,\mathbf{x}_j)\right)$$

$$\chi^2(\mathbf{x}_i,\mathbf{x}_j)=\frac{1}{2}\sum_{p=1}^{d}(x_{ip}-x_{jp})^2/(x_{ip}+x_{jp}), \tag{5}$$

where χ^2 is the chi-square distance. Obviously, the kernel values depend on both the data and the parameter γ, and the kernel distances only depend on the data. With one copy of the kernel distances, the kernel values with a specified γ can be computed. As different models are trained in the same dataset with different γ in parameter selection, the kernel distance pre-computation will efficiently reduce the computational time of model selection. Additionally, if we plan to train a number of concept detectors in the same dataset, these SVM trainings also share the same kernel distance matrix. The more concepts to learn, the more benefits we can gain from pre-computation of kernel distance.

In our implement, all elements of $\chi^2(\mathbf{x}_i,\mathbf{x}_j)$ are pre-computed in parallel and stored in hard disk. Before training SVM on subset $\{S_t^+,S_t^-\}$, its $2l^+\times 2l^+$ kernel matrix is loaded to memory from the pre-computed $l\times l$ kernel distance matrix. For fast access, the pre-computed kernel distance matrix is stored as in a number of binary files and indexed by hashing. The detail is presented in Algorithm 3.

Algorithm 3. Loading kernel matrix in BNU-SVMs ensemble

Input: Training set: S; Training subset: $\{S_t^+,S_t^-\}$, which are sampled from S; Index mapping function from $\{S_t^+,S_t^-\}$ to S: $idx(i)\in\{1,\cdots,l\}, i=1,\cdots,2l^+$, where i is the example index in $\{S_t^+,S_t^-\}$ and $idx(i)$ returns its corresponding index in S; Parameter of $K(\bullet)$: γ; Kernel distances of S are stored in binary files: $\{\mathbf{B}_i\}_{i=1}^{\lceil l/k\rceil}$, where $\mathbf{B}_i$ stores a $k\times l$ kernel distance matrix and $\chi^2(\mathbf{x}_i,\mathbf{x}_j)$ is stored from $(i\%k\times l+j-1)\times m$ to $(i\%k\times l+j)\times m$ bytes in file $\mathbf{B}_{\lceil i/k\rceil}$ (assume each value is stored by m bytes);

1: **for** i=1 to $2l^+$ **do**
2: load $\left[(idx(i)\%k)\times l\times m,(idx(i)\%k+1)\times l\times m\right]$ in file $\mathbf{B}_{\lceil idx(i)/k\rceil}$ to $\mathbf{d}$, where $\mathbf{d}$ is a array of size l and $\mathbf{d}(j)=\chi^2(\mathbf{x}_{idx(i)},\mathbf{x}_j)$
3: **for** j==1 to $2l^+$ **do**
4: $\mathbf{K}(i,j)=\exp(-\gamma\mathbf{d}(idx(j)))$
5: **end for**
6: **end for**

Output: kernel matrix $\mathbf{K}$ of $\{S_t^+,S_t^-\}$, where $\mathbf{K}(i,j)=K(\mathbf{x}_{idx(i)},\mathbf{x}_{idx(j)})$

3.3. Computational complexity

The computation of the proposed BNU-SVMs can be divided into two parts: the kernel distance pre-computation and the model training.

Firstly, let us assume that each kernel distance calculation costs $O(d)$, and the kernel distance pre-computation of the training dataset S takes $O(dl^2)$. This can be run in parallel by computer cluster.

Secondly, in the model training part, the BNU-SVMs trains T base SVM classifiers and each of them is trained on a training subset with size $2l^+$. As discussed in Section 2.1, with pre-computed kernel distances, the time cost of training one base classifier is $O((2l^+)^2)$. Suppose we use a k-fold cross-validation to select best parameter from p parameters, the total time cost becomes $T(pkO((2l^+/k)^2)+O((2l^+)^2))$. With the same kernel distance pre-computation technique, the time complexity of the traditional SVM training is $pkO((l/k)^2)+O(l^2)$. Given a imbalanced large-scale dataset, l^+ is much smaller than l. Assuming $l/l^+=R$, the time complexity of traditional SVM training is $R^2/(4T)$ times of the proposed BNU-SVMs. When R is 50 and T is 10, the proposed BNU-SVMs is 62.5 times faster than the traditional SVM. Note that, with the kernel distance pre-computation technique, the space cost of the traditional SVM is $O(l^2)$. The space cost of our proposed BUN-SVMs is $O((2l^+)^2)$, which is $R^2/4$ times less than the traditional SVM. Also due to the high space cost, the kernel calculation is often embedded in optimization process as LIBSVM [8] does, which makes the traditional SVM cannot take advantage of the kernel distance pre-computation technique.

4. Experiments

4.1. Datasets

To evaluate the performance of our proposed method, we conducted experiments on TRECVID 2011 benchmark collection for semantic indexing [7]. This collection includes two parts: the training (development) dataset and the testing (evaluation) dataset. The training dataset contains around 400-h videos of 236,697 shots (examples). The testing dataset contains around 200-h videos of 137,327 shots. The TRECVID organizers select 346 test concepts and provide their annotations in training dataset. Of the 346 test concepts the organizers select a subset of 50 for evaluation in testing data set. Since our study focus on the imbalanced large-scale datasets, out of 50 concepts we selected 11 concepts whose imbalance ratio is larger than 50 (l^-/l^+).The details of the 11 concepts are listed in Table 1.

Three of BoW visual features are extracted to represent the visual content of shots: SIFT [33], Color SIFT [33] and Motion SIFT [34]. For each feature, a visual codebook of size 4096 is built by K-Means. Using the Spatial-Pyramid Matching technique [35], we extract 8 regions for one shot and calculate a BoW vector for each region. At the end, we obtain a $8 \times 4,096 = 32,768$ dimensional BoW vector for each feature. The chi-squared distance matrix of 236,697 training examples is pre-computed for all of these three features. The final kernel distance matrix used in our experiments is the average weighted fusion of the kernel distance matrixes of the three features. These distances are saved in 789 files in binary format and each one stores a $300 \times 236,697$ distance matrix.

Table 1
Details of the 11 concepts for evaluation.

ID	Concept	#Positive	#Negative	Imbalance Ratio
1027	Cheering	376	56,187	149.43
1038	Dancing	870	57,517	66.11
1041	Demonstration or Protest	599	55,733	93.04
1053	Flowers	516	46,006	89.16
1086	Old people	697	55,327	79.38
1089	People marching	510	59,263	116.20
1100	Running	350	69,804	199.44
1431	Skating	751	38,828	51.70
1443	Speaking to camera	506	39,365	77.80
1454	Studio with anchorperson	622	55,316	88.93
1478	Traffic	753	38,912	51.68

4.2. Experimental setup

For a thorough empirical comparison, we implemented 8 algorithms as listed in Table 2. These algorithms are divided into 3 groups. The first group is the baseline group, which includes two methods: SVM and DEC-SVM [14]. We take those two methods as our baseline as they are the most popular methods for concept detection [1]. The second group is the pure under-sampling group, which only utilizes the under-sampling technique and includes RU-SVM and NU-SVM. The last group is the under-sampling ensemble group, which is a combination of the under-sampling and ensemble techniques and includes RU-SVMs, BU-SVMs, NU-SVMs and the proposed BNU-SVMs. NU-SVMs is implemented to further study the combination of previous distribution-based under-sampling and ensembles techniques, which combines the NeaMiss2 under-sampling [31] with ensembles.

Note that the parameters C and γ for SVM training in all algorithms are selected by 2-fold cross-validation from $C \in \left\{2^{-5}, 2^{-2}, 2^{1}, 2^{4}, 2^{7}, 2^{10}, 2^{13}\right\}$ and $\gamma \in \left\{2^{-15}, 2^{-12}, 2^{-9}, 2^{-6}, 2^{-3}, 2^{0}, 2^{3}\right\}$. Without specific notification, T, the number of base SVM classifiers, is set to 20 for all the methods in the under-sampling

Table 2
The algorithms for comparison.

Group	Abbreviation	Short description
Baseline	SVM	The basic SVM classifier without considering the class-imbalance issue.
	DEC-SVM	The different error cost method [14], and C^+/C^- is set as l^-/l^+.
Under-sampling	RU-SVM	Random under-sampling SVM: one SVM is trained on a balanced dataset, where the negative examples are selected by random under-sampling.
	NU-SVM	Near-miss under-sampling SVM: one SVM is trained on a balanced dataset, where the negative examples are selected by NearMiss2 under-sampling [31].
Under-sampling ensembles	RU-SVMs	Random under-sampling on SVM ensembles: it is the same as RUSE discussed in Section 2.2 and the base classifier is SVM.
	BU-SVMs	Boosted under-sampling on SVM ensembles: it is the same as BUSE discussed in Section 2.2 and the base classifier is SVM.
	NU-SVMs	Near-miss Under-sampling on SVM ensembles: Several SVM classifiers are trained in parallel and each of them is trained on a balanced dataset by NearMiss2 under-sampling [31].
	BNU-SVMs	Boosted Near-miss Under-sampling on SVM ensembles: our proposed method describe in Section 3.1.

Table 3
The xinfAP comparison of algorithms.

ID	SVM	DEC-SVM	RU-SVM	NU-SVM	RU-SVMs	BU-SVMs	NU-SVMs	BNU-SVMs
1027	0.0630	0.0762	0.0688	0.0929	0.0506	0.0930	0.0941	***0.1041***
1038	0.0558	0.0613	0.0471	0.0513	0.0531	0.0669	0.0574	***0.0708***
1041	0.1364	***0.1424***	0.0504	0.0727	0.0612	0.1264	0.0593	0.1357
1053	0.0387	***0.0907***	0.0426	0.0485	0.0538	0.0812	0.0476	0.0856
1086	0.0759	***0.1147***	0.0728	0.0729	0.0825	0.1032	0.0892	0.1108
1089	0.0358	0.0378	0.0247	0.0309	0.0411	0.0348	0.0334	***0.0414***
1100	0.0649	0.0821	0.0343	0.0262	0.0359	0.0712	0.0470	***0.0843***
1431	0.2609	0.3035	0.1882	0.2385	0.2365	0.2501	0.2263	***0.3339***
1443	0.1011	0.1244	0.1402	0.1636	0.138	0.1454	0.1589	***0.1685***
1454	0.4148	0.4086	0.3967	0.3927	0.4093	0.4397	0.4063	***0.4437***
1478	0.1708	0.2089	0.1816	0.1771	0.1977	0.1999	0.2148	***0.2466***
Mean xinfAP	0.1289	0.1501	0.1134	0.1243	0.1236	0.1465	0.1304	***0.1659***

ensemble group. Moreover, given the high-dimensional visual features, we applied the kernel distance pre-computation technique for all those methods. Due to the high space complexity of SVM and DEC-SVM, we randomly discarded some negative examples to keep the training size less than 40,000.

The evaluation criteria used here is extended inferred average precision (xinfAP) [36], which an estimation of average precision (AP) and officially used in TRECVID.

4.3. Experimental results

For a thorough comparison of these methods, the experimental results are presented from different angles. Firstly, Table 3 presents the xinfAP of different methods on each concept and also theirs mean xinfAPs, which gives the most details of the performances of these methods. Secondly, we illustrate their average ranking values in Fig. 2, which shows their overall performances. Furthermore, we use a paired t-test on xinfAPs to find out whether there exist significant differences between a pair of algorithms, and their p-values are shown in Table 4. If the p-value of a pair of algorithms is smaller than 0.05, then the differences between their xinfAPs are significant and it confirms that the algorithm with higher mean xinfAP significantly outperforms the other one. Finally, to evaluate the efficiency of different methods, we compare their average run time costs as illustrated in Fig. 3.

Based on these experimental results, we further analyze the performances of the eights algorithms.

4.3.1. Baseline

Based on the comparison between SVM and DEC-SVM, we observe that DEC-SVM is consistently better than SVM as shown in Table 3 and this improvement is confirmed to be significant by checking p-value in Table 4. Since DEC-SVM is an imbalanced version of SVM by assigning a high cost for positive examples to make SVM less bias to negative examples, its outperformance shows that: 1) the traditional SVM is indeed sensitive to the imbalanced datasets; 2) by taking care of the imbalance issue, its performance could be significantly improved. Therefore, the imbalance issue becomes the first point of our study.

As for the efficiency evaluation, with the kernel-distance pre-computation technique, DEC-SVM and SVM take 289.92 and 285.41 min, respectively, as shown in Fig. 3. Note that, without the kernel-distance pre-computation technique, the time cost will be linearly increased with the dimension of visual feature. Considering the high dimension of visual feature, that time cost is not acceptable in practice. Therefore, even thought DEC-SVM can handle the imbalance issue, it still suffers that the same efficiency problem from large-scale datasets as SVM does. The large-scale issue becomes the second point of our study.

4.3.2. Pure under-sampling

As shown in Fig. 3, the most outstanding feature of pure under-sampling methods is their fast speed. The total time costs of RU-SVM and NU-SVM are 0.57 and 0.80 min, respectively, which are only about 0.2% of SVM's cost. However, their high efficiencies results in their low performance. Comparing their mean xinfAPs and average rankings with the others, RU-SVM and NU-SVM are the worst two methods. We are not surprised by their low performances. Because by abandoning negative examples to balance and reduce the training dataset, pure under-sampling methods also increase the probability of losing useful information, which hurts the performance of classifier. These observations demonstrate that under-sampling techniques have the potential to handle the large-scale imbalanced problem, but there is still some work to improve their performances.

4.3.3. The previous under-sampling ensemble methods

RU-SVMs: As shown in Tables 3 and 4, the performance of RU-SVMs is significantly better than RU-SVM. This result indicates that the performance of under-sampling technique could be improved by combining with ensemble technique, which shows the effectiveness of under-sampling ensemble methods in dealing with imbalance issue. As for the efficiency evaluation, the average run time of RU-SVMs is 11.82 min, which is only 4.0% of SVM's cost. That indicates that the under-sampling ensemble methods inherit the efficiency from under-sampling and presents their potential to handle large-scale datasets. The above observations demonstrate that the under-sampling ensemble method is a promising solution for concept detection in large-scale imbalanced datasets, as it

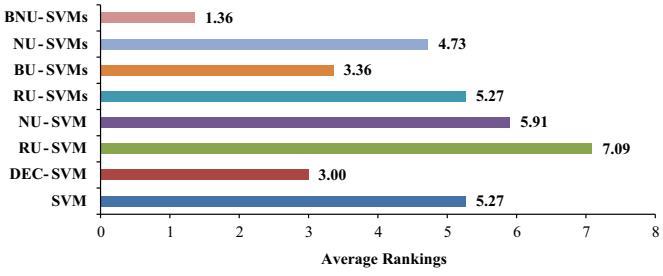

Fig. 2. Average rankings of different algorithms.

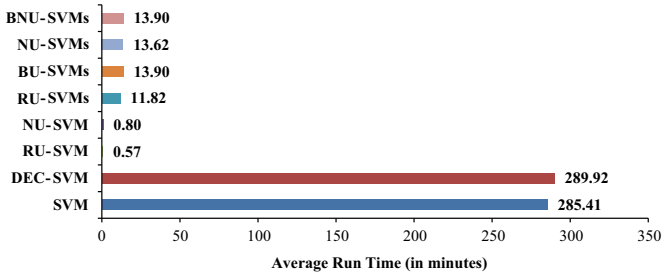

Fig. 3. Total run time of different algorithms.

Table 4
P-values of paired algorithms on xinfAPs.

	SVM	DEC-SVM	RU-SVM	NU-SVM	RU-SVMs	BU-SVMs	NU-SVMs	BNU-SVMs
SVM	––	***0.0022***	0.0935	0.3292	0.2888	***0.0069***	0.4494	***0.0005***
DEC-SVM	***0.0022***	––	***0.0052***	***0.0157***	***0.0075***	0.3083	0.0556	***0.0105***
RU-SVM	0.0935	***0.0052***	––	***0.0325***	***0.0307***	***0.0002***	***0.0002***	***0.0004***
NU-SVM	0.3292	***0.0157***	**0.0325**	––	0.4531	***0.0038***	0.1059	***0.0004***
RU-SVMs	0.2888	***0.0075***	***0.0307***	0.4531	––	***0.0019***	0.0933	***0.0002***
BU-SVMs	***0.0069***	0.3083	***0.0002***	***0.0038***	***0.0019***	––	***0.0247***	***0.0132***
NU-SVMs	0.4494	0.0556	***0.0002***	0.1059	0.0933	***0.0247***	––	***0.0018***
BNU-SVMs	***0.0005***	***0.0105***	***0.0004***	***0.0004***	***0.0002***	***0.0132***	***0.0018***	––

combines the merits of under-sampling and ensemble techniques. That motivates us to develop a method under this framework.

NU-SVMs: As shown in Table 3 and Fig. 2, the performance of NU-SVMs is only slightly better than NU-SVM, where the mean xinfAP is increased by 4.9% and its average ranking is improved from 5.91 to 4.73. However, this improvement is not significant in statistics because the p-value is 0.1059. Even though, as a pure under-sampling technique, NU-SVM is well-performed (compared with RU-SVM), its performance cannot be improved by ensemble technique as random under-sampling (RU-SVM vs. RU-SVMs) does. One possible reason is that, the distribution-based under-sampling could not achieve the diversity of base classifiers, as we have discussed in Section 2.3. This observation indicates the potential success of the proposed BNU-SVMS, where the diversity of base classifiers is achieved by under-sampling on data distribution in output space.

BU-SVMs: As shown in Table 3 and Fig. 2, BU-SVMs is the best performer among the previous under-sampling ensemble methods, compared with BU-SVMs and NU-SVMs. As the BU-SVMs selects the easily misclassified negative examples, its outperformance shows that the easily misclassified negative examples contain more useful information than others. That indicates the potential success of the proposed BNU-SVMs, which also considers the data accuracy information by under-sampling in output space.

4.3.4. The proposed BNU-SVMs method

Compared with NU-SVMs: As shown in Table 3 and Fig. 3, the mean xinfAP of BNU-SVMs is increased by 27.27% and its average ranking is improved from 4.73 to 1.36. By checking the *p*-value in Table 4, we can confirm that BNU-SVMs significantly outperform NU-SVMs. As we discussed in Section 3, the outperformance of BUN-SVMs is expected for two reasons: 1) the BNU-SVMs will focus on the negative examples which are not only close to the positive ones but also still have not been separated from positive ones by previous base classifiers. Those representative examples contain the most useful information in dataset. 2) The data distribution in output spaces changes with the classifiers which ensures the diversity of training subsets and makes BNU-SVMs benefit from the ensemble method. All of these explain the outperformance of BNU-SVMs.

Compared with BU-SVMs: As shown in Table 3 and Fig. 3, compared with RU-SVMs, the mean xinfAP of BNU-SVMs is increased about 13.25% and its average ranking is improved from 3.36 to 1.36. By checking the *p*-value in Table 4, we confirm that BNU-SVMs significantly outperforms BU-SVMs. That makes BNU-SVMs become the best performers in the under-sampling ensemble group. As we have pointed out, the near miss negative examples selected by BNU-SVMs not only covers the easily misclassified negative examples selected by BU-SVMs, but also includes the negative examples which are ambiguous with misclassified positive examples and the main cause for misclassifying those positive examples. Those useful negative examples make BNU-SVMs outperform BU-SVMs.

Compared with DEC-SVM: As shown in Table 3 and Fig. 3, the performance of BNU-SVMs is also significantly better than DEC-SVM, where the mean xinfAP is increased by 10.59%, its average ranking improved from 3.00 to 1.36, and the significance of their differences is confirmed by checking the *p*-value ($0.0105 < 0.05$). These observations further affirm the outstanding performance of the proposed BNU-SVMs in dealing with imbalanced datasets. Moreover, the BNU-SVMs inherits the efficiency from the under-sampling technique. As shown in Figure 4, the average run time cost of BNU-SVMs is 13.90 min which is only 4.80% of DEC-SVM's cost. That shows the capability of BNU-SVMs in handling large-scale datasets. Additional, to evaluate the performance of accessing the pre-computed distances, we separately count the run time of loading distances in training process. The average time cost of distance loading for each concept is about 0.66 min, which is only 4.74% of the training process. The binary storage and hashing index should take the credits for the high efficiency of distance loading.

All of these above observations demonstrate that, the proposed BNU-SVMs method with the kernel-distance pre-computation technique provides a both efficient and effective solution for concept learning in large-scale imbalance datasets.

5. Conclusions

In this paper, we proposed a new under-sampling ensemble method: Boosted Near-miss Under-sampling on SVM ensembles (BNU-SVMs) for concept detection in large-scale imbalanced datasets. Under the framework of under-sampling ensemble methods, the BNU-SVMs is capable to handle the large-scale imbalanced dataset, since the training dataset is balanced and reduced by under-sampling and the performance of classifier is improved by combining several of them. More specifically, the proposed BNU-SVMs selects the most near-miss examples in output space of base classifiers. From the data distribution perspective, since the data distribution in output space is a reflection of its distribution in original input space, BNU-SVMs picks the negative examples which not only are close to the positive ones but also cannot be separated from positive examples by previous classifiers. From the data accuracy perspective, the near miss negative examples selected by BNU-SVMs not only cover the easily misclassified negative examples selected by BU-SVMs, but also contain the negative examples which are the main cause for the misclassified positive examples. Therefore, by benefiting from both data distribution and data accuracy, the BNU-SVMs is expected to outperform previous methods. In addition, considering computation cost caused by high-dimensional visual features, we also proposed a kernel-distance pre-computation technique to further improve the efficiency of the BNU-SVMs, where the time-consuming kernel distances can be computed in parallel with computer cluster and the repetitively distance computation during training process can be avoided. Experiments are designed on TRECVID benchmark datasets to compare the proposed method with previous works. The results show the proposed BNU-SVMs significantly improves the performance. Therefore we can conclude that the BNU-SVMs is an effective and efficient solution to concept detection with large-scale imbalanced datasets.

Acknowledgments

This work is supported by National High Technology Research and Development Program of China (2014AA015202), the National Natural Science Foundation of China (61172153, 61100087), and the National Key Technology Research and Development Program of China (2012BAH39B02).

References

[1] C.G.M. Snoek, M.a. Worring, Concept-based video retrieval, Found. Trends Inf. Retr. 2 (4) (2009) 215–332.
[2] S. Ayache and G. Qu´enot, Video corpus annotation using active learning, in: Proceedings of European Conference on Information Retrieval, Glasgow, UK, 2008, pp. 187–198.
[3] Zheng-Jun Zha, Meng Wang, Yan-Tao Zheng, Y.i. Yang, Richang Hong, Tat-Seng Chua: interactive video indexing with statistical active learning, IEEE Trans. Multimed. 14 (1) (2012) 17–27.

[4] Zheng-Jun Zha, Linjun Yang, T.a.o. Mei, Meng Wang, Zengfu Wang, Visual query suggestion, ACM Multimed. (2009) 15–24.
[5] A.F. Smeaton, P. Over, and W. Kraaij, Evaluation campaigns and TRECVid, in Proceedings of the ACM SIGMM International Workshop on Multimedia Information Retrieval, 2006, pp. 321–330.
[6] A.F. Smeaton, P. Over, W. Kraaij, High Level Feature Detection From Video in TRECVid: A 5-Year Retrospective of Achievements, in: A. Divakaran (Ed.), Multimedia Content Analysis, Theory and Applications, Springer, 2008.
[7] P. Over, G. Awad, T. Rose, J. Fiscus, W. Kraaij, and A.F. Smeaton, TRECVID 2011—goals, tasks, data, evaluation mechanisms and metrics, in: Proceedings of the TRECVID Workshop, Gaithersburg, USA, 2011.
[8] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (27) (2011) 1–27.
[9] Yang Liu, Aijun An, and Xiangji Huang. Boosting prediction accuracy on imbalanced datasets with SVM ensembles, in; Proceedings of the 10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD'06), Berlin, Heidelberg, 2006, pp. 107–118.
[10] J. Yuan, J. Li,, and B. Zhang, Learning concepts from large scale imbalanced data sets using support cluster machines, in: Proceedings of the 14th Annual ACM International Conference on Multimedia, ACM, 2006, pp. 441–450.
[11] Q. Yang, X. Wu, 10 challenging problems in data mining research, Int. J. Inf. Technol. Decis. 5 (4) (2006) 597–604.
[12] R. Polikar, Ensemble based systems in decision making, IEEE Circuits Syst. Mag. 6 (3) (2006) 21–45.
[13] L. Rokach, Ensemble-based classifiers, Artif. Intell. Rev. 33 (2010) 1–39.
[14] K. Veropoulos, C. Campbell, and N. Cristianini, Controlling the sensitivity of support vector machines, in: Proceedings of the International Joint Conference on Artificial Intelligence, 1999, pp. 55–60.
[15] G. Wu and E. Chang, Adaptive feature-space conformal transformation for imbalanced-data learning, in: Proceedings of the 20th International Conference on Machine Learning, 2003, pp. 816–823.
[16] J.C. Platt, Probabilities for SV machines, in: A.J. Smola, P.L. Bartlett, B. Scholkopf, D. Schuurmans (Eds.), Advances in Large Margin Classifiers, The MIT Press, Cambridge, USA, 2000, pp. 61–74.
[17] J.C. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Sch•olkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, MIT Press, Cambridge, MA, 1998.
[18] A. Smeaton, P. Over, and W. Kraaij, Evaluation campaigns and TRECVID, in: Proceedings of ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR), 2006.
[19] M. Everingham, J. Winn, The PASCALVisual Object Classes Challenge 2007 (VOC 2007) Development Kit Technical Report, University of Leeds, 2007.
[20] Yuxin Peng and Jia Yao. AdaOUBoost: adaptive over-sampling and under-sampling to boost the concept learning in large scale imbalanced data sets, in: Proceedings of the International Conference on Multimedia information retrieval (MIR '10). ACM, New York, NY, USA, 2010, pp. 111–118.
[21] L. Breiman, Bagging predictors, Mach. Learn. 24 (1996) 123–140.
[22] J. Friedman and P. Hall," On bagging and nonlinear estimation. 2000.available at ⟨http://www.stat.stanford.edu/~jhf/⟩, .
[23] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[24] Dacheng Tao, Xiaoou Tang, Xuelong Li, Xindong Wu, Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval, Pattern Anal. Machine Intell. IEEE Trans. 28 (7) (2006) 1088–1099.
[25] P. Kang, S.EUS Cho, SVMs: ensemble of under-sampled SVMs for data imbalance problems, Lect. Notes Comput. Sci. 4232 (2006) 837–846.
[26] Xu-Ying Liu, Jianxin Wu, Zhi-Hua Zhou, Exploratory undersampling for class-imbalance learning, Syst. Man Cybern. B: Cybern. IEEE Trans. 39 (2) (2009) 539–550.
[27] ZhiYong Lin, ZhiFeng Hao, XiaoWei Yang, XiaoLan Liu, Several SVM ensemble methods integrated with under-sampling for imbalanced data learning, Adv. Data Min. Appl. Lecture Notes Comput. Sci. 5678 (2009) 536–544.
[28] C. Seiffert, T. Khoshgoftaar, J. Van Hulse, A. Napolitano, Rusboost: a hybrid approach to alleviating class imbalance, IEEE Trans. Syst., Man Cybern. A: Syst. Hum. 40 (1) (2010) 185–197.
[29] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, Syst. Man Cybern. C: Appl. Rev. IEEE Trans. 42 (4) (2012) 463–484.
[30] Haibo He, E.A. Garcia, Learning from imbalanced data, Knowl. Data Eng. IEEE Trans. 21 (9) (2009) 1263–1284.
[31] J. Zhang and I. Mani, KNN approach to unbalanced data distributions: a case study involving information extraction, in: Proceedings of International Conference on Machine Learning (ICML 2003), Workshop on Learning from Imbalanced Data Sets, 2003.
[32] Y.e.n. Show-Jane, Yue-Shi Lee, Cluster-based under-sampling approaches for imbalanced data distributions, Expert Syst. Appl. 36 (3) (2009) 5718–5727.
[33] K.E.A. van de Sande, T. Gevers, C.G.M. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1582–1596.
[34] M.-Y. Chen and A. Hauptmann, Mosift: recognizing human actions in surveillance videos, 2009.
[35] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, CVPR 2 (2006) 2169–2178.
[36] Yilmaz, Emine, Evangelos Kanoulas, and Aslam Javed A. . A simple and efficient sampling method for estimating AP and NDCG, in: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2008.
[37] B.E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers, in: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, ACM Press, 1992, pp. 144–152.
[38] Héla Zouari, Laurent Heutte, Yves Lecourtier, Controlling the diversity in classifier ensembles through a measure of agreement, Pattern Recognit. 38 (11) (2005) 2195–2199.

Lei Bao, born in 1984, Ph. D. candidate. Her research interests focus on multimedia retrieval and machine learning.

Juan Cao, born in 1980, associate professor. Her research interests focus on multimedia retrieval and large scale social media analysis.

Jintao Li, born in 1962, professor, Ph.D. Supervisor. His major field includes multimedia processing and VR technology.

Yongdong Zhang, born in 1973, professor, Ph.D. Supervisor. His major field includes image processing and video processing.