Multimed Tools Appl (2016) 75:12627–12644
DOI 10.1007/s11042-014-2423-2

Multi-view multi-label learning for image annotation

Fuhao Zou · Yu Liu · Hua Wang · Jingkuan Song · Jie Shao · Ke Zhou · Sheng Zheng

Received: 31 July 2014 / Revised: 12 December 2014 / Accepted: 17 December 2014 /
Published online: 8 January 2015

Abstract Image annotation is posed as multi-class classification problem. Pursuing higher accuracy is a permanent but not stale challenge in the field of image annotation. To further improve the accuracy of image annotation, we propose a multi-view multi-label (abbreviated by MVML) learning algorithm, in which we take multiple feature (i.e., view) and ensemble learning into account simultaneously. By doing so, we make full use of the complementarity among the views and the base learners of ensemble learning, leading to higher accuracy of image annotation. With respect to the different distribution of positive and negative training examples, we propose two versions of MVML: the Boosting and Bagging versions of MVML. The former is suitable for learning over balanced examples while the latter applies to the opposite scenario. Besides, the weights of base learner is evaluated on validation data instead of training data, which will improve the generalization ability of the final ensemble classifiers. The experimental results have shown that the MVML is superior to the ensemble SVM of single view.

Keywords Image Annotation · Ensemble learning · Multi-view learning · Multi-label learning

F. Zou
School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China

Y. Liu (✉) · H. Wang · K. Zhou · S. Zheng
Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan, 430074, China
e-mail: lightyear416@gmail.com

J. Song
Department of Information Engineering and Computer Science, University of Trento, 38100 Trento, Italy

J. Shao
School of Computer Science and Technology, University of Electronic Science and Technology of China, Wuhan, 610054, China

1 Introduction

With the popularity of image capture devices, edit tools and sharing websites, creating, editing and spreading images have become more convenient than ever. These bring us more benefits but lead to explosive growth of the amount of images at the same time [31, 33]. Taking the online image sharing website, Flickr, for example, about 4.5 million photos are uploaded every day. Therefore, managing such large scale images have been a challenging issue. At present, content-based image retrieval is an effective way to manage the large scale image data, which falls under two categories: instance-based and keyword-based methods. The fundamental difference between them is that the former uses an image as query instead of the keywords adopted by the latter. Comparatively speaking, the later is more convenient since offering keywords is easier than that of images. However, the keywords based method requires that the images in the reference database has been annotated. It follows that image annotation plays a significant role in content based image retrieval [28].

The task of image annotation is to assign a set of semantic descriptions to an image that the human has visually perceived. Generally, it can be classified into two categories: hand-crafted and automatic image annotations [25, 26]. As is well known, annotating the image in a hand-crafted way is labor intensive and impractical to handle large scale image data. Therefore, automatic image annotation is an ideal option when the scale of images is quite large. Without specification, the terminology of image annotation refers to the automatic one hereafter. The goal of image annotation is to bridge the gap between the upper-layer semantic abstractions and low-layer visual features [17, 29, 30]. Mathematically, image annotation can be posed as a classification problem, which can be realized in various classification methods, such as support vector machine (SVM) [7, 32], random forest (RS) [4], logistic regression (LR) [14], and neural network (NN) [13], etc. In the machine learning field, pursuing higher classification accuracy has been a permanent but not stale topic. For image annotation , of course, there is no exception.

To further improve the accuracy of image annotation, we intend to simultaneously take the multi-view (i.e., multi-feature) and ensemble learning into account. The motivation of introducing the multi-view learning is as follows. Regardless of how perfect the single feature is, it often has its advantages and disadvantages simultaneously. Adopting multi-feature to characterize an image is to make full use of their complementarity such that the disadvantages of individual feature can be conquered [19]. As to ensemble learning, it has been an effective way to better the performance of a specific classification algorithm. Though a large number of accurate classifiers are available to us, there doesn't exist a single learning algorithm that always yields the most accurate learner in any domain. This is because each learning algorithm is dictated to a certain model with a set of assumptions, resulting in the corresponding model bias [9]. If those assumptions do not fit the data, the model bias leads to error. Through ensemble learning, the uncorrelated errors of individual classifiers can be eliminated by averaging. Consequently, the ensemble classifier has higher accuracy than individual learner. Besides, the generalization ability of an ensemble is usually much superior to that of a single learner [34].

Based on the above-mentioned consideration, we propose an image annotation scheme, also called multi-view multi-label learning (MVML) algorithm. The term of multi-label comes from the fact that each image is probably assigned with multiple labels since it may contains multiple semantic objects. Multi-view means each image is represented by multiple features in this algorithm, each of which refers to a view. The basic idea of the MVML is as follows. For the simplicity of description, we take annotating one label for instance. The SVM algorithm is selected as base learners. For each view, we use ensemble learning

to learn a set of base learners on top of SVM. After this, we ensemble these base learners of all views to yield a fused classifier, which is used to predict a label for an image. In this algorithm, according to the divergence of the ratio between the negative and positive examples, two different ensemble learning methods including boosting and bagging versions are proposed. The boosting version is chosen when the number of the negative and positive examples approximately equals. Otherwise, we prefer to the bagging version. Accordingly, the fusion strategies associated with the two methods are also different. The former takes all base learners into account while the latter only consider the base learners with prediction accuracy bigger than 50 percent. Note that the weights of base learners associated with two versions are all computed based on their performance on the validation data rather than training data. In short, it is worth highlighting the properties of the MVML as follows.

- **Higher accuracy**: Combing three powerful learning methods including SVM, multi-view and ensemble learning helps to make full use of complementarities among views and base learners, leading to higher classification accuracy;
- **Good applicability**: Offering two candidates for ensemble learning according to the divergence of examples distribution will enhance the applicability of the proposed MVML.
- **Superior generalization ability**: For two versions of MVML, evaluating the example weights over the validation data instead of training data is beneficial for improving the generalization ability of the MVML.

The rest of this paper is organized as follows. In Section 2, we review the two key techniques of ensemble learning and SVM. Next, we present our image annotation method named MVML in Section 3. In Section 4, the extensive experiments are conducted to demonstrate the superiority of the proposed algorithm. Finally, we draw a conclusion in Section 5.

2 Related work

In this section, we will review the techniques closely related to this work, such as ensemble learning, and their combination with SVM, also called Ensemble of SVM.

2.1 Ensemble learning

Recall that ensemble learning is not a specific learning method, such as regression, dimension reduction, clustering [27],classification, but a aggregation (or fusion) method over several specific learners. Let's have a look at how the ensemble learning is built over specific learning algorithms. Generally, s specific learning method can be formulated as conducting a searching job in a hypothesis space to obtain a approximately optimal hypothesis that will yield accurate predictions for a specific problem. However, even if there exists a very well-suited hypotheses for such a specific problem in the hypothesis space, it is probably quite difficult to find an ideal one. Ensemble learning combines multiple hypotheses to generate a better hypothesis. That is, ensemble learning is a technique for fusing many base learners to yield a stronger learner. From perspective of statistics or machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than any of the base learning algorithms. Theoretically, it needs infinite number of base learners to approximate the better one. But, in practice, a concrete finite number of base learners is sufficient

to approximate such goal [8]. There are many effective ensemble methods [2, 18]. In the following, we just introduce three representative methods: bagging, boosting, and stacking.

- **Bagging** [3]: Bagging is also called bootstrap aggregating, which trains a number of base learners each from a different bootstrap sample by invoking a base learning algorithm. A bootstrap sample is generated by sub-sampling the training data set with replacement, where the sizes of all bootstrap sample are equal. After obtaining the base learners, Bagging combines them by majority voting and the most-voted class is chosen. By averaging misclassification errors on different bootstrap sample, it can offer a better estimate of the predictive ability of a learning method.
- **Boosting** [10]: Boosting involves incrementally building an ensemble by training each new base learner to emphasize the training examples that have been misclassified by the previous base learner. Sometimes, boosting has been shown to yield better accuracy than bagging, but it also tends to suffer from the over-fitting issue. Thus far, the most popular version of Boosting is Adaboost, although some newer algorithms are reported to achieve better results.
- **Stacking** [21]: Stacking (also called stacked generalization) trains a learning algorithm in two levels. First, all of the first-level algorithms are trained using the available data, then a combiner algorithm (i.e., second-level algorithm) is trained to make a final prediction using all the predictions of the first-level algorithms as additional inputs. Stacking usually performs better than any single one of the trained models. It has been successfully adopted by both supervised learning tasks (i.e., regression) and unsupervised learning tasks(i.e., density estimation).

2.2 Ensemble of SVM

Ensembles of SVM algorithms have been introduced in various domains. Bagging is the most frequently used method for producing ensemble classifiers. Bagging introduces randomness in the training examples. Recently a number of SVM ensemble on top of bagging have appeared. For example, Kim et al. [16] and Yan et al. [23] proposed SVM ensembles based bagging to improve the classification accuracy. Tao et al. [24] proposed a SVM ensemble method based on bagging and random subspace to improve the user relevance feedback performance for content based image retrieval. The experimental results show that above-mentioned Ensemble SVM effectively improve the classification accuracy.

Besides, Valentini et al. [20] presented a low bias SVM ensemble based on bagging. The aim is to reduce bias of the SVMs before performing bagging. To handle the issue that the existing SVM ensembles are computationally intensive especially when the size of training datasets is large, Alham et al. [1] presents a MapReduce based distributed SVM ensemble algorithm for image annotation, which re-samples the training dataset via bootstrapping and trains SVM on each dataset in parallel using a cluster of computers. In literature [22], Zhihua Zhou proposed an ensemble multi-instance multi-label learning method to deal with class imbalance issue frequently occurred in real world data. Experiments show that the proposed method outperforms a number of state of the art methods.

In short, research on SVM ensemble algorithms has been carried out from various perspectives, which mainly focuses on improving classification accuracy. It has been shown that improving the accuracy of SVM ensemble remains an permanent but not stale challenge problem. This motivates us to design the MVML which takes the multi-view and ensemble SVM learning into account simultaneously.

3 The proposed image annotation scheme

Image annotation can be posed as a multi-class classification problem. To further improve the accuracy of annotation, we propose a multi-view multi-label (MVML) learning algorithm to learn a set of binary classifiers, each of which is used to predict a specific label. The general framework is shown in Fig. 1. For the simplicity of introducing, the following only introduces the procedure of generating a specific label classifiers since multi-label annotation can be extended by directly combining multiple single-label classifiers. First, each image is represented by $\mathcal{M}$ features (also known as views). For each view, we train $\mathcal{T}$ base learners on top of SVM using ensemble learning. Next, all base learners from the $\mathcal{M}$ views are combined to form a final classifier, which is used to predict label for an image. The purpose of introducing the multi-views and ensemble learning simultaneously is to make full use of complementarity among the views as well as the base learners, leading to the higher annotation accuracy.

In practice, there exists a divergence between the number of the positive and negative examples. Therefore, we propose two candidates version: Boosting version and Bagging version of MVML. The former is suitable for the case with balanced examples while the latter for that with imbalanced examples, which will be introduced in the following.

3.1 Boosting version of MVML

The boosting version of MVML is mainly based on Adaboosting and SVM. Let $\mathcal{D}_T = \left\{(x_1^t, y_1^t), (x_2^t, y_2^t), ..., (x_n^t, y_n^t)\right\}_{t=1}^{\mathcal{M}}$ denote the training data of $\mathcal{M}$ views. Let $\{W_k^t(i)\}_{i=1}^n$ denote the examples weights of the $t-th$ view and $k-th$ iteration. For the

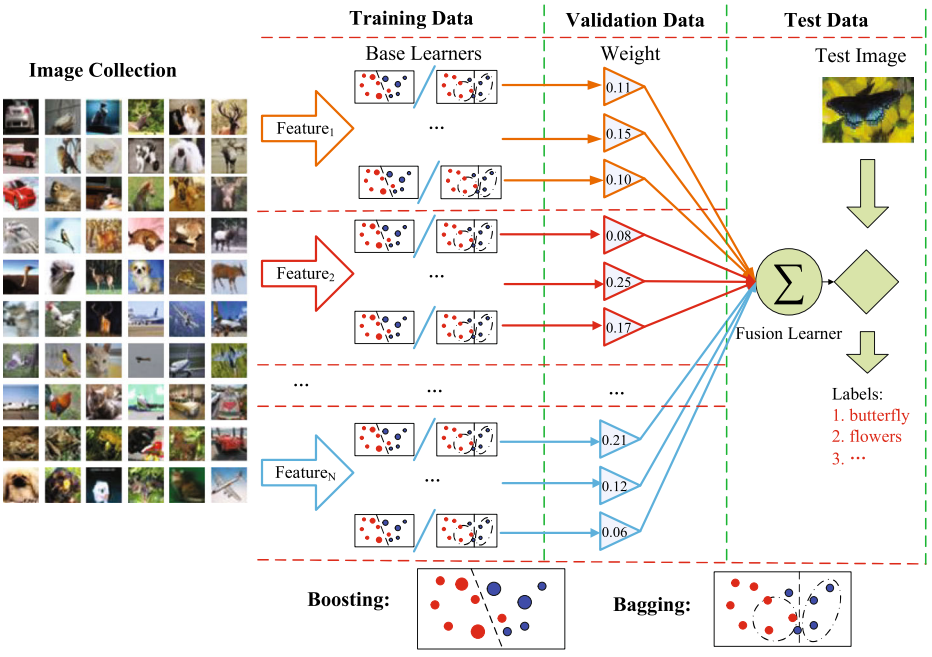


Fig. 1 Illustration of image annotation scheme via MVML

$t-th$ view and $k-th$ iteration, we learn a base learner h_k^t on top of SVM. Given examples $\mathcal{D}_T^{(t)}$ and associated examples weights W_k^t, the base learner h_k^t is trained using modified SVM, whose objective has the form

$$\begin{aligned} &\underset{\mathbf{w}_k^t,\xi_k^t,b_k^t}{argmin}: \frac{1}{2}\|\mathbf{w_k^t}\|^2 + \sum_{i=1}^{n} W_k^t(i)(\xi_i)_k^t \\ &subject\ to: y_i^t(\mathbf{w_k^t}\cdot\mathbf{x_i^t} - b_k^t) \geq 1-(\xi_i)_k^t \\ &(\xi_i)_k^t \geq 0 \end{aligned} \tag{1}$$

where the parameters $\mathbf{w}_k^t, \xi_k^t, b_k^t$ share same definition with parameters $\mathbf{w}, \xi, b$ in SVM [6]. Compared with soft margin version defined in literatures [5, 7], the base learner h_k^t takes the weights of examples into account. Usually, the misclassified examples in last iteration will be assigned larger weights. Accordingly, misclassifying these examples will result in heavier penalty, leading to a classifier with larger margin and higher accuracy. During each iteration, we will calculate the prediction error

$$\epsilon_k^t = \sum_{j=1}^{n} \mathcal{W}_k^t(j) 1\{h_k^t(x_j^t) \neq y_j^t\} \tag{2}$$

and the weight of base learner h_k^t as

$$\alpha_k^t = \frac{1}{2}\ln\frac{1-\epsilon_k^t}{\epsilon_k^t}, \tag{3}$$

where $1\{\cdot\}$ returns 1 if $1\{true\}$ and 0 otherwise. In the sequel, the weights of next iteration is derived as

$$\mathcal{W}_{k+1}^t(j) = \frac{\mathcal{W}_k^t(j)\exp(-\alpha_k^t y_j^t h_k^t(x_j^t))}{\sum_{j=1}^{n}\mathcal{W}_k^t(j)\exp(-\alpha_k^t y_j^t h_k^t(x_j^t))}. \tag{4}$$

Remember that the weights of examples first iteration are initialized as $\mathcal{W}_1^t(j) = \frac{1}{n}$.

Note that if basing on performance of base learner on training data, the final ensemble classifier readily leads to overfiting issue. To avoid this issue, the weights of base learner of MVML is determined according to its prediction ability on validation data rather than training data such that the obtained base learner has stronger generalization ability. Given validation data $\mathcal{D}_V = \{(v_1^t, u_1^t), (v_2^t, u_2^t), ..., (v_m^t, u_m^t)\}_{t=1}^{\mathcal{M}}$, we first compute the perdition error

$$\eta_k^t = \frac{1}{m}\sum_{j=1}^{m} 1\left\{h_k^t(v_j^t) \neq u_j^t\right\}, \tag{5}$$

and then calculate the weight as

$$\beta_k^t = \frac{1}{2}\ln\frac{1-\eta_k^t}{\eta_k^t}. \tag{6}$$

With the obtained based learners and their weights, we can get a ensemble classifier defined as

$$H\left(\left[x^1, x^2, ..., x^{\mathcal{M}}\right]\right) = sign\left(\sum_{t=1}^{\mathcal{M}}\sum_{k=1}^{\mathcal{T}}\beta_k^t h_k^t(x^t)\right), \tag{7}$$

where $\left[x^1, x^2, ..., x^{\mathcal{M}}\right]$ denotes the multiple features of test example.

3.2 Bagging version of MVML

For classification learning task, the imbalance of training data will result in ill-posed classifier. In literature [11, 15], they train a series of base learners over the subset with different proportion of positive and negative samples and then use them to perform classification task. It observes that base learners with balanceable samples play more key role. According to this observation, we propose a bagging version of MVML, in which the balance of training data is preserved by random sample. For convenience, we assume that the notations $\mathcal{D}_T, \mathcal{D}_V, \mathcal{L}, \mathcal{T}$ share the same definition as Boosting version of MVML. For the $t-th$ view and $k-th$ iteration, we start with randomly sampling a subtset $\mathcal{D}_T^{*(t)}$ having balance examples from $\mathcal{D}_{T^{(t)}}$ and learn a base learner $h_k^t = \mathcal{L}(\mathcal{D}_T^{*(t)})$ over $\mathcal{D}_T^{*(t)}$ using SVM. Analog to the Boosting version of MVML, the weights of base learners are also evaluated according the base learners performance on validation data. Let $\mathcal{D}_V = \{(v_1^t, u_1^t), (v_2^t, u_2^t), ..., (v_m^t, u_m^t)\}_{t=1}^{\mathcal{M}}$ denote the validation data. We randomly sample a subtset $\mathcal{D}_V^{*(t)}$ having balanced examples from $\mathcal{D}_V^{(t)}$. Next, we compute prediction error over $\mathcal{D}_V^{*(t)}$ as

$$\epsilon_k^t = \frac{1}{n^*} \sum_{j=1}^{n^*} 1\{h_k^t((v_j^t)^*) \neq (u_j^t)^*\} \tag{8}$$

and the weight of base learner as

$$\alpha_k^t = \frac{1}{2} \ln \frac{1-\epsilon_k^t}{\epsilon_k^t}. \tag{9}$$

In the end, we produce a ensemble classifier as

$$H([x^1, x^2, ..., x^{\mathcal{M}}]) = sign\left(\sum_{t=1}^{\mathcal{M}} \sum_{k=1}^{\mathcal{T}} \alpha_k^t h_k^t(x^t) 1\{\epsilon_k^t < 0.5\}\right), \tag{10}$$

where $\left[x^1, x^2, ..., x^{\mathcal{M}}\right]$ denotes the multiple features of test example. Unlike the Boosting version of MVML, the base learners whose prediction errors are larger than 50 % are not considered in the process of ensemble.

3.3 Complexity analysis

There are two versions of MVML proposed in this work. The detailed complexity analysis is introduced in the following.

For the Boosting version of MVML, let n denote the size of training samples, d_t denote feature dimensions of the $t-th$ view, $\mathcal{T}$ denote the number of iteration of ensemble learning, m denote the size of validation samples, and $\mathcal{M}$ denote the number of views (or features). In each iteration, the time complexity of training base learner using SVM is $O(n^{2.2})$ [5]. The time complexity of computing prediction error and weights of base learners over training and validation data is totally $O(2(n+m))$. After $\mathcal{T}$ iterations over $\mathcal{M}$ views, the time complexity of training ensemble classifier is $\mathcal{O}(\mathcal{MT}(n^{2.2} + 2(n+m)))$. The time complexity of predicting a label for test image is $\mathcal{O}\left(\sum_{t=1}^{\mathcal{M}} d_t\right)$. During the training procedure, we have to store training data of size $d_t \times n$ and validation data of size $d_t \times m$ for the $t-th$ view. The space cost for training SVM is quadratic in the size of training data [7] and then its space cost is $\mathcal{O}\left(\sum_{t=1}^{\mathcal{M}} (nd_t)^2\right)$. Apart from this, the space complexity of storing the medium variables

such as prediction error and weights of base learners is $\mathcal{O}(4n)$. Totally, the space complexity is $\mathcal{O}\left(\sum_{t=1}^{\mathcal{M}}((nd_t)^2 + md_t) + 4n\right)$. During the test procedure, the space complexity is $\mathcal{O}\left(\sum_{t=1}^{\mathcal{M}} d_t\right)$.

For the Bagging version of MVML, let n^* and m^* denote the size of subset randomly sampling from training data and validation data,$\mathcal{T}$ denote the number of iteration of ensemble learning, $\mathcal{M}$ denote the number of views (or features), and d_t denote the dimensions of $t-th$ feature. For each iteration, the time complexity is $\mathcal{O}((n^*)^{2.2} + 2m^*)$. Totally, the time complexity of training a ensemble classifier is $\mathcal{O}(\mathcal{M}\mathcal{T}((n^*)^{2.2} + 2m^*))$. Similar to the Boosting version, the time complexity for prediction a label also is $\mathcal{O}\left(\sum_{t=1}^{\mathcal{M}} d_t\right)$. The space complexity for training a ensemble classifier is $\mathcal{O}\left(\sum_{t=1}^{\mathcal{M}}((n * d_t)^2 + m * d_t\right)$. The space complexity of the test procedure is $\mathcal{O}\left(\sum_{t=1}^{\mathcal{M}} d_t\right)$.

4 Experiment

To fairly evaluate the proposed method, without loss of generality, three bag model-like features: SIFT-based BOF, VLAD and PHOG are selected to extract features from each image. We conduct a series of experiments to verify the performance of the proposed MVML in terms of annotation accuracy.

4.1 Experiment setting

The experiments are operated on benchmark data sets of LabelMe and VOC2012. The short description and preprocessing of each data set is as follows:

LabelMe: LabelMe is a image databases for computer vision research which is built by online annotation tool. There are 50000 photos presenting 12 different objects including person, cars, building, window, tree, sign, door, bookshelf, chair, table, keyboard and head. We randomly choose 4000 images as training set and the rest as testing set. This data set is to verify the performance of Boosting version of MVML, which requires the training and validation data with balanced examples. To that end, the image set that doesn't satisfy this requirement has been removed. After this processing, only three kinds of labeled data: person, cars or window are left. The final data set contains 12723 samples, in which 4000 examples are randomly selected as training set, 3000 examples are randomly selected as validation set, and the other examples are used as test examples.

VOC2012: VOC2012 is abbreviation of Visual Object Classes Challenge 2012. There are 17125 photos in its devkit package of VOC2012 with different size. This data set is used to demonstrate that Bagging version of MVML can perform well in the data set with imbalanced examples. To handle the issue of imbalanced examples, we adopt randomly sampling strategy to preserving the balance of final training and validation data, which is used to training base learners. In the end, there are 4585 photos selected for experiment. Three labeled images including walking, reading and ridinghorse, are selected, which

Table 1 The setting of the constructed training set, validation set and test set over VOC2012

Label	Training set		Validating set		Testing set	
	Positive	Negative	Positive	Negative	Positive	Negative
Walking	300	300	250	250	378	4207
Reading	400	400	300	300	463	4122
Ridinghorse	350	350	250	250	408	4177

only have 378, 463 and 408 positive samples respectively. The setting of constructed subset of training set, validation set and testing set of each lablel are listed in Table 1.

We perform k-means for BOF with 2400 centers and 32 centers for VLAD. In the meanwhile, we set *bin*=8, *level*=3 and *angle*=360 for PHOG. After feature extraction, we obtain BOF, VLAD, and PHOG features with 2400, 3200 and 680 dimensions respectively.

4.2 Performance of the proposed method

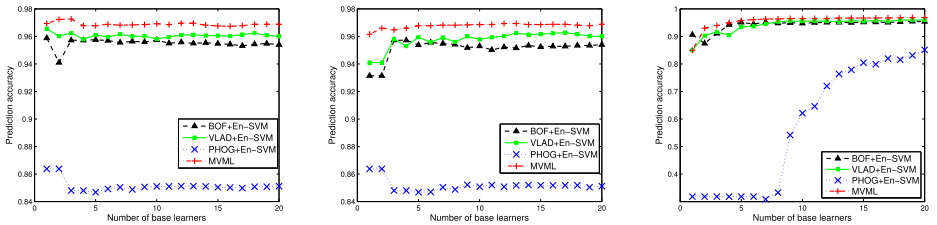
4.2.1 Results on LabelMe

In this section, two kinds of experiments will be conducted. First, we compare the performance between muli-view and single-view of Boosting version. Note that single view stands for ensemble over single view. Another type of experiment is to check whether the ensemble of base learners with higher weights have better performance. For this purpose, we construct three variants of Boosting version: Sequential order, Ascendent order, and Descendent order, which can be constructed as follows. We begin by conducting K iterations to get $\mathcal{K}$ base learners. Following, we select first $\mathcal{T}$ ($\mathcal{T} < \mathcal{K}$) base learners according to their sequential order, ascendent order, and descendent orders. In other words, the Sequential variant refers to ensemble of the first $\mathcal{T}$ came out base learners while the Ascendent and Descendent variants refer to the ensemble of the base learners associated with Top $\mathcal{T}$ weights in Ascendent and Descendent order respectively.

Within the experiment, we set $\mathcal{T} = 20$ and $\mathcal{K} = 35$. Table 2 shows all details of experimental results on LabelMe. Remember that BOF+En-SVM, VLAD+En-SVM, and PHOG+En-SVM refer to ensemble of SVM over features BOF, VLAD, and PHOG respectively. In addition, notations of 'seq', 'desc', and 'asc' are the abbreviations of Sequential order, Ascendent order, and Descendent order respectively. From Table 2, it can be observed that the Boosting version of MVML has higher average annotation accuracy while keeping smaller standard deviation than any of single views. Besides, we find that the sequential variants is superior to ascendent and descendent variants. This means that the ensemble of base learners with high prediction accuracy doesn't leads to best ensemble classifiers. This phenomenon reflects that sequential order of selecting base learners helps to make full use of the complementarity more than other two methods.

Table 2 Annotation accuracy results over LabelMe datasets among the Boosting variants and Boost version of MVML

Label	Method		Mean±Std	Max AC	Min AC
Person	BOF+En-SVM	seq	0.9550±0.0036	0.9588	0.9410
		desc	0.9513±0.0070	0.9572	0.9314
		asc	0.9423±0.0207	0.9544	0.8746
	VLAD+En-SVM	seq	0.9607±0.0016	0.9656	0.9580
		desc	0.9577±0.0062	0.9628	0.9410
		asc	0.9412±0.0280	0.9600	0.8498
	PHOG+En-SVM	seq	0.8514±0.0044	0.8638	0.8468
		desc	0.8518±0.0045	0.8638	0.8468
		asc	0.5770±0.2288	0.8512	0.3084
	MVML	seq	0.9689±0.0014	0.9674	0.9726
		desc	0.9677±0.0019	0.9694	0.9616
		asc	0.9558±0.0269	0.9688	0.8498
Cars	BOF+SVM	seq	0.9429±0.0076	0.9490	0.9128
		desc	0.9401±0.0096	0.9482	0.9128
		asc	0.9363±0.0086	0.9438	0.9050
	VLAD+En-SVM	seq	0.9490±0.0036	0.9598	0.9432
		desc	0.9412±0.0086	0.9482	0.9214
		asc	0.9435±0.0045	0.9616	0.9568
	PHOG+En-SVM	seq	0.8955±0.0030	0.9022	0.8906
		desc	0.8877±0.0097	0.8950	0.8670
		asc	0.8945±0.0021	0.8964	0.8866
	MVML	seq	0.9608±0.0007	0.9630	0.9598
		desc	0.9583±0.0040	0.9612	0.9456
		asc	0.9595±0.0013	0.9494	0.9344
Window	BOF+En-SVM	seq	0.9615±0.0028	0.9680	0.9544
		desc	0.9568±0.0057	0.9620	0.9418
		asc	0.9598±0.0061	0.9636	0.9346
	VLAD+En-SVM	seq	0.9642±0.0017	0.9682	0.9618
		desc	0.9605±0.0047	0.9638	0.9478
		asc	0.9474±0.0297	0.9634	0.8398
	PHOG+En-SVM	seq	0.8652±0.0038	0.8762	0.8632
		desc	0.8665±0.0036	0.8762	0.8634
		asc	0.4431±0.2938	0.8642	0.1882
	MVML	seq	0.9720±0.0007	0.9740	0.9706
		desc	0.9698±0.0020	0.9714	0.9626
		asc	0.9673±0.0082	0.9716	0.9346



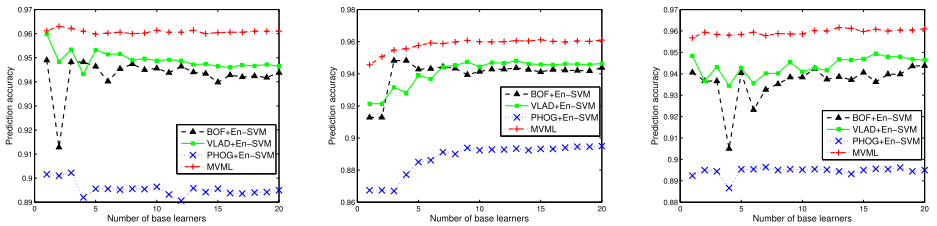
(a) seq (b) desc (c) asc

Fig. 2 The prediction accuracy of person label in LableMe using Boosting version of MVML and three single-view Boosting version

Under different setting of combination of views and selection order of base learners, we test the annotation accuracy by varying the number of iterations. Figures 2, 3 and 4 have shown the performance of different settings with respect to the task of annotating person, cars, and window respectively. The above three figures have shown that the accuracy of annotation increases with the number of base learners and the ensemble of all versions approximate optimum status when number of base learners equals 20. The observation is fully consistent with the principle behind the ensemble learning, which can decease the invariance by increasing the number of base learners.

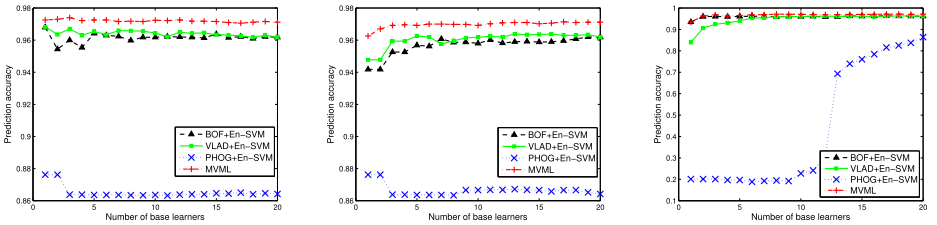
4.2.2 Results on VOC2012

In this section, we mainly test the advantage of Bagging version of MVML over imbalanced data. The setting of data for Bagging version of MVML is listed in Table 1. At the same time, we also evaluate the Boosting version of MVML and single view over VOC2012. Table 3 shows all the details of experimental results on VOC2012 data set. It has shown that the Bagging version of MVML is superior to the Boosting version of MVML by up to about 25 percent over VOC2012. This means that the Bagging version of MVML can be viewed as an effective alternative to Boosting version of MVML over the imbalanced data. In addition, among all settings, the MVML performs better than any single view version.



(a) seq (b) desc (c) asc

Fig. 3 The prediction accuracy of cars label in LableMe using Boosting version of MVML and three single-view Boosting version



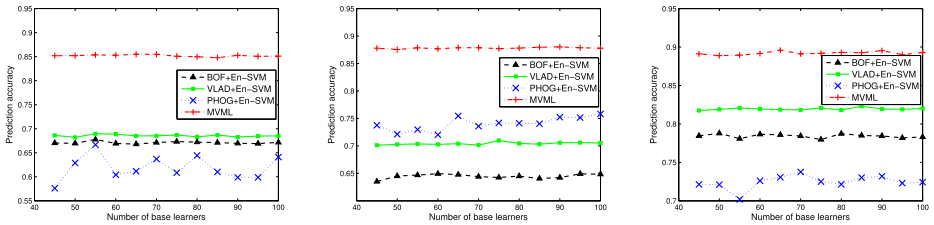
(a) seq (b) desc (c) asc

Fig. 4 The prediction accuracy of window label in LableMe using Boosting version of MVML and three single-view Boosting version

Also, we test the performance of Bagging version of MVML by changing the number of base learners. Figure 5 shows the results of prediction accuracy of walking, reading and ridinghorese label under different settings. It can be seen that the bagging version of MVML approximates the optimum status when the number of base learners equals 50. When the number of base learners is larger than a threshold, say 50, the bagging version of MVML can hardly increase the accuracy further.

Table 3 Annotation accuracy results on VOC2012 by Boosting and Bagging versions of MVML as well as variants of Boosting and Bagging

Label	Method		Mean±Std	Max AC	Min AC
Walking	Boosting	BOF+En-SVM	0.7136±0.0303	0.7852	0.6728
		VLAD+En-SVM	0.7211±0.0417	0.8251	0.6696
		PHOG+En-SVM	0.5617±0.0440	0.6321	0.4349
		MVML	**0.7222±0.0163**	**0.7579**	**0.6829**
	Bagging	BOF+En-SVM	0.6711±0.0025	0.6776	0.6680
		VLAD+En-SVM	0.6855±0.0023	0.6896	0.6820
		PHOG+En-SVM	0.6188±0.0252	0.6667	0.5760
		MVML	**0.8519±0.0021**	0.8550	**0.8478**
Reading	Boosting	BOF+En-SVM	0.7144±0.0519	0.8079	0.6113
		VLAD+En-SVM	0.7136±0.0107	0.7269	0.6755
		PHOG+En-SVM	0.6130±0.0774	0.7003	0.5093
		MVML	**0.7344±0.0147**	**0.7679**	**0.7101**
	Bagging	BOF+En-SVM	0.6447±0.0041	0.6493	0.6351
		VLAD+En-SVM	0.7042±0.0023	0.7097	0.7012
		PHOG+En-SVM	0.7404±0.0124	0.7581	0.7202
		MVML	**0.8781±0.0013**	**0.8803**	**0.8755**
Ridinghorse	Boosting	BOF+En-SVM	0.6548±0.0169	0.6883	0.6253
		VLAD+En-SVM	0.6496±0.0108	0.6595	0.6076
		PHOG+En-SVM	0.4929±0.0552	0.5802	0.3965
		MVML	**0.6582±0.0069**	**0.6680**	**0.6321**
	Bagging	BOF+En-SVM	0.7842±0.0025	0.7878	0.7797
		VLAD+En-SVM	0.8195±0.0015	0.8229	0.8174
		PHOG+En-SVM	0.7245±0.0088	0.7376	0.7019
		MVML	**0.8920±0.0021**	**0.8957**	**0.8888**



(a) Walking (b) Reading (c) Ridinghorse

Fig. 5 The prediction accuracy of different label in VOC2012 using Bagging version of MVML and Boosting version of Single view

4.2.3 Comparison between the proposed method with MKL-SVM

To verify the superior ability of feature fusion of the proposed method, the multiple kernel learning-SVM (abbrivated by MKL-SVM) [12] has been selected as comparison object. Note that the MKL-SVM is regarded as state-of-the-art work in multiple kernel learning field. In three obtained feature sets (such as BOF, VLAD and PHOG), the radial basis function (RBF) is exploited as kernel function. The parameters of MKL-SVM are tuned such that it approximates the optimal status. Recall that here the training and validation examples are balanced for boosting version and MKL-SVM via random sampling, leading to equal number of negative and positive examples for boosting version and MKL-SVM. The result is listed in Table 4.

According to the results, it can be observed that Boosting version totally performs best over the LableMe data (which have more balance training and validation data after example balancing) and the second rank is MKL-SVM. And, over the VOC2012, which have less balance training and validation data, Boosting version is winner most of the time. Though the MKL-SVM stably locates in medium place among three algorithms, the boosting and bagging versions can defeat it in having more and less balance data cases respectively.

Table 4 Annotation accuracy results among Boosting and Bagging versions, as well as MKL-SVM

	Boosting Version	Bagging Version	MKL-SVM	
LableMe	Person	0.9696	0.9622	0.9125
	Cars	**0.9606**	0.8251	0.9033
	Window	0.9726	**0.9737**	0.9075
	Building	**0.9633**	0.8734	0.9081
	Tree	**0.9617**	0.8937	0.9027
	Sign	**0.9315**	0.8140	0.8817
VOC2012	Walking	0.7725	**0.8534**	0.8332
	Reading	0.7280	**0.8179**	0.8022
	Ridinghorse	0.6574	**0.8840**	0.8680
	Jumping	0.6970	**0.8813**	0.7255
	UsingComputer	0.7136	**0.8257**	0.8007
	phoning	0.7257	0.8181	**0.8632**

5 Conclusions

In this paper, we propose a MVML leaning algorithm for image annotation. According to distribution of positive and negative samples, we provide Boosting and Bagging version of MVML. It is worth noting that the two variants are not the simple extension of Adaboost and Bagging. The contributions are as follows.

– Each image to be annotated is characterized as multiple features (i.e., views), which help to take advantage of complementarity among features.
– Bagging version of MVML can handle the challenge arisen by imbalance of positive and negative samples in training data, making the applicability of MVML much broader.
– When fusing the base learners, their weights are determined by the performance of base learners on validation data instead of training data, avoiding the overfitting issue.

During fulfilling this work, we find that ensemble learning can be used to handle lots of issues of machine learning algorithm. In the future, we will enlarge the power of ensemble learning in more areas.

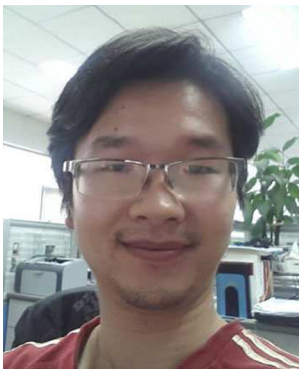
Acknowledgment This work is supported in part by the National Basic Research Program (973 Program) of China under Grant No. 2011CB302305, the National Natural Science Foundation of China under Grant No. 61232004. The authors appreciate the valuable suggestions from the anonymous reviewers and the Editors.

References

1. Alham NK, Li M, Liu Y, Ponraj M, Qi M (2012) A distributed SVM ensemble for image classification and annotation. 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), IEEE, pp 1581–1584
2. Bauer E, Kohavi R (1999) An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning. Springer 36(1-2):105–139
3. Breiman L (1996) Bagging predictors. Mach Learn, Springer 24(2):123–140
4. Breiman Leo V (2001) Random forests. Mach Learn 45(1):5–32
5. Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Disc, Springer 5(2):121–167
6. Chang C-C, Lin C-J (2011) LIBSVM: A library for support vector machines. ACM Trans Intell Syst Technol 2(3):1–27
7. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
8. Dietterich TG (2000) Ensemble methods in machine learning, Multiple classifier systems, pp 1–15
9. Dietterichl TGS (2002) Ensemble learning. The handbook of brain theory and neural networks, pp 405–408
10. Freund Y, Schapire RE (1995) A desicion-theoretic generalization of on-line learning and an application to boosting, Computational learning theory. Springer, pp 23–37
11. Galar M, Alberto F, Tartas EB, Sola HB, Herrera F (2012) A review on ensembles for the class imbalance problem: Bagging-, Boosting-, and Hybrid-Based approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C, IEEE, pp 463–484
12. Gonen M, Alpayd E (2011) multiple kernel learning algorithms. The Journal of Machine Learning Research. JMLR.org, pp 2211–2268
13. Haykin S (2004) A comprehensive neural networks. Neural Netw 2(2004)
14. Hosmer DW, Lemeshow S, Sturdivant RX (2000) Introduction to the logistic regression model, Wiley Online Library Inc.
15. Khoshgoftaar TM, Van Hulse J, Napolitano A (2011) comparing boosting and bagging techniques with noisy and imbalanced data. IEEE Transactions on Systems, Man, and Cybernetics, Part A, pp 552–568
16. Kim H-C, Pang S, Je H-M, Kim D, Bang S-Y (2002) Support vector machine ensemble with bagging, Pattern recognition with support vector machines. Springer, pp 397–408

17. Muda Z (2007) classification and image annotation for bridging the semantic gap. In: Proceedings of the summer school on multimedia semantics, vol 2007, pp 15–21
18. Sewell M (2008) Ensemble learning. RN, Citeseer 11(2):1–15
19. Song J, Yang Y, Huang Z, Shen HT, Hong R (2011) Multiple feature hashing for real-time large scale near-duplicate video retrieval. In: Proceedings of the 19th ACM international conference on multimedia, pp 423–432
20. Valentini G, Dietterich TG (2003) Low bias bagged support vector machines, ICML, pp 752–759
21. Wolpert DH (1992) Stacked generalization. Neural Netw Elsevier 5(2):241–259
22. Xu X-S, Xue X, Zhou Z-H (2011) Ensemble multi-instance multi-label learning approach for video annotation task. In: Proceedings of the 19th ACM international conference on multimedia. ACM, pp 1153–1156
23. Yan G, Ma G, Zhu L (2006) Support vector machines ensemble based on fuzzy integral for classification. Advances in Neural Networks-ISNN 2006. Springer, pp 974–980
24. Yan G, Ma G, Zhu L (2006) Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE, pp 1088–1099
25. Yang Y, Huang Z, Yang Y, Liu J, Shen HT, Luo J (2013) local image tagging via graph regularized joint group sparsity. Pattern Recogn, Elsevier Sc Inc 46(5):1358–1368
26. Yang Y, Yang Y, Huang Z, Shen HT (2011) Tag localization with spatial correlations and joint group sparsity. In: proceedings of IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 881–888
27. Yang Y, Yang Y, Shen HT, Zhang Y, Du X, Zhou X (2013) discriminative nonnegative spectral clustering with out-of-sample extension. IEEE Trans Data Knowl Eng (TKDE) 25(8):1760–1771
28. Yang Y, Zha Z-J, Gao Y, Zhu X, Chua T-S (2014) exploiting web images for semantic video indexing via robust sample-speci?c loss. IEEE Trans Multimed 16(6):1677–1689
29. Zhang L, Gao Y, Xia Y, Dai Q, Li X (2014) a fine-grained image categorization system by cellet-encoded spatial pyramid modeling. Transactions on Industrial Electronics, IEEE, pp 1-8
30. Zhang L, Han Y, Yang Y, Song M, Yan S, Tian Q (2013) discovering discriminative graphlets for aerial image categories recognition. IEEE Transactions on Image Processing, pp 5071–5084
31. Zhang L, Song M, Zhao Q, Liu X, Bu J, Chen C (2013) IEEE, probabilistic graphlet transfer for photo cropping. IEEE Trans Image Process 22(2):802–815
32. Zhang L, Yi Y, Gao Y, Yu Y, Wang C, Li X (2014) A probabilistic associative model for segmenting weakly supervised images. IEEE Trans Image Process 23(9):4150–4159
33. Zhang L, Xia Y, Ji R, Li O (2014) IEEE, spatial-aware object-level saliency prediction by learning graphlet hierarchies. IEEE Trans Ind Electron 99:1–8
34. Zhou Z-H (2009) Ensemble learning. Encyclopedia of Biometrics. Springer, pp 270–273

Fuhao Zou received B.E. degree in computer science from Huazhong Normal University, Wuhan, Hubei, China, in 1998. And received M.S. and Ph.D. in computer science and technology from Huazhong University of Science and Technology (HUST), Wuhan, Hubei, China, in 2003 and 2006. Currently, he is an associate professor with the college of computer science and technology, HUST. His research interests include machine learning, multimedia understanding and analysis, big data analysis, semantic based storage, and cloud storage. He is senior member of China Computer Federation (CCF) and member of IEEE, ACM.



Yu Liu received the B.S. degree and the M.S. degree from Computer Science and Technology, Wuhan Institute of Technology, Wuhan, China. He is currently pursuing the Ph.D at Huazhong University of Science and Technology, Wuhan, China. His current research interests include machine learning,big data and storage etc.

Hua Wang received the BE, ME, and PhD degrees in computer science and technology from Huazhong University of Science and Technology (HUST), Wuhan, China. She is a lecturer of Wuhan National Laboratory for Optoelectronics, HUST. Her specific interests include computer architecture and cloud storage.

Jingkuan Song received his PhD degree in Information Technology from The University of Queensland, Australia. He received his BS degree in Software Engineering from University of Electronic Science and Technology of China. Currently, he a postdoctoral researcher in the Dept. of Information Engineering and Computer Science, University of Trento, Italy. His research interest includes large-scale multimedia search and machine learning.

Jie Shao is a professor at the School of Computer Science and Engineering, University of Electronic Science and Technology of China. He received a Ph.D. degree from the University of Queensland, Australia in 2009, and a bachelor degree from Southeast University, China in 2004, both in computer science. From 2009 to 2013 he was a research fellow in the University of Melbourne, Australia and National University of Singapore respectively. His research interests include multimedia information retrieval as well as spatial databases and their applications.

Ke Zhou received the BE, ME, and PhD degrees in computer science and technology from Huazhong University of Science and Technology (HUST), China, in 1996, 1999, and 2003, respectively. He is a full professor of the School of Computer Science and Technology, HUST. His main research interests include computer architecture, cloud storage, parallel I/O and storage security. He has more than 50 publications in journals and international conferences, including Performance Evaluation, FAST, MSST, ACM MM, SYSTOR, MASCOTS and ICC. He is a member of IEEE.

Sheng Zheng received the M.S. degree and the PH.D from School of Electronic Information of Wuhan University, Wuhan, China. His current research interests include system structure and distributed storage system etc.