



## ISABOOST: A weak classifier inner structure adjusting based AdaBoost algorithm—ISABOOST based application in scene categorization

Xueming Qian<sup>a,b,\*</sup>, Yuan Yan Tang<sup>b</sup>, Zhe Yan<sup>a</sup>, Kaiyu Hang<sup>a</sup>

<sup>a</sup> School of Electronic and Information Engineering, Xi'an Jiaotong University, Xianning Road, Xi'an, China

<sup>b</sup> Faculty of Science and Technology, The University of Macau, Macau, China

### ARTICLE INFO

#### Article history:

Received 17 November 2011

Received in revised form

20 September 2012

Accepted 22 September 2012

Available online 27 October 2012

#### Keywords:

AdaBoost

Scene categorization

Pattern classification

Back-propagation networks

SVM

Weight learning

### ABSTRACT

AdaBoost algorithms fuse weak classifiers to be a strong classifier by adaptively determine fusion weights of weak classifiers. In this paper, an enhanced AdaBoost algorithm by adjusting inner structure of weak classifiers (ISABOOST) is proposed. In the traditional AdaBoost algorithms, the weak classifiers are not changed once they are trained. In ISABOOST, the inner structures of weak classifiers are adjusted before their fusion weights determination. ISABOOST inherits the advantages of the AdaBoost algorithms in fusing weak classifiers to be a strong classifier. ISABOOST gives each weak classifier a second chance to be adjusted stronger. The adjusted weak classifiers are more contributive to make correct classifications for the hardest samples. To show the effectiveness of the proposed ISABOOST algorithm, its applications in scene categorization are evaluated. Comparisons of ISABOOST and AdaBoost algorithms on three widely utilized scene datasets show the effectiveness of ISABOOST algorithm.

Crown Copyright © 2012 Published by Elsevier B.V. All rights reserved.

### 1. Introduction

Generally speaking, it is hard to train a very robust classifier [1–15,23–27]. It is very easy to train dozens of weak classifiers with their performances are just better than random guessing. How to fuse dozens of weak classifiers to be a strong one is as important as to train a single strong classifier. AdaBoost algorithms can fulfill this function by assigning appropriate weights for the weak classifiers [31,32,35–37,39]. AdaBoost algorithms turn out to be very effective in machine learning and pattern recognition. In AdaBoost algorithms, weak classifiers are not expected to be perfect. Many of them are just better than random guessing. Weak classifiers can be fused to be a strong one by properly determining their fusion weights [31,32,35–37,39]. Usually, the effective classifiers have large weights and the poor classifiers have small weights. The weights of weak classifiers are determined by AdaBoost algorithms with respect to their error rates to the weighted training samples [31,32,35–37,39]. The incorrectly classified samples by the previous classifiers are given large weights during re-weighting. The training samples with

larger weights are harder to be correctly classified by the previous classifiers. This will allow the rest classifiers focusing on the hardest samples. AdaBoost algorithms do not expect that all the classifiers are robust enough but expect each classifier should have some contribution to make correct classifications for the hardest samples [31].

Li and Zhang proposed FloatBoost algorithm and applied it to multi-view face detection [41]. FloatBoost uses a back-track mechanism in each iteration of AdaBoost learning to minimize the error rate. FloatBoost selects the best weak classifier by a stage-wise approximation of the posterior probability. Thus, compared to the conventional AdaBoost algorithms, FloatBoost achieves lower error rates by fewer weak classifiers.

Tieu and Viola [42] used boosting methods for feature selection and applied them in image retrieval. Each feature corresponds to a weak classifier. In the training proceeds, the algorithm selects a new feature, determines its weights and adds it to the ensemble. Finally, a strong classifier is computed as weighted linear combination of the weak classifiers.

Lu et al. [43] proposed an i.Boosting approach which integrating feature re-weighting into AdaBoost. It merges AdaBoost, feature re-weighting and relevance feedback into a unified framework. In i.Boosting, not only the samples but also the feature elements are weighted. The basic idea of often utilized AdaBoost algorithms is training sample re-weighting.

The training sample reweighting can be carried out by resampling [44,45]. In re-sampling based AdaBoost approaches, the

\* Corresponding author at: School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China. This work is partly performed when Xueming Qian is in his visiting at The University of Macau.

E-mail address: [qianxm@mail.xjtu.edu.cn](mailto:qianxm@mail.xjtu.edu.cn) (X. Qian).

<sup>1</sup> This work is partly performed when Xueming Qian is in his visiting at The University of Macau.

training data is resampled with replacement to generate a new training set according to the weights distribution of the training examples. Examples with larger weights are more likely to be selected and examples with small weights are less likely to be selected for weak classifiers' training. Seiffert et al. [45] systematically evaluated the resampling and reweighting approaches utilized in nine AdaBoost algorithms. Their empirical experimental results show that the resampling based AdaBoost approaches outperform the corresponding reweighting based AdaBoost approaches.

AdaBoost algorithms can be implemented online or offline [37]. The online AdaBoost algorithms train weak classifiers and determine their fusion weights in a unified framework using the same training set [31,32]. The block diagram of online AdaBoost algorithm is shown in Fig. 1(a). Online AdaBoost algorithm calls weak learner to train a weak classifier with respect to the distribution of weighted training samples. In the offline AdaBoost algorithm, weak classifiers are trained before assigning their fusion weights [35–37]. The block diagram of offline AdaBoost algorithm is shown in Fig. 1(b). The difference of online and offline trainings are clear by comparing Fig. 1(a) and (b). The online algorithm trains the weak classifiers and determines their fusion weights using the same weighted training samples. There are two independent training sets in the offline AdaBoost algorithms. One is the weak classifiers training set and the other is weak classifiers' weights learning set. The offline algorithm works iteratively by selecting the best weak classifier with respect to the error rate of the weight training samples [35–37].

In the offline and online AdaBoost algorithms, the inner structures of the weak classifiers do not changed once they are trained. AdaBoost algorithms only determine the fusion weights of weak classifiers. However, if weak classifiers are not well trained, then they influence the performances of AdaBoost algorithms significantly. This is often happened in the applications of multi-class pattern classification problem [37]. Multi-class pattern classification is completed by cascading several component classifiers. Each component classifier carries out discrimination of some patterns from the others [29].

The on-line boosting algorithms train weak classifiers according to the weighted training samples. Oza and Russell had pointed out that [46], if off-line and on-line boosting are given the same training set, then the weak classifiers returned by on-line boosting converges statistically to the one obtained by off-line boosting as the number of iterations tend to be infinite. Therefore, for repeated presentation of the training set on-line boosting and off-line boosting deliver the same result. The on-line algorithm

requires that the number of weak classifiers is fixed at the beginning. In the off-line AdaBoost all samples are used to determine the fusion weight of weak classifiers. In the on-line AdaBoost one sample is used to train all weak classifiers and determine the corresponding weights. An on-line boosting based feature selection framework is proposed [47]. It allows on-line feature selection using boosting. The basic idea is that the difficulties of samples can be estimated by propagating them through the set of weak classifiers. Given a fixed set of selectors, they are updated by estimating the probability distributions of training samples and generated hypothesis when a new training sample arrives. The weak classifier with the smallest error is selected by the selector.

The motivation of ISABOOST is based on the fact that if inner structures of weak classifiers can be adjusted, then the influences of the poor component classifiers to AdaBoost can be reduced. In this paper, the inner structures of weak classifiers are adjusted before assigning their outer weights. Thus the poor weak classifiers have some chances to be adjusted more contributive in combining a stronger classifier. The contributions of this paper are summarized as follows: (1) the proposed ISABOOST algorithm improve traditional AdaBoost algorithm by adjusting the inner structures of the trained weak classifiers adaptively before assigning their fusion weights; (2) the proposed ISABOOST algorithm iteratively selects an optimal weak classifier after inner structure adjusting and determines its weight in a unified optimization framework; (3) ISABOOST based application in scene categorization is proposed; (4) Impacts of the type of weak classifier to ISABOOST algorithms are evaluated. Given the total number of training samples, the impacts of the assignment of training samples numbers for weak classifier training, inner structure adjusting and adjusting validation, and fusion weight determination are evaluated. These are helpful for showing the effectiveness of the proposed ISABOOST algorithm and provide some guidelines for using ISABOOST.

This paper is extended from our previous works [39]. The improvements of it over our previous work are as follows: (1) more experiments are conducted which make the conclusion more confidence. (2) in our previous work, the type of weak classifier is back-propagation networks (BPN), while in this paper, not only BPN but also support vector machines (SVM [28,38]) are served as weak classifiers.

The rest of this paper is organized as follows: In Section 2 the proposed ISABOOST algorithm is presented. In Section 3 ISABOOST based application in scene categorization is illustrated in detail. In Section 4 experiments and discussions are given. Finally, conclusions are drawn in Section 5.

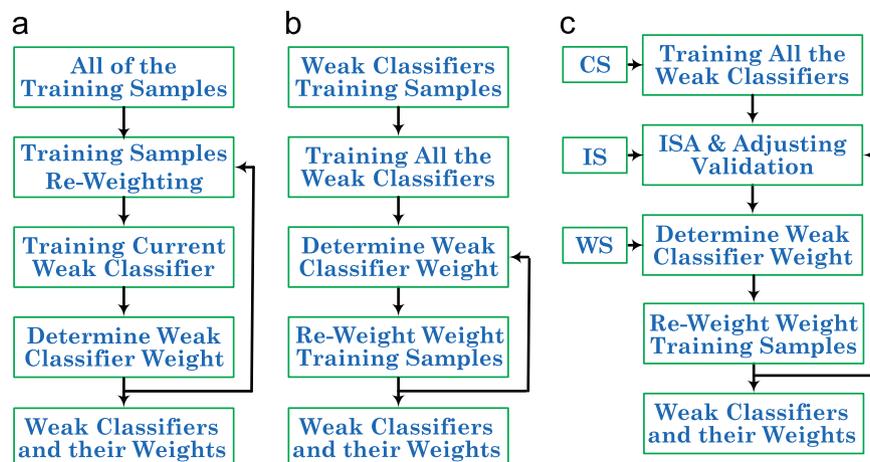


Fig. 1. Diagrams of (a) online AdaBoost, (b) offline AdaBoost and (c) ISABOOST.

## 2. The proposed ISABoost algorithm

In this section, we firstly brief overview the traditional AdaBoost algorithms for multi-class pattern classification. Then the proposed ISABoost algorithm is illustrated in details. Block diagrams of the traditional online AdaBoost, offline Adaboost and the ISABoost algorithms are shown in Fig. 1(a)–(c), respectively. For the online AdaBoost algorithm, a weak classifier is trained and its fusion weight is determined on the weighted training samples at a step. The training samples are utilized both for training weak classifiers and determining their fusion weights. For the offline AdaBoost algorithm, the weak classifiers and the fusion weights are determined using two independent training sets. The weak classifier training set (denoted CS) is utilized to train a set of weak classifiers and the fusion weights training set (denoted WS) is utilized to determine the fusion weights of learned weak classifiers. The proposed ISABoost algorithm consists of four parts: (1) weak classifiers training; (2) inner structure adjusting for the trained weak classifiers and adjusting validation; (3) optimal weak classifier selection and fusion weights determination; (4) training samples re-weighting.

The unfolded drawings of the AdaBoost algorithm for multi-class pattern classification is shown in Fig. 2(a). The  $N$ -Class pattern classification problem can be fulfilled by cascading  $N$  one-versus-all component classifiers. Assuming that  $M$  weak classifiers (denoted Weak  $C_k$ ,  $k = 1, \dots, M$ ) are trained and their outer weights  $\alpha_k$  ( $k = 1, \dots, M$ ) are determined. Each weak classifier consists of  $N$  component classifiers (denoted CompC $_n$ ,  $n = 1, \dots, N$ ). In this paper, the component classifier CompC $_n$  ( $n = 1, \dots, N$ ) discriminates the  $n$ -th class and the other  $N-1$  classes. From the unfolded drawings as shown in Fig. 2, it is clear that the traditional AdaBoost algorithms do not change the inner structures of the trained weak classifiers. They only determine their outer fusion weights.

### 2.1. AdaBoost algorithms

AdaBoost algorithms determine the fusion weights of the trained  $M$  weak classifiers by a set of weight training samples [31,32,37,39]. In the classification stage, take AdaBoost.M1 for example, each weak classifier outputs a hard decision for the test sample  $\mathbf{x}$ . The final output  $H(\mathbf{x})$  is related to weighted votes of all weak classifiers.

$$H(\mathbf{x}) = \arg \max_{y \in Y} \sum_{t=1}^M \alpha_t [h_t(\mathbf{x}) = y]; \quad Y = \{1, \dots, N\} \quad (1)$$

where  $\alpha_t$  is the weight of weak classifier  $h_t$ , and

$$[h_t(\mathbf{x}) = y] = \begin{cases} 1 & h_t(\mathbf{x}) = y \\ 0 & h_t(\mathbf{x}) \neq y \end{cases}; \quad y \in Y = \{1, \dots, N\} \quad (2)$$

where  $h_t(\mathbf{x})$  outputs a hard label

$$h_t(\mathbf{x}) = \arg \max_{y \in Y} \{h_t(\mathbf{x}, y)\}; \quad Y = \{1, \dots, N\} \quad (3)$$

where  $h_t(\mathbf{x}, y)$  is the response of weak classifier  $h_t$  to the label  $y$ .

Winner takes all approach is utilized in AdaBoost.M1. It makes a hard decision for the input sample. Generally, the hard decision is too strict to perform better. In order to make weak classifiers contributive to multi-class pattern classification, AdaBoost.M2 and AdaBoost.MT make full use of the responses of component classifiers of a weak classifier to make decision. Different from AdaBoost.M1, in AdaBoost.M2 and AdaBoost.MT each weak classifier outputs a response vector rather than a hard decision [31,37]. Each element of the response vector is in the range  $[0, 1]$ . For a given test sample  $\mathbf{x}$ , AdaBoost.M2 or AdaBoost.MT outputs the estimated label as follows

$$H(\mathbf{x}) = \arg \max_{y \in Y} \sum_{t=1}^T \alpha_t h_t(\mathbf{x}, y) \quad (4)$$

where  $\alpha_t h_t(\mathbf{x}, y)$  is the weighted response of weak classifier  $h_t$  to the label  $y$ .

### 2.2. ISABoost algorithms

Now we turn to introduce the proposed ISABoost algorithm in details. The unfolded drawings of the weak classifiers in ISABoost are shown in Fig. 2(b). In this paper, each weak classifier is adjusted by an inner weight adjusting vector  $\beta$  and a bias vector  $\eta$ . The dimensions of them are both  $1 \times N$  (note that  $N$  is the multi-class number). ISABoost fuses the adjusted weak classifiers which have been trained under either online or offline AdaBoost frameworks. The training samples reweighting and weak classifiers fusion weights determination of ISABoost algorithm are identical to the standard AdaBoost algorithms. Thus, we only illustrate the corresponding classification of ISABoost for a test sample, inner structure adjusting and adjusting validation, and optimal adjusting parameters determination.

#### 2.2.1. Classification of ISABoost for a test sample

Let  $\mathbf{R}_q = \langle R_q^1, \dots, R_q^N \rangle$  denote the response vector of the  $q$ -th weak classifier to the input vector  $\mathbf{x}$ , and  $R_q^k$  denote the corresponding response of the  $k$ -th ( $k = 1, \dots, N$ ) component classifier of the  $q$ -th ( $q = 1, \dots, M$ ) weak classifier. ISABoost adjusts the inner structure of a weak classifier by finding optimal biases vector  $\eta_q = \langle \eta_q^1, \dots, \eta_q^N \rangle$  and inner weight adjusting vector  $\beta_q = \langle \beta_q^1, \dots, \beta_q^N \rangle$ .

In ISABoost algorithm, the training samples are utilized to fulfill weak classifiers training, inner structure adjusting and adjusting validation, and outer weights determination as shown in Fig. 1(c). Assume that  $h'_t(\mathbf{x}, y)$  is the  $t$ -th ( $t = 1, \dots, M$ ) fused optimal weak classifier, and  $H(\mathbf{x})$  is the output of ISABoost for the test sample  $\mathbf{x}$  is as follows

$$\begin{aligned} H(\mathbf{x}) &= \arg \max_{y \in Y} \sum_{i=1}^M \alpha_i \times h'_i(\mathbf{x}, y) \\ &= \arg \max_{y \in Y} \sum_{i=1}^M \alpha_i \times [\beta_i^y (R_i^y(\mathbf{x}) - \eta_i^y)]; \quad Y = \{1, \dots, N\} \end{aligned} \quad (5)$$

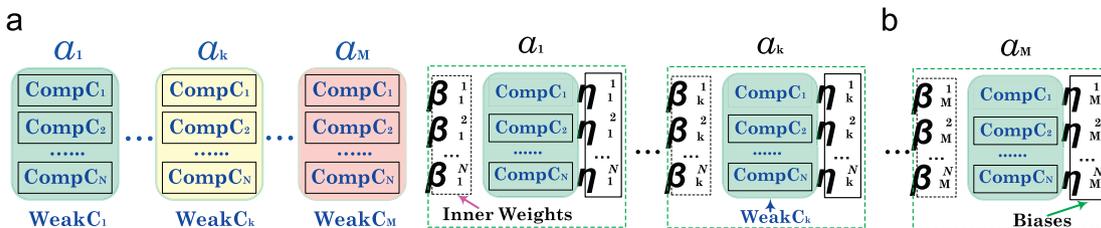


Fig. 2. Unfolded drawings of (a) AdaBoost and (b) ISABoost.

If we use the hard outer weights determination approach utilized in AdaBoost.M1, then the corresponding output  $H(\mathbf{x})$  of ISABOOST is as follows

$$H(\mathbf{x}) = \arg \max_{y \in Y} \sum_{t=1}^M \alpha_t [h'_t(\mathbf{x}, y)]; \quad Y = \{1, \dots, N\} \quad (6)$$

where  $[h'_t(\mathbf{x}, y)]$  is a hard decision function which is expressed as follows

$$[h'_t(\mathbf{x}, y)] = \begin{cases} 1 & \beta_t^y (R_t^y(\mathbf{x}) - \eta_t^y) = \max_{k \in Y} \{ \beta_t^k (R_t^k(\mathbf{x}) - \eta_t^k) \} \\ 0 & \beta_t^y (R_t^y(\mathbf{x}) - \eta_t^y) \neq \max_{k \in Y} \{ \beta_t^k (R_t^k(\mathbf{x}) - \eta_t^k) \} \end{cases}; Y = \{1, \dots, N\}; \quad t = 1, \dots, M \quad (7)$$

### 2.2.2. Inner structure adjusting and adjusting validation

In order to determine the parameters of ISABOOST, all the samples in CS (the sample number is denoted as  $S$ ) are utilized for inner structure adjusting.  $\nu$  samples  $\{(\mathbf{x}^1, y^1), \dots, (\mathbf{x}^\nu, y^\nu)\}$  with their labels  $y^i \in Y = \{1, \dots, N\}$  are utilized for adjusting validation. The objective is to find the optimal adjusting parameters  $\theta_q^* = \langle \boldsymbol{\eta}_q^*, \boldsymbol{\beta}_q^* \rangle$ ,  $q \in \{1, \dots, M\}$ , for each weak classifier. In this paper, ISABOOST iteratively selects an optimal inner structure adjusted weak classifier with respect to the error rates of the weighted training samples. The corresponding objective function in weak classifier inner structure adjusting is to find optimal inner structure adjusting parameters  $\theta_q^* = \langle \boldsymbol{\eta}_q^*, \boldsymbol{\beta}_q^* \rangle$  as follows

$$\theta_q^* = \langle \boldsymbol{\eta}_q^*, \boldsymbol{\beta}_q^* \rangle = \arg \max_{\theta_q} f(\theta_q); \quad \theta_q = \{ \theta_q^0, \theta_q^1, \dots, \theta_q^K \}, \quad q = 1, \dots, M \quad (8)$$

s.t. adjusting validation

where  $f(\theta_q)$  is the correct recognition rate of the weighted training samples. It is expressed as follows

$$f(\theta_q) = \frac{1}{S} \sum_{s=1}^S [h'_q(\mathbf{x}^s, y^s)]; \quad q = 1, \dots, M \quad (9)$$

where  $[h'_q(\mathbf{x}^s, y^s)]$  is a function of  $\beta_q$  and  $\eta_q$  as shown in Eq. (7). In Eq. (8),  $\theta_q^0$  (i.e., without inner structure adjusting  $\theta_q^0 = \langle \boldsymbol{\eta}_q^0, \boldsymbol{\beta}_q^0 \rangle$ ;  $\boldsymbol{\eta}_q^0 = \mathbf{0}$ ;  $\boldsymbol{\beta}_q^0 = \mathbf{1}$ ;  $q = 1, \dots, M$ ) and  $\theta_q^n$  ( $n = 1, \dots, K$ ) the initial parameters of the  $q$ -th weak classifier and the parameters of  $K$  adjusting weak classifiers for the  $q$ -th weak classifier. To make sure that the adjusted weak classifier  $q$  is with the best performances among all the adjusted weak classifiers, i.e.

$$f(\theta_q^*) = \max_{n=0, \dots, K} f(\theta_q^n); \quad q = 1, \dots, M \quad (10)$$

In order to make sure the adjusting is valid, the adjusted weak classifiers must pass through adjusting validation. The adjusting validation is evaluated by  $\nu$  samples as follows

$$\varphi(\theta_q^*) \geq \varphi(\theta_q^0); \quad q = 1, \dots, M \quad (11)$$

where  $\varphi(\theta_q)$  is performances of the adjusted classifier for the  $\nu$  validation samples, we have

$$\varphi(\theta_q) = \frac{1}{\nu} \sum_{i=1}^{\nu} [h'_q(\mathbf{x}^i, y^i)]; \quad q = 1, \dots, M \quad (12)$$

If  $\varphi(\theta_q^*) \geq \varphi(\theta_q^0)$  is not satisfied, then the original weak classifier is utilized in the final fusion stage. The importance of adjusting validation is discussed in Section 4.

### 2.2.3. Optimal inner structure adjusting parameters determination

To solve the optimal problem for inner structure adjusting as shown in Eq. (8). Gradient descent based approaches, and

genetic algorithms (GA) can be utilized [39]. GA is very flexible in designing objective functions. In this paper, we use GA to find the optimization solution. The inner weights are assigned in the range  $\beta_q^k \in [0.8, 1.2]$  ( $q = 1, \dots, M$ ;  $k = 1, \dots, N$ ). The biases are in the range  $\eta_q^k \in [-0.2, 0.2]$  ( $q = 1, \dots, M$ ;  $k = 1, \dots, N$ ). We set the parameters in limited ranges to save computational cost [39]. The flowchart of GA based parameter optimization is as follows.

- 1) Initialize parameters of GA: crossover probability  $P_c = 0.8$ , mutation probability  $P_m = 0.05$ , population size  $P_s = 400$ , maximum iteration times  $l_m = 50,000$ , minimum error variation  $Em = 10^{-6}$ , initial evolution step  $gen = 1$ .
- 2) Generate  $P_s$  individuals and encode them into chromosomes;
- 3) Calculate fitness values of each chromosome according to Eq. (9); The individuals with large fitness values correspond to the parameters with high correct recognition rates.
- 4) Update evaluation generations  $gen = gen + 1$ ; Select  $P_s$  chromosomes to the next generation according to the fitness. The selection probability of each chromosome is calculated as follows

$$P(k) = \text{fit}(k) / \sum_{k=1}^{P_s} \text{fit}(k) \quad (13)$$

where  $\text{fit}(k)$  and  $P(k)$  are the fitness value and selection probability of the  $k$ -th chromosome.

- 5) Generate  $P_s$  new individuals by genetic operations (crossover and mutation) according to the crossover probability  $P_c$  and mutation probability  $P_m$ ;
- 6) Repeat step (3) if the evaluation step  $gen$  less than  $l_m$  and the performance improvement of neighboring two generations is larger than  $Em$ ;
- 7) Select the chromosomes with highest fitness as the final output  $\theta_q^*$ .

For  $M$  weak classifiers, each classifier is adjusted separately. In order to make sure the final scene categorization can benefit from the adjusted weak classifiers, adjusting validation is needed. We evaluate the performances of the inner structure adjusted weak classifiers using the adjusting validation set. If  $\varphi(\theta_q^*) \geq \varphi(\theta_q^0)$  is not satisfied, then the original weak classifier is utilized in the final fusion stage.

### 2.3. Relationship of AdaBoost and ISABOOST

From the unfolded drawings of ISABOOST and AdaBoost algorithms, for any weak classifier  $q$ , if  $\beta_q = 1$  and  $\eta_q = 0$  (1 and 0 are vectors with dimensions  $1 \times N$ , each elements in them are with sample values 1 and 0.), then AdaBoost is identical to ISABOOST. Thus ISABOOST is a generalized case of AdaBoost.

## 3. ISABOOST based scene categorization

In order to show the effectiveness of the proposed ISABOOST approach, its applications are test on scene categorization. Before illustrating the proposed ISABOOST based scene categorization approach, we briefly review the related works on scene categorization.

### 3.1. Related work on scene categorization

Scene categorization is a specified application of multi-class pattern categorization. Recently BOW (Bag-of-Words) based approaches and salient feature based approaches are often utilized. The BOW based approaches are convenient to model scenes by geometric structures. BOW based approaches model objects in

a scene/image as geometric-free structures [1–7,11,16,24]. In [11,14], BOW based approaches represent objects with rigorous geometric structures by modeling the relationships of different parts. The co-occurrences, dependences and linkages of the salient parts of images are also modeled to improve scene categorization performances [4,5,7,12,13,17–22]. Modeling the spatial dependency between neighboring patches of a scene by machine learning models, such as hidden Markov model [8], Markov random fields [10], and conditional random field models [11] can improve scene categorization performances. These methods aim at training robust classifiers to fulfill scene categorization. Despite of assigning an image with a category, some research works aim at assigning more relevant labels for each image [48–51]. For example, Zha et al. fuse multi-label and multi-instance learning for robust image classification [48]. Semantic distance learning and graph based approach are adopted to determine more refined categories of the image [50,51].

Despite of constructing robust models [1–14], effective feature representation approaches are also important for scene categorization [15,23–27,40]. Salient shape [23], local scale and transform invariant feature [15], local binary pattern and its extensions in spatial pyramid spaces [40], visual appearance [24] and multi-resolution texture descriptors [25–27] have shown their effectiveness in scene categorization. Pyramid histogram of oriented gradients (PHOG) is good at representing the shapes and spatial layouts of scenes [23]. It has the advantages to represent image with certain global and local shape information. Lazebnik et al. represent the appearance feature of a scene using BOW histogram and carry out scene categorization by spatial pyramid matching (SPM) [24]. SPM is robust to the variations of rotation, resolution and illumination [24]. HWVP improves the discrimination power for images by utilizing sub-bands filtering in hierarchical wavelet packet domain [27]. GIST is robust to represent image texture information with oriented multiple-scale based filtering [25,26]. In this paper, we utilize the feature based approach to test the effectiveness of the proposed ISABOOST algorithm. Weak classifiers are trained independently using those features. ISABOOST and AdaBoost based scene categorization performances are compared in fusing the trained weak classifiers.

### 3.2. ISABOOST based scene categorization approach

The flowchart of training of ISABOOST based scene categorization is shown in Fig. 3(a) and the flowchart of ISABOOST based scene categorization for a test image is shown in Fig. 3(b). ISABOOST based scene categorization approach consists of the following 4 steps: (1) low-level features extraction; (2) weak

classifier training; (3) weak classifier inner structure adjusting and adjusting validation; (4) classifier selection and fusion weight learning. When the training process is completed, we get ISABOOST parameters. The flowchart of the ISABOOST based scene categorization consists of following 2 steps: (1) low-level feature extraction; (2) Scene categorization results for the input image with visual feature  $x$  by utilizing Eq. (5). The corresponding ISABOOST algorithm is illustrated in Section 2. Now we express the first two steps in ISABOOST based scene categorization.

ISABOOST and AdaBoost share the same basic weak classifiers. The weak classifiers are trained using the training set CS. In ISABOOST, the inner structure adjusting is carried out by using IS and outer weight of weak classifier is determined by WS. In AdaBoost the training samples IS and WS are both utilized for fusion weights determination.

#### 3.2.1. Low-level features extraction

Four features: SPM [24], PHOG [23], GIST [25,26] and HWVP [27] are served as the input features to train weak classifiers and to carry out scene categorization. **SPM** feature is a local scale invariant descriptor by using spatial pyramid transforms [24,40]. In the extraction of **SPM** [24], the 128 dimensional SIFT features [15] are converted into visual words. SPM is composed of visual words histograms of an image at various spatial pyramids. **PHOG** represents an image with histograms of orientation gradients over spatial pyramids. Each bin in PHOG represents the number of edges that have orientations within a certain angular range. **HWVP** represents texture information of an image using hierarchical wavelet packet transform [27]. **GIST** is an effective high dimensional texture descriptor. This feature takes advantages of Gabor transform [26].

In the extraction of SPM [24], the local appearance features are converted into D visual vocabularies. Then the visual vocabulary histogram of each image in spatial pyramid domain is constructed. In this paper, we set the spatial pyramid level to be two and the vocabulary size D to be 300, thus the dimension of SPM is 6300.

In the extraction of PHOG [23], the local shape is captured by the distribution over edge orientations within a region. In this paper we set the spatial level  $S=3$  and orientations  $J=10$ , then the dimension of PHOG is 850.

In this paper, we use the descriptor HWVP [27] under local partitioning pattern Local5 (the image is partitioned into  $2 \times 2$  grids and a centralized grid) to represent the hierarchical wavelet packet texture. We set the wavelet packet basis to be db2. The mean and standard deviation of each sub-band are utilized for

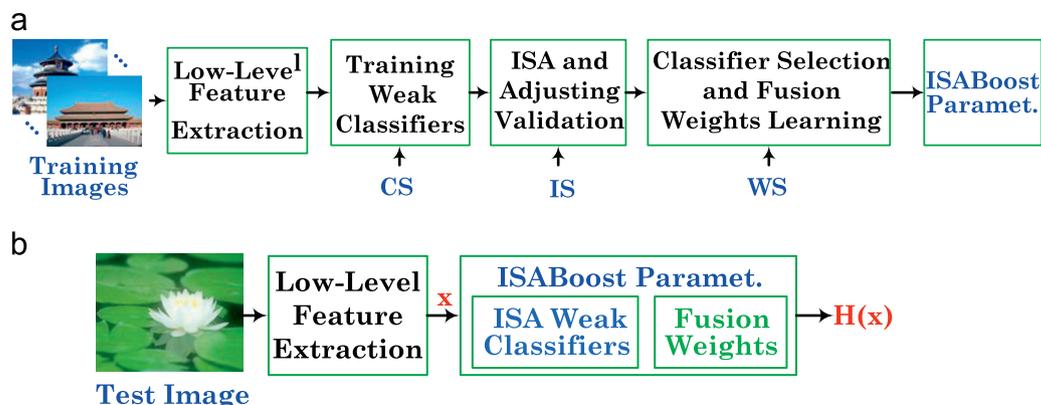


Fig. 3. Flowcharts of ISABOOST based Scene categorization. (a) Flowchart of the training of ISABOOST based scene categorization. (b) Flowchart of the ISABOOST based scene categorization for a test image.

texture descriptions [27]. In this paper we set  $L=3$ , thus the dimension of HWVP is 850.

In the extraction of GIST feature[26], firstly, each image is segmented into  $4 \times 4$  grids and each grid (with sizes  $32 \times 32$ ) is decomposed by a bank of multi-scale oriented filters (in this paper 8 orientations and 5 scales are utilized). Finally, the magnitude of each sub-band is utilized for feature representation. Thus the dimension of GIST of a gray-level image is  $5 \times 8 \times 16=640$ .

In following part of this paper, let  $\mathbf{x}$  denote the input feature of an image.  $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$  is a  $d$  dimensional vector. The dimension  $d$  is a feature related parameter. The dimensions of SPM, PHOG, HWVP, GIST are  $d=6300, 850, 850, 640$ , respectively [37,39].

### 3.2.2. Training weak classifiers

For the  $N$ -class scene categorization problem,  $N$  one-versus-all component classifiers are combined to determine the exact label [30]. During training,  $b$  samples per category are randomly selected and served as the weak classifiers' training set CS. For the  $M$ -class scene categorization problem, we combine the  $M$  one-against-all classifiers to determine the accurate scene label index.

Let complexes;  $= \{C^k\}_{k=1}^M$  denotes the parameters of a weak classifier. Let  $R_k$  ( $k = 1, \dots, M$ ) denotes the response of the  $k$ -th one-versus-all classifier with input feature  $X$ ,  $R_k \in [0,1]$ . We can estimate the label  $k_0$  of the input image with its feature  $X$  by a weak classifier as follow

$$k_0 = \arg \max_{k=1, \dots, M} \{R_k\} \quad (14)$$

$V$  weak classifiers for each of the four features are trained by running the training process  $V$  times. In this paper, we set  $V=20$ . In each training process,  $b$  samples per category are randomly selected from CS to train a weak classifier. For the training of the  $k$ -th CompC of a WeakC, the  $b$  training images from the  $k$ -th class are severed as positive samples and  $b \times (N - 1)$  images from the other  $N - 1$  classes are severed as negative samples. Some of the samples may be utilized more than once and some of them may not be selected in each process.

## 4. Experiments and discussion

Five experiments are conducted to evaluate the performance of boosted scene categorization approach by adjusting the inner structures and determining outer weights of weak classifiers. The first three experiments are on OT [25], Scene-13 [3] and Sport Event [34] datasets. Comparisons of AdaBoost and ISABOOST based scene categorization approaches are made. Back-propagation networks are served as weak classifiers. ISABOOST under various outer weight determination approaches are also compared. ISABOOST.M1, ISABOOST.M2 and ISABOOST.MT are the ISABOOST algorithms with the weak classifiers' fusion weights determined by the approaches AdaBoost.M1, AdaBoost.M2 and AdaBoost.MT. When the total training samples is fixed, the impact of number assignment for weak classifier training, inner structure adjusting and adjusting validation, and fusion weights determination to ISABOOST is discussed. In order to show the impacts of the type of weak classifier to ISABOOST, the fourth experiment is conducted by utilizing SVM as basic weak classifier. And the last experiment is to show the importance of adjusting validation. Accurate recognition rate (AR) is utilized to evaluate scene categorization performance which is expressed as follows

$$AR = \frac{NC}{NC + NM} \times 100\% \quad (15)$$

where  $NC$  and  $NM$  are the correct and missing detections.

**Table 1**

ISABOOST based scene categorization performances for OT dataset under various testing conditions. Totally,  $T$  training samples per category are utilized for weak classifiers training, inner structure adjusting and outer weights learning. The samples numbers of weak classifiers training, inner structure adjusting validation, and weight learning are  $b$ ,  $a$  and  $g$ , respectively.  $g=0$  corresponds to the AdaBoost algorithms and  $g \neq 0$  denotes the proposed inner structure adjusting and outer weights leaning based approach.

$T$	$b$	$a$	$G$	ISABOOST.M1	ISABOOST.M2	ISABOOST.MT	
100	50	50	0	75.44	82.16	82.60	
		30	20	76.67 ± 0.43	82.34 ± 0.17	83.69 ± 0.51	
		20	30	76.58 ± 0.39	82.46 ± 0.29	82.99 ± 0.14	
	80	50	25	25	76.01 ± 0.18	83.20 ± 0.22	84.14 ± 0.62
			30	0	76.01	83.20	83.39
			20	10	78.07 ± 0.56	83.72 ± 0.13	84.55 ± 0.32
50	40	15	15	77.55 ± 0.34	83.81 ± 0.23	84.56 ± 0.29	
		10	20	77.39 ± 0.36	83.56 ± 0.19	83.78 ± 0.11	
		10	0	75.29	80.94	79.98	
		6	4	77.57 ± 0.53	81.29 ± 0.16	80.39 ± 0.13	
		5	5	78.14 ± 0.86	81.08 ± 0.14	80.56 ± 0.18	
		6	6	78.06 ± 0.55	81.21 ± 0.14	80.12 ± 0.27	
	30	25	25	0	69.60	76.50	77.28
			15	10	73.12 ± 0.52	76.83 ± 0.09	77.75 ± 0.13
			10	15	72.36 ± 0.48	76.96 ± 0.12	77.62 ± 0.11
			10	0	65.72	73.99	74.50
			8	2	70.50 ± 0.63	74.45 ± 0.21	74.81 ± 0.19
			7	3	70.27 ± 0.51	74.46 ± 0.16	74.77 ± 0.15
25	20	6	4	70.14 ± 0.45	74.62 ± 0.14	75.06 ± 0.36	
		5	5	69.79 ± 0.97	74.56 ± 0.15	74.68 ± 0.16	
		4	6	68.53 ± 0.58	74.49 ± 0.18	74.75 ± 0.19	
		3	7	68.24 ± 0.75	74.55 ± 0.14	74.67 ± 0.13	
		2	8	69.36 ± 0.63	74.67 ± 0.16	74.73 ± 0.20	
		5	0	73.75	77.46	77.02	
	1	4	4	1	75.74 ± 0.43	77.86 ± 0.23	77.22 ± 0.28
			3	2	75.85 ± 0.38	77.71 ± 0.13	77.18 ± 0.12
			2	3	75.26 ± 0.46	77.65 ± 0.19	77.12 ± 0.19
			1	4	75.35 ± 0.52	78.09 ± 0.36	77.05 ± 0.16

**Table 2**

ISABOOST based scene categorization performances for Sport Event dataset under various testing conditions. Totally,  $T$  training samples per category are utilized for weak classifiers training, inner structure adjusting and outer weights learning. The samples numbers of weak classifiers training, inner structure adjusting validation, and weight learning are  $b$ ,  $a$  and  $g$ , respectively.  $g=0$  corresponds to the AdaBoost algorithms and  $g \neq 0$  denotes the proposed inner structure adjusting and outer weights leaning based approach.

$T$	$b$	$a$	$g$	ISABOOST.M1	ISABOOST.M2	ISABOOST.MT
20	15	5	0	50.90	64.29	64.69
		3	2	51.47 ± 0.16	64.48 ± 0.12	66.36 ± 0.38
		2	3	51.10 ± 0.13	64.84 ± 0.21	66.16 ± 0.35
50	40	10	0	57.76	77.22	74.55
		6	4	60.89 ± 0.72	77.65 ± 0.16	75.47 ± 0.28
		5	5	60.02 ± 0.53	77.46 ± 0.18	75.34 ± 0.13
		4	6	60.65 ± 0.66	77.99 ± 0.22	75.74 ± 0.29
		4	0	57.54	70.78	71.33
70	30	30	10	58.20 ± 0.31	71.28 ± 0.21	71.80 ± 0.12
		25	15	58.38 ± 0.26	71.11 ± 0.16	71.95 ± 0.11
		20	20	58.93 ± 0.44	71.53 ± 0.32	71.57 ± 0.08
		15	25	57.72 ± 0.12	71.06 ± 0.11	72.48 ± 0.12
		10	30	58.35 ± 0.27	71.32 ± 0.25	71.83 ± 0.27
		20	0	77.78	97.34	95.75
70	50	15	5	81.84 ± 0.99	97.57 ± 0.15	96.08 ± 0.09
		10	10	82.14 ± 1.26	97.54 ± 0.17	95.95 ± 0.12
		5	15	81.52 ± 0.83	97.59 ± 0.19	95.89 ± 0.14

In Tables 1–3, scene categorization performances by utilizing totally  $T$  training samples per category are shown. The numbers of samples for weak classifier training and inner structure adjusting, adjusting validation, and outer weight determination are  $b$ ,  $g$ , and  $a$ , respectively. In Tables 1–3, the average accurate recognition rates and their standard deviations of 10 times run of the

**Table 3**

ISABoost based scene categorization performances for Scene-13 dataset under various testing conditions. Totally,  $T$  training samples per category are utilized for weak classifiers training, inner structure adjusting and outer weights learning. The samples numbers of weak classifiers training, inner structure adjusting validation, and weight learning are  $b$ ,  $a$  and  $g$ , respectively.  $g=0$  corresponds to the AdaBoost algorithms and  $g \neq 0$  denotes the proposed inner structure adjusting and outer weights leaning based approach.

$T$	$b$	$A$	$g$	ISABoost.M1	ISABoost.M2	ISABoost.MT
50	40	10	0	68.98	79.73	79.34
		6	4	$71.91 \pm 0.81$	$79.76 \pm 0.06$	$79.42 \pm 0.07$
		5	5	$70.79 \pm 0.79$	$79.73 \pm 0.08$	$79.64 \pm 0.18$
		4	6	$74.96 \pm 0.97$	$79.78 \pm 0.08$	$80.53 \pm 0.26$

proposed ISABoost algorithms are shown.  $g=0$  corresponds to the AdaBoost and  $g \neq 0$  corresponds to ISABoost.

#### 4.1. Experiments on OT dataset

The OT dataset has 2688 images with eight categories [33,25]: 360 **coast**, 328 **forest**, 374 **mountain**, 260 **highway**, 308 **insidicity**, 410 **open country**, 292 **street**, and 356 **tallbuilding**. Each image in this dataset is with the same sizes  $256 \times 256$ .

Table 1 shows ISABoost based scene categorization performances (average recognition rate and its standard deviation) under  $T=100$ , 80, 50 and 30. In the circumstance that  $T=100$ ,  $b=50$  and  $a+g=50$ , the performances of ISABoost under several combinations of  $a$  and  $g$  are shown in Table 1. The average recognition rates of AdaBoost.M1, AdaBoost.M2 and AdaBoost.MT are 75.44%, 82.16% and 82.60%, respectively. The average scene categorization performances of ISABoost algorithms outperform those of the AdaBoost algorithms by about 0.98%, 0.47% and 1.10%, respectively.

Under  $T=80$ ,  $b=50$  and  $a+g=30$ , the ISABoost algorithms outperform the corresponding AdaBoost algorithms by about 1.66%, 0.50% and 0.91% in average, respectively. Under  $T=50$ ,  $b=40$  and  $a+g=10$ , the average performances of ISABoost.M1, ISABoost.M2 and ISABoost.MT are improved by about 2.63%, 0.53% and 0.37%, respectively over the corresponding AdaBoost algorithms. While under  $T=50$ ,  $b=25$  and  $a+g=25$ , the improvements of ISABoost algorithms over the corresponding AdaBoost algorithms are 3.14%, 0.40% and 0.44%, respectively.

The performances of ISABoost under the case that  $(T,b)=(30,25)$ , outperform those of the case  $(T,b)=(30,20)$  by about 8.03%, 3.47%, 2.52%, respectively for ISABoost.M1, ISABoost.M2 and ISABoost.MT. The average recognition rates for the proposed approach are 69.55%, 74.54% and 74.78%, respectively for the ISABoost.M1, ISABoost.M2 and ISABoost.MT algorithms under  $(b,a+g)=(20,10)$ . While under  $(b,a+g)=(25,5)$ , the average recognition rates are 75.55%, 77.83% and 77.14%, respectively for the original ISABoost.M1, ISABoost.M2 and ISABoost.MT algorithms.

By comparing the performances of ISABoost algorithms under the following 4 cases: (c1)  $T=50$ ,  $b=40$ ,  $a+g=10$ ; (c2)  $T=50$ ,  $b=25$ ,  $a+g=25$ ; (c3)  $T=30$ ,  $b=20$ ,  $a+g=10$ ; (c4)  $T=30$ ,  $b=25$ ,  $a+g=5$ , we find that ISABoost performances is influenced by the learning samples significantly. For (c1) and (c2), performances of ISABoost algorithms under (c1) outperform these of (c2) by about 5.2%, 4.3% and 2.7%, respectively for ISABoost.M1, ISABoost.M2 and ISABoost.MT. For (c3) and (c4), performances of ISABoost algorithms under (c4) outperform these of (c3) by about 6.00%, 3.29% and 2.36%, respectively for ISABoost.M1, ISABoost.M2 and ISABoost.MT. From above comparisons, it is clear that, if and only if the weak classifiers are well trained (i.e., more samples are utilized to train weak classifiers), AdaBoost and ISABoost algorithms have better performances.

#### 4.2. Experiments on sport event dataset

The Sport Event dataset contains 1579 images of eight sport event classes [34]: 200 **badminton**, 137 **bocce**, 236 **croquet**, 182 **polo**, 194 **rock climbing**, 250 **rowing**, 190 **sailing**, and 190 **snowboarding**.

Table 2 shows ISABoost based scene categorization performances under  $T=20$ , 50 and 70. When  $T=20$ ,  $b=15$ ,  $a=5$  and  $g=0$ , the corresponding categorization performance of AdaBoost.M1, AdaBoost.M2 and AdaBoost.MT are 50.9%, 64.29% and 64.69%, respectively. ISABoost algorithms under  $T=20$ ,  $b=15$ ,  $a+g=5$ , get performances improvement by 0.39%, 0.37% and 1.57%, respectively over the corresponding AdaBoost algorithms.

For the cases  $T=50$ ,  $b=40$  and  $a+g=10$ , ISABoost algorithms get improvements by about 2.76%, 0.48% and 0.97% in average over AdaBoost.M1, AdaBoost.M2 and AdaBoost.MT. For the cases  $T=70$   $b=30$  and  $a+g=40$ , the average improvements are about 0.78%, 0.48% and 0.53%, respectively.

From the comparative analysis, ISABoost based scene categorization approaches get better performances than the corresponding AdaBoost algorithms. It is clear that the weak classifiers' training is very important for ISABoost. When the total numbers of the training samples are equal, better performances are achieved for the cases that the training sample number for weak classifiers are sufficient.

#### 4.3. Experiments on scene-13 dataset

The Scene-13 dataset consists of the 2688 images of the eight categories of the OT dataset and another five categories with 1071 images [3]: 241 **suburb**, 174 **bedroom**, 151 **kitchen**, 289 **living room**, and 216 **office**. Totally there are 3759 images in this dataset.

For Scene13 dataset, scene categorization performances under  $T=50$ ,  $b=40$ ,  $a+g=10$  are shown in Table 3. The average recognition rates of AdaBoost.M1, AdaBoost.M2 and AdaBoost.MT are 68.98%, 79.73%, and 79.34%, respectively. The ISABoost algorithms improve scene categorization performances by about 3.57%, 0.04% and 0.52% in average over AdaBoost algorithms. In this case, the average recognition rates of ISABoost.M1, ISABoost.M2 and ISABoost.MT are 72.55%, 79.77% and 79.86%, respectively. This also shows that ISABoost algorithms comparatively have high performances than their corresponding AdaBoost algorithms.

#### 4.4. Impacts of weak classifier type on ISABoost

The above three experiments are carried out under the conditions that the basic weak classifier is BP networks. In this section, the impacts of the type of weak classifier to ISABoost performances are discussed. Experiments for the ISABoost algorithms using SVM and BPN as weak classifiers are carried out on OT, and Sport Event datasets under  $T=50$ ,  $b=40$ ,  $a+g=10$ . The performances of ISABoost with weak classifiers BPN (denoted ISA@BPN) and SVM [28,38] (denoted ISA@SVM) are shown in Fig. 4, respectively. For OT dataset, performances of ISABoost algorithms outperform those of AdaBoost (denoted BSL@BPN and BSL@SVM) by 2.56%, 0.55% and 0.72%, respectively in average. For Sport Event dataset, scene categorization performances improved by 2.85%, 0.57% and 0.44%, respectively. This shows that the ISABoost algorithms under different type of weak classifier are all better than the corresponding AdaBoost algorithms.

#### 4.5. Impacts of adjusting validation

The above experiments, ISABoost based scene categorization is carried out under adjusting validation. If the constraints are not satisfied, then the original weak classifier is utilized in the fusion stage. In order to make sure the inner structure adjusting is valid, we use the  $a+g$  samples (randomly selected from the  $T$  training samples) per category in our experiments. Now we turn to discuss scene

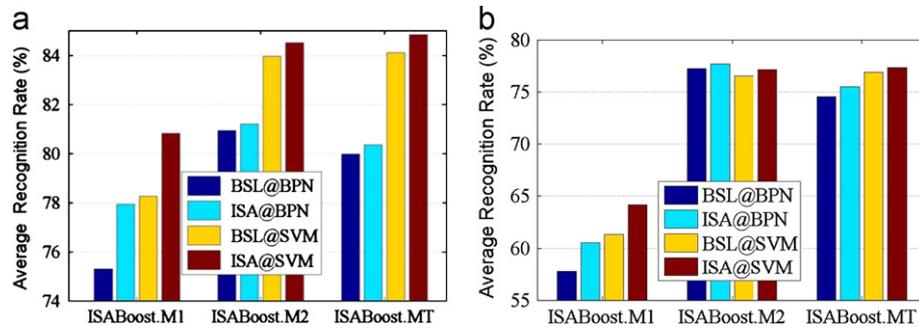


Fig. 4. The impacts of type of weak classifier to ISABoost performances on (a) OT dataset, (b) Sport Event dataset under  $T=50$ ,  $b=40$  and  $a+g=10$ . BSL@BPN and BSL@SVM are the AdaBoost algorithms with the type of weak classifier BPN and SVM. ISA@BPN and ISA@SVM are the ISABoost algorithms with the type of weak classifier BPN and SVM.

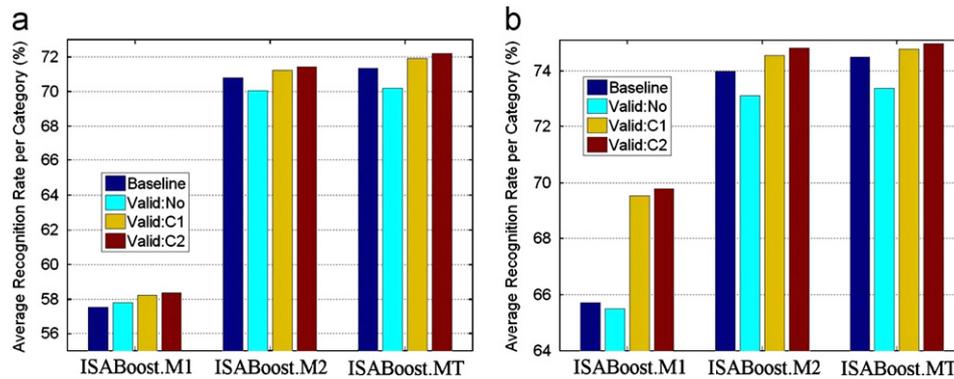


Fig. 5. ISABoost based scene categorization performances versus validation for (a) Sport Event dataset under  $T=70$ ,  $b=30$ ,  $a+g=40$ ; (b) OT under  $T=30$ ,  $b=20$ ,  $a+g=10$ .

categorization performance versus different validation approaches. The first approach is that no validation is adopted (denoted Valid:No). The second approach is that the  $a+g$  samples (the training samples except the samples for weak classifier training) per category are utilized (denoted Valid:C1) for validation. The experimental results shown in Tables 1–3, Fig. 4(a) and (b) are obtained under Valid:C1. The third approach is that  $a+g+b$  samples per category are all utilized for adjusting validation (denoted Valid:C2). In this circumstance, the weak classifiers training samples are also utilized in validation. Correspondingly, scene categorization performances of above 3 approaches on Sport Event dataset under  $T=70$ ,  $b=30$ ,  $a+g=40$  and OT dataset under  $T=30$ ,  $b=20$ ,  $a+g=10$  are shown in Fig. 5(a) and (b), respectively. ISABoost based scene categorization performances under Valid:No are inferior to those of AdaBoost algorithms (denoted Baseline).

The performances of ISABoost.M1 under Baseline and Valid:No are almost the same. This is the fact that, in ISABoost.M1, many weak classifiers are viewed as useless. After inner structure adjusting, some weak classifiers with their performances inferior to  $1/2$  are adjusted to be better (with their performances higher than  $1/2$ ), while some of better classifiers are adjusted to be poor. This makes the valid weak classifier number of Valid:No and Baseline are almost the same. Thus performances of ISABoost.M1 under Validation:No and Baseline are very close.

When no inner structure adjusting validation is adopted, performances of ISABoost.M2 and ISABoost.MT are decreased by about 0.8%. This is caused by the fact that some of the best classifiers are degraded after inner structures adjusting if no adjusting validation is adopted. The validation is important to make sure the performances of the selected **optimal** classifiers are not poor than their originals. From the classifier fusion point of

view, the more samples are utilized in adjusting validation, the less likely the inferior classifiers are selected during fusion. This is revealed by the performances of Valid:C1 and Valid:C2. When the training samples of weak classifiers are also utilized in adjusting validation, further improvements are achieved. Actually it is not strange that ISABoost under Valid:C2 outperforms ISABoost under Valid:C1. If all the training samples are correctly classified by the trained weak classifier, then they are useless in adjusting validation. However, some hard samples which are incorrectly classified by the weak classifiers have some positive contribution for adjusting validation.

When extra samples are utilized for adjusting validation, their performances outperform that of the Baseline. When the weak classifiers' training samples are also utilized for validation (Valid:C2), performances of ISABoost can further be improved by about 0.3% in average over Valid:C1.

## 5. Conclusion

In this paper, an enhanced AdaBoost algorithm: ISABoost is proposed. It is a generalized AdaBoost algorithm. ISABoost adjusts inner structures of the trained weak classifiers. It selects a best classifier from a set of adjusted weak classifiers. Adjusting validation makes sure the ISABoost better than AdaBoost algorithm. Properly assigning the number of weak classifier training, inner structure adjusting and fusion weight determination is important in ISABoost. The advantages of ISABoost over AdaBoost do not influenced by the type of weak classifiers. Under the same testing conditions, ISABoost is better than AdaBoost. This can be revealed by the performances of ISABoost and AdaBoost based scene categorization. However, how to make the proposed ISABoost algorithms to large scale dataset with

is also a very challenge problem, because the parameters needed to be adjusted increase linearly and the computational costs increase exponentially.

## Acknowledgments

This work is supported by the Natural Science Foundation of China No.60903121, No.61173109 and Microsoft Research Asia.

## Appendix A. Supplementary information

Supplementary data associated with this article can be found in the online version at <http://dx.doi.org/10.1016/j.neucom.2012.09.011>.

## References

- [1] F. Monay, D. Gatica-Perez, PLSA-based image auto-annotation: constraining the latent space, in Proc. ACM Multimedia, 2004.
- [2] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky, Describing Visual Scenes Using Transformed Dirichlet Processes, in NIPS, 2005.
- [3] F. Li, and P. Perona, A Bayesian hierarchy model for learning natural scene categories, in Proc. CVPR, 2005.
- [4] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman, Discovering object categories in image collections, in Proc. ICCV, 2005.
- [5] Y. Zheng, M. Zhao, S. Neo, T. Chua, and Q. Tian, Visual synset: towards a higher-level visual representation, in Proc. CVPR, 2008.
- [6] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, Local features and kernels for classification of texture and object categories: a comprehensive study, *Int. J. Comput. Vision*, 2007.
- [7] A. Bosch, A. Zisserman, X. Munoz, Scene classification using a hybrid generative/discriminative approach, *IEEE Trans. Pattern Anal. Mach. Intell.* 30 (4) (2008) 712–727.
- [8] J. Li, J. Wang, Automatic linguistic indexing of pictures by a statistical modeling approach, *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (9) (2003) 1075–1088.
- [9] J. Bi, Y. Chen, and J. Wang, A sparse support vector machine approach to region-based image categorization, in Proc. CVPR, 2005.
- [10] D. Larlus, and F. Jurie, Combining appearance models and markov random fields for category level object segmentation, in Proc. CVPR, 2008.
- [11] A. Quattoni, M. Collins, and T. Darrell, Conditional random fields for object recognition, in NIPS, 2004.
- [12] C. Galleguillos, A. Rabinovich, and S. Belongie, Object categorization using co-occurrence, location and appearance, in Proc. CVPR, 2008.
- [13] L. Cao, and F. Li, Spatially coherent latent topic model for concurrent object segmentation and classification, in Proc. ICCV, 2007.
- [14] A. Holub and P. Perona, A discriminative framework for modeling object classes, in Proc. ICCV, 2005.
- [15] D. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vision* 60 (2) (2004) 91–110.
- [16] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, Visual categorization with bags of keypoints, in Proc. ECCV, 2004.
- [17] D. Crandall, P. Felzenszwalb, and D. Huttenlocher, Spatial priors for part-based recognition using statistical models, in Proc. CVPR, 2005.
- [18] G. Wang, Y. Zhang, and F. Li, Using dependent regions for object categorization in a generative framework, in Proc. CVPR 2006.
- [19] S. Savarese, J. Winn, and A. Criminisi, Discriminative object class models of appearance and shape by correlations, in Proc. CVPR, 2006, pp. 2033–2040.
- [20] Y. Teh, M. Jordan, M. Beal, D. Blei, Hierarchical dirichlet processes, *J. Am. Statist. Assoc.* (2006).
- [21] P. Gosselin, M. Cord, S. Philipp-Foliguet, Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval, *Comput. Vision Image Understanding* 110 (2008) 403–417.
- [22] L. Wu, Y. Hu, M. Li, N. Yu, X. Hua, Scale-invariant visual language modeling for object categorization, *IEEE Trans. Multimedia* 11 (2) (2009) 286–294.
- [23] A. Bosch, A. Zisserman, and X. Munoz, Representing shape with a spatial pyramid kernel, in Proc. CIVR, 2007.
- [24] S. Lazebnik, C. Schmid, and J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in Proc. CVPR, 2006.
- [25] A. Torralba, K. William, T. Freeman, and M. Rubin, Context-based vision system for place and object recognition, in Proc. ICCV 2003.
- [26] A. Oliva, A. Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope, *IJCV* 42 (3) (2001) 145–175.
- [27] X. Qian, G. Liu, D. Guo, Z. Li, Z. Wang, and H. Wang, Object categorization using hierarchical wavelet packet texture descriptors, in Proc. ISM 2009, pp.44–51.
- [28] H. Zhang, A. Berg, M. Maire, and J. Malik, Svm-knn: discriminative nearest neighbor classification for visual category recognition, in Proc. CVPR, 2006.
- [29] G. Ou, Y. Murphey, Multi-class pattern classification using neural networks, *Pattern Recognit.* 40 (2007) 4–18.
- [30] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, 1995.
- [31] Y. Freund, R. Schapire, a decision-theoretic generalization of online learning and an application to boosting, *J. Comput. Syst. Sci.* 55 (1997) 119–139.
- [32] Y. Freund, and R. Schapire, Experiments with a new boosting algorithms. *Machine learning: Proceedings of the 13th International Conference*, 1996.
- [33] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, Object in context, in Proc. ICCV, 2007.
- [34] L. Li, and F. Li, What, where and who? Classifying events by scene and object recognition, in Proc. ICCV, 2007.
- [35] Paul A Viola, Michael J Jones, Robust real-time face detection, *Int. J. Comput. Vision* 57 (2) (2004) 137–154.
- [36] Paul A. Viola, Michael J. Jones, Rapid object detection using a boosted cascade of simple features, *CVPR* 1 (2001) 511–518.
- [37] X. Qian, Z. Yan, K. Hang, G. Liu, H. Wang, Z. Wang, Z. Li, Scene categorization using boosted back-propagation neural networks, in Proc. PCM 2010.
- [38] C. Chang, C. Lin, LIBSVM: a library for support vector machines, DOI=<http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [39] X. Qian, Z. Yan, K. Hang, Boosted scene categorization approach by adjusting inner structures and outer weights of weak classifiers, in Proc. MMM, 2011, pp.413–423.
- [40] X. Qian, X.S. Hua, P. Chen, L. Ke, PLBP: an effective local binary patterns texture descriptor with pyramid representation, *Pattern Recognit.* 44 (2011) 2502–2515.
- [41] Stan Li, Z. Zhang, FloatBoost learning and statistical face detection, *IEEE Trans. PAMI* 26 (9) (2004) 1112–1123.
- [42] K. Tieu and P. Viola, Boosting image retrieval. In Proc. CVPR, pp. 228–235, 2000.
- [43] Y. Lu; T. Zhang; Q. Tian, i.Boosting for image classification, in Proc. ICME, 2007, pp.1711–1714.
- [44] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. Bowyer. SMOTEBoost: Improving prediction of the minority class in boosting, in Proc. Principles of Knowledge Discovery in Databases, 2003.
- [45] C. Seiffert, T. Khoshgoftaar, J. Hulse, A. Napolitano, Resampling or reweighting: a comparison of boosting implementations, in Proc. ICTAI, 2008, pp.445–451.
- [46] N. Oza and S. Russell. Experimental comparisons of online and batch versions of bagging and boosting. In Proc. ACM SIGKDD, 2001.
- [47] H. Grabner and H. Bischof, On-line boosting and vision, in Proc. CVPR, 2006.
- [48] E. Moxley, T. Mei, B. Manjunath, Video annotation through search and graph reinforcement mining, *IEEE Trans. Multimedia* 12 (3) (2010) 184–193.
- [49] X. Qian, X. Hua, and X. Hou, Tag filtering based on similar compatible principle, In Proc. ICIP, 2012.
- [50] T. Mei, Y. Wang, X. Hua, S. Gong, and S. Li, Coherent image annotation by learning semantic distance, in Proc. CVPR, 2008.
- [51] Z. Zha, X. Hua, T. Mei, J. Wang, G. Qi, and Z. Wang, Joint multi-label multi-instance learning for image classification, in Proc. CVPR, 2008.



**Xueming Qian** (M'10) received the B.S. and M.S. degrees in Xi' an University of Technology, Xi' an, China, in 1999 and 2004, respectively, and the Ph.D. degree in the School of Electronics and Information Engineering, Xi' an Jiaotong University, Xi' an, China, in 2008. He was awarded Microsoft fellowship in 2006. From 1999 to 2001, he was at Shaanxi Daily. From 2008 till now, he is an associate professor of the School of Electronics and Information Engineering, Xi' an Jiaotong University. He was a visit scholar at Microsoft research Asia from Aug. 2010 to March 2011. His research interests include video/image analysis and retrieval, social and mobile multimedia search and mining. He publishes more than 40 papers in international conferences and journals. He is the reviewers of IEEE TCSVT, IEEE TMM, Signal Processing: Image Comm., Pattern Recognition Letters, IET Image Processing, Neurocomputing, Journal of Visual Communication and Image Representation, Journal of Computer Science and Technology and many international conferences. He is a session chair of VIE 2008. He holds 15 Chinese patents or pending applications. His researches are founded by NSFC, National "863" Project, and many companies, including Huawei, Microsoft Research Asia.



**Yuan Yan Tang** (F'04) received the B.S. degree in electrical and computer engineering from Chongqing University, Chongqing, China, the M.S. degree in electrical engineering from the Beijing University of Post and Telecommunications, Beijing, China, and the Ph.D. degree in computer science from Concordia University, Montreal, QC, Canada. He is currently a Professor with the Department of Computer Science, Chongqing University, a Chair Professor with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, and an Adjunct Professor of computer science with Concordia University. He is an Honorary Lecturer with the University of Hong Kong and an Advisory

Professor with many institutes in China. He is the Founder and Editor-in-Chief of the *International Journal on Wavelets, Multiresolution, and Information Processing* and the Associate Editor for several international journals on pattern recognition and artificial intelligence. He has published more than 250 technical papers and is the author or a coauthor of 21 books and book chapters on several subjects, e.g., electrical engineering and computer science. His research interests include wavelet theory and applications, pattern recognition, image processing, document processing, artificial intelligence, parallel processing, Chinese computing and VLSI architecture. Prof. Tang is a Fellow of the International Association for Pattern Recognition. He has been the General Chair, the Program Chair, and a Committee Member for many international conferences. He was the General Chair of the 19th International Conference on Pattern Recognition (ICPR 2006).



**Kaiyu Hang**, received the B.S. degree in Electronic and Information Engineering on 2010, Xi'an Jiaotong University, China. He is now pursuing master's degree in communication systems program in ICT (school of information and communication technology) at KTH, Sweden. He served as researcher at Ericsson Stockholm AB, Sweden from No.2011 to Apr. 2012.



**Zhe Yan**, received the B.S. degree in Electronic and Information Engineering on 2010, Xi'an Jiaotong University, China. He is engineer at Hand Corporation.