Cooperative Sparse Representation in Two Opposite Directions for Semi-supervised Image Annotation

Zhong-Qiu Zhao, Herve Glotin, Zhao Xie, Jun Gao, and Xindong Wu, *Fellow, IEEE*

Abstract

Recent studies have shown that sparse representation (SR) can deal well with many computer vision problems, and its kernel version owns more powerful classification capability. In this paper, we address the application of a cooperative sparse representation (Co-SR) in semi-supervised image annotation which can grow the amount of labeled images for further use in training image classifiers. Provided a set of labeled (training) images and a set of unlabeled (test) images, the usual SR method, which we call forward SR, is to represent each unlabeled image with several labeled ones, and then to annotate the unlabeled image according to the annotations of these labeled ones. However, to the best of our knowledge, the SR method in an opposite direction, which we call backward SR, to represent each labeled image with several unlabeled images, and then to annotate any unlabeled image according to the annotations of the labeled images which the unlabeled image is selected by the backward SR to represent, has not been addressed by researchers. In this paper, we explore how much the backward SR can contribute to image annotation, and be complementary to the forward SR. The co-training, which has been proved to be a semi-supervised method improving each other only if two classifiers are relatively independent, is then adopted to testify this complementary nature between two SRs in opposite directions. Finally, the co-training of two SRs in kernel space builds a cooperative kernel sparse representation (Co-KSR) method for image annotation.

Zhong-Qiu Zhao, Zhao Xie and Jun Gao are with the College of Computer Science and Information Engineering, Hefei University of Technology, China. E-mail: z.zhao@hfut.edu.cn

Herve Glotin is with the LSIS CNRS Lab, University of Sud-Toulon Var, and Institut Universitaire de France. E-mail: glotin@univ-tln.fr

Xindong Wu, corresponding author, is with the College of Computer Science and Information Engineering, Hefei University of Technology, China, and the Department of Computer Science, University of Vermont, USA. E-mail: xwu@cs.uvm.edu

Experimental results and analyses show that two KSRs in opposite directions are complementary, and Co-KSR improves much over either single one of them with an image annotation performance better than other state-of-the-art semi-supervised classifiers such as TSVM (Transductive Support Vector Machine), LGC (Local and Global Consistency) and GFHF (Gaussian Fields and Harmonic Functions). Comparative experiments with a non-sparse solution are also performed to show that the sparsity plays an important role in the cooperation of image representations in two opposite directions. Our work will extend the application of sparse representation in image annotation and retrieval.

Index Terms

image annotation, sparse representation, co-training, semi-supervised learning, image retrieval

I. Introduction

In the past decade, the number of images available online and offline has dramatically increased. TBIR (Text-Based Image Retrieval) search engines, which are popular for image retrieval, retrieve relevant images without using any content information. Recently, many machine learning methods have been developed to annotate new images automatically by making use of available training images with annotations [6][25][26][37][47][44]. Thus, assigning images with correct class labels or relevant keywords by visual contents in the images is very significant for further learning of image classification and annotation models. However, manual annotation is becoming more and more time-consuming and difficult due to the increasing size of image databases, and the lack of annotated images hinders the efficient and reliable construction of various classifiers. Therefore, semi-supervised techniques, which propagate labels from a limited number of labeled images to unlabeled ones, has recently attracted lots of research attention [18][45][46][48][19][49][39][38][10].

In essence, the task of machine learning is to associate keywords with visual contents in images, and establish a mapping from low-level visual features to high-level semantic concepts. Given some labeled images with annotations, this mapping for an unlabeled image can be implemented by approximately representing the unlabeled image with several labeled images according to its visual content similarity to the labeled images, and the annotations of the unlabeled image are then indicated by those of the labeled images (Figure 1(a)). However, in a dense solution, the number of labeled images which are selected to approximately represent the unlabeled image is large, and these images may be from many classes. So the dense solution is not especially informative for classifying the unlabeled image [40]. Sparse representation (SR) [40] can automatically select a small number of labeled images and approximate

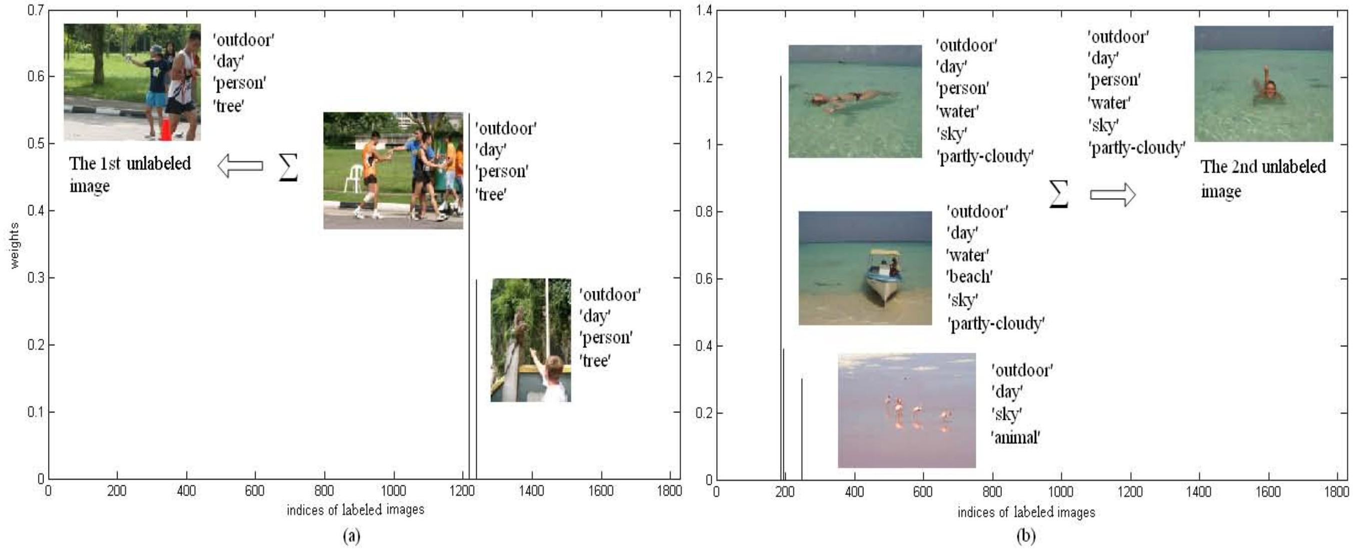


Fig. 1. Comparison of two examples of the forward SR on the ImageCLEF-VCDT 2008 dataset. We can see that 1) any one image is more or less similar to the images which are utilized to represent it by the SR in the feature space, and the test image can be annotated according to the annotations of the labeled images; 2) The 1st unlabeled image (a) is approximately equal to the weighted sum of two labeled images both with the 'person' annotation, while the 2nd unlabeled image (b) is approximately equal to the weighted sum of three labeled images, only one with the 'person' annotation and two without the 'person' annotation; Thereby, it can be deduced that the 1st test image owns a higher confidence with which it belongs to the concept 'person' than the 2nd one.

the unlabeled image with a weighted sum of them. Due to its good performance, SR has been applied into many computer vision topics such as image annotation [37], image restoration [28][29], and image classification [15][40][42][17].

In this paper, we discuss the SR's application in image auto-annotation. Apparently, in an ideal situation, an unlabeled image associated with the target concept or class can be thoroughly represented by a few labeled images annotated with the corresponding keywords, and the weights to the labeled images annotated without the keywords are all zeros in sparse representation. However, this fact doesn't hold in most cases due to the existence of other annotations (objects) and noise in the unlabeled image. Instead, an unlabeled image related to the target class is usually represented by both relevant and irrelevant labeled images. In these cases, the irrelevant components in sparse representation can be considered as noise deviating from the target class. So the confidence or score with which the unlabeled image belongs to the target class depends on the proportion of the irrelevant components in the whole sparse representation. For example in Figure 1, the first unlabeled image is approximately represented as the weighted sum of two

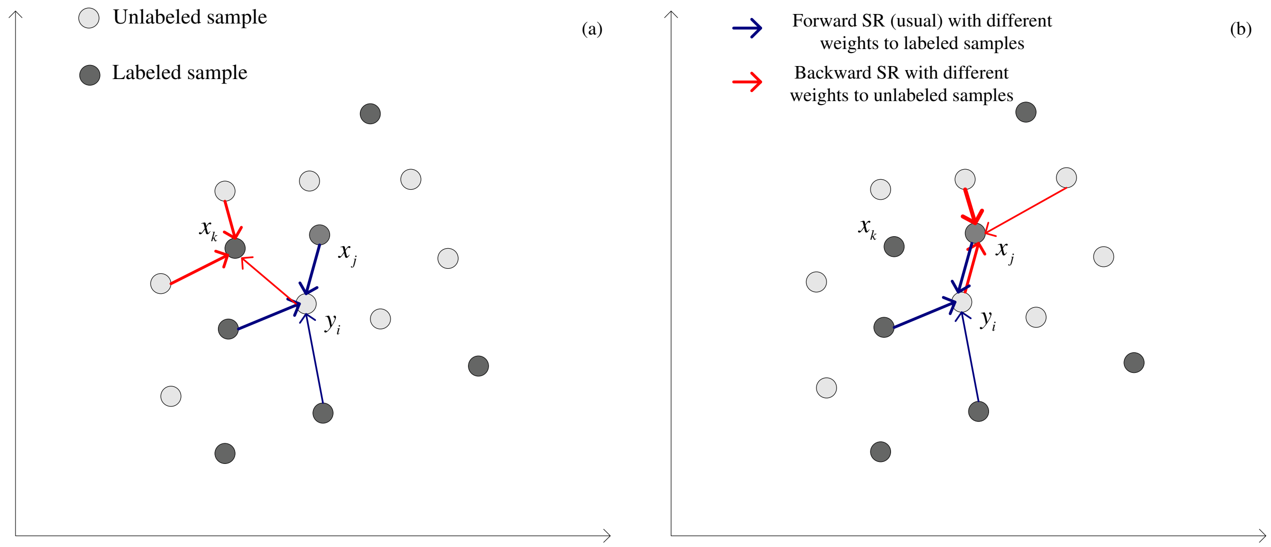
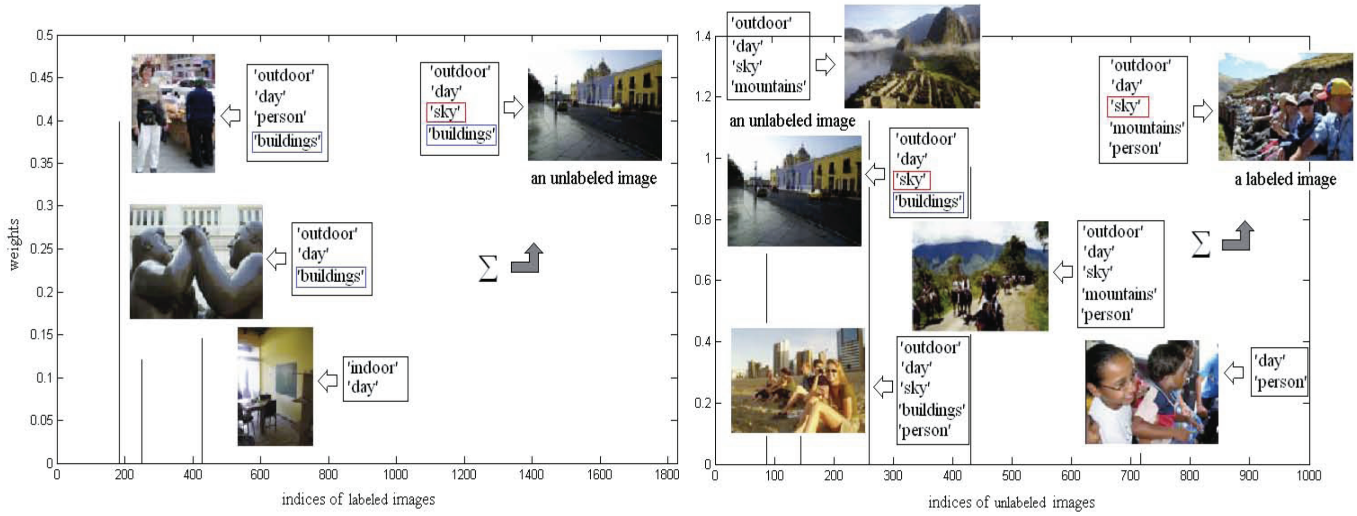

Unlabeled sample
Labeled sample
(a)
x_k
x_j
y_i
Forward SR (usual) with different weights to labeled samples
Backward SR with different weights to unlabeled samples
(b)
x_k
x_j
y_i

Fig. 2. Sketches of two examples of the forward and backward sparse representation being complementary to each other: (a) on covering new concepts; (b) on strengthening the associations.

labeled images both with the ’person’ annotation, while the second unlabeled image is approximately represented as the weighted sum of three labeled images, only one with the ’person’ annotation and two without the ’person’ annotation. Thereby, it can be deduced that the first test image owns a higher confidence with which it’s associated with the concept ’person’ than the second one.

Provided a set of labeled images and a set of unlabeled images, as a counter case of the usual SR which represents each unlabeled (test) image with a few labeled (training) images, each labeled image can also be represented with a few unlabeled images by the SR in a backward direction. In this case, if one certain unlabeled image is related to the target class, it’s apt to be selected to represent the relevant labeled images. On the contrary, if the unlabeled image is not related to the target class, it’s apt not to be selected to represent the relevant labeled images. Obviously, the backward SR contains useful information for the image annotation, and the unlabeled image can be annotated according to the annotations of the labeled images which it is selected to represent by the backward SR. It seems very interesting to find out how much this backward SR can contribute to the image annotation. Unfortunately, to the best of our knowledge, this problem has not been addressed by existing research efforts. The motivation of this paper also lies in that the SRs in two opposite directions may provide complementary information to each other. For example, in Figure 2(a), when the backward SR associates the unlabeled image y_i with



(a) The forward SR: representing unlabeled images with labeled images

(b) The backward SR: representing labeled images with unlabeled images

Fig. 3. An example of comparison between the forward and backward sparse representations on the ImageCLEF-VCDT 2008 dataset. (a) The forward SR represents an unlabeled image with several labeled images; (b) The backward SR represents a labeled image with several unlabeled images. We can see that the backward SR can supplement the annotation 'sky' which the forward SR misses, while the forward SR can supplement the annotation 'buildings' which the backward SR misses.

the labeled image x_k which the forward SR cannot associate y_i with, x_k may contain some new concepts which the labeled images utilized by the forward SR to represent y_i do not cover. Thereby, the backward SR can be a supplement to the forward SR. And in Figure 2(b), the labeled image x_j is utilized by the forward SR to represent the unlabeled image y_i, and on the other side, y_i is utilized by the backward SR to represent x_j. So the association between y_i and x_j is then strengthened, which implies that the probability with which y_i contains the same concepts as x_j is higher. This is also a complementary case. Figure 3 is an example of the case of Figure 2(a), which shows that the backward SR remedies the annotation 'sky' which the forward SR misses for the unlabeled image, and the forward SR remedies the annotation 'buildings' which the backward SR misses. So, exploring whether the sparse representations in two opposite directions can really provide this complementary information or not for image annotation is not only interesting from a theoretical point of view, but also very important for the SR's practical applications in image classification, annotation and retrieval.

The contribution of this paper is to design for image annotation two kinds of SR algorithms in opposite directions, to explore their complementary nature and then to build a cooperative sparse representation method which effectively combines them, with the expectation of the performance improvement of image annotation. Labeled images and unlabeled ones are both utilized to build our classification model to annotate the unlabeled images, so we will compare it by experiments with other state-of-the-art semi-supervised methods such as TSVM (Transductive Support Vector Machine) [20] and two graph-based semi-supervised methods: LGC (Local and Global Consistency) [48] and GFHF (Gaussian Fields and Harmonic Functions) [49]. We will also compare our approach with a fusion of non-sparse solutions in opposite directions to show that the sparsity of SR plays an important role in the cooperative image annotation.

The rest of this paper is organized as follows. Section II surveys related work. The forward and backward SR algorithms for image annotation are depicted in Section III. Section IV proposes the cooperative sparse representation algorithm which combines the forward SR with the backward SR by co-training. Section V depicts the kernel trick we adopt to implement the cooperative sparse representation. Section VI presents our experiments and discussions. Finally, some conclusions are drawn in Section VII.

II. Related Work

Recently, we have witnessed many applications of sparse representation in visual recognition. Wright et al. [40] exploited the sparse representation classification (SRC) method for robust face recognition and Wagner et al. [36] further pushed its practical application. Gao et al. proposed Laplacian sparse representation and obtained better performance for image classification in [16]. Yuan et al. [43] proposed a joint sparse representation model which combines multiple features for image classification. Zhang et al. [44] introduced sparse representation into feature selection for image annotation. They all assumed that any test image can be approximately a linear combination of a basis set formed by the training images, namely, $y = Xa$, where $X = [x_1, x_2, ..., x_m] \in \mathbb{R}^{d \times m}$ denotes m bases and $y \in \mathbb{R}^d$ is a new test image. If $m > d$, the solution of a which satisfies $y = Xa$ is not unique. This difficulty can be conquered by assuming that the test image y can be sufficiently represented by using only the training images from the same class and with the most similar visual features to the test image. In this case, the solution of a should be sparse if the number of classes or the size of the image database is relatively large while the percentage of positive images for each class is low. For instance, in the dataset of PASCAL-VOC 2010, the percentages of positive images are on average only about **5.0%** for all 20 classes, and for the class 'Cow', which owes the fewest positive images, the percentage of positive images is only about **1.7%**. So on the condition of the above assumption, a very small percentage of the entries of the solution of a should be nonzero. Moreover, this proportion could be further reduced by the assumption that the test image is represented using only the training images with the most similar visual features to the test image even though these images belong to the same class, due to high complexity and diversity of real scene images. The sparsity can be solved by minimizing the ℓ_1-norm of a. Compared with the ℓ_1-norm minimization, the ℓ_2-norm one usually results in a dense solution, while the ℓ_0-norm one is NP-hard and difficult to compute. The ℓ_1-norm minimization problem can be solved by efficient methods such as standard linear programming [5], homotopy algorithms [11], and so on.

If the forward and backward sparse representations complementarily annotate the images, those images with the most confident predictions of two SRs in different directions can be iteratively moved from the unlabeled set to the labeled set, which is expected to improve step by step the classification and annotation performances. This iterative training process is called co-training [4], which is a semi-supervised learning algorithm firstly proposed by Blum and Mitchell for web-page classification with only a small size of labeled examples and a large size of unlabeled examples. The co-training of Blum and Mitchell was based on the assumption that two feature sets of each example are conditionally independent given the class and each feature set is sufficient for classification. This idea of two different feature sets can be transformed to that of two different classifier models. Krogel and Scheffer [22] further showed in 2004 that co-training is beneficial only if two classifications are relatively independent, namely, one classifier correctly labels some examples which the other classifier misclassifies. Otherwise, if both classifiers agree on all the unlabeled data, co-training does not create any new information for single classifiers. Therefore, in this paper, we will adopt co-training to explore the difference and complementarity between the forward and backward SRs, and to build a cooperative sparse representation method which effectively combines them to improve image annotation performance.

Another important recent effort related to sparse representation is the kernel sparse representation proposed by Gao et al. for image classification in [15]. The kernel trick [32] maps non-linear separable features into a higher dimensional feature space, in which features of the same class are closely grouped together and those of different classes become linearly separable. Similarly, an image could be better linearly represented with a small number of other ones in the kernel space than in the original lower dimensional feature space. The kernel sparse representation(KSR) assumes that the sparse representation for the images can be more easily found, and the reconstruction error may be reduced as well. The KSR was applied into image classification and face recognition, and achieved better performance than SR in the original feature space. In our work, we will also implement the cooperative sparse representation in the kernel space and build a cooperative kernel sparse representation (Co-KSR), in order to obtain better image annotation performance.

III. Image Annotation Based on Sparse Representations

Given m bases $X = [x_1, x_2, ..., x_m] \in \mathbb{R}^{d \times m}$ and a new test signal $y \in \mathbb{R}^d$, sparse representation aims to find a sparse vector $a \in \mathbb{R}^m$ where y can be represented as:

$$y = Xa \tag{1}$$

This sparse representation problem can be solved by minimizing the following objective function:

$$\begin{gathered} \min_a \|y - Xa\|^2 + \lambda \|a\|_1 \\ subject\ to \quad \|x_i\|^2 \leq 1 \end{gathered} \tag{2}$$

where $\|.\|$ denotes the ℓ_2-norm, $\|.\|_1$ denotes the ℓ_1-norm; the first term is to minimize the reconstruction error, while the second one is to control the sparsity of the vector a, with the tradeoff parameter λ. A larger λ usually results in a sparser a solution.

Actually, real-world data are complex and noisy, so the test signal y usually cannot be exactly represented by a sparse superposition of the bases. Instead, the test signal y can be precisely represented as:

$$y = Xa + e \tag{3}$$

where the vector $e \in \mathbb{R}^d$ is the signal noise.

In this paper, we will design for image annotation two kinds of SR algorithms in opposite directions. We call them the forward SR and the backward SR, respectively.

A basic problem of image annotation can be described as follows. Given a class, a set of labeled images $X = [x_1, x_2, ..., x_m] \in \mathbb{R}^{d \times m}$ with their annotations $L = [l_1, l_2, ..., l_m]$ indicating whether they belong to the class or not, and a set of unlabeled images $Y = [y_1, y_2, ..., y_n] \in \mathbb{R}^{d \times n}$, where each image can be expressed by a d-dimension vector, m and n are the numbers of labeled and unlabeled images, respectively, the image annotation task is to assign each unlabeled image a confidence or score with which it's related to the class, or to determine whether the unlabeled image belongs to the class or not.

A. Forward Sparse Representation (Forward-SR)

As described in Eqn. (3), any unlabeled image y_i can be represented by a linear combination of a few labeled images as

$$\begin{aligned} y_i &= Xa_i + e_i \\ &= \alpha_{1i}x_1 + \alpha_{2i}x_2 + ... + \alpha_{mi}x_m + e_i \end{aligned} \tag{4}$$

where $a_i = [\alpha_{1i}, \alpha_{2i}, ..., \alpha_{mi}]^T$. This method which represents unlabeled images with labeled images is called forward sparse representation (Forward-SR) in our paper.

If the labeled set is divided into two subsets: $X^+ = [x_1^+, x_2^+, ..., x_u^+]$ and $X^- = [x_1^-, x_2^-, ..., x_v^-]$, where $u + v = m$, X^+ and X^- are mutually exclusive, consisting of the samples belonging to the annotation class and not, respectively, then the sparse representation of the unlabeled image y_i in Eqn. (4) can also be written as:

$$\begin{aligned} y_i = \alpha_{1i}^+ x_1^+ + \alpha_{2i}^+ x_2^+ + ... + \alpha_{ui}^+ x_u^+ \\ + \alpha_{1i}^- x_1^- + \alpha_{2i}^- x_2^- + ... + \alpha_{vi}^- x_v^- + e_i \end{aligned} \tag{5}$$

where α_{ji}^+ and α_{ji}^- are two elements of a_i, and denote the weights to x_j^+ and x_j^-, respectively. If the unlabeled image y_i belongs to the class, in an ideal situation where y_i doesn't contain any noise deviating from the concept which the class indicates, y_i can be thoroughly represented by a few positive labeled samples from X^+, while the weights to the labeled samples from X^- are all zeros, namely, $\alpha_{ji}^- = 0$ for all $j = 1, 2, ..., v$. On the contrary, if the unlabeled image y_i doesn't belong to the annotation class, in an ideal situation where y_i doesn't contain any contents associated with the concept which the class indicates, y_i can be thoroughly represented by a few negative labeled samples from X^-, while the weights to the labeled samples from X^+ are all zeros, namely, $\alpha_{ji}^+ = 0$ for all $j = 1, 2, ..., u$.

However, in most situations, the unlabeled image y_i is represented by both a few positive labeled samples and a few negative labeled ones due to noise or concept correlations. In these cases, the proportion of the positive components in the whole sparse representation

$$\frac{\sum_j \alpha_{ji}^+}{\sum_j \alpha_{ji}^+ + \sum_j \alpha_{ji}^-} \tag{6}$$

can be assigned to y_i as the score[1] or degree with which it belongs to the annotation class.

Note that the reconstruction error $\|e_i\|_2 = \|y_i - X a_i\|_2$ is invariant to simultaneously multiply X by a scalar and a_i by the inverse of the scalar, and the non-zero elements α_{ji}^+ or α_{ji}^- in a_i are used as the similarities between the unlabeled image y_i and the corresponding labeled samples. So the ℓ_2 norm of each basis x_j^+ or x_j^- should be normalized to the same (usually as 1) to ensure the scoring by Eqn. (6) to be fair. Then the objective function (2) is modified to be:

$$\min_{a_i} \|y_i - X\Lambda_X a_i\|^2 + \lambda \|a_i\|_1 \tag{7}$$

[1] The literature [40] used the residual $\|y_i - \Sigma_j \alpha_{ji}^+ x_j^+\|_2$ for human face classification, by which the score can also be defined as $1/\|y_i - \Sigma_j \alpha_{ji}^+ x_j^+\|_2$. Our experiments show that this definition is slightly worse than that of Eqn. (6) defined in our paper by the results evaluated by the AUC and EER criteria.

And the unlabeled image y_i can be represented as:

$$y_i = X\Lambda_X a_i + q_i \tag{8}$$

where $\Lambda_X = diag\{1/\|x_1\|_2, 1/\|x_2\|_2, ..., 1/\|x_m\|_2\}$ is the diagonal matrix to normalize each basis in the X , and the vector $q_i \in \mathbb{R}^d$ denotes the noise vector.

Considering all unlabeled images $Y = [y_1, y_2, ..., y_n]$, then

$$Y = X\Lambda_X A + Q \tag{9}$$

where $Y \in \mathbb{R}^{d\times n}$, $A = [a_1, a_2, ..., a_n] \in \mathbb{R}^{m\times n}$, and $Q = [q_1, q_2, ..., q_n] \in \mathbb{R}^{d\times n}$ are the image noise.

The image annotation procedure based on the forward sparse representation is summarized as Algorithm 1.

Algorithm 1 Forward Sparse Representation (Forward-SR) for Image Annotation.

```
Solve the Forward SR.
    for i = 1:n
    (i): Input the matrix of labeled images $X = [x_1,$
            $x_2, ..., x_m]$ and the unlabeled image $y_i$.
    (ii): Solve the minimization problem depicted by
          Eqn. (7) and obtain the sparse vector $a_i$.
    end
    Obtain the sparse matrix $A = [a_1, a_2, ..., a_n]$.
Annotate unlabeled images.
    for each annotation class
    (i): Input the class labels of all labeled images $L =$
          $\{l_1, l_2, ..., l_m\}$.
          for i=1:n
    (ii):       Assign $y_i$ the score computed by Eqn. (6)
                  using $a_i$ and $L$.
          end
    (iii): Annotate the unlabeled images by their scores.
    end each annotation class
```

B. Backward Sparse Representation

Provided a set of unlabeled images $Y = [y_1, y_2, ..., y_n] \in \mathbb{R}^{d \times n}$, any labeled image x_j can also be represented by sparse representation as:

$$x_j = Y \Lambda_Y b_j + r_j \tag{10}$$

where the sparse vector b_j can be solved by minimizing the following objective function:

$$\min_{b_i} \|x_i - Y \Lambda_Y b_i\|^2 + \lambda \|b_i\|_1 \tag{11}$$

This method which represents labeled images with unlabeled images is called backward sparse representation (Backward-SR) in this paper.

Considering all labeled images $X = [x_1, x_2, ..., x_m] \in \mathbb{R}^{d \times m}$, we obtain

$$X = Y \Lambda_Y B + R \tag{12}$$

where

$$B = [b_1, b_2, ..., b_m] \in \mathbb{R}^{n \times m}, \tag{13}$$

$R = [r_1, r_2, ..., r_m] \in \mathbb{R}^{d \times m}$ is the image noise , and $\Lambda_Y = diag\{1/\|y_1\|_2, 1/\|y_2\|_2, ..., 1/\|y_n\|_2\}$ is the diagonal matrix to normalize Y.

Here, Eqn. (13) can be written as

$$B = \begin{bmatrix} b_1^{'} \\ b_2^{'} \\ ... \\ b_n^{'} \end{bmatrix} \in \mathbb{R}^{n \times m}, \tag{14}$$

where $b_i^{'}$ denotes the ith row of B. Considering the elements in $b_i^{'}$ and resorting them, then we can obtain

$$b_i^{'} = [\beta_{i1}^{+}, \beta_{i2}^{+}, ..., \beta_{iu}^{+}, \beta_{i1}^{-}, \beta_{i2}^{-}, ..., \beta_{iv}^{-}], \tag{15}$$

where $u + v = m$, β_{ij}^{+} and β_{ij}^{-} denote the connection weight between y_i and x_j^{+}, and that between y_i and x_j^{-}, respectively. Different from the case of the Forward-SR, if the unlabeled image y_i belongs to the annotation class, in an ideal situation, y_i can only be selected to represent the positive labeled images, so $\beta_{ij}^{-} = 0$ for all $j = 1, 2, ..., v$. On the contrary, if the unlabeled image y_i does not belong to the annotation class, in an ideal situation, y_i can only be selected to represent the negative labeled images, so $\beta_{ij}^{+} = 0$ for all $j = 1, 2, ..., u$. Thereby, the proportion of β_{ij}^{+} in all elements of the vector $b_i^{'}$ expresses the score or degree with which the unlabeled image y_i belongs to the annotation class, and y_i can be assigned the score

$$\frac{\sum_j \beta_{ij}^{+}}{\sum_j \beta_{ij}^{+} + \sum_j \beta_{ij}^{-}} \tag{16}$$

Algorithm 2 Backward Sparse Representation (Backward-SR) for Image Annotation.

```
1: Solve the Backward SR.
   for i = 1:m
   (i): Input the matrix of unlabeled images $Y = [y_1, y_2,$
        $..., y_n]$ and the labeled image $x_i$.
   (ii): Solve the minimization problem depicted by
        Eqn. (11) and obtain the sparse vector $b_i$.
   end
   Obtain the sparse matrix $B = [b_1, b_2, ..., b_m]$.
2: Annotate unlabeled images.
   for each annotation class
   (i): Input the class labels of all labeled images $L =$
        $\{l_1, l_2, ..., l_m\}$.
        for i=1:n
   (ii):     Assign $y_i$ the score computed by Eqn. (16)
             using $b_i^{'}$ and $L$.
        end
   (iii): Annotate the unlabeled images by their scores.
   end each annotation class
```

The image annotation procedure based on the backward sparse representation is summarized as Algorithm 2.

IV. Cooperative Sparse Representations for Image Annotation

In this section, a cooperative sparse representation method which combines the forward-SR and backward-SR by using co-training is proposed for image annotation.

A. Co-training of Forward and Backward Sparse Representations

Co-training [4], which is an iterative semi-supervised learning algorithm, can enhance the classification performances of two classifiers which are complementary and relatively independent to each other. Here, it is utilized to demonstrate the difference and complementarity between the forward and backward SRs, and to build an effective cooperative sparse representation method to improve image annotation performance. The co-training of forward and backward SRs is described as Algorithm 3.

Algorithm 3 Co-training of Forward and Backward Sparse Representations.

```
Given a labeled image set $X = [x_1, x_2, ..., x_m]$ with class labels and an unlabeled image set $Y = [y_1, y_2, ..., y_n]$ to annotate,
for each annotation class:
Loop for $k$ iterations:
    Solve the forward-SR with $X$ and $Y$, and obtain the sparse matrix $A$ which satisfies Eqn. (9).
    Solve the backward-SR with $X$ and $Y$, and obtain the sparse matrix $B$ which satisfies Eqn. (12).
    Assign the unlabeled images in $Y$ the scores computed by Eqn. (6) using the sparse matrix $A$.
    Assign the unlabeled images in $Y$ the scores computed by Eqn. (16) using the sparse matrix $B$.
    Select the unlabeled image with the highest score and the one with the lowest score by the forward-SR, remove them from $Y$, and add them into $X$.
    Select the unlabeled image with the highest score and the one with the lowest score by the backward-SR, remove them from $Y$, and add them into $X$.
end Loop
end each annotation class
```

Here, only one image with the highest score and one with the lowest score are selected by each of the forward-SR and the backward-SR, namely, only totally 4 images are moved from Y to X for each iteration. The varying of the numbers of selected positive and negative images per iteration in a reasonable range results in little effect on the annotation performance, according to our experiments. The number of iterations k is usually predefined by experience [4][22].

Also note that for the datasets in which each image has multiple class labels, the co-training has to be run by class since the transmitted images from the unlabeled set to the labeled set differ between classes.

B. Fusion of Forward and Backward Sparse Representations

A simple average fusion is usually performed when the co-training of two classifiers stops at the kth iteration [4][22][41]. And the previous works [4][22] have proved that the average fusion after co-training can further improve the performance. We follow these works, and propose an average fusion method to combine the outputs of the forward-SR with the backward-SR for image annotation after the co-training.

As described in Section III, the connection weight between the ith test image and the jth training image is defined as the jth-row-ith-column element of the matrix A, α_{ji}, by the forward-SR, and defined as the ith-row-jth-column element of the matrix B, β_{ij}, by the Backward-SR. So the weighted-averaging fusion of these two connection weights is as follows:

$$\gamma_{ji} = (1-w)\alpha_{ji} + w\beta_{ij} \tag{17}$$

where w is the fusion weight which is usually set to be 0.5. Note that we normalize α_{ji} and β_{ij} to be in the range of [0, 1] by the min-max normalization method: $\alpha_{ji} = \frac{\alpha_{ji} - \min_j \alpha_{ji}}{\max_j \alpha_{ji} - \min_j \alpha_{ji}}$, $\beta_{ij} = \frac{\beta_{ij} - \min_j \beta_{ij}}{\max_j \beta_{ij} - \min_j \beta_{ij}}$ before the average fusion is performed. Then the connection weights between the ith unlabeled image and all m labeled images are $c_i = [\gamma_{1i}, \gamma_{2i}, ..., \gamma_{mi}]^T$. Considering all n unlabeled images, we obtain:

$$\begin{aligned} C &= [c_1, c_2, ..., c_n] \\ &= (1-w)A + wB^T \end{aligned} \tag{18}$$

Like the case of the forward-SR, the connection weights in C depict the similarities and relevance degrees between the unlabeled images and the labeled ones. So the unlabeled image y_i is then assigned the score

$$\frac{\sum_j \gamma_{ji}^+}{\sum_j \gamma_{ji}^+ + \sum_j \gamma_{ji}^-} \tag{19}$$

with which it belongs to the annotation class, where γ_{ji}^+ and γ_{ji}^- denote the connection weight between y_i and x_j^+, and that between y_i and x_j^-, respectively. Then we can annotate unlabeled images by their scores.

This average fusion process could further exploit the complementary information between two SRs in opposite directions and enhance the annotation performance if the co-training has not collected all complementary information.

C. Independence between Two Opposite SRs

The key requirement of co-training is that the two classifiers are with difference or at least not so tightly correlated [4][2][22]. The conditional independence of two opposite SRs depends on the sparsity of SR and the diversity of scene images. The forward SR associates the test image y_i with a very small subset of labeled images $X_f \subset X$ and their class labels $L_{X_f} \subset L$ thanks to the sparse nature, while the backward SR associates y_i with another very small subset of labeled images $X_b \subset X$ and their class labels $L_{X_b} \subset L$. In the meantime, because of the diversity of scene images, the connections between y_i and any labeled image in two opposite directions usually does not simultaneously appear. So the sparsity of the SR and the diversity of scene images together ensure that the labeled subsets $\{X_f, L_{X_f}\}$ and $\{X_b, L_{X_b}\}$ differ from each other or have few overlaps. According to the score definitions in Eqn. (6) and Eqn. (16), the forward SR classifier and the backward SR classifier are actually based on the label subsets L_{X_f} and L_{X_b}, respectively, so the difference between L_{X_f} and L_{X_b} would lead to different predictions of the forward SR and the backward SR.

V. KERNEL TRICK

The kernel trick [32][1] maps nonlinearly separable features into a higher dimensional feature space in which the features may be linearly separable. Similarly, the kernel trick could map nonlinearly or non-sparsely representable features into a higher dimensional feature space in which the features may be linearly and sparsely representable. Gao et al. [15] proposed kernel sparse representation(KSR), assuming that the sparse representation for the images can be more easily found, and the reconstruction error may be reduced as well. They applied KSR into image classification and face recognition, and achieved better performance than SR in the original feature space. In our work, we also implement our cooperative sparse representation in the kernel space.

Assume that there is a feature mapping $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^H, \quad d < H$, which maps all image features to a higher dimensional space, namely $y \rightarrow \phi(y)$, $X = [x_1, x_2, ..., x_m] \rightarrow \chi = [\phi(x_1), \phi(x_2), ..., \phi(x_m)]$. Then the objective function of sparse representation in the kernel space can be written as:

$$\min_a \|\phi(y) - \chi a\|^2 + \lambda \|a\|_1 \tag{20}$$

In our work, the Gaussian function $\kappa(x_1, x_2) = \exp(-\nu\|x_1 - x_2\|^2)$ is used as the kernel function. Note that $\phi(x_i)^T \phi(x_i) = \kappa(x_i, x_i) = 1$, so the normalization constraint on the bases can be ignored.

The objective function (20) can be expanded and written as follows:

$$\begin{aligned} &\|\phi(y) - \chi a\|^2 + \lambda\|a\|_1 \\ &= (\phi(y) - \chi a)^T(\phi(y) - \chi a) + \lambda\|a\|_1 \\ &= \phi(y)^T\phi(y) + a^T\chi^T\chi a - \phi(y)^T\chi a - a^T\chi^T\phi(y) + \lambda\|a\|_1 \\ &= 1 + a^T K_{XX} a - 2a^T K_X(y) + \lambda\|a\|_1 \\ &= E(a) + \lambda\|a\|_1 \end{aligned} \tag{21}$$

where $E(a) = 1 + a^T K_{XX} a - 2a^T K_X(y)$, K_{XX} is an $m*m$ matrix whose elements $\{K_{XX}\}_{ij} = \kappa(x_i, x_j)$, and $K_X(y)$ is an $m * 1$ vector whose elements $\{K_X(y)\}_i = \kappa(x_i, y)$. We can see that the changes of the kernelized objective function to the original space only lies in the uses of K_{XX} and $K_X(y)$. So the homotopy algorithm [3] can be easily extended to solve this kernel sparse representation problem, with the extra computations of K_{XX} and $K_X(y)$.

Since X is always changing by iterations of the co-training, it seems necessary to compute K_{XX} and $K_X(y)$ repeatedly by iterations. However, we can avoid it by matrix decomposition. For the forward kernel sparse representation, at the kth iteration of the co-training,

$$E_k(a) = 1 + a^T K_{X_k X_k} a - 2a^T K_{X_k}(y) \tag{22}$$

where X_k is the labeled image set at the kth iteration, and at the $(k+1)$th iteration,

$$\begin{aligned} E_{k+1}(a) =& 1 + a^T K_{X_{k+1} X_{k+1}} a - 2a^T K_{X_{k+1}}(y) \\ =& 1 + a^T \begin{bmatrix} K_{X_k X_k} & K_{X_k X_{new}} \\ K_{X_{new} X_k} & K_{X_{new} X_{new}} \end{bmatrix} a \\ &- 2a^T \begin{bmatrix} K_{X_k}(y) \\ K_{X_{new}}(y) \end{bmatrix} \end{aligned} \tag{23}$$

where $X_{k+1} = [X_k, X_{new}]$, X_{new} denotes the newly added images to the labeled set in the co-training. Comparing Eqn. (23) with (22), we can see that at the $(k+1)$th iteration, $K_{X_{new} X_k}$ is the transpose of $K_{X_k X_{new}}$, so we only need to compute $K_{X_k X_{new}}$ and $K_{X_{new} X_{new}}$ in order to obtain $K_{X_{k+1} X_{k+1}}$, with the existing knowledge of $K_{X_k X_k}$ at the kth iteration. Similarly, at the $(k+1)$th iteration, we only need to compute $K_{X_{new}}(y)$ in order to obtain $K_{X_{k+1}}(y)$, with the existing knowledge of $K_{X_k}(y)$ at the kth iteration.

As for the backward kernel sparse representation, the kernelized bases are decreasing with the co-training iterations. Apparently, $K_{Y_{k+1} Y_{k+1}}$ is a square submatrix of $K_{Y_k Y_k}$, and $K_{Y_{k+1}}(y)$ is a subvector of $K_{Y_k}(y)$. So the kernelized bases of the $(k+1)$th iteration can be easily extracted from those of the kth iteration by removing the components corresponding to the samples moved from the unlabeled set to the labeled set.

VI. Experiments and Discussions

In this section, we explore the effectiveness of the co-training of the forward and backward KSRs, and evaluate the performance of the proposed cooperative kernel sparse representation (Co-KSR) by comparing it with other state-of-the-art semi-supervised methods such as the TSVM (Transductive Support Vector Machine) [20], LGC (Local and Global Consistency) [48] and GFHF (Gaussian Fields and Harmonic Functions) [49]. We stop the co-training of the Co-KSR at the 100th iteration, and then calculate its performance.

A. Experimental Setup

Datasets: Two image annotation datasets of famous campaigns: ImageCLEF-VCDT 2008 [35] and PASCAL-VOC 2010 (Classification Task) [13] are used to perform our comparative study.

The ImageCLEF-VCDT task provides a training set of 1827 images with annotations from a set of total 17 keywords, and a test set of 1000 images. We merge them together, and obtain a whole set of 2827 images. Then we randomly select $(10\%, 20\%, ..., 60\%)$ of images, respectively, from the whole set, and use them as the labeled images while the rest as the unlabeled ones [2]. So we have totally 6 experiments for this dataset.

The image classification task of PASCAL-VOC 2010 [13] provides a training set of 4998 images with the annotations from a set of total 20 keywords. We randomly select $(10\%, 20\%, ..., 60\%)$, respectively, from the training set, and use them as the labeled images while the rest as the unlabeled ones. We also have totally 6 experiments for this dataset.

Image Features: Feature selection is an open problem and might have a great impact on the results. The traditional and usual features are global ones such as color [30][34][33][8] and texture [7]. Recently, BoF (Bag-of-Features) models such as SIFT (scale-invariant feature transform) [27], spin [23], RIFT (rotation-invariant feature transform) [24], and so on, have been investigated and proved to owe more

[2] The groundtruth of the test images of ImageCLEF-VCDT 2008 was released, which allows this process probably to use some of the test images as the labeled ones.

powerful image description capabilities. In our work, we use the usual SIFT image descriptors [3], whose parameter settings are as follows. The dense grid sampling strategy is used, and the grid spacing and patch size are set to be 8 and 16, respectively. As for the codebook, we set the number of clusters to be 1024 in k-means, and randomly select $8 * 10^4$ features from the whole training feature set to generate codebook for each data set. Then the ScSPM [42] method is used for coding, in which the spatial block number on each level of the pyramid is set as $[1, 2, 4]$, the weight for features on each level is set as $[1, 1, 1]$, the sparsity regularization parameter 0.15, and the smooth regularization for sparse coding is set as 10^{-3}. Finally, the PCA [21][14] is utilized to reduce the SIFT codes to 200 dimensions, which produces the image vectors with $d = 200$.

ℓ_1-Minimization Solution: We use the homotopy algorithm which has shown to be with a relatively good performance and a low time complexity in [40][3]. We use the matlab toolbox of BPDN-homotopy provided on the webpage http://users.ece.gatech.edu/%7Esasif/homotopy/ (The same is also provided on http://www.eecs.berkeley.edu/∼yang/software/l1benchmark/). We set the tradeoff parameter λ in Eqn. (7) and Eqn. (11) to be 0.005 (actually, according to our experiments, the BPDN-homotopy algorithm is very stable and gives almost the same results when the parameters meet $0 < \lambda \leq 0.01$). According to our experiments, an ℓ_1-minimization solution with thousands of bases in a 200 dimensional feature space takes only about $1/20$ second with the BPDN-homotopy algorithm implemented by the Matlab 7.1.0.183 (R14) Service Pack 3 running on a computer with the CPU of 16 Quad-Core AMD Opteron(tm) Processor 8382, the Memory of 66176824 kB, and the OS of Debian GNU/Linux 5.0.

Performance Evaluation: The ROC curves, which plot the true positive rate vs the false positive rate with different decision thresholds, are used for performance evaluation. Two typical measures derived from the ROC curves are used to evaluate the performances: area under curve (AUC) and equal error rate (EER). The AUC is the area under the ROC curve, and a larger AUC means a better ROC curve. The EER denotes the error rate at which the false positive rate is equal to the false negative rate, and smaller EER means better classification results. Furthermore, for comparison convenience, the average AUC and EER over all annotation terms of each dataset are calculated for each method.

[3] We also have some results with a concatenation of the traditional global features: color histogram [34], color moment [33] and wavelet texture [7]. The comparative results and conclusions are consistent with those over SIFT features which are provided in the paper, though the average AUC and EER performances of the global features are worse than those of SIFT.

B. Experimental Results and Discussions

We predefine the number of iterations of the co-training as $k = 100$, and run for 20 times with different random selections of the labeled images from the whole image set [9]. Then the ROC curves and confusion matrices of the proposed Co-KSR at the end (the 100th iteration) of the co-training on two datasets are given in Figure 4 and Figure 5, respectively, from which we can see that the Co-KSR can achieve a good performance for each annotation class. These results also show that the AUCs increase, while the EERs decrease with the enlargement of the ratio of labeled images; the variation of AUCs and EERs is sharp when the number of labeled images is small, while becomes flattened when the number of labeled images increases.

To explore the complementarity of two KSRs in opposite directions and the effectiveness of the co-training, we compute the average AUC and EER of all classes from the ROC curve by iteration of the co-training, and plot the average AUC and EER in Figure 6 and Figure 7, respectively. Though our approach switches to the average fusion process at the end of the co-training, in order to better show the complementary nature of two opposite SR, we also plot the average fusion result for each iteration in these figures. From these figures, we have the following observations:

(1) The AUC increases while the EER decreases by the number of iterations of the co-training, which shows that two KSRs in opposite directions are different and independent to some extent according to the conclusions in [2][4][22];

(2) The Co-KSR improves the AUC evaluation about 6.1% averagely on two datasets over the forward KSR without the co-training and about 23.6% averagely over the backward KSR, while reduces the EER evaluation about 3.7% averagely on two datasets over the forward KSR and about 9.8% averagely over the backward KSR. And even at the 1st iteration before the co-training starts, the simple average fusion of two KSRs in opposite directions can achieve better performance than either one of them. These results show that two KSRs in opposite directions contain complementary information to each other for image annotation;

(3) The backward KSR is relatively weak for image annotation when the number of labeled images is small. For a large-scale image dataset, the backward KSR with a small number of labeled images can annotate only a small number of unlabeled images due to the nature of sparsity of SR, which will result in the weakness of the backward KSR and its fusion with the forward KSR. This problem can be solved by firstly dividing the whole unlabeled set into several smaller subsets, then performing the backward KSR on each pair of the whole labeled set and one of unlabeled subsets, and finally combining the results

of all pairs to achieve the result of the whole unlabeled set;
(4) The performances of the forward and backward KSRs are both improved with the increase of the number of labeled images, but the backward KSR is improved more than the forward one. The backward KSR can achieve a performance approximately equal to the forward KSR when the percentage of the labeled images is larger than 50%.

We compare our Co-KSR with the usual KSR (the single forward kernel sparse represnetation), and other methods such as TSVM [20], LGC [48] and GFHF [49]. TSVM is a state-of-the-art semi-supervised model adopting the same RBF kernel as that in our Co-KSR and has also been applied in image annotation [46]. LGC and GFHF are both state-of-the-art graph-based semi-supervised methods adopting the Gaussian function and LGC has been used in image classification [19]. In addition, to address the doubt about whether the SR is appropriate or not for image classification [31], we also compare Co-KSR with a non-sparse solution with the ℓ_2-norm constraint in the RBF kernel space (K-L2), which can be simply computed via the pseudoinverse of the base matrix [12]. We use cross-validation to set the width parameter in the RBF or Gaussian functions. Finally, we plot the average AUC and EER for different semi-supervised methods on two datasets in Figure 8 and Figure 9, respectively. From these figures, we can see that Co-KSR achieves average AUC and EER performances much better than KSR. For the ImageCLEF-VCDT dataset, Co-KSR outperforms other state-of-the-art semi-supervised methods such as TSVM, LGC and GFHF, while for the PASCAL-VOC dataset, the proposed Co-KSR method outperforms other semi-supervised methods when the ratio of the labeled images is more than 20%, and achieves a performance approximately equal to the best (GFHF) of other semi-supervised methods when the labeled images are fewer.

These results also show that the non-sparse solution (K-L2) works better than KSR, which confirms the conclusion in [31]. But our Co-KSR remedies the performance loss brought by the sparsity, and achieves better annotation performances than K-L2. The comparison with K-L2 raises the argument that the fusion of K-L2 in the forward and backward directions also works and may perform better than that of KSR. So we also run experiments of the co-training with K-L2 in two opposite directions on the ImageCLEF-VCDT 2008 dataset, and plot the average AUC and EER by iteration of the co-training in Figure 10 (to save space, the experimental results on the PASCAL-VOC 2010 dataset are omitted). Figure 10 shows that the AUC decreases while the EER increases by the number of iterations of the co-training, and even the simple average fusion of two K-L2 in opposite directions performs worse than the single forward K-L2, which confirms the conclusion in Subsection IV-C that the sparsity is one of the necessary conditions by which the image representations in opposite directions differ enough from each other and the co-training of them can work well. The results also show that the single backward K-L2 performs worse and worse when the number of labeled images increases. Actually, the backward K-L2 to any unlabeled image can be considered as the fusion of image representations of m labeled images with the unlabeled ones, as shown in Algorithm 2. Thereby, this further confirm that the sparsity plays an important role in the diversification and fusion of image representations.

In addition, to explore the robustness of our proposed method to noisy image labels, addressing the real-life scenario where some of labels or keywords are not related with the images they are associated to, we randomly select $[10\%, 20\%, 30\%, 40\%, 50\%]$ labels of all labeled images, and negate them, namely for each annotation class, we set the image not associated if it belongs to the class while associated if it doesn't belong to the class. Then we perform experiments on these noisy datasets. We repeat the above process for 20 times. The curves of the average AUC and EER of various methods vs. the percentage of noisy labels for the ImageCLEF-VCDT 2008 and PASCAL-VOC 2010 datasets are plotted in Figures 11 and 12, respectively. The experimental results show that on the whole, the proposed Co-KSR achieves the best performance, while similar to other methods when the noisy percentage is larger than 40%. The results also show that KSR performs better than K-L2 on the noisy data, contrary to the case without label noise, which indicate that the sparse solution is more robust than the non-sparse one to image annotation when the image labels are partially incorrect.

VII. Conclusions

In this paper, we explored the complementary nature for image annotation between the sparse representation in two opposite directions. We firstly designed the forward and backward SR algorithms for image annotation, respectively. Then we adopted co-training to explore the difference and complementarity between the forward and backward SRs, and to build a novel semi-supervised learning model called cooperative sparse representation which effectively combines them to improve image annotation performance. Finally, the proposed method was implemented in a nonlinear kernel space. Experimental results and analyses showed that the backward KSR also works on image annotation and can do almost equally to the forward (usual) KSR when the percentage of the labeled images is larger than 50%; two KSRs in opposite directions are relatively different and independent classifiers, and the Co-KSR which combines them by co-training is effective on image annotation with a high performance improvement over either single one of the forward and backward KSRs without co-training with each other; the proposed Co-KSR works better than the K-L2 method which is a non-sparse solution; it also outperforms other state-of-the-art semi-supervised classifiers such as the TSVM, GFHF, and LGC, and is robust to image label noise. Thereby, our proposed Co-KSR method can be an effective method for semi-supervised image annotation, and especially, it is suitable to expand the size of training set for building any image classification model while a small number of images are provided with labels.

Acknowledgments

This research has been supported by the National Natural Science Foundation of China (61005007, 60875012, 61100161), the National 863 Program of China (2012AA011005), the US National Science Foundation (CCF-0905337), the French ANR Foundation (AVEIR ANR-06-MDCA-002), the Science and Technological Fund of Anhui Province for Outstanding Youth (10040606Y09) and the China Postdoctoral Science Foundation (20100470825, 201104324).

The authors would like to thank M. Salman Asif and Justin Romberg, who provided their BPDN-homotopy codes on the Internet for sharing, and to thank Jianchao Yang, who provided the SIFT ScSPM codes for sharing. The authors would also like to thank the anonymous reviewers who provided extensive constructive comments that improved our work.

References

[1] M. E. Abbasnejad, D. Ramachandram, and R. Mandava. A survey of the state of the art in learning the kernels. *Knowledge and Information Systems*, DOI: 10.1007/s10115-011-0404-6, 2011.

[2] S. Abney. Bootstrapping. *40th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference (ACL)*, pages 360–367, 2002.

[3] M. S. Asif and J. Romberg. Dynamic updating for ℓ_1 minimization. *IEEE Journal of Selected Topics in Signal Processing*, 4(2), 2010.

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. *COLT: Proceedings of the Workshop on Computational Learning Theory*, pages 92–100, 1998.

[5] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM REVIEW*, 43(1):129–159, 2001.

[6] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2), 2008.

[7] I. Daubechies. Ten lectures on wavelets. *CBMS-NSF Conference Series in Applied Mathematics, SIAM Ed.*, 1992.

[8] Y. Deng, B. S. Manjunath, C. Kenney, M. S. Moore, and H. Shin. An efficient color representation for image retrieval. *IEEE Transactions on Image Processing*, 10(1):140–147, 2001.

[9] G. R. Doddington, M. A. Przybocki, A. F. Martin, and D. A. Reynolds. The nist speaker recognition evaluation - overview, methodology, systems, results, perspective. *Speech Communication*, 31(2):225–254, 2000.

[10] C. Domeniconi, J. Peng, and B. Yan. Composite kernels for semi-supervised clustering. *Knowledge and Information Systems*, 28(1):99–116, 2011.

[11] D. L. Donoho and Y. Tsaig. Fast solution of l1-norm minimization problems when the solution may be sparse. *IEEE Transactions on Information Theory*, 54(11):4789–4812, 2008.

[12] M. Elad. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Section 1.2, 2010.

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html.

[14] J. E. Fowler. Compressive-projection principal component analysis. *IEEE Transactions on Image Processing*, 18(10):2230–2242, 2009.

[15] S. Gao, I. W.-H. Tsang, and L.-T. Chia. Kernel sparse representation for image classification and face recognition. *ECCV*, pages 1–14, 2010.

[16] S. Gao, I. W.-H. Tsang, L.-T. Chia, and P. Zhao. Local features are not lonely c laplacian sparse coding for image classification. *CVPR*, pages 3555–3561, 2010.

[17] J. Gui, Z. Sun, W. Jia, R. Hu, Y. Lei, and S. Ji. Discriminant sparse neighborhood preserving embedding for face recognition. *Pattern Recognition*, 45(8):2884–2893, 2012.

[18] M. Guillaumin, J. Verbeek, and C. Schmid. Multimodal semi-supervised learning for image classification. *CVPR*, 36:902–909, 2010.

[19] J. He, M. Li, H.-J. Zhang, H. Tong, and C. Zhang. Generalized manifold-ranking-based image retrieval. *IEEE Transactions on Image Processing*, 15(10):3170–3177, 2006.

[20] T. Joachims. Transductive inference for text classification using support vector machines. *ICML*, pages 200–209, 1999.

[21] I. Jolliffe. *Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed.* Springer, 2002.

[22] M.-A. Krogel and T. Scheffer. Multi-relational learning, text mining, and semi-supervised learning for functional genomics. *Machine Learning*, 57:61–81, 2004.

[23] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using affine-invariant regions. *CVPR*, 2:319–324, 2003.

[24] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. *Proceedings of the British Machine Vision Conference*, 2004.

[25] J. Li and J. Wang. Real-time computerized annotation of pictures. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(6):985–1002, 2008.

[26] J. Liu, M. Li, Q. Liu, H. Lu, and S. Ma. Image annotation via graph learning. *Pattern Recogn.*, 42(2):218–228, 2009.

[27] D. Lowe. Object recognition from local scale-invariant features. *ICCV*, 2:1150–1157, 1999.

[28] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, 2008.

[29] J. Marial, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. *ICCV*, pages 2272–2279, 2009.

[30] G. Pass. Comparing images using color coherence vectors. *ACM Int. Conf. Multimedia*, pages 65–73, 1997.

[31] R. Rigamonti, M. Brown, and V. Lepetit. Are sparse representations really relevant for image classification. *CVPR*, pages 1545–1552, 2011.

[32] B. Scholkopf, A. Smola, and K. Muller. Kernel principal component analysis. *International Conference on Artificial Neural Networks*, pages 583–588, 1997.

[33] M. Stricker and M. Orengo. Similarity of color images. *SPIE Storage and Retrieval for Image and Video Databases*, pages 381–392, 1995.

[34] M. Swain and D. Ballard. Color indexing. *Int. J. Comput. Vis.*, 7(1):11–32, 1991.

[35] D. Thomas and H. Allan. The visual concept detection task in imageclef08. *Evaluating Systems for Multilingual and*

Multimodal Information Access, 2008.

[36] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, and Y. Ma. Towards a practical face recognition system: Robust alignment and illumination by sparse representation. *IEEE Trans. PAMI.*, 34(2):372–386, 2012.

[37] C. Wang, S. Yan, L. Zhang, and H. Zhang. Multi-label sparse coding for automatic image annotation. *CVPR*, pages 1643–1650, 2009.

[38] M. Wang, X. S. Hua, R. Hong, J. Tang, G. J. Qi, and Y. Song. Unified video annotation via multi-graph learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(5):733–746, 2009.

[39] M. Wang, X. S. Hua, J. Tang, and R. Hong. Beyond distance measurement: Constructing neighborhood similarity for video annotation. *IEEE Transactions on Multimedia*, 11(3):465–476, 2009.

[40] J. Wright, A. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. PAMI.*, 31(2):210–227, 2009.

[41] Y. Yan, L. Chen, and W.-C. Tjhi. Semi-supervised fuzzy co-clustering algorithm for document categorization. *Knowledge and Information Systems*, DOI: 10.1007/s10115-011-0454-9, 2011.

[42] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. *CVPR*, pages 1794–1801, 2009.

[43] X. Yuan and S. Yan. Visual classification with multi-task joint sparse representation. *CVPR*, pages 3493–3500, 2010.

[44] S. Zhang, J. Huang, Y. Huang, Y. Yu, H. Li, and D. N. Metaxas. Automatic image annotation using group sparsity. *CVPR*, pages 3312–3319, 2010.

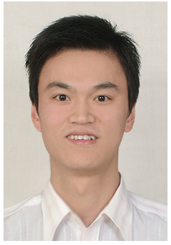
[45] Y. F. Zhao, Y. Zhao, and Z. Zhu. Co-training for search-based automatic image annotation. *Journal of Digital Information Management*, 6(2):214–218, 2008.

[46] Y. F. Zhao, Y. Zhao, and Z. Zhu. Tsvm-hmm: Transductive svm based hidden markov model for automatic image annotation. *Expert Systems with Applications*, 36(6):9813–9818, 2009.

[47] Z. Zhao and H. Glotin. Diversifying image retrieval by affinity propagation clustering on visual manifolds. *IEEE Multimedia*, 16(4):34–43, 2009.

[48] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, pages 321–328. MIT Press, 2004.

[49] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. *ICML*, pages 912–919, 2003.



Zhong-Qiu Zhao is an associate professor at Hefei University of Technology, China. He obtained the Master's degree in Pattern Recognition & Intelligent System at Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei, China, in 2004, and the PhD degree in Pattern Recognition & Intelligent System at University of Science and Technology, China, in 2007. From April 2008 to November 2009, he held a postdoctoral position in image processing in CNRS UMR6168 Lab Sciences de l'Information et des Systmes, France. Now he works in Laboratory of Data Mining and Intelligent Computing, Hefei University of Technology, China. His research is about pattern recognition, image processing, and computer vision.

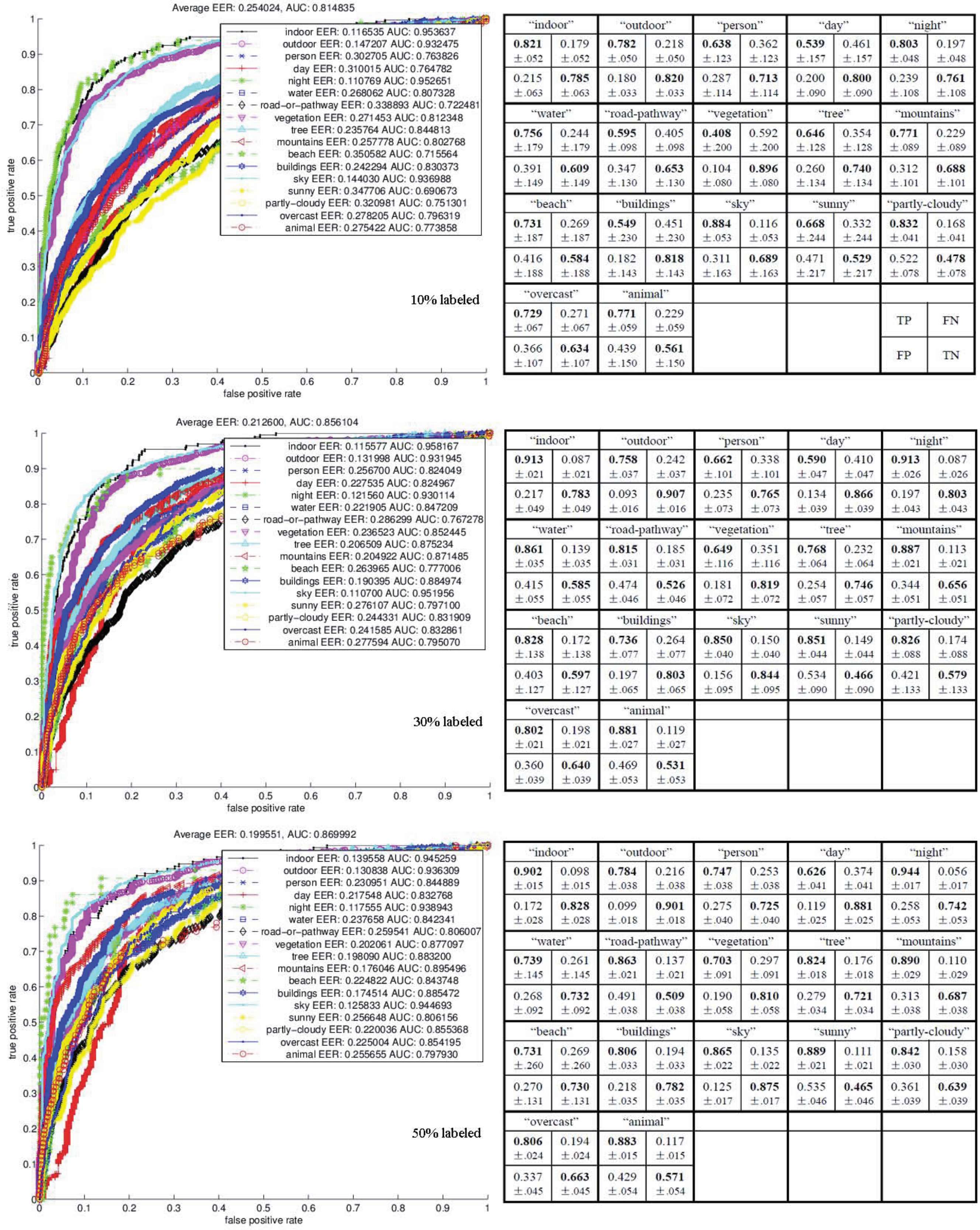
Herve Glotin is a professor at Universite du Sud Toulon Var, leader of the Information Dynamics and Integration team (Dyni), in the Systems and Information Sciences CNRS Lab. (LSIS). He has been awarded for outstanding research, and is a member, of the Institut Universitaire de France (IUF). He received his master in Artificial Intelligence from UPMC-Paris, his PhD on automatic audiovisual speech recognition from IDIAP-EPFL-CH and Inst. Polytech. Grenoble-FR. He has been invited in 2000 in IBM Via-Voice - J.Hopkins expert group. After two year as research ing. in the Syntax and Semantics CNRS lab, he has been associate professor from 2003 to 2010, and got his habilitation in 2007 on multimodal information retrieval. He organizes since eight years the summer school in Multimodal Information Retrieval (ERMITES). His research interests include signal processing, scene Understanding, cognitive Systems and machine learning.

Zhao Xie is an associate professor at Hefei University of Technology, China. He obtained the B.Sc in Computer Science at Hefei University of Technology, China, in 2002, and the PhD degree in computer vision in the Department of Computer Science at Hefei University of Technology, China, in 2007. His research interests include image understanding and pattern recognition.

Jun Gao is a professor at Hefei University of Technology, China. He obtained Bachelor's degree in Electronic Engineering at HeFei University of Technology in 1985, Master's degree in Signal & Information Processing at HeFei University of Technology in 1991, Doctor's degree in Information and Communication Engineering at University of Science and Technology, China in 1999. From March 1995 to October 1996, he was invited to work in University of Stuttgart, Germany. Now he works in Laboratory of Image Information Processing, Hefei University of Technology, China. His work is about image processing and intelligent information processing.

Xindong Wu is a Yangtze River Scholar in the School of Computer Science and Information Engineering at the Hefei University of Technology (China), a Professor of Computer Science at the University of Vermont (USA), and a Fellow of the IEEE. He received his Bachelor's and Master's degrees in Computer Science from the Hefei University of Technology, China, and his Ph.D. degree in Artificial Intelligence from the University of Edinburgh, Britain. His research interests include data mining, knowledge-based systems, and Web information exploration.

Dr. Wu is the Steering Committee Chair of the IEEE International Conference on Data Mining (ICDM), the Editor-in-Chief of Knowledge and Information Systems (KAIS, by Springer), and a Series Editor of the Springer Book Series on Advanced Information and Knowledge Processing (AI&KP). He was the Editor-in-Chief of the IEEE Transactions on Knowledge and Data Engineering (TKDE, by the IEEE Computer Society) between 2005 and 2008. He served as Program Committee Chair/Co-Chair for ICDM '03 (the 2003 IEEE International Conference on Data Mining), KDD-07 (the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining), and CIKM 2010 (the 19th ACM Conference on Information and Knowledge Management).

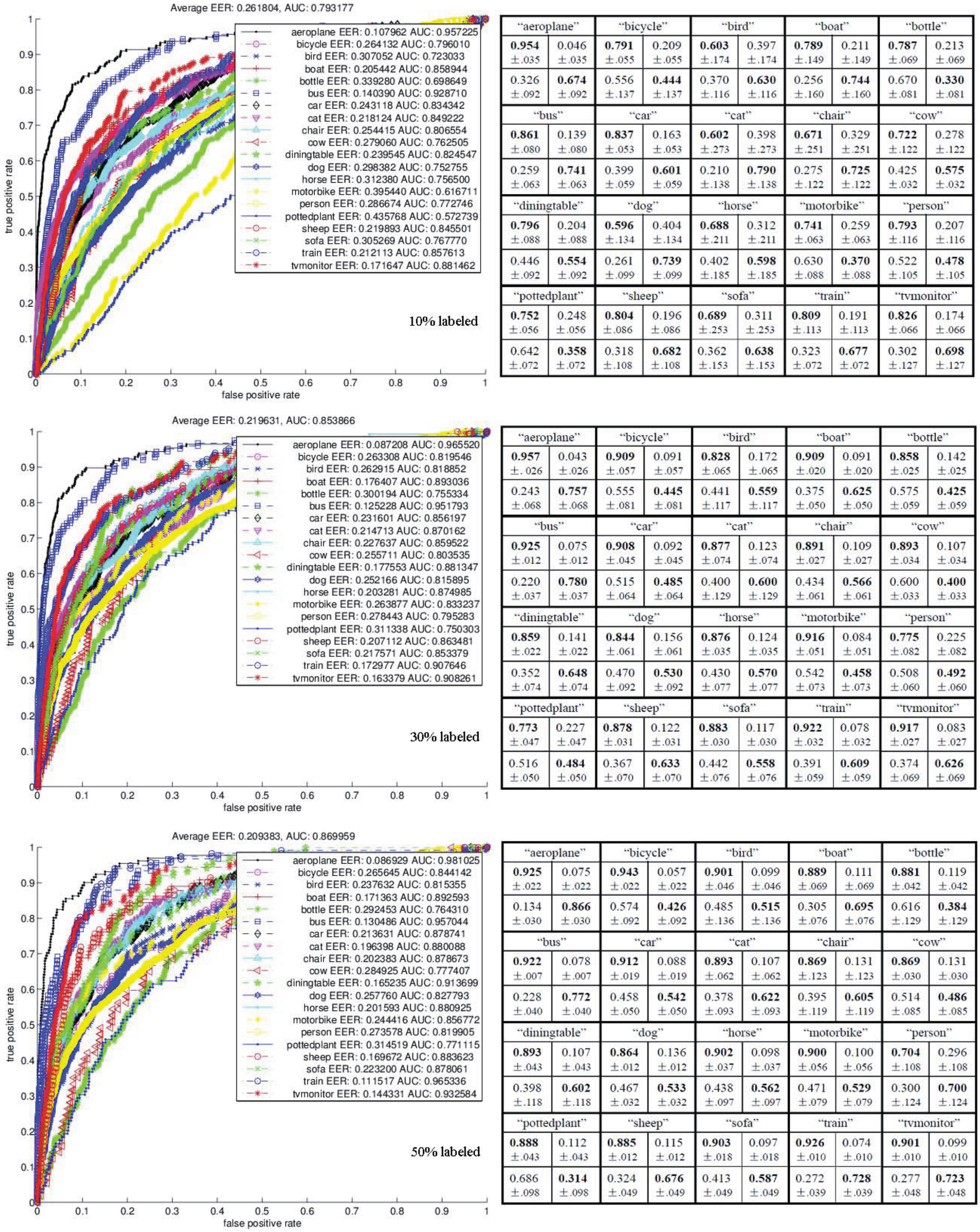


“indoor”		“outdoor”		“person”		“day”		“night”	
0.821 ±.052	0.179 ±.052	**0.782** ±.050	0.218 ±.050	**0.638** ±.123	0.362 ±.123	**0.539** ±.157	0.461 ±.157	**0.803** ±.048	0.197 ±.048
0.215 ±.063	**0.785** ±.063	0.180 ±.033	**0.820** ±.033	0.287 ±.114	**0.713** ±.114	0.200 ±.090	**0.800** ±.090	0.239 ±.108	**0.761** ±.108
“water”		“road-pathway”		“vegetation”		“tree”		“mountains”	
0.756 ±.179	0.244 ±.179	**0.595** ±.098	0.405 ±.098	**0.408** ±.200	0.592 ±.200	**0.646** ±.128	0.354 ±.128	**0.771** ±.089	0.229 ±.089
0.391 ±.149	**0.609** ±.149	0.347 ±.130	**0.653** ±.130	0.104 ±.080	**0.896** ±.080	0.260 ±.134	**0.740** ±.134	0.312 ±.101	**0.688** ±.101
“beach”		“buildings”		“sky”		“sunny”		“partly-cloudy”	
0.731 ±.187	0.269 ±.187	**0.549** ±.230	0.451 ±.230	**0.884** ±.053	0.116 ±.053	**0.668** ±.244	0.332 ±.244	**0.832** ±.041	0.168 ±.041
0.416 ±.188	**0.584** ±.188	0.182 ±.143	**0.818** ±.143	0.311 ±.163	**0.689** ±.163	0.471 ±.217	**0.529** ±.217	0.522 ±.078	**0.478** ±.078
“overcast”		“animal”							
0.729 ±.067	0.271 ±.067	**0.771** ±.059	0.229 ±.059					TP	FN
0.366 ±.107	**0.634** ±.107	0.439 ±.150	**0.561** ±.150					FP	TN

“indoor”		“outdoor”		“person”		“day”		“night”	
0.913 ±.021	0.087 ±.021	**0.758** ±.037	0.242 ±.037	**0.662** ±.101	0.338 ±.101	**0.590** ±.047	0.410 ±.047	**0.913** ±.026	0.087 ±.026
0.217 ±.049	**0.783** ±.049	0.093 ±.016	**0.907** ±.016	0.235 ±.073	**0.765** ±.073	0.134 ±.039	**0.866** ±.039	0.197 ±.043	**0.803** ±.043
“water”		“road-pathway”		“vegetation”		“tree”		“mountains”	
0.861 ±.035	0.139 ±.035	**0.815** ±.031	0.185 ±.031	**0.649** ±.116	0.351 ±.116	**0.768** ±.064	0.232 ±.064	**0.887** ±.021	0.113 ±.021
0.415 ±.055	**0.585** ±.055	0.474 ±.046	**0.526** ±.046	0.181 ±.072	**0.819** ±.072	0.254 ±.057	**0.746** ±.057	0.344 ±.051	**0.656** ±.051
“beach”		“buildings”		“sky”		“sunny”		“partly-cloudy”	
0.828 ±.138	0.172 ±.138	**0.736** ±.077	0.264 ±.077	**0.850** ±.040	0.150 ±.040	**0.851** ±.044	0.149 ±.044	**0.826** ±.088	0.174 ±.088
0.403 ±.127	**0.597** ±.127	0.197 ±.065	**0.803** ±.065	0.156 ±.095	**0.844** ±.095	0.534 ±.090	**0.466** ±.090	0.421 ±.133	**0.579** ±.133
“overcast”		“animal”							
0.802 ±.021	0.198 ±.021	**0.881** ±.027	0.119 ±.027						
0.360 ±.039	**0.640** ±.039	0.469 ±.053	**0.531** ±.053						

“indoor”		“outdoor”		“person”		“day”		“night”	
0.902 ±.015	0.098 ±.015	**0.784** ±.038	0.216 ±.038	**0.747** ±.038	0.253 ±.038	**0.626** ±.041	0.374 ±.041	**0.944** ±.017	0.056 ±.017
0.172 ±.028	**0.828** ±.028	0.099 ±.018	**0.901** ±.018	0.275 ±.040	**0.725** ±.040	0.119 ±.025	**0.881** ±.025	0.258 ±.053	**0.742** ±.053
“water”		“road-pathway”		“vegetation”		“tree”		“mountains”	
0.739 ±.145	0.261 ±.145	**0.863** ±.021	0.137 ±.021	**0.703** ±.091	0.297 ±.091	**0.824** ±.018	0.176 ±.018	**0.890** ±.029	0.110 ±.029
0.268 ±.092	**0.732** ±.092	0.491 ±.038	**0.509** ±.038	0.190 ±.058	**0.810** ±.058	0.279 ±.034	**0.721** ±.034	0.313 ±.038	**0.687** ±.038
“beach”		“buildings”		“sky”		“sunny”		“partly-cloudy”	
0.731 ±.260	0.269 ±.260	**0.806** ±.033	0.194 ±.033	**0.865** ±.022	0.135 ±.022	**0.889** ±.021	0.111 ±.021	**0.842** ±.030	0.158 ±.030
0.270 ±.131	**0.730** ±.131	0.218 ±.035	**0.782** ±.035	0.125 ±.017	**0.875** ±.017	0.535 ±.046	**0.465** ±.046	0.361 ±.039	**0.639** ±.039
“overcast”		“animal”							
0.806 ±.024	0.194 ±.024	**0.883** ±.015	0.117 ±.015						
0.337 ±.045	**0.663** ±.045	0.429 ±.054	**0.571** ±.054						

Fig. 4. The ROC curves (left column, an example of 20 runs) and confusion matrices (right column, mean values ± standard deviations on 20 runs) of the Co-KSR at the 100th iteration of the co-training on the ImageCLEF-VCDT 2008 dataset (10%, 30%, 50% labeled); The confusion matrices are all shown by class with a 2 × 2 size, where the two elements of the 1st row are TP (true positive) and FN (false negative) rates, while the two elements of the 2nd row are FP (false positive) and TN (true

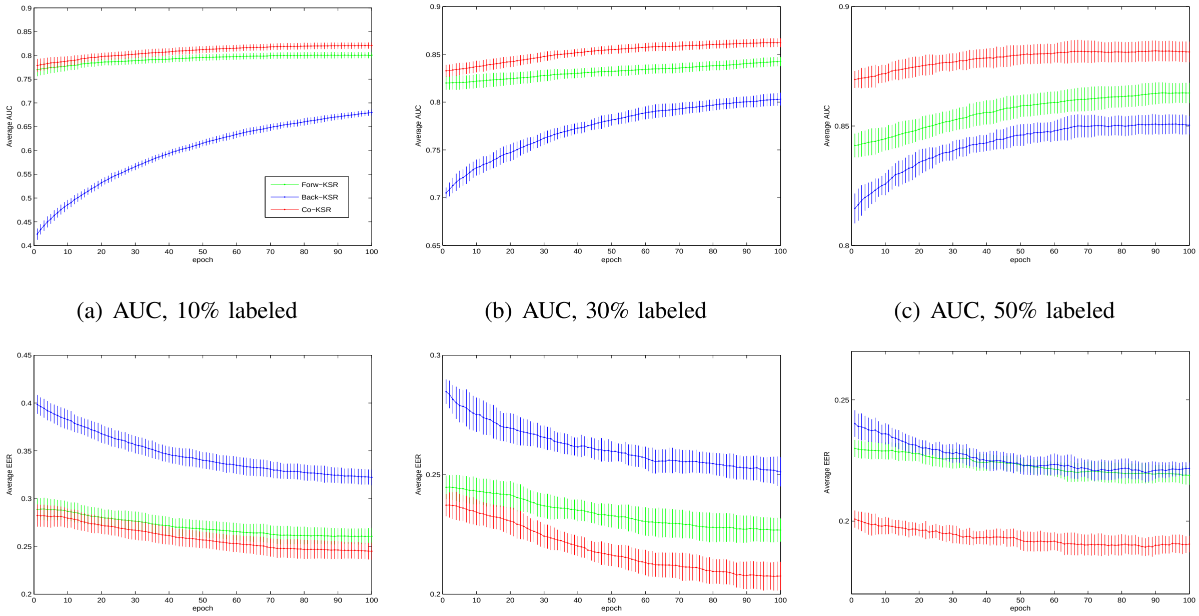


“aeroplane”		“bicycle”		“bird”		“boat”		“bottle”	
0.954 ±.035	0.046 ±.035	**0.791** ±.055	0.209 ±.055	**0.603** ±.174	0.397 ±.174	**0.789** ±.149	0.211 ±.149	**0.787** ±.069	0.213 ±.069
0.326 ±.092	**0.674** ±.092	0.556 ±.137	**0.444** ±.137	0.370 ±.116	**0.630** ±.116	0.256 ±.160	**0.744** ±.160	0.670 ±.081	**0.330** ±.081
“bus”		“car”		“cat”		“chair”		“cow”	
0.861 ±.080	0.139 ±.080	**0.837** ±.053	0.163 ±.053	**0.602** ±.273	0.398 ±.273	**0.671** ±.251	0.329 ±.251	**0.722** ±.122	0.278 ±.122
0.259 ±.063	**0.741** ±.063	0.399 ±.059	**0.601** ±.059	0.210 ±.138	**0.790** ±.138	0.275 ±.122	**0.725** ±.122	0.425 ±.032	**0.575** ±.032
“diningtable”		“dog”		“horse”		“motorbike”		“person”	
0.796 ±.088	0.204 ±.088	**0.596** ±.134	0.404 ±.134	**0.688** ±.211	0.312 ±.211	**0.741** ±.063	0.259 ±.063	**0.793** ±.116	0.207 ±.116
0.446 ±.092	**0.554** ±.092	0.261 ±.099	**0.739** ±.099	0.402 ±.185	**0.598** ±.185	0.630 ±.088	**0.370** ±.088	0.522 ±.105	**0.478** ±.105
“pottedplant”		“sheep”		“sofa”		“train”		“tvmonitor”	
0.752 ±.056	0.248 ±.056	**0.804** ±.086	0.196 ±.086	**0.689** ±.253	0.311 ±.253	**0.809** ±.113	0.191 ±.113	**0.826** ±.066	0.174 ±.066
0.642 ±.072	**0.358** ±.072	0.318 ±.108	**0.682** ±.108	0.362 ±.153	**0.638** ±.153	0.323 ±.072	**0.677** ±.072	0.302 ±.127	**0.698** ±.127

“aeroplane”		“bicycle”		“bird”		“boat”		“bottle”	
0.957 ±. 026	0.043 ±.026	**0.909** ±.057	0.091 ±.057	**0.828** ±.065	0.172 ±.065	**0.909** ±.020	0.091 ±.020	**0.858** ±.025	0.142 ±.025
0.243 ±.068	**0.757** ±.068	0.555 ±.081	**0.445** ±.081	0.441 ±.117	**0.559** ±.117	0.375 ±.050	**0.625** ±.050	0.575 ±.059	**0.425** ±.059
“bus”		“car”		“cat”		“chair”		“cow”	
0.925 ±.012	0.075 ±.012	**0.908** ±.045	0.092 ±.045	**0.877** ±.074	0.123 ±.074	**0.891** ±.027	0.109 ±.027	**0.893** ±.034	0.107 ±.034
0.220 ±.037	**0.780** ±.037	0.515 ±.064	**0.485** ±.064	0.400 ±.129	**0.600** ±.129	0.434 ±.061	**0.566** ±.061	0.600 ±.033	**0.400** ±.033
“diningtable”		“dog”		“horse”		“motorbike”		“person”	
0.859 ±.022	0.141 ±.022	**0.844** ±.061	0.156 ±.061	**0.876** ±.035	0.124 ±.035	**0.916** ±.051	0.084 ±.051	**0.775** ±.082	0.225 ±.082
0.352 ±.074	**0.648** ±.074	0.470 ±.092	**0.530** ±.092	0.430 ±.077	**0.570** ±.077	0.542 ±.073	**0.458** ±.073	0.508 ±.060	**0.492** ±.060
“pottedplant”		“sheep”		“sofa”		“train”		“tvmonitor”	
0.773 ±.047	0.227 ±.047	**0.878** ±.031	0.122 ±.031	**0.883** ±.030	0.117 ±.030	**0.922** ±.032	0.078 ±.032	**0.917** ±.027	0.083 ±.027
0.516 ±.050	**0.484** ±.050	0.367 ±.070	**0.633** ±.070	0.442 ±.076	**0.558** ±.076	0.391 ±.059	**0.609** ±.059	0.374 ±.069	**0.626** ±.069

“aeroplane”		“bicycle”		“bird”		“boat”		“bottle”	
0.925 ±.022	0.075 ±.022	**0.943** ±.022	0.057 ±.022	**0.901** ±.046	0.099 ±.046	**0.889** ±.069	0.111 ±.069	**0.881** ±.042	0.119 ±.042
0.134 ±.030	**0.866** ±.030	0.574 ±.092	**0.426** ±.092	0.485 ±.136	**0.515** ±.136	0.305 ±.076	**0.695** ±.076	0.616 ±.129	**0.384** ±.129
“bus”		“car”		“cat”		“chair”		“cow”	
0.922 ±.007	0.078 ±.007	**0.912** ±.019	0.088 ±.019	**0.893** ±.062	0.107 ±.062	**0.869** ±.123	0.131 ±.123	**0.869** ±.030	0.131 ±.030
0.228 ±.040	**0.772** ±.040	0.458 ±.050	**0.542** ±.050	0.378 ±.093	**0.622** ±.093	0.395 ±.119	**0.605** ±.119	0.514 ±.085	**0.486** ±.085
“diningtable”		“dog”		“horse”		“motorbike”		“person”	
0.893 ±.043	0.107 ±.043	**0.864** ±.012	0.136 ±.012	**0.902** ±.037	0.098 ±.037	**0.900** ±.056	0.100 ±.056	**0.704** ±.108	0.296 ±.108
0.398 ±.118	**0.602** ±.118	0.467 ±.032	**0.533** ±.032	0.438 ±.097	**0.562** ±.097	0.471 ±.079	**0.529** ±.079	0.300 ±.124	**0.700** ±.124
“pottedplant”		“sheep”		“sofa”		“train”		“tvmonitor”	
0.888 ±.043	0.112 ±.043	**0.885** ±.012	0.115 ±.012	**0.903** ±.018	0.097 ±.018	**0.926** ±.010	0.074 ±.010	**0.901** ±.010	0.099 ±.010
0.686 ±.098	**0.314** ±.098	0.324 ±.049	**0.676** ±.049	0.413 ±.049	**0.587** ±.049	0.272 ±.039	**0.728** ±.039	0.277 ±.048	**0.723** ±.048

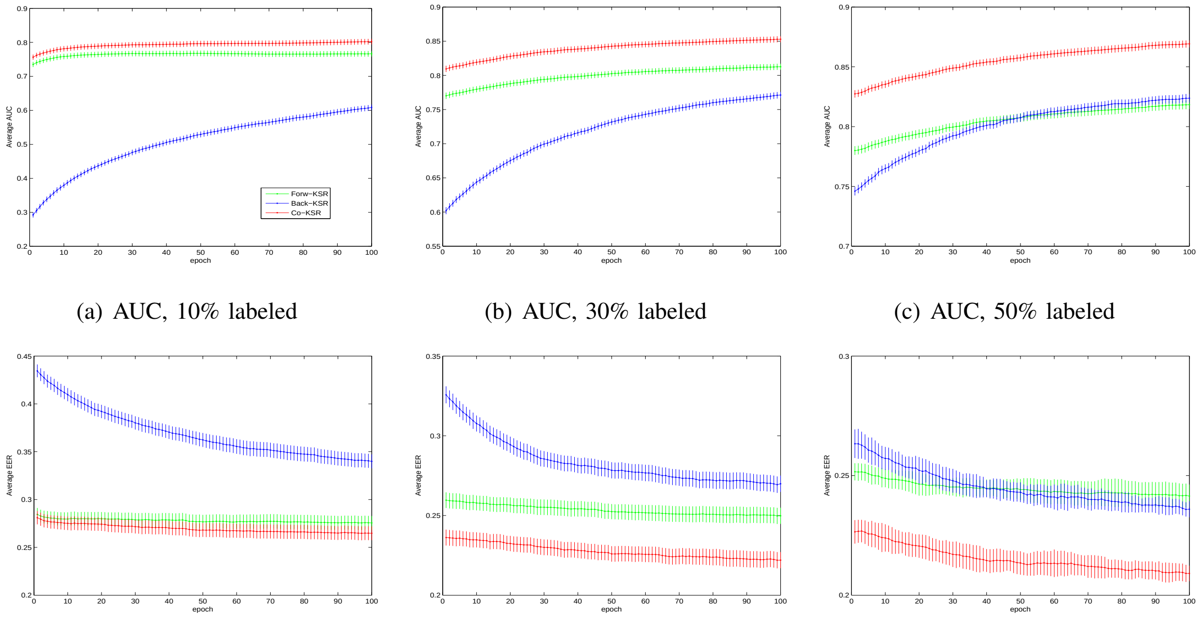
Fig. 5. The ROC curves (left column, an example of 20 runs) and confusion matrices (right column, mean values ± standard deviations on 20 runs) of the Co-KSR at the 100th iteration of the co-training on the PASCAL-VOC 2010 dataset (10%, 30%, 50% labeled); The confusion matrices are all shown by class with a 2 × 2 size, where the two elements of the 1st row are TP (true positive) and FN (false negative) rates, while the two elements of the 2nd row are FP (false positive) and TN (true



(a) AUC, 10% labeled (b) AUC, 30% labeled (c) AUC, 50% labeled

(d) EER, 10% labeled (e) EER, 30% labeled (f) EER, 50% labeled

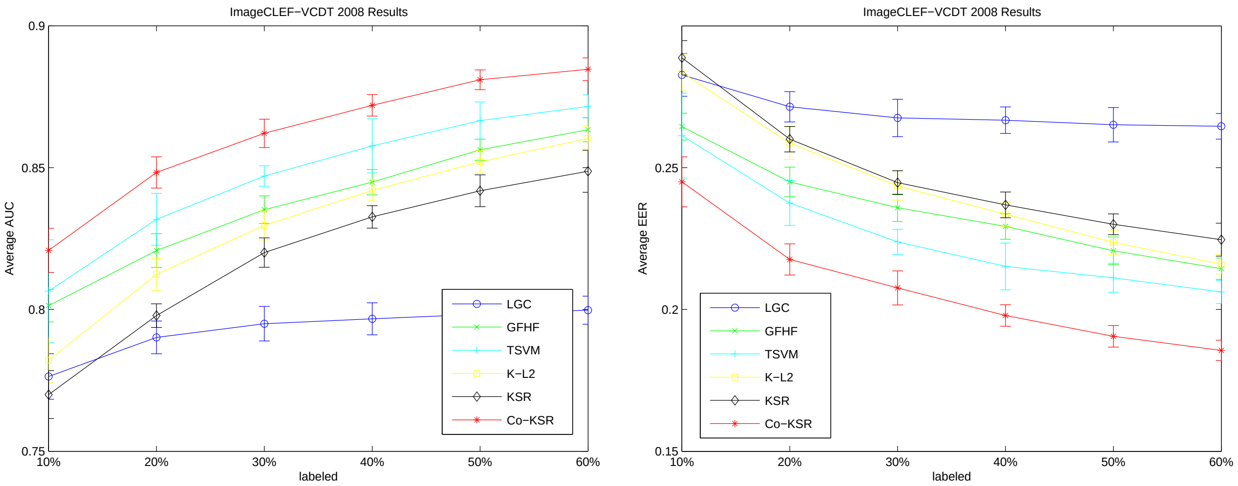
Fig. 6. The curve of the average AUC and EER (mean values with standard deviations on 20 runs) changing by the number of iterations of the KSR co-training on the ImageCLEF-VCDT 2008 dataset (10%, 30%, 50% labeled).



(a) AUC, 10% labeled (b) AUC, 30% labeled (c) AUC, 50% labeled

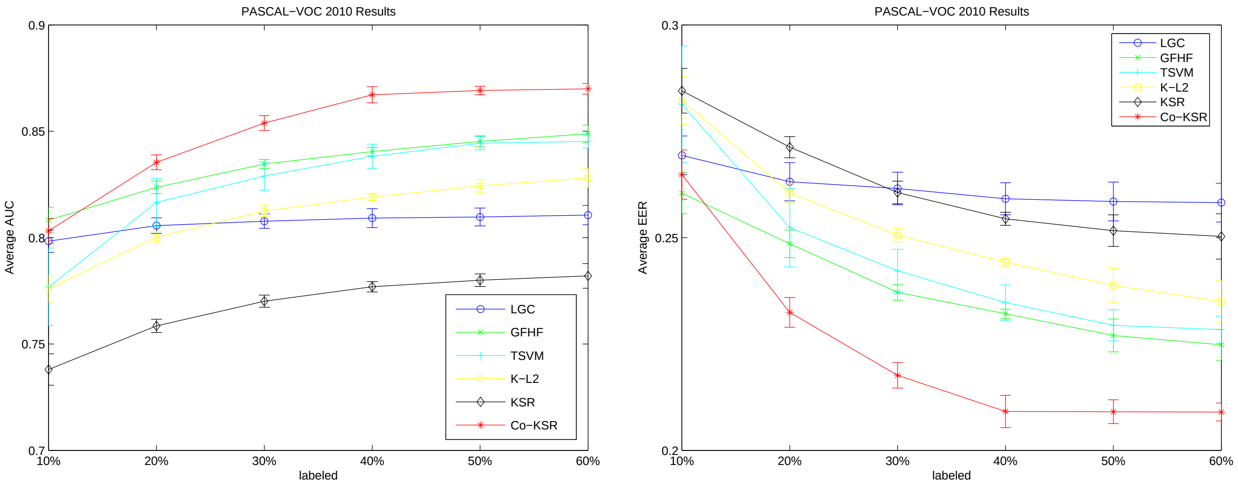
(d) EER, 10% labeled (e) EER, 30% labeled (f) EER, 50% labeled

Fig. 7. The curve of the average AUC and EER (mean values with standard deviations on 20 runs) changing by the number of iterations of the KSR co-training on the PASCAL-VOC 2010 dataset (10%, 30%, 50% labeled).



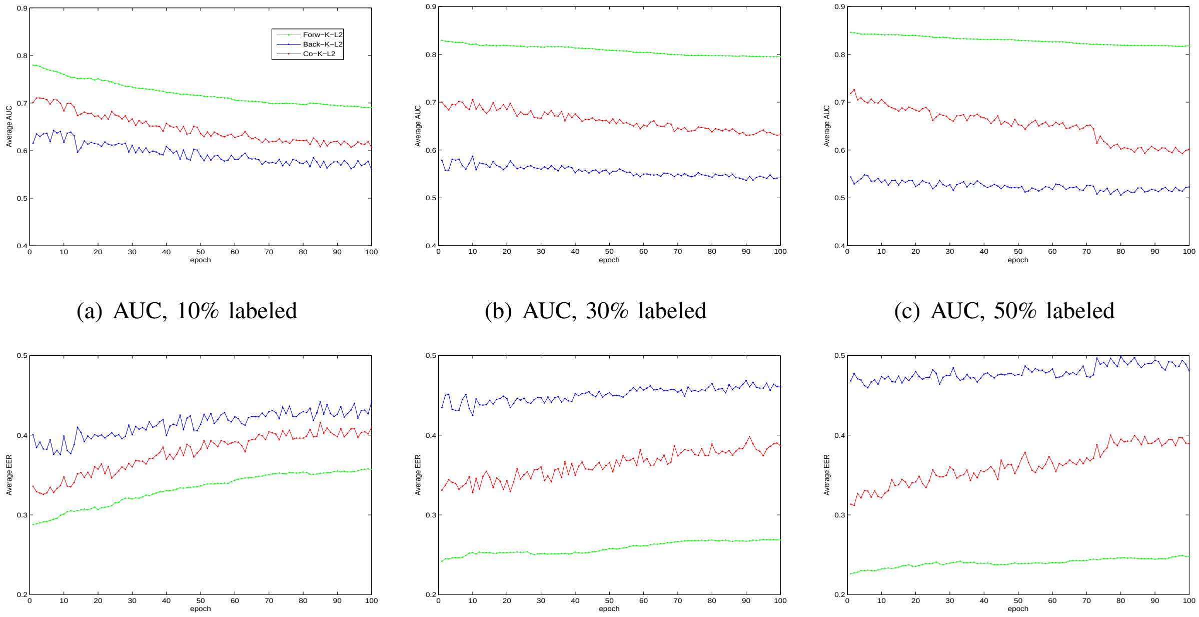
(a) AUC (b) EER

Fig. 8. The curve of the average AUC and EER (mean values with standard deviations on 20 runs) of various methods vs. the percentage of labeled images on the ImageCLEF-VCDT 2008 dataset, where the LGC and GFHF are both with the affinity weights defined by the Gaussian function, and the TSVM, K-L2, KSR and Co-KSR are all with the RBF kernel.



(a) AUC (b) EER

Fig. 9. The curve of the average AUC and EER (mean values with standard deviations on 20 runs) of various methods vs. the percentage of labeled images on the PASCAL-VOC 2010 dataset, where the LGC and GFHF are both with the affinity weights defined by the Gaussian function, and the TSVM, K-L2, KSR and Co-KSR are all with the RBF kernel.



(a) AUC, 10% labeled

(b) AUC, 30% labeled

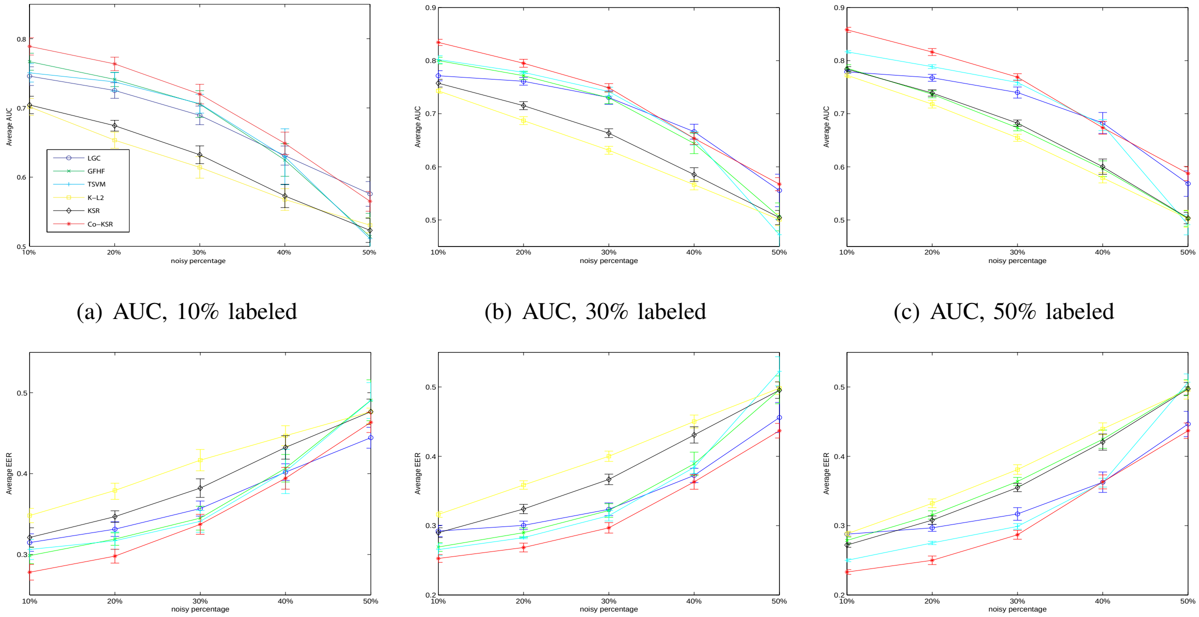
(c) AUC, 50% labeled

(d) EER, 10% labeled

(e) EER, 30% labeled

(f) EER, 50% labeled

Fig. 10. The curve of the average AUC and EER changing by the number of iterations of the K-L2 co-training on the ImageCLEF-VCDT 2008 dataset.



(a) AUC, 10% labeled

(b) AUC, 30% labeled

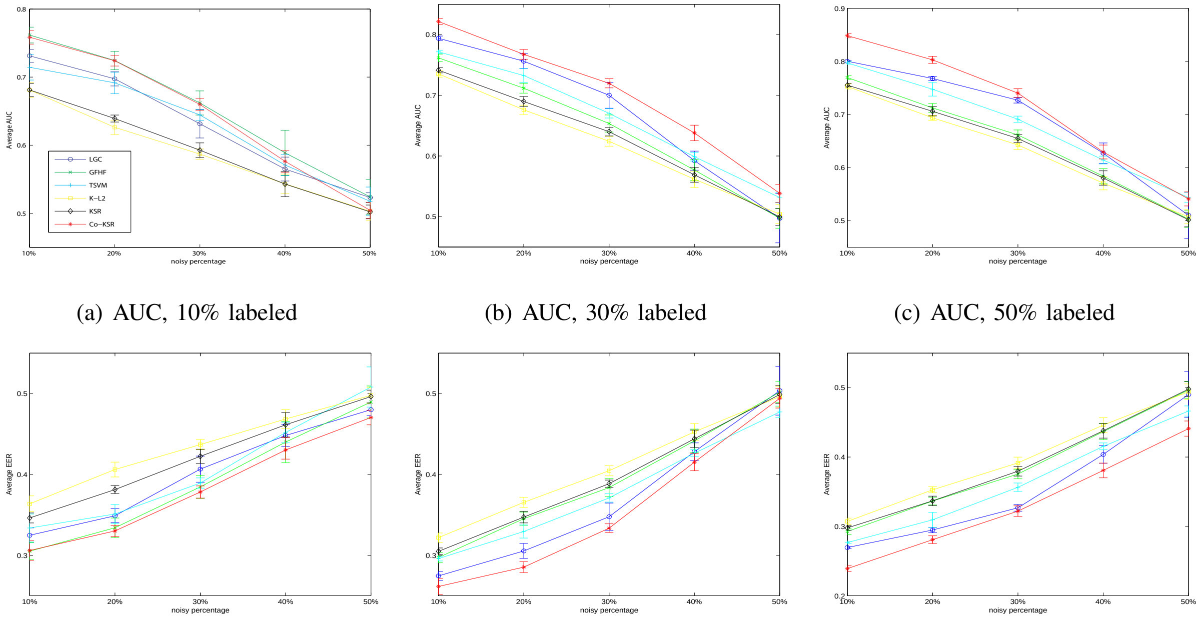
(c) AUC, 50% labeled

(d) EER, 10% labeled

(e) EER, 30% labeled

(f) EER, 50% labeled

Fig. 11. The curve of the average AUC and EER (mean values with standard deviations on 20 runs) of various methods vs. the percentage of noisy labels among those of labeled images of the ImageCLEF-VCDT 2008 dataset.



(a) AUC, 10% labeled

(b) AUC, 30% labeled

(c) AUC, 50% labeled

(d) EER, 10% labeled

(e) EER, 30% labeled

(f) EER, 50% labeled

Fig. 12. The curve of the average AUC and EER (mean values with standard deviations on 20 runs) of various methods vs. the percentage of noisy labels among those of labeled images of the PASCAL-VOC 2010 dataset.