Semi-Supervised Active Learning with Cross-Class Sample Transfer*

Yuchen Guo, Guiguang Ding, Yue Gao, Jianmin Wang
Tsinghua National Laboratory for Information Science and Technology (TNList)
School of Software, Tsinghua University, Beijing 100084, China
{yuchen.w.guo,kevin.gaoy}@gmail.com,{dinggg,jimwang}@tsinghua.edu.cn

Abstract

To save the labeling efforts for training a classification model, we can simultaneously adopt Active Learning (AL) to select the most informative samples for human labeling, and Semi-supervised Learning (SSL) to construct effective classifiers using a few labeled samples and a large number of unlabeled samples. Recently, using Transfer Learning (TL) to enhance AL and SSL, i.e., T-SS-AL, has gained considerable attention. However, existing T-SS-AL methods mostly focus on the situation where the source domain and the target domain share the same classes. In this paper, we consider a more practical and challenging setting where the source domain and the target domain have different but related classes. We propose a novel cross-class sample transfer based T-SS-AL method, called CC-SS-AL, to exploit the information from the source domain. Our key idea is to select samples from the source domain which are very similar to the target domain classes and assign pseudo labels to them for classifier training. Extensive experiments on three datasets verify the efficacy of the proposed method.

1 Introduction

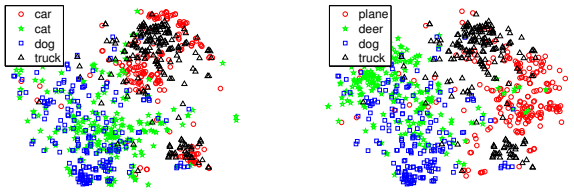
Generally, training effective classifiers require adequate labeled samples. However, manual label annotation is expensive and time consuming. To save labeling efforts, two learning strategies have been widely adopted. The first is Active Learning (AL) [Settles, 2009] which selects the most informative samples for labeling from a large pool of unlabeled samples. With the elaborate selection, even a few samples can provide sufficient information for supervised learning. The second is Semi-supervised Learning (SSL) [Zhu, 2005] which trains classifiers using both labeled and unlabeled samples. Using the information from a large number of unlabeled samples, SSL can achieve promising performance given a small number of labeled samples. Based on the power of AL and SSL, recent studies have demonstrated that the combination of these two strategies, i.e., Semi-supervised Active Learning (SS-AL), leads to better performance than either of them [Leng *et al.*, 2013; Zhang *et al.*, 2014; Wang *et al.*, 2016], and SS-AL has been applied to many applications, like document analysis [Bouguelia *et al.*, 2013], image classification [Tang *et al.*, 2012] and retrieval [Feng *et al.*, 2012], and sentiment classification [Zhou *et al.*, 2013].

Besides the target task and the corresponding samples, some auxiliary data sources are always available. For example, when our goal is to train an object recognizer/detector for YouTube videos, we can collect some images from Flickr which are similar to the target object and able to help train the model. In fact, by transferring the knowledge from auxiliary data sources (the source domain) into the target task (the target domain), the performance of the model can be further improved. This learning strategy is termed as Transfer Learning (TL) [Pan and Yang, 2010]. Motivated by the success of transfer learning, some attempts [Li *et al.*, 2012; 2013; Chattopadhyay *et al.*, 2013] have been made to simultaneously build classifiers in the target domain by SS-AL, and transfer knowledge from other source domains, called Transfer Semi-supervised Active Learning (T-SS-AL). T-SS-AL can significantly reduce the number of labeled samples in the target domain and achieve satisfactory performance simultaneously. However, existing T-SS-AL methods mostly require the classes in the source domain and the target domain to be identical. It is noted that this requirement is too demanding in many cases. For example, when we want to train image classifiers for some uncommon classes, like "lophius litulon" and "euchoreutes naso", it is very difficult, if not impossible, to collect auxiliary samples exactly belonging to the same classes. On the other hand, collecting auxiliary images from some other classes, like "dolphin" and "rabbit" is relatively easier. Under such circumstances, transferring knowledge across different classes can further enhance the power of T-SS-AL because the class limitation to the source domain is relaxed.

1.1 Motivation and Contribution

In this paper, we investigate T-SS-AL under the cross-class setting where the classes in the source domain and the target domain are different but related. To transfer knowledge across classes, we propose a sample transfer method based

*Corresponding author: Guiguang Ding. This research was supported by the National Natural Science Foundation of China (Grant No.61271394 and 61571269), and the Royal Society Newton Mobility Grant (IE150997).



(a) (b)

Figure 1: Observation and motivation.

on the sample-class similarity. We observe that some samples from other auxiliary classes can contribute to the property description of a target class. To demonstrate this, we select some images from CIFAR10 [Krizhevsky, 2009] and visualize them by tSNE [Van der Maaten and Hinton, 2008] in Figure 1. Here, we suppose we are constructing a dog truck classifer. In Figure 1(a), we can observe that if some labeled images from cat and car are available, they can well capture the characteristics of dog and truck. In fact, if we directly assign pseudo labels "dog" to all cat images, and "truck" to all car images and then train a linear SVM classifier with them, the classification accuracy is 92.8% for dog and truck. In Figure 1(b), we use labeled samples from "deer" and "plane" for evaluation. Although the distributions in the source domain and the target domain vary a lot, we can still observe that many (not all) samples in the source domain are very similar to "dog" and "truck". If we select these similar samples (which will be introduced detailedly later) and assign pseudo labels to them, the obtained classifier produces 92.4% accuracy. The result indicates that we do not need the samples exactly belonging to the target classes, and samples which are similar enough to the target classes (e.g., a cat image to the class "dog") can yield effective target domain classifiers.

To take the advantage of cross-class description as shown above,, we propose a novel method based on cross-class **sample** transfer, called CC-SS-AL. The key idea is to "borrow" samples from source domain for each target class and assign the corresponding pseudo labels. Then, the selected samples are regarded as the labeled samples in target domain and the classification model is trained using both the transferred samples and the labeled target domain samples. To select good samples, we adopt feature semantic embedding [Socher *et al.*, 2013] which maps samples and class labels into a common semantic space where the similarity between samples and labels can be directly measured. Then, based on the sample-class similarity and the separability, some samples in the source domain are selected to assign pseudo labels. With the transferred samples, SS-AL is performed by training semi-supervised classifiers in the target domain and selecting samples for human labeling by the graph based uncertainty sampling. In summary, we make the following contributions.

- We extend T-SS-AL into a challenging cross-class setting where the classes in the source domain and the target domain are different. By transferring samples across classes, the labeling effort can be significantly reduced.
- A novel sample transfer based method, CC-SS-AL, is proposed. Samples in the source domain are selected and assigned pseudo labels from the target domain. The sample selection procedure is based on the sample-class similarity and separability such that the selected samples can well capture the characteristics of the target classes.
- We carry out comprehensive empirical analysis on three benchmark datasets. The results show that the proposed CC-SS-AL requires much fewer labeled samples in the target domain than the conventional SS-AL methods to achieve the same accuracy, which validates its effecacy.

2 Background

Active Learning assumes that the learning system is allowed and able to select samples from a large unlabeled pool for human labeling. In fact, the information in samples is different and thus even a few labeled samples can provide sufficient information if they are the most informative ones. Generally, the informativeness is measured by representativeness [Yu *et al.*, 2006], i.e., the samples that best fit the data distribution are selected, or uncertainty [Yang *et al.*, 2014], i.e., the samples that the current system is most uncertain about are selected. The latter, i.e., uncertainty sampling, has attracted much attention recently and we also make use of this strategy.

As the large unlabeled pool is available during the whole active learning procedure, we can utilize not only the labeled samples, but also a large number unlabeled samples for classifier training, which is a semi-supervised schema. Some representative Semi-supervised learning algorithms are transductive SVM [Joachims, 1999] and Laplacian SVM [Belkin *et al.*, 2006]. Recent studies further investigate the combination of SSL and AL, leading to SS-AL. Leng *et al.* [2013] made use of the complementarity between SSL and AL and proposed a SS-AL method which queried the most uncertain samples and trained SVM with the labeled samples and the unlabeled class central samples. Tang *et al.* [2012] proposed to use the sparse-graph-based SSL method in AL. Wang *et al.* [2016] proposed to combine manifold regularization and AL. They showed that SS-AL can yield better results than either SSL or AL using the same number of labeled samples.

In many real-world applications, auxiliary data sources which have abundant label information are available and we can utilize them to improve the learning in the target domain. By simultaneously transferring knowledge to the target domain and selecting the most informative samples for human labeling, several T-SS-AL methods have be proposed recently. Shi *et al.* [2008] proposed to use the transferred knowledge as often as possible and the human labeling was triggered only when necessary. Li *et al.* [2012] proposed to find a shared common space for different domains such that the knowledge can be effectively transferred. Chattopadhyay *et al.* [2013] proposed to simultaneously reweight the source domain samples and select target domain samples to minimize the distribution difference between the two domains. Li *et al.* [2013] proposed to construct two classifiers for the source domain and the target domain respectively and the final classification was performed based on both classifiers. It

is noted that existing T-SS-AL methods mostly require the richly labeled source domain to have the same classes as the target domain and thus they cannot deal with the cross-class setting.

3 The Proposed Method

3.1 Problem and Notation

In this part, we define the problem and important notations. In the target domain, we have k_t classes $\mathcal{C}^t = \{c_1^t, ..., c_{k_t}^t\}$ and a large pool of data $\mathcal{D}^p = \{(\mathbf{x}_1^p, \mathbf{y}_1^p), ..., (\mathbf{x}_{n_p}^p, \mathbf{y}_{n_p}^p)\}$, where $\mathbf{x}_i^p \in \mathbb{R}^m$ is the feature vector for sample i. $\mathbf{y}_i^p \in \{0,1\}^{k_t}$ is the label vector where $y_{ij} = 1$ if sample i belongs to class j or $y_{ij} = 0$ otherwise. This pool consists of two disjoint sets, i.e., the labeled set $\mathcal{L}$ where the label vector is known and the unlabeled set $\mathcal{U}$ where the label vector is unknown. In AL, we progressively select samples from $\mathcal{U}$ and manually label them, i.e., add them to $\mathcal{L}$. The goal of AL is to achieve satisfactory classification accuracy and keep $\mathcal{L}$ as small as possible. Finally, the obtained model is tested on an i.i.d. test set in the target domain $\mathcal{D}^t = \{(\mathbf{x}_1^t, \mathbf{y}_1^t), ..., (\mathbf{x}_{n_t}^t, \mathbf{y}_{n_t}^t)\}$. As a T-SS-AL problem, we are given a set of labeled data in the source domain $\mathcal{D}^s = \{(\mathbf{x}_1^s, \mathbf{y}_1^s), ..., (\mathbf{x}_{n_s}^s, \mathbf{y}_{n_s}^s)\}$ and they belong to classes $\mathcal{C}^s = \{c_1^s, ..., c_{k_s}^s\}$. Existing T-SS-AL methods require $\mathcal{C}^s = \mathcal{C}^t$, while in this paper, we consider a more challenging and practical setting where $\mathcal{C}^s \cap \mathcal{C}^t = \emptyset$. In addition, to transfer knowledge across classes, for each class $c \in \mathcal{C}^s \cup \mathcal{C}^t$, we have a label semantic vector $\mathbf{a}_c \in \mathbb{R}^r$ for it.

3.2 Bridging Samples and Classes

As mentioned above, the key idea is to select samples from source domain which are very similar to the target classes. Feature Semantic Embedding (FSE) [Socher *et al.*, 2013] is an effective method to build the similarity measure between samples and classes. In many cases, the class labels are semantic meaningful, like "dog" and "truck". Based on some Natural Language Processing techniques, like [Huang *et al.*, 2012], we can build vectorial representations in the semantic space for class labels which reflects the semantic relationship between them, i.e., $\mathbf{a}_c$. Then, the feature vectors of the samples can be projected into the semantic space. Because the projected samples and class labels are in the same space, we can directly measure their similarity/distance, like the Euclidean distance. The projection is learned using the labeled samples by minimizing the distance between the embedded feature and the corresponding label semantic vector as below

$$\min_{\mathcal{P}} \sum\nolimits_i d(\mathcal{P}(\mathbf{x}_i), \mathbf{a}_{c(\mathbf{x}_i)}) \tag{1}$$

where $\mathcal{P}$ is the embedding function, $c(\mathbf{x}_i)$ denotes the class that $\mathbf{x}_i$ belongs to, and $d(\cdot,\cdot)$ is a distance measure which is Euclidean distance in this paper. As the source domain is richly labeled, we have abundant labeled training data to construct an effective embedding function by solving the above problem. In addition, as the semantic space is *shared* by the labels from all classes, including both $\mathcal{C}^s$ and $\mathcal{C}^t$, the projection learned using the source domain also works in the target domain, i.e., we can use the learned projection to assist the similarity measure between the source domain samples and the target domain classes. Furthermore, in the AL framework, we have some labeled data in the target domain. We incorporate them into the above problem. Previous works [Socher *et al.*, 2013; Guo *et al.*, 2016] have shown that simple linear function works well and thus we adopt the linear function and squared Euclidean distance, which leads to the solution below

$$\mathbf{P} = (\mathbf{X}'\mathbf{X} + \epsilon \mathbf{I}_m)^{-1}\mathbf{X}'\mathbf{A} \tag{2}$$

where $\mathbf{P} \in \mathbb{R}^{m \times r}$ is the linear embedding, $\mathbf{X} = [\mathbf{x}_1; ...; \mathbf{x}_n]$ is the feature matrix for all labeled samples from both source and target domains, $\mathbf{A} = [\mathbf{a}_{c(\mathbf{x}_1)}; ...; \mathbf{a}_{c(\mathbf{x}_n)}]$, $\mathbf{I}_m$ is a m-dimensional identity matrix, ϵ is a small positive value to avoid numeric problem, and $\mathbf{X}'$ denotes the transpose of $\mathbf{X}$. Given $\mathbf{P}$, the distance between any source domain sample and any target domain class is measured in the semantic space as

$$d_i^{c_j^t} = \|\mathbf{a}_{c_j^t} - \mathbf{x}_i\mathbf{P}\|^2 \tag{3}$$

a smaller distance indicates the sample $\mathbf{x}_i$ from the source domain is more similar to the target domain class c_j^t. Based on the FSE, we build the cross-class similarity between the source domain samples and the target domain classes, which acts as the building block for the cross-class sample transfer.

3.3 Cross-class Sample Transfer

As we illustrated in Figure 1, there are many samples in the source domain that can well describe the properties of the target domain classes. Therefore, the cross-class sample transfer aims to select samples from the source domain that can help distinguish one target domain class from the others. This goal indicates two criteria, 1) the selected samples should be similar to the target domain classes, and 2) the selected samples should be separatable from the other samples/classes. Based on these criteria, we propose the objective function as below

$$\begin{aligned}
&\min_{\mathbf{w}^c, s_j^c} \|\mathbf{w}^c\|^2 + C_1 \sum_{i=1}^{n_l} \xi_i + C_2 \sum_{j=1}^{n_s} s_j^c \xi_j^* + \beta \sum_{j=1}^{n_s} s_j^c d_j^c \\
&s.t.\ l_i^c(\mathbf{w}^c \mathbf{x}_i^{t'}) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, ..., n_l \\
&(2s_j^c - 1)(\mathbf{w}^c \mathbf{x}_j^{s'}) \geq 1 - \xi_j^*, \xi_j^* \geq 0, j = 1, ..., n_s \\
&s_j^c \in \{0,1\}, \sum\nolimits_j s_j^c \geq Q
\end{aligned} \tag{4}$$

where n_l denotes the size of the labeled set in the target domain, i.e., $\mathcal{L}$, l_i^c is the class-specific label vector for class $c \in \mathcal{C}^t$ where $l_i^c = 1$ if $y_{ic} = 1$ or $l_i^c = -1$ otherwise, and s_j^c is the indicator index for class c where $s_j^c = 1$ means that sample $\mathbf{x}_j^s$ is selected for class c, i.e., we transfer it to class c and assign c as its pseudo class, or $s_j^c = 0$ otherwise. In the above objective function, we perform class-wise sample transfer, i.e., we select samples for each target domain class independently. In fact, it is not expensive to collect "negative" samples, while labeling "positive" samples is costly, and thus our objective function mainly focuses on transferring positive samples for each target domain class from the source domain.

As mentioned above, both similarity and separability are considered simultaneously in Eq. (4). Specifically, minimizing the last term $\sum_{j=1}^{n_s} s_j^c d_j^c$ requires selecting samples with small distance (large similarity) to class c. Minimizing the

third term results in that the selected "positive" samples are separatable from the labeled negative samples in the target domain. Different from conventional max-margin formulation, $s_j^c\xi_j^c$ is employed to take the place of ξ_j^c as the loss. Besides transferring positive samples, another important reason is that most of the source domain samples are not useful such that the negative ones may dominate the objective function if we adopt the latter loss. In addition, we also incorporate the information form the labeled target domain samples into the objective function. This is to guarantee that the separation is consistent between the labeled samples and the transferred samples. Furthermore, the constraint $\sum_j s_j^c \geq Q$ is to prevent the optimization task to be a trivial solution which assigns 0 to all s_j^c, i.e., no sample is transferred. This constraint guarantees that there are at least Q samples transferred for c.

Optimization

The optimization task in Eq. (4) can be solved iteratively, just like in the transductive SVM [Joachims, 1999]. Specifically, the optimization algorithms consists of the following steps.

Fix s_j^c and update $\mathbf{w}^c$. With s_j^c fixed, Eq. (4) w.r.t. $\mathbf{w}^c$ is

$$\begin{aligned} &\min_{\mathbf{w}^c} \|\mathbf{w}^c\|^2 + C_1 \sum_{i=1}^{n_l} \xi_i + C_2 \sum_{j=1}^{n_s'} \xi_j^* \\ &s.t.\ l_i^c(\mathbf{w}^c \mathbf{x}_i^{t'}) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, ..., n_l \\ &\mathbf{w}^c \mathbf{x}_j^{s'} \geq 1 - \xi_j^*, \xi_j^* \geq 0, j = 1, ..., n_s' \end{aligned} \tag{5}$$

where n_s' is the number of samples in the source domain with $s_j^c = 1$ at the current iteration. This problem is a weighted SVM training problem [Yang *et al.*, 2007] which can be transformed into its dual problem, a constrained quadratic programming problem. It can be solved efficiently by ready-made QP software, like `quadprog`[1] function in MATLAB.

Fix $\mathbf{w}^c$ and update s_j^c. Given $\mathbf{w}^c$, we first update $\xi_j^* = 1 - \mathbf{w}^c \mathbf{x}_j^{s'}$, and then Eq. (4) w.r.t. s_j^c can be written as follows,

$$\min_{s_j^c} C_2 \sum_{j=1}^{n_s} s_j^c \xi_j^* + \beta \sum_{j=1}^{n_s} s_j^c d_j^c,\ s.t.\ \sum_{j=1}^{n_s} s_j^c \geq Q \tag{6}$$

Denote $\eta_j^c = C_2 \xi_j^* + \beta d_j^c$. Then, we can rank all η_j^c ascendingly and the solution to Eq. (6) is the top Q ranked samples.

It is straightforward to observe that the objective function value is non-increasing in both steps. Hence, we can iterate the above steps until convergence to obtain the final solution.

Initialization. In the above steps, we assume that one variable is provided when updating the other one. Now we address the initialization problem where no variable is provided. We can set $C_2 = 0$ and solve Eq. (5) first, i.e., we initialize $\mathbf{w}^c$ using only the labeled samples in the target domain. We can also set $C_2 = 0$ and solve Eq. (6) first, i.e., we initialize s_j^c by considering only the similarity. Empirically, the latter strategy leads to better performance and faster convergence.

[1] http://cn.mathworks.com/help/optim/ug/quadprog.html

3.4 Semi-supervised Active Learning

Graph-based Classifier Learning

In this paper, we consider the multi-class problem and the binary classification is just a special case. With the transferred samples, the number of (pseudo) labeled samples for each target domain class is significantly enlarged. Specifically, for each class $c \in \mathcal{C}^t$, we solve Eq. (4) and select some samples from source domain and assign label c. Finally, we obtain a pseudo labeled set $\tilde{\mathcal{L}}$ containing the transferred samples for each class. To perform multi-class classification, we train a one-vs-all classifier for each class [Hsu and Lin, 2002] which regards samples from one class as positive and the other as negative. Formally, for class q, we construct $\mathcal{L}_q^+ = \{\mathbf{x} | \mathbf{x} \in \mathcal{L} \wedge c(\mathbf{x}) = q, \text{or } \mathbf{x} \in \tilde{\mathcal{L}} \wedge \tilde{c}(\mathbf{x}) = q\}$ as the positive set, and $\mathcal{L}_q^- = \mathcal{L} \cup \tilde{\mathcal{L}} \backslash \mathcal{L}_q^+$ as the negative set where c and $\tilde{c}$ denote the label and the pseudo label of $\mathbf{x}$ respectively.

To make use of the unlabeled samples in $\mathcal{U}$, we train a semi-supervised classifier where a graph based classifier [Belkin *et al.*, 2006] is adopted here. Based on the manifold assumption, similar samples should have similar label. We first construct a k nearest neighbor graph on $\mathcal{L}_q \cup \mathcal{U}$ as

$$S_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in \mathcal{N}_k(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in \mathcal{N}_k(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Then, we construct a diagonal matrix $\mathbf{D}$ with diagonal element $D_{ii} = \sum_j S_{ij}$ and the graph laplacian $\mathbf{L} = \mathbf{D} - \mathbf{S}$. The graph-based SVM classifier (LapSVM) is trained as follows,

$$\min_{\mathbf{w}^q} \|\mathbf{w}^q\|^2 + C \sum_{\mathbf{x}_i \in L_q} (1 - l_i^q \mathbf{w}^q \mathbf{x}_i')_+ + C_g \mathbf{w}^q \mathbf{X}' \mathbf{L} \mathbf{X} \mathbf{w}^{q'} \tag{8}$$

where $l_i^q \in \{-1, 1\}$ is the label vector for class q as we introduced before. It can be efficiently solved by conjugate gradient schemas. Please refer to [Belkin *et al.*, 2006] for details.

For each class $c \in \mathcal{C}^t$, we can obtain the corresponding one-vs-all classifier parameters $\mathbf{w}^c$ by solving Eq. (8). Then, the multi-class classification for a new sample $\mathbf{x}$ is given by

$$c(\mathbf{x}) = \text{argmax}_c \mathbf{w}^c \mathbf{x}' \tag{9}$$

Graph-based Uncertainty Sampling

To select samples from $\mathcal{U}$ for human labeling, we adopt the uncertainty sampling strategy considering its effectiveness in active learning. In this paper, we follow the best-vs-second-best (BvSB) strategy [Joshi *et al.*, 2012] for uncertainty measurement. Specifically, for $\mathbf{x} \in \mathcal{U}$, suppose that $\mathbf{w}^c \mathbf{x}'$ produces the largest and the second largest responses on classes c_1 and c_2, we compute $p_1 = e^{\mathbf{w}^{c_1} \mathbf{x}'} / Z$ and $p_2 = e^{\mathbf{w}^{c_2} \mathbf{x}'} / Z$ where $Z = e^{\mathbf{w}^{c_1} \mathbf{x}'} + e^{\mathbf{w}^{c_2} \mathbf{x}'}$ is the normalization factor. The entropy is defined as $E(\mathbf{x}) = -\sum_{j=1}^{2} p_i \log p_i$. The larger entropy is, the more uncertain the sample is. Intuitively, we can compute the entropy for all unlabeled samples and select the ones with the largest entropy. However, this strategy 1) fails to consider the relation between the uncertainty between samples because if we label one sample, the uncertainties of its neighbors may also decrease significantly, and 2) leads to redundancy because similar samples have similar uncertainty.

Algorithm 1 CC-SS-AL

Input: Source domain data $\mathcal{D}^s$, target domain pool $\mathcal{D}^p$; Label semantic vector $\mathbf{a}_c$ for $\forall c \in \mathcal{C}^s \cup \mathcal{C}^t$;
Output: Classifiers $\mathbf{w}_c$ for target domain, $\forall c \in \mathcal{C}^t$;

1: Initialize $\mathcal{L}$ by random seed, $\mathcal{U} = \{1, ..., n_p\} \backslash \mathcal{L}$;
2: **for** $iter = 1 : max\_iter$ **do**
3: Construct feature semantic embedding $\mathbf{P}$ by Eq. (2);
4: Initialize pseudo labeled set $\tilde{\mathcal{L}} = \emptyset$;
5: **for** $c \in \mathcal{C}^t$ **do**
6: Compute sample-class similarity by Eq. (3);
7: Select samples $\mathcal{S}^c = \{j | s_j^c = 1\}$ for c by Eq. (4);
8: Assign pseudo label $\tilde{c}(\mathbf{x}^s_{j \in \mathcal{S}^c}) = c$, $\tilde{\mathcal{L}} = \tilde{\mathcal{L}} \cup \mathcal{S}^c$;
9: **end for**
10: Train LapSVM parameters $\mathbf{w}^c$ for $\forall c \in \mathcal{C}^t$ by Eq. (8);
11: Select top ranked samples $\mathcal{S}$ by Eq. (10) for labeling;
12: Update $\mathcal{L} = \mathcal{L} \cup \mathcal{S}$ and $\mathcal{U} = \mathcal{U} \backslash \mathcal{S}$;
13: **end for**
14: Return $\mathbf{w}^c$, $\forall c \in \mathcal{C}^t$;

Therefore, we propose a graph-based uncertainty sampling strategy, which minimizes the objective function as follows,

$$\min_{r_i} -\mathbf{r}\mathbf{S}\mathbf{E}' + \lambda \mathbf{r}\mathbf{S}\mathbf{r}', \ s.t. \ \mathbf{r}\mathbf{1}' = \rho > 0, \mathbf{r} \succeq 0 \qquad (10)$$

where r_i is the ranking score for $\mathbf{x}_i \in \mathcal{U}$ and $\mathbf{S}$ is the k-NN graph on $\mathcal{U}$ and $S_{ii} = 1$. This problem can be solved by QP software or the augmented Lagrange multipliers algorithm [Bertsekas, 1999]. By solving Eq. (10), we obtain the ranking scores for unlabeled samples and we select the top ranked ones for human labeling. In the first term, the ranking score of $\mathbf{x}_i$ considers not only its own uncertainty, also its neighbors'. The second term removes the redundancy. If $\mathbf{x}_i$ and $\mathbf{x}_j$ are neighbors ($S_{ij} = 1$), and if r_i is large which indicates that it may be selected, the term $r_i S_{ij} r_j$ enforces to assign a small value to r_j for minimizing the whole function.

3.5 Summarize

We summarize the whole procedure of CC-SS-AL in Algorithm 1. Specifically, from line 2 to 9, we select samples from the source domain for cross-class sample transfer, which is the main difference between our work and existing T-SS-AL methods. In the line 10, the one-vs-all graph-based semi-supervised classifiers are trained using the labeled samples, transferred samples with pseudo labels, and unlabeled samples. In line 12 and 13, we perform graph-based uncertainty sampling to select informative samples for human annotation.

4 Experiment

4.1 Settings

To demonstrate the effectiveness of the proposed method, we conduct experiments on three benchmark datasets. The first is CIFAR10 [Krizhevsky, 2009], which consists of 10 classes like "plane" and "dog", and each class has $6,000$ images. In each source-target split, we use 8 classes as source domain and the other 2 classes as target domain. Thus we have $C_{10}^2 = 45$ different splits and the average result is reported.. The second dataset is Animals with Attributes (AwA) [Lampert *et al.*, 2014]. It has 50 different animal classes and $30,475$ images. This dataset provides a standard source-target split where 40 classes with $24,295$ images belong to source domain and 10 classes with $6,180$ images belong to target domain. The third dataset is aPascal-aYahoo (aPY) dataset [Farhadi *et al.*, 2009] containing two subsets. The first subset is aPascal from PASCAL VOC2008 challenge that has $12,695$ samples from 20 different categories like "people" and "dog". The second subset is aYahoo which is collected from Yahoo image search. aYahoo has 12 categories with $2,644$ images that are similar but different from the categories in aPascal, such as "centaur" and "wolf". In aPY, we follow the standard setting where aPascal works as the source domain and aYahoo is the target domain. To extract features for each image, we utilize the pre-trained Deep CNN tool Caffe [Donahue *et al.*, 2014] and we use the output of the fc7 layer which is a $4,096$-dimensional vector for each image. For each label, we use the 50-dimensional word vector provided by Huang *et al.* [2012] as the label semantic vector.

Because the existing T-SS-AL methods cannot address the cross-class problem, we compare our method to SS-AL methods. The first is LapSVM-R [Belkin *et al.*, 2006], a widely used SSL method, which uses LapSVM and random sampling. The second is SVM-AL [Joshi *et al.*, 2012], a conventional AL method, which adopts SVM classifier and uncertainty sampling. The third is LapSVM-AL [Wang *et al.*, 2016]. a SS-AL method which adopts LapSVM as the semi-supervised classifier and active learning to select unlabeled samples. It is noted that the main difference between the proposed method and these existing works, which is also our main contribution, lies in that our work is able to transfer knowledge across different class while other methods cannot.

To evaluate the performance, we follow the metrics in [Joshi *et al.*, 2012]. Specifically, we split the data in target domain equally into two parts, and one part acts as the pool $\mathcal{D}^p$ where the methods select samples for human labeling, and the other part is the test set $\mathcal{D}^t$. Each method iteratively selects samples from $\mathcal{D}^p$ for labeling by the corresponding sampling strategy, e.g., uncertainty sampling, and the model is retrained on the $\mathcal{D}^p$ with the labeled samples $\mathcal{L}$ and the unlabeled samples $\mathcal{U}$. Then we evaluate the model on $\mathcal{D}^t$. Hence, we can draw a curve which reflects the classification accuracy on $\mathcal{D}^t$ of each method w.r.t. the number of iterations which is equivalent to the number of labeled samples in $\mathcal{D}^p$. In each iteration, 2, 10, and 12 samples are selected for labeling for CIFAR10, AwA, and aPY, respectively. For fair comparison, at the first iteration, all methods share the same random seed.

In addition, to remove the influence of initial seeds, we use 50 different random seeds and the average result is reported.

To determine the model parameters for each model, e.g., the parameter C for SVM, the cross-validation (CV) strategy is employed here. Specifically, for three baselines, we use the labeled source domain for CV. The parameter C for SVM and C_g for Laplacian regularization are chosen from $\{10^{-3}, 10^{-2}, ..., 10^2\}$. Following Guo *et al.*[2016], we use cross-class CV for our method. For CIFAR10 which has 8 classes in source domain, we use 2 classes to simulate the target domain and the other as the source domain. The other two

datasets are processed in similar way. In CV, C_1 and C_2 in Eq. (4), C and C_g in Eq. (8) are selected from $\{0.1, 1, 10\}$. In addition, we simply set β in Eq. (4) and λ in Eq. (10) to 1.

4.2 Results

First we compare the proposed method to baselines. In this experiment, we set Q, the number of transferred samples for each target domain class, to 100, 100, and 50 for CIFAR10, AwA, and aPY, respectively. The performance curves on three datasets are shown in Figure 2. We can observe that the proposed CC-SS-AL significantly outperforms the other baselines, which verifies its effectiveness. Specifically, at the 5-th iteration, our method has 86.72%, 90.11%, and 90.04% accuracy on three datasets. The improvements upon the best baseline, LapSVM-AL, are $\mathbf{7.17}\%$, $\mathbf{6.62}\%$, and $\mathbf{4.64}\%$ on three datasets, which indicates that our method achieves error reductions of $\mathbf{35.06}\%$, $\mathbf{40.10}\%$, and $\mathbf{31.78}\%$, respectively.

An interesting observation is that the proposed CC-SS-AL method can achieve much higher performance when only a few labeled samples are available. The performance gains are $\mathbf{17.43}\%$, $\mathbf{8.01}\%$, and $\mathbf{6.67}\%$ compared to the best baseline at the first iteration. This phenomenon indicates that the transferred and pseudo labeled samples can indeed capture the characteristics of target domain classes which validates 1) our motivation that the elaborately selected samples from other classes can well describe the target class and 2) our selection algorithm is indeed effective. In addition, CC-SS-AL requires much fewer labeled samples in the target domain than the other baselines to achieve the same performance. For example, in AwA and aPY, CC-SS-AL needs only 5 iterations (50 and 60 labeled samples respectively) to achieve 90% accuracy, while LapSVM-AL, needs 17 and 12 iterations (170 and 144 labeled samples respectively), which indicates CC-SS-AL saves $\mathbf{70.58}\%$ and $\mathbf{58.33}\%$ labeling efforts. Furthermore, CC-SS-AL only needs samples from related and different classes which are very easy to obtain from Web, such that CC-SS-AL is more practical than existing T-SS-AL methods.

In the second experiment, we investigate the influence of cross-class sample transfer. Q in Eq. (4) determines how many samples are transferred for each target domain class. We plot the performance curves of CC-SS-AL with different value of Q on three datasets in Figure 3. Even we set Q as a small value, such as 5, CC-SS-AL can also outperform baselines, which validates again that the transferred samples provide valuable information. When we increase Q (e.g., to 50 and 100), CC-SS-AL performs better because more knowledge is transferred. Interestingly, when we increase Q to a large value (e.g., 500 for AwA), the performance of CC-SS-AL degrades rapidly. For example, the accuracy at the 10-th iteration on AwA decreases from 92.42% to 69.57% when we increase Q from 100 to 500. In fact, the underlying assumption of CC-SS-AL is that there are some samples in the source domain that are very similar to the target domain classes. When Q is too large and the source domain is small, the selection algorithm is forced to choose dissimilar samples which introduce negative information for training classifiers.

5 Conclusion

In this paper, we extend the T-SS-AL into a new cross-class setting where the auxiliary source domain has different but related classes to the target domain. We propose a novel cross-class sample transfer based method, dubbed CC-SS-AL. It selects sample from the source domain which can well capture the characteristics of the target domain classes and assign pseudo labels to them. The information in target domain can be enhanced by incorporating such samples with pseudo labels. Then, a semi-supervised classifier is trained and a graph-based uncertainty sampling method is proposed to select samples for human labeling. Experiments on three datasets demonstrate that CC-SS-AL can achieve satisfactory performance with only a few labeled samples in the target domain, which is much superior to existing SS-AL methods.

References

[Belkin *et al.*, 2006] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7:2399–2434, 2006.

[Bertsekas, 1999] D. Bertsekas. *Nonlinear programming*. Belmont,MA: Athena Scientific, 1999.

[Bouguelia *et al.*, 2013] M. Bouguelia, Y. Belaïd, and A. Belaïd. A stream-based semi-supervised active learning approach for document classification. In *ICDAR*, 2013.

[Chattopadhyay *et al.*, 2013] Rita Chattopadhyay, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Joint transfer and batch-mode active learning. In *ICML*, pages 253–261, 2013.

[Donahue *et al.*, 2014] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 2014.

[Farhadi *et al.*, 2009] Ali Farhadi, Ian Endres, Derek Hoiem, and David A. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

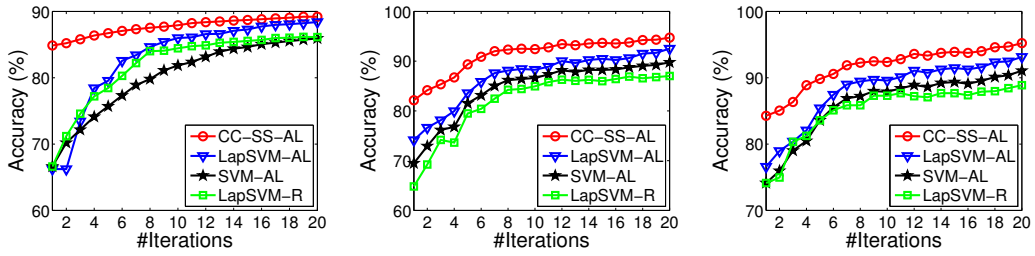
[Feng *et al.*, 2012] Y. Feng, J. Xiao, Z. Zha, H. Zhang, and Y. Yang. Active learning for social image retrieval using locally regressive optimal design. *Neurocomp.*, 2012.

[Guo *et al.*, 2016] Yuchen Guo, Guiguang Ding, Xiaoming Jin, and Jianmin Wang. Transductive zero-shot recognition via shared model space learning. In *AAAI*, 2016.

[Hsu and Lin, 2002] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE TNN*, 13(2):415–425, 2002.

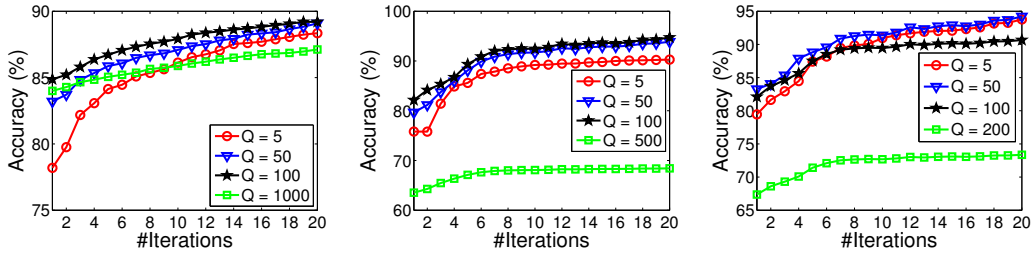
[Huang *et al.*, 2012] Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Improving word representations via global context and multiple word prototypes. In *ACL*, pages 873–882, 2012.

[Joachims, 1999] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *ICML*, volume 99, pages 200–209, 1999.



(a) CIFAR10 (b) AwA (c) aPY

Figure 2: Classification accuracy w.r.t. the number of iterations (labeled samples).



(a) CIFAR10 (b) AwA (c) aPY

Figure 3: The effect of the number of transferred samples (Q).

[Joshi *et al.*, 2012] Ajay J. Joshi, Fatih Porikli, and Nikolaos P. Papanikolopoulos. Scalable active learning for multiclass image classification. *TPAMI*, 2012.

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Tech Report. Univ. of Toronto*, 2009.

[Lampert *et al.*, 2014] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *TPAMI*, 2014.

[Leng *et al.*, 2013] Yan Leng, Xinyan Xu, and Guanghui Qi. Combining active learning and semi-supervised learning to construct svm classifier. *KBS*, 44:121–131, 2013.

[Li *et al.*, 2012] Lianghao Li, Xiaoming Jin, Sinno Jialin Pan, and Jian-Tao Sun. Multi-domain active learning for text classification. In *SIGKDD*, pages 1086–1094, 2012.

[Li *et al.*, 2013] Shoushan Li, Yunxia Xue, Zhongqing Wang, and Guodong Zhou. Active learning for cross-domain sentiment classification. In *IJCAI*, 2013.

[Pan and Yang, 2010] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *TKDE*, 2010.

[Settles, 2009] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[Shi *et al.*, 2008] X. Shi, W. Fan, and J. Ren. Actively transfer domain knowledge. In *ECMLPKDD*, 2008.

[Socher *et al.*, 2013] Richard Socher, Milind Ganjoo, Christopher D. Manning, and Andrew Y. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, 2013.

[Tang *et al.*, 2012] Jinhui Tang, Zheng-Jun Zha, Dacheng Tao, and Tat-Seng Chua. Semantic-gap-oriented active learning for multilabel image annotation. *TIP*, 2012.

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[Wang *et al.*, 2016] Xibin Wang, Junhao Wen, Shafiq Alam, Zhuo Jiang, and Yingbo Wu. Semi-supervised learning combining transductive support vector machine with active learning. *Neurocomputing*, 173:1288–1298, 2016.

[Yang *et al.*, 2007] X. Yang, Q. Song, and Y. Wang. A weighted support vector machine for data classification. *IJPRAI*, 2007.

[Yang *et al.*, 2014] Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G Hauptmann. Multi-class active learning by uncertainty sampling with diversity maximization. *IJCV*, 2014.

[Yu *et al.*, 2006] Kai Yu, Jinbo Bi, and Volker Tresp. Active learning via transductive experimental design. In *ICML*, pages 1081–1088, 2006.

[Zhang *et al.*, 2014] Yihao Zhang, Junhao Wen, Xibin Wang, and Zhuo Jiang. Semi-supervised learning combining co-training with active learning. *ESA*, 2014.

[Zhou *et al.*, 2013] Shusen Zhou, Qingcai Chen, and Xiaolong Wang. Active deep learning method for semi-supervised sentiment classification. *Neurocomp.*, 2013.

[Zhu, 2005] Xiaojin Zhu. Semi-supervised learning literature survey. 2005.