Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Active Learning with Cross-Class Similarity Transfer*

Yuchen Guo,† Guiguang Ding,† Yue Gao,† Jungong Han‡
†Tsinghua National Laboratory for Information Science and Technology (TNList)
School of Software, Tsinghua University, Beijing 100084, China
‡Northumbria University, Newcastle, NE1 8ST, UK
yuchen.w.guo@gmail.com, {dinggg,gaoyue}@tsinghua.edu.cn,jungong.han@northumbria.ac.uk

Abstract

How to save labeling efforts for training supervised classifiers is an important research topic in machine learning community. Active learning (AL) and transfer learning (TL) are two useful tools to achieve this goal, and their combination, i.e., transfer active learning (T-AL) has also attracted considerable research interest. However, existing T-AL approaches consider to transfer knowledge from a source/auxiliary domain which has the same class labels as the target domain, but ignore the relationship among classes. In this paper, we investigate a more practical setting where the classes in source domain are related/similar to but different from the target domain classes. Specifically, we propose a novel cross-class T-AL approach to simultaneously transfer knowledge from source domain and actively annotate the most informative samples in target domain so that we can train satisfactory classifiers with as few labeled samples as possible. In particular, based on the class-class similarity and sample-sample similarity, we adopt a similarity propagation to find the source domain samples that can well capture the characteristics of a target class and then transfer the similar samples as the (pseudo) labeled data for the target class. In turn, the labeled and transferred samples are used to train classifiers and actively select new samples for annotation. Extensive experiments on three datasets demonstrate that the proposed approach outperforms significantly the state-of-the-art related approaches.

Introduction

When training supervised classifiers, we always expect that there are sufficient labeled samples available for the target classes (Bishop and others 2006). However, this requirement seems too demanding in some real-world applications. For example, many objects "in the wild" follow a long-tailed distribution such that they do not occur frequently enough to collect and label a large set of representative exemplars to build the corresponding recognizers (Changpinyo et al. 2016). In addition, the labeling effort for many objects can be very expensive because the expert knowledge is required, like in the fine-grained bird recognition (Wah et al. 2011). Under these circumstances, it is always expected to train effective classifiers with as few labeled samples as possible.

Therefore, saving efforts for labeling data in a supervised learning is an important topic in the machine learning community. Two strategies are widely adopted. The first is active learning (AL) (Settles 2009) which selects the most informative samples for expert labeling. In fact, the information in each sample is different. Therefore, if the most representative/informative samples are selected and labeled, even a few labeled samples can provide sufficient knowledge to construct effective classifiers. The second is transfer learning (TL) (Pan and Yang 2010) which transfers knowledge from related auxiliary source domains to the target domain. By using the knowledge from auxiliary domains, we can save the labeling efforts paid to the target domain. On the top of them, transfer active learning (T-AL) (Chattopadhyay et al. 2013; Li et al. 2013), which attempts to simultaneously transfer knowledge from auxiliary domains and actively selects samples for expert labeling in the target domain, has gained increasing attention and achieved promising results recently.

The current T-AL approaches mostly focus on the **intra-class** transfer where the source domain and the target domain share the same class labels but have different marginal and conditional distributions. However, sometimes it is difficult to collect a fully labeled source domain that has exactly the same classes as the target domain, especially when the target domain contains uncommon or newly defined classes such as images of the Tesla's Model S. On the other hand, collecting labeled samples from some different but related common classes is much easier in many cases (Zhu et al. 2011). Therefore, if we can extend T-AL into the **inter-class** transfer setting where the knowledge is transferred across classes, T-AL can be applied to more practical situations.

Motivation and Contribution

The reason why existing T-AL approaches fail to transfer knowledge across classes is because they treat each class independently without taking the relationship between classes into account, thus giving rise to the fact that only the same class can build correspondence. In reality, the classes in many real-world applications, such as object recognition, are strongly or weakly related to each other. For example, class "dolphin" is strongly related to "shark" but weakly to

*This research was supported by the National Natural Science Foundation of China (Grant No. 61571269 and 61671267) and the Royal Society Newton Mobility Grant (IE150997). Corresponding author: Guiguang Ding.

"tiger", while class "lion" is strongly related to "tiger" but weakly to "shark". Based on the relationship, it is possible to transfer knowledge from the labeled samples of "dolphin" and "lion" to help construct classifier between "shark" and "tiger" (Lampert, Nickisch, and Harmeling 2014).

Motivated by this observation, in this paper we propose a novel cross-class T-AL approach which simultaneously transfers knowledge from source domain that includes different but similar/related classes to the target domain and actively selects informative samples in the target domain for expert labeling by using the relationship between classes. Specifically, both class-class similarity and sample-sample similarity enable us to adopt a similarity propagation strategy to build the similarity between the labeled sample in the source domain and each class in the target domain. Then, the source domain samples that are highly similar to a target class are "borrowed" by the class by assigning pseudo label to them and treating them as the labeled samples of the class. Although the borrowed samples are not exactly from the target class, they can well capture the characteristics of the target class and regarding them as labeled samples can help build more effective classifier (Lim, Salakhutdinov, and Torralba 2011; Choi et al. 2013). In this way, the knowledge is transferred across classes from the source domain to the target domain. Subsequently, we can build classifiers with the labeled target domain samples and the transferred and pseudo labeled source domain samples. With the classifiers, the most informative samples can be selected for labeling. In summary, we make the following contributions in this paper:

- We investigate the T-AL in the challenging cross-class setting and a novel approach is proposed which can simultaneously transfer knowledge from related source domain classes into target domain classes and actively select the most informative target domain samples for annotation.
- In view of the relationship between classes, we adopt a similarity propagation method to build the similarity between source domain samples and target domain classes. Then the source domain samples which are the most similar to the target classes are selected and assigned by pseudo labels. With the knowledge of the cross-class transferred samples, more effective classifiers can be build even with just a few labeled samples in the target domain.
- We conduct extensive empirical analysis on three benchmark datasets. The experimental results demonstrate that the proposed cross-class T-AL approach can achieve higher accuracy by using much fewer labeled samples in the target domain than the state-of-the-art related approaches.

Background

Different from passive learning where the labeled samples are given in advance, active learning allows the learning system to select unlabeled samples for expert labeling. The underlying assumption in active learning is that the samples have different information and only a small portion of samples can provide sufficient information for supervised learning. This idea has been also applied in many applications, such as hard negative mining (Shrivastava, Gupta, and Girshick 2016). There are two ways to measure the informativeness of samples. The first is based on the representativeness (Yu, Bi, and Tresp 2006) which considers how the selected samples can capture the distribution of data. The second is based on uncertainty of samples given the current model (Joshi, Porikli, and Papanikolopoulos 2012; Yang et al. 2015). Specifically, it considers how the current model is uncertain about the sample and selects the most uncertain samples, which are also the hardest samples for the current model, for expert labeling. If the model can well handle difficult samples, it is reasonable that it can handle easy samples as well. The recent researchers mostly focus on the second strategy because of its superior performance.

If there are auxiliary knowledge/data sources available that have abundant label information, we can consider to transfer knowledge from them to further reduce the labeling cost in the target domain, which is the focus of transfer active learning. Shi, Fan, and Ren (2008) proposed to transfer knowledge from auxiliary sources as often as possible and labeling target domain samples was triggered only when the likelihood that the unlabeled samples in target domain can be correctly classified became too low. Li et al. (2012) proposed to find a shared latent space for auxiliary source domain and target domain such that the label information in source domain could be well exploited. Then the most informative samples were selected by considering the information from the latent space. Chattopadhyay et al. (2013) proposed to re-weight the source domain samples and select the target domain samples to reduce the distribution difference between domains such that the knowledge could be transferred more effectively. Li et al. (2013) proposed a disjoint framework where individual classifiers for source domain and target domain were trained independently and the prediction was made from both classifiers. They adopted Query by Committee strategy to select the most informative samples. Intuitively, utilizing more information from auxiliary sources can save the labeling effort in target domain and result in better performance, which has been demonstrated empirically by previous works. However, they make a strong assumption that the source domain and target domain share the same class labels. But in real world, most of applications violate this assumption. Therefore, to enhance the generalization of T-AL, we further relax the share-class assumption in this paper and investigate T-AL in the cross-class setting.

The Proposed Approach

Problem Definition and Notations

In this paper, we consider the cross-class T-AL. Specifically, the problem is described as follows. In the target domain, there are a large sample pool $\mathcal{D}^p = \{\mathbf{x}_1^p, ..., \mathbf{x}_{n_p}^p\}$ where $\mathbf{x}_i^p \in \mathbb{R}^d$ is a d-dimensional feature vector, and k_t classes $\mathcal{C}^t = \{c_1^t, ..., c_{k_t}^t\}$. Each sample belongs to one class in $\mathcal{C}^t$. $\mathcal{D}^p$ consists of two disjoint set, the labeled set $\mathcal{L}$ and the unlabeled set $\mathcal{U}$. There is another test set $\mathcal{D}^t = \{\mathbf{x}_1^t, ..., \mathbf{x}_{n_t}^t\}$ which share the same distribution and class set with $\mathcal{D}^p$. We progressively select some samples from $\mathcal{U}$ for expert labeling (i.e., move them to $\mathcal{L}$), and then train a classifier with the labeled samples. The task of active learning

is to construct a classifier that yields satisfactory performance on $\mathcal{D}^t$ with as few labeled samples in $\mathcal{D}^p$ as possible (i.e., the size of $\mathcal{L}$ should be small). Furthermore, in the cross-class transfer setting, we are given another auxiliary source domain for knowledge transfer, which contains a set of labeled samples $\mathcal{D}^s = \{(\mathbf{x}_1^s, \mathbf{y}_1^s), ..., (\mathbf{x}_{n_s}^s, \mathbf{y}_{n_s}^s)\}$ and k_s classes $\mathcal{C}^s = \{c_1^s, ..., c_{k_s}^s\}$. In the cross-class setting, we have $\mathcal{C}^s \cup \mathcal{C}^t = \emptyset$ while previous T-AL approaches require $\mathcal{C}^s = \mathcal{C}^t$. Each source domain sample belongs to one class in $\mathcal{C}^s$ and $\mathbf{y}_i^s \in \{-1, 1\}^{k_s}$ is the label vector where $y_{ij}^s = 1$ if sample $\mathbf{x}_i^s$ belongs to c_j^s or -1 otherwise. In addition, to connect source domain and target domain, we are given a class similarity matrix $\mathbf{G} \in \mathbb{R}^{(k_s+k_t)\times(k_s+k_t)}$ where g_{ij} denotes the similarity between two classes. This matrix can be defined by experts or computed from auxiliary information. For example, in the object recognition task, we can use the word hierarchy (Fellbaum 1998) based on object's name or the word vector (Huang et al. 2012) to define the similarity.

Cross-class Similarity Transfer

In this paper, we propose a similarity based sample transfer method to perform cross-class knowledge transfer. Specifically, we select some samples that can well capture the characteristics of target domain classes and assign pseudo labels to them so as to expand the small labeled set in the target domain. In fact, although the classes in source domain and target domain are different, there may exist some samples in the source domain that are highly similar to the target classes. For example, in the shark-tiger task we mentioned above, although not all the dolphin (lion) images are similar to shark (tiger), it is reasonable to assume that there are a portion of dolphin (lion) images that can well describe the shark (tiger) class. In fact, if there are more classes and samples in the source domain, it is more likely that there exists similar samples. Augmenting $\mathcal{L}$ by adding those samples in would improve the classifiers, since more knowledge is exploited although they have pseudo labels (Choi et al. 2013).

To build the similarity between source domain samples and target domain classes, we first adopt similarity propagation on the class-class similarity graph. This is motivated by the graph-based random walk for information propagation (Lin, Ding, and Hu 2015). The relationship between classes is given by $\mathbf{G}$, but the relationship between a sample to all classes is unknown. Denote $\mathbf{r}_i \in [0, 1]^{k_s}$ as the relatedness scores (similarity) between source domain sample $\mathbf{x}_i^s$ and all source domain classes where r_{ic} is the initial score between $\mathbf{x}_i^s$ and class c. Because $\mathbf{x}_i^s$ is labeled, one simple way is to assign $r_{ic} = 1$ where c is the class of $\mathbf{x}_i^s$ and the other elements to 0, i.e., hard assignment. However, this strategy ignores 1) the relationship between classes (e.g., an image with label "horse" may also have label "grass"), and 2) the intra-class diversity (Guo et al. 2015) (e.g., dolphin images may vary a lot from each other). Therefore, we adopt a soft assignment strategy. Specifically, because the source domain samples are fully labeled, we can train k_s one-vs-all probability classifiers (e.g., Logistic regression classifier), in which each classifier f_c^s outputs the probability that the sample belongs to c. Then we use the outputs of these classifiers for $\mathbf{r}_i$. In this way, both problems above can be addressed.

Now we need to propagate the similarity to target domain classes. Our goal is to connect source domain samples and target domain classes. We formulate this goal as a propagation procedure where the sample is the source node, all target domain classes are the sink nodes, all source domain classes are the transient nodes, and the transient probability matrix for these nodes is derived from $\mathbf{r}_i$ and $\mathbf{G}$ as follows:

$$\mathbf{T} = \begin{pmatrix} 0_{1\times 1}^{x\to x} & (\tilde{\mathbf{r}}_i)_{1\times k_s}^{x\to s} & \mathbf{0}_{1\times k_t}^{x\to t} \\ \mathbf{0}_{k_s\times 1}^{s\to x} & \tilde{\mathbf{G}}_{k_s\times k_s}^{s\to s} & \tilde{\mathbf{G}}_{k_s\times k_t}^{s\to t} \\ \mathbf{0}_{k_t\times 1}^{t\to x} & \mathbf{0}_{k_t\times k_s}^{t\to s} & \mathbf{I}_{k_t\times k_t}^{t\to t} \end{pmatrix} \tag{1}$$

where the symbols x, s, and t denote sample (source node), source domain class (transient node), and target domain class (sink node) respectively, and $(\cdot)^{a\to b}$ is the transient matrix between node a and b. $\mathbf{G}^{a\to b}$ is a sub-matrix of $\mathbf{G}$ that contains the similarity between classes in a and b. We further perform ℓ_1-norm normalization to each row of $\mathbf{T}$ such that it satisfies the definition of transient probability matrix. Here, we require that the source node has no input, sink nodes have no output, and transient nodes have both input and output. Then, given an object at source node, it can randomly walk on the similarity graph and the probability that it walks from node i to j is given by T_{ij}. After enough steps, it will finally reach a sink node. Intuitively, the probability that the object reaches target domain class c can be regarded as the similarity between the sample and c. For example, if $\mathbf{x}_i$ is very similar to source domain class c_p^s (r_{ip} is large) and c_p^s is very similar to target domain class c_q^t (G_{pq} is large), it is very likely that the object can walk from $\mathbf{x}_i$ to c_p^s and then to c_q^t. Using similarity-based random walk to build the relationship between sample and class has been adopted in many applications, like image annotation (Guillaumin et al. 2009), as it is able to discover complicated relationships.

Now we need to compute the probability that the random walk stops at each target domain class c, i.e., the similarity between $\mathbf{x}_i$ and c. We first reorganize $\mathbf{T}$ into a simpler way:

$$\mathbf{T} = \begin{pmatrix} \mathbf{Q}_{(k_s+1)\times(k_s+1)}^{x,s\to x,s} & \mathbf{R}_{(k_s+1)\times k_t}^{x,s\to t} \\ \mathbf{0}_{k_t\times(k_s+1)}^{t\to x,s} & \mathbf{I}_{k_t\times k_t}^{t\to t} \end{pmatrix} \tag{2}$$

where

$$\mathbf{Q} = \begin{pmatrix} 0_{1\times 1}^{x\to x} & (\tilde{\mathbf{r}}_i)_{1\times k_s}^{x\to s} \\ \mathbf{0}_{k_s\times 1}^{s\to x} & \tilde{\mathbf{G}}_{k_s\times k_s}^{s\to s} \end{pmatrix}, \mathbf{R} = \begin{pmatrix} \mathbf{0}_{1\times k_t}^{x\to t} \\ \tilde{\mathbf{G}}_{k_s\times k_t}^{s\to t} \end{pmatrix} \tag{3}$$

This is a transient probability matrix for the standard absorbing Markov chain (Grinstead and Snell 1997). Based on its theory, if the random walk starts at node $q \in \{x, s\}$, the probability that it stops at sink node c is computed as below:

$$p_{qc} = \mathbf{M}_{q*}\mathbf{R}_{*c} \tag{4}$$

where $\mathbf{M}_{q*}$ is the q-th row and $\mathbf{R}_{*c}$ is the c-th column. The matrix $\mathbf{M}$ is defined as $\mathbf{M} = (\mathbf{I} - \mathbf{Q})^{-1}$. Specifically, it can be computed by the block matrix invention formula as below

$$\mathbf{M} = \begin{pmatrix} 1 & \tilde{\mathbf{r}}_i(\mathbf{I} - \tilde{\mathbf{G}}_{k_s\times k_s}^{s\to s})^{-1} \\ 0 & (\mathbf{I} - \tilde{\mathbf{G}}_{k_s\times k_s}^{s\to s})^{-1} \end{pmatrix} \tag{5}$$

In our propagation procedure, the random walk starts at the sample. Therefore, we only care about the first row in $\mathbf{M}$

based on our definition. By using Eq. (5), the probability that the random walk stops at each target domain class is

$$\mathbf{p}_i^c = \tilde{\mathbf{r}}_i(\mathbf{I} - \tilde{\mathbf{G}}_{k_s \times k_s}^{s \to s})^{-1}\tilde{\mathbf{G}}_{k_s \times k_t}^{s \to t} \tag{6}$$

where p_{ij}^c is the probability for class c_j^t. Because $\tilde{\mathbf{G}}_{k_s \times k_s}^{s \to s}$ and $\tilde{\mathbf{G}}_{k_s \times k_t}^{s \to t}$ is given in advance, the term $(\mathbf{I} - \tilde{\mathbf{G}}_{k_s \times k_s}^{s \to s})^{-1}\tilde{\mathbf{G}}_{k_s \times k_t}^{s \to t}$ can be pre-computed. For each sample, we just need to compute its $\mathbf{r}_i$ and then apply a simple matrix multiplication operation to obtain $\mathbf{p}_i^c$, which is quite efficient. This score can be regarded as the similarity between the source domain sample $\mathbf{x}_i$ and each target domain class.

The class-class similarity graph focuses on the general characteristics of class. On the other hand, in active learning, some labeled samples of target domain classes are available. Although its number is quite small, these samples can also provide some specific information about target domain classes. Therefore, we also consider the similarity propagation on the sample-sample similarity graph. The sample based propagation is analogous to the class based one, and we just need some proper modifications. Specifically, we regard the source domain sample $\mathbf{x}_i$ as source node, target domain classes as sink nodes, as in the class based propagation, and we regard the labeled samples in $\mathcal{L}$ and some source domain samples obtained by random sampling as the transient nodes. The transient probability between samples (including the source node and transient nodes) is defined by the heat-kernel similarity (Lu, Yuan, and Yan 2014) $h_{ij} = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma^2)$ where σ is set to the mean Euclidean distance between feature vectors in the training set. The transient probability between transient nodes and sink nodes (target domain classes) is defined as follows. If the sample is from source domain, then its probabilities to all sink nodes are 0. On the other hand, if the sample is from $\mathcal{L}$, because its label is available, its probability to the corresponding class is 1 and the others are 0. Therefore, the transient probability matrix for all nodes is written as follows,

$$\mathbf{T} = \begin{pmatrix} 0_{1\times 1}^{x\to x} & \tilde{\mathbf{H}}_{1\times n_s'}^{x\to s} & \tilde{\mathbf{H}}_{1\times n_l}^{x\to t} & \mathbf{0}_{1\times k_t}^{x\to c} \\ \mathbf{0}_{n_s'\times 1}^{s\to x} & \tilde{\mathbf{H}}_{n_s'\times n_s'}^{s\to s} & \tilde{\mathbf{H}}_{n_s'\times n_l}^{s\to t} & \mathbf{0}_{n_s'\times k_t}^{s\to c} \\ \mathbf{0}_{n_l\times 1}^{t\to x} & \tilde{\mathbf{H}}_{n_l\times n_s'}^{t\to s} & \tilde{\mathbf{H}}_{n_l\times n_l}^{t\to t} & (\tilde{\mathbf{Y}}_{\mathcal{L}}^{t})_{n_l\times k_t}^{t\to c} \\ \mathbf{0}_{k_t\times 1}^{c\to x} & \mathbf{0}_{k_t\times n_s'}^{c\to s} & \mathbf{0}_{k_t\times n_l}^{c\to t} & \mathbf{I}_{k_t\times k_t}^{c\to x} \end{pmatrix} \tag{7}$$

where the symbols x, s, t, and c denote the sample $\mathbf{x}_i$ to be considered (source node), the picked samples from source domain (transient node) whose number is $n_s' \ll n_s$, labeled samples in the target domain (transient node) whose number is n_l, and the target domain classes (sink node). Also, we normalize each row such that the summation is 1. Then we perform random walk for similarity propagation on the sample-sample similarity graph from source node to sink nodes and the probability that it stops at each sink node can be computed analogous to the class-based case as follows:

$$\mathbf{p}_i^s = (\tilde{\mathbf{H}}_{1\times n_s'}^{x\to s}, \tilde{\mathbf{H}}_{1\times n_l}^{x\to t})(\mathbf{I} - \tilde{\mathbf{H}}^{s,t\to s,t})^{-1}\tilde{\mathbf{H}}^{s,t\to c} \tag{8}$$

where

$$\tilde{\mathbf{H}}^{s,t\to s,t} = \begin{pmatrix} \tilde{\mathbf{H}}_{n_s'\times n_s'}^{s\to s} & \tilde{\mathbf{H}}_{n_s'\times n_l}^{s\to t} \\ \tilde{\mathbf{H}}_{n_l\times n_s'}^{t\to s} & \tilde{\mathbf{H}}_{n_l\times n_l}^{t\to t} \end{pmatrix}$$

$$\tilde{\mathbf{H}}^{s,t\to c} = \begin{pmatrix} \mathbf{0}_{n_s'\times k_t}^{s\to c} \\ (\tilde{\mathbf{Y}}_{\mathcal{L}}^{t})_{n_l\times k_t}^{t\to c} \end{pmatrix} \tag{9}$$

In this way, we obtain the similarity between a source domain sample and each target domain class as $\mathbf{p}_i^s$ from the sample similarity graph. Now, we can combine the similarity from both perspectives in Eq. (6) and Eq. (8) as follows:

$$\mathbf{p}_i = \lambda\mathbf{p}_i^c + (1-\lambda)\mathbf{p}_i^s \tag{10}$$

where λ is a balance parameter. Now, for each class j in the target domain, we rank the similarity scores $p_{ij}(\forall i)$ of all source domain samples, and the top ranked samples are selected and transferred to class j, i.e., we assign pseudo label j to them to expand $\mathcal{L}$. Although the transferred samples do not belong to the target domain classes based on the original labels, they are highly similar to the target domain classes so that they can well capture the characteristics of these classes.

Active Learning

By the cross-class transferred samples, the labeled set $\mathcal{L}$ is expanded into $\tilde{\mathcal{L}}$. Now we can utilize $\tilde{\mathcal{L}}$ to train a model for target domain. Specifically, we consider the one-vs-all SVM classifier for the multi-class problem. With the labels for samples in $\mathcal{L}$ and pseudo labels for transferred samples, we can rewrite $\tilde{\mathcal{L}} = \{(\mathbf{x}_i, \mathbf{y}_i, \theta_i)\}$, where θ_i is the weight for the sample i. In the standard SVM, we set $\theta_i = 1$ for all samples. However, because the transferred samples have pseudo labels which are not true labels, it is necessary to consider their influence. From Eq. (10), we obtain the similarity between samples and classes. Obviously, if p_{ij} is large, it is reasonable to trust its pseudo label j. Therefore, suppose a transferred sample $\mathbf{x}_i$ is assigned by pseudo label c_j^t, we directly set $\theta_i = p_{ic_j^t}$, and we assign $\theta_i = 1$ for the samples from $\mathcal{L}$ because they have true labels. The one-vs-all SVM classifier can be trained by the following dual formulation:

$$\min_{\alpha_1^c,\ldots,\alpha_l^c} \sum\nolimits_{i=1}^{l} \alpha_i^c - \frac{1}{2}\sum\nolimits_{i,j=1}^{l} \alpha_i^c\alpha_j^c y_{ic}y_{jc}K(\mathbf{x}_i,\mathbf{x}_j)$$
$$s.t.\ \sum\nolimits_{i=1}^{l} \alpha_i^c y_{ic} = 0,\ 0 \le \alpha_i^c \le C\theta_i \tag{11}$$

where α^c is the classifier parameter for class $c \in \mathcal{C}^t$, y_{ic} is the label vector that $y_{ic} = 1$ if the sample is labeled by c or $y_{ic} = -1$ otherwise, $K(\mathbf{x}_i, \mathbf{x}_j)$ is the kernel for SVM and we adopt linear kernel in this paper, and l is the size of the expanded training set $\tilde{\mathcal{L}}$. For class c, its classifier is constructed as $f_c^t(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i^c y_{ic}K(\mathbf{x}_i, \mathbf{x})$ and a larger output indicates that $\mathbf{x}$ is more likely to belong to c. This is a weighted SVM formulation (Yang, Song, and Wang 2007) which can be easily solved by well-established quadratic programming solvers, like quadprog function in MATLAB.

By Eq. (11), we can build the one-vs-all classifier for each target domain class. The multi-class prediction is given by

$$f^t(\mathbf{x}) = \text{argmax}_c \sum\nolimits_{i=1}^{l} \alpha_i^c y_{ic}K(\mathbf{x}_i, \mathbf{x}) \tag{12}$$

Algorithm 1 AL with Cross-class Similarity Transfer

Input: Source domain samples $\mathcal{D}^s$, target domain pool $\mathcal{D}^p$; Class-class similarity matrix $\mathbf{G}$;
Output: Classifiers f_c^t for the target domain classes

Initialize $\mathcal{L}$ by random seed, $\mathcal{U} = \mathcal{D}^p \backslash \mathcal{L}$;
Construct probability classifier f_c^s for source domain;
for $iter = 1 : max\_iter$ **do**
  Initialize $\mathbf{r}_i (i \in \mathcal{D}^s)$ using $f_c^s (c \in \mathcal{C}^s)$;
  Compute $\mathbf{p}_i^c$ on the class similarity graph by Eq. (6);
  Compute $\mathbf{p}_i^s$ on the sample similarity graph by Eq.(8);
  Compute $\mathbf{p}_i$ for each $i \in \mathcal{D}^s$ by Eq. (10);
  Initialize the expanded set $\tilde{\mathcal{L}} = \mathcal{L}$;
  for $c \in \mathcal{C}^t$ **do**
    Select samples $\mathcal{S}^c$ with largest $p_{ic} (\forall i)$ for c;
    Expand $\tilde{\mathcal{L}} = \tilde{\mathcal{L}} \cup \{(\mathbf{x}_i, c, \theta_i = p_{ic})\}, i \in \mathcal{S}^c$;
  end for
  Train classifiers $f_c^t (c \in \mathcal{C}^t)$ by Eq. (11) using $\tilde{\mathcal{L}}$;
  Compute entropy $E_i (i \in \mathcal{U})$ using current models f_c^t, heat-kernel similarity matrices $\mathbf{K}^{uu}$ and $\mathbf{K}^{us}$;
  Compute ranking score $r_i (i \in \mathcal{U})$ by Eq. (13);
  Select top ranked samples $\mathcal{S}^U$ for expert labeling;
  Update $\mathcal{L} = \mathcal{L} \cup \mathcal{S}^U, \mathcal{U} = \mathcal{U} \backslash \mathcal{S}^U$;
end for
Return f_c^t;

Then we can select samples from $\mathcal{U}$ for expert labeling based on the current model by uncertainty sampling. We adopt the best-vs-second-best strategy in the multi-class scenario considering its effectiveness (Joshi, Porikli, and Papanikolopoulos 2012). Specifically, given an unlabeled sample $\mathbf{x}_i^u \in \mathcal{U}$, the current model can output a value $o_i^c = f_c^t(\mathbf{x}_i^u)$ on each target domain class. Suppose c_1 and c_2 are the two classes that output the largest values ($\mathbf{x}_i^u$ is most likely to belong to them), we can compute the entropy of the sample as $E_i = -\sum_{j=1}^{2} p_i^j \log p_i^j$ where $p_i^j = \exp(o_i^{c_j}) / (\sum_{m=1}^{2} \exp(o_i^{c_m}))$ based on the soft-max operation. One simple method is to select the unlabeled samples with the largest entropy (uncertainty) for expert labeling. However, this method 1) may lead to redundant selection (Yang et al. 2015) and 2) fails to consider the information from source domain. The former can be addressed by considering the diversity of selected samples. In fact, the latter is an important issue in the cross-class setting. Because we are going to transfer source domain samples based on their similarity to the labeled samples, labeling one sample in the target domain will affect the transfer procedure in the next round. If one target domain sample has very few similar source domain samples and is diverse to the other labeled samples, labeling it may impose an outlier transient node in the sample graph defined in Eq. (7) such that it has little influence on the propagation. Therefore, we hope the selected sample to have some neighbors in the source domain. Formally, we first define a heat-kernel similarity matrix for the samples in $\mathcal{U}$ as $\mathbf{K}^{uu} \in \mathbb{R}^{n_u \times n_u}$, and another similarity matrix between $\mathcal{U}$ and a randomly sampled subset (to reduce the complexity) from $\mathcal{D}^s$ as $\mathbf{K}^{us} \in \mathbb{R}^{n_u \times n'_s}$ where $n'_s \ll n_s$ is the size of the subset. Then we can formulate the sample selection procedure as the following optimization problem:

$$\min_{r_i \geq 0,\ \sum_i r_i = 1} \quad -\mathbf{r}\mathbf{K}^{uu}\mathbf{E}' - \tau \mathbf{r}\mathbf{K}^{us}\mathbf{1}'_{n'_s} + \eta \mathbf{r}\mathbf{K}^{uu}\mathbf{r}' \quad (13)$$

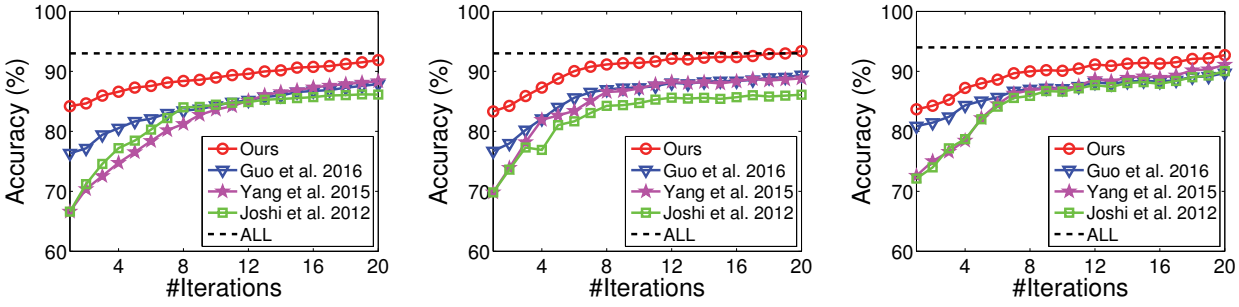
which can be efficiently solved by the quadratic problem solvers. r_i is the ranking score for $\mathbf{x}_i^u$ and the samples with largest ranking scores are selected for expert labeling. The formulation covers three aspects. The first term is $-\mathbf{r}\mathbf{K}^{uu}\mathbf{E}' = -\sum_i r_i (\sum_j K_{ij}^{uu} E_j)$, which considers the uncertainty. Note that this uncertainty is transferable from one sample to its related samples. For example, if sample i and j are similar (K_{ij}^{uu} is large), labeling sample i will significantly reduce the uncertainty of sample j because it is common that similar samples have similar labels. The second term is $-\mathbf{r}\mathbf{K}^{us}\mathbf{1}'_{n'_s} = -\sum_i r_i \sum_j K_{ij}^{us}$. As discussed above, the samples that have many similar samples in the source domain are preferred. The third term is $\mathbf{r}\mathbf{K}^{uu}\mathbf{r}' = \sum_{ij} K_{ij}^{uu} r_i r_j$. If sample i and j are similar (K_{ij}^{uu} is large), assigning large values to r_i and r_j simultaneously will lead to large penalty. Therefore, minimizing this term can lead to diverse selection. We summarize the procedure of the proposed cross-class T-AL approach in Algorithm 1.

Experiment

Settings

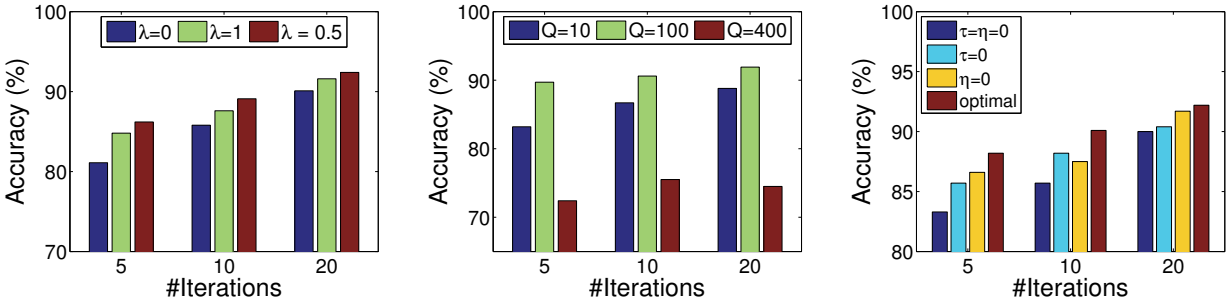
We select three datasets to demonstrate the effectiveness of the proposed approach. The first is CIFAR10 (Krizhevsky 2009) which has 10 object classes and each class has $6,000$ images. We select 8 classes as source domain and 2 classes as target domain which leads to $C_{10}^2 = 45$ different splits. We report the average result on the 45 splits. The second is Animals with Attributes (AwA) (Lampert, Nickisch, and Harmeling 2014) which has 50 animal classes. Following the standard split in (Lampert, Nickisch, and Harmeling 2014), 40 classes with $24,295$ images are regarded as source domain, and the other 10 classes with $6,180$ images as target domain. The third is aPascal-aYahoo (aPY) (Farhadi et al. 2009) which has two subsets. The first subset is aPascal from Pascal VOC2008 challenge that has 20 classes and $12,695$ images, which is used as source domain. The second subset is aYahoo collected from Yahoo image search which has 12 classes and $2,644$ images that are similar but different from aPascal, and it is used as target domain. For CIFAR10, each class is described by the word vector from (Huang et al. 2012). For AwA and aPY, each class has an attribute vector provided by the dataset. To construct the class similarity matrix $\mathbf{G}$, we adopt the cosine similarity between the classes' attribute/word vectors and negative similarity is set to 0. To make the matrix more discriminative, we perform square transformation on each element and then perform ℓ_1-norm normalization. For each image, we adopt the Caffe tool (Donahue et al. 2014) with the pretrained AlexNet (Krizhevsky, Sutskever, and Hinton 2012) and use the $4,096$-dimensional output of fc7 layer as feature vector.

We use two state-of-the-art AL approaches (Joshi, Porikli, and Papanikolopoulos 2012; Yang et al. 2015) as baselines. Because the traditional T-AL approaches cannot perform



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

Figure 1: Benchmark comparison.



(a) Effect of λ (CIFAR10) (b) Effect of Q (AwA) (c) Effect of τ and η (aPY)

Figure 2: Effect of parameters.

cross-class transfer, it is not possible to make comparison. In addition, a cross-class approach (Guo et al. 2016) is also included in the baselines even though it requires the class attributes for knowledge transfer which is more difficult to obtain than the class similarity. Following (Yang et al. 2015), to evaluate each approach, we split the target domain samples equally into $\mathcal{D}^p$ and D^t. For each round, we select 2, 10, and 12 samples from D^p for CIFAR10, AwA, and aPY respectively for labeling and training, and we use the classification accuracy on D^t as the evaluation metric.

When implementing our approach, we use the following settings. The parameter λ in Eq. (10) is set to 0.5 in all experiments. The values of τ and η in Eq. (13) are chosen by class-wise cross validation (Guo et al. 2016) which splits the source domain by **class** and uses some classes as source domain and the other classes as validation domain to simulate the cross-class setting and uses the label information for parameter selection. These parameters are chosen from $\{0.1, 1, 10\}$. In line 10 of Algorithm 1, we select and transfer $Q = |\mathcal{S}^c| = 200, 100, 100$ samples for each target domain class for three datasets. To construct sample similarity graph in Eq. (7), we randomly choose 500 samples from $\mathcal{D}^s$. For the matrix $\mathbf{K}^{us}$ in Eq. (13), we randomly choose $1,000$ samples from $\mathcal{D}^s$. For the baselines, we also utilize the cross-validation on source domain to find the optimal parameters. In addition, we show the result that uses a fully labeled $\mathcal{D}^p$ to train classifiers. We report the average results over 20 runs.

Results

The performance curves on three datasets are plotted in Figure 1. It can be observed that our approach significantly outperforms the baselines, which verifies its effectiveness. Specifically, the accuracy of our approach at 10-th iteration on three datasets are 88.98%, 90.98% and 91.01%, which improves upon the best baselines with $\mathbf{5.62}\%$, $\mathbf{5.33}\%$, and $\mathbf{4.27}\%$, indicating relative error reductions of $\mathbf{33.7}\%$, $\mathbf{37.1}\%$, and $\mathbf{32.2}\%$. In addition, after only 20 iterations, our approach reaches and even surpasses the accuracy of using fully labeled $\mathcal{D}^p$ for classifier training (ALL), which validates that our approach indeed selects the most informative samples for labeling and the samples transferred from source domain can well capture the characteristics of target domain classes and provide valuable knowledge. To achieve 90% accuracy, our approach needs 12, 7, and 7 iterations (24, 70, and 84 labeled samples), while the best baseline needs 24, 18, and 14 iterations (48, 180, 168 samples), which means our approach saves $\mathbf{50}\%$, $\mathbf{55.6}\%$, $\mathbf{50}\%$ labeling efforts.

The effect of λ in Eq. (10) is shown in Figure 2(a). We can see that when we ignore the class graph ($\lambda = 0$) or sample graph ($\lambda = 1$), the performance drops significantly, implying that both graphs are important for building similarity between source domain samples and target domain classes. Moreover, the results show that class graph ($\lambda = 1$) performs better than sample graph ($\lambda = 0$). This is because the

class similarity can provide more comprehensive information about the class. But there are only a few labeled samples such that the sample graph may miss important information.

The effect of the number of transferred samples for each target domain class (Q) is shown in Figure 2(b). When Q is small, increasing it can improve performance because more useful information is transferred. However, when it is too large (e.g., 400), many dissimilar samples will be transferred such that they may decrease the performance dramatically.

The effect of τ and η in Eq. (13) is shown in Figure 2(c). We can see that the diversity and information from source domain are both necessary for choosing valuable samples.

Conclusion

In this paper, we propose a novel cross-class T-AL approach which simultaneously transfers samples from source domain that are very similar to target domain classes based on the class-class similarity and sample-sample similarity propagation, and selects the most informative samples in the target domain for expert labeling. Comprehensive experiments on three datasets demonstrate the superiority of the proposed approach over several state-of-the-art related approaches.

References

Bishop, C. M., et al. 2006. *Pattern recognition and machine learning*, volume 1. Springer, New York.

Changpinyo, S.; Chao, W.; Gong, B.; and Sha, F. 2016. Synthesized classifiers for zero-shot learning. In *CVPR*.

Chattopadhyay, R.; Fan, W.; Davidson, I.; Panchanathan, S.; and Ye, J. 2013. Joint transfer and batch-mode active learning. In *ICML*.

Choi, J.; Rastegari, M.; Farhadi, A.; and Davis, L. S. 2013. Adding unlabeled samples to categories by learned attributes. In *CVPR*.

Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; and Darrell, T. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*.

Farhadi, A.; Endres, I.; Hoiem, D.; and Forsyth, D. A. 2009. Describing objects by their attributes. In *CVPR*.

Fellbaum, C. D. 1998. Wordnet: An electronic lexical database. Technical report, MIT Press.

Grinstead, C. M., and Snell, J. L. 1997. *Introduction to Probability*. American Mathematical Society.

Guillaumin, M.; Mensink, T.; Verbeek, J. J.; and Schmid, C. 2009. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *ICCV*.

Guo, Y.; Ding, G.; Jin, X.; and Wang, J. 2015. Learning predictable and discriminative attributes for visual recognition. In *AAAI*.

Guo, Y.; Ding, G.; Wang, Y.; and Jin, X. 2016. Active learning with cross-class knowledge transfer. In *AAAI*.

Huang, E. H.; Socher, R.; Manning, C. D.; and Ng, A. Y. 2012. Improving word representations via global context and multiple word prototypes. In *ACL*.

Joshi, A. J.; Porikli, F.; and Papanikolopoulos, N. P. 2012. Scalable active learning for multiclass image classification. *IEEE TPAMI*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

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

Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2014. Attribute-based classification for zero-shot visual object categorization. *IEEE TPAMI*.

Li, L.; Jin, X.; Pan, S. J.; and Sun, J. 2012. Multi-domain active learning for text classification. In *KDD*.

Li, S.; Xue, Y.; Wang, Z.; and Zhou, G. 2013. Active learning for cross-domain sentiment classification. In *IJCAI*.

Lim, J. J.; Salakhutdinov, R.; and Torralba, A. 2011. Transfer learning by borrowing examples for multiclass object detection. In *NIPS*.

Lin, Z.; Ding, G.; and Hu, M. 2015. Image auto-annotation via tag-dependent random search over range-constrained visual neighbours. *MTAP*.

Lu, X.; Yuan, Y.; and Yan, P. 2014. Alternatively constrained dictionary learning for image superresolution. *IEEE Trans. Cybernetics*.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE TKDE*.

Settles, B. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, Univ. of Wisconsin–Madison.

Shi, X.; Fan, W.; and Ren, J. 2008. Actively transfer domain knowledge. In *ECML*.

Shrivastava, A.; Gupta, A.; and Girshick, R. B. 2016. Training region-based object detectors with online hard example mining. In *CVPR*.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical report.

Yang, Y.; Ma, Z.; Nie, F.; Chang, X.; and Hauptmann, A. G. 2015. Multi-class active learning by uncertainty sampling with diversity maximization. *IJCV*.

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

Yu, K.; Bi, J.; and Tresp, V. 2006. Active learning via transductive experimental design. In *ICML*.

Zhu, Y.; Chen, Y.; Lu, Z.; Pan, S. J.; Xue, G.; Yu, Y.; and Yang, Q. 2011. Heterogeneous transfer learning for image classification. In *AAAI*.