Active Transfer Learning

Vaibhav Nagar
14785

Prawaan Singh
14495

Ishika Soni
14275

Problem Statement

Active learning (AL) and transfer learning (TL) are one of the prominent tools for saving the labelling effort for training supervised classification models. One obvious way that one can think of would be to combine both of their power in an Transfer-Active learning (T-AL) setup, this has attracted some interest in recent past.

T-AL algorithms are used when one has sufficient training data for one supervised learning task (the source task) but only very limited training data for a second task (the target task) that is similar but not identical to the first and we can also ask for more labelled data in the target class. These algorithms use varying assumptions about the similarity between the tasks to carry information from the source to the target task. Most of the existing approaches in this field 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. But a more interesting setting is where source domain are related/similar to but different from the target domain classes.

So we are investigating this setting : A cross-class approach 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 as done in [1]

Prior Work

Both Active and transfer learning model a lot of unsupervised labels but have their own strengths and weaknesses. It possible to combine their strengths to model a distribution and obtain potentially better results[9]. The key intuition in it was to use the knowledge transferred from other domain as often as possible to help learn the current domain, and query experts only when necessary.

Shi, Fan, and Ren (2008) [4] proposed to transfer knowledge from other source as often as possible and true labels were queried only when the likelihood that the unlabeled samples in target domain can be correctly classified became too low.

Li et al. (2012) [5] used a shared latent space for source domain and target domain such that the information in source domain could be used well. Then the most informative samples were selected by considering the information from the latent space.

Chattopadhyay et al. (2013) [6] 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

Utilizing more information from source domain can save the labeling effort in target domain and result in better performance, which has been demonstrated neatly by these 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. The paper the we implemented and tried to made some changes to is *Active Learning with Cross-Class Similarity Transfer* [1] that mainly address this problem.

Tools/Softwares used

- PyTorch: To get embedding/features of images from pretrained imagenet models.
- Gensim: Word2Vec model trained on GoogleNews-vectors is used to compute similarities between classes.
- Python (mainly Numpy, cvxopt and sklearn packages): The whole active-transfer algorithm is implemented in numpy and classifiers like SVM and MLP are taken from sklearn.

Experiments

Datasets

- CIFAR 10 [2]
  - 10 animal classes
  - Each class with 6,000 images
  - 8 classes as source and 2 as target domain
- MNIST handwritten digits [3]
  - 10 object classes
  - Total images 60000
  - 8 classes as source and 2 as target domain

Pretrained Models:
We use the following pretrained models (trained on Imagenet) to generate embedding/feature vectors of images.

- AlexNet [12]
- ResNet18 [13]

Approach

The following aproach is taken from '*Active Learning with Cross-Class Similarity Transfer*' [1]. We have implemented this approach from scratch using the aforementioned tools and softwares.

Notations

Target domain data - $\mathcal{D}^p = \left\{x_1^p, x_2^p, ..., x_{n_p}^p\right\}$ where $x_i^p \in R^d$ and k_l classes $\mathcal{C}^l = \left\{c_1^p, c_2^p, ..., c_{k_t}^t\right\}$. Each sample belongs to one class in $\mathcal{C}^t$.

$\mathcal{D}^p$= $\mathcal{L}$ (labeled set) $\cup$ $\mathcal{U}$ (unlabeled set)

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.

Cross-class transfer setting

Given another **Auxiliary Source domain** for knowledge transfer :
Contains a set of labeled samples $\mathcal{D}^s = \left\{\left(x_1^s, y_1^s\right), ..., \left(x_{n_s}^s, y_{n_s}^s\right)\right\}$ and k_s classes $\mathcal{C}^s = \left\{c_1^s, ..., c_{k_s}^s\right\}$.
Cross class assumption $\Longrightarrow \mathcal{C}^s \cap \mathcal{C}^t = \phi$
Each source domain sample belongs to one class in $\mathcal{C}^s$ and $y_i^s \in \{-1, 1\}^{k_s}$ is the label vector where $y_{ij}^s = 1$ if sample x_i^s belongs to c_j^s or -1 otherwise.

Class similarity matrix - $\boldsymbol{G} \in R^{(k_s+k_t)\times(k_s+k_t)}$, where g_{ij} denotes the similarity between two class

Class-Class Similarity Matrix

To construct class similarity matrix G, we use the Word2Vec model trained on "GoogleNews-vectors" and compute the cosine similarity between the class attribute vectors generated from this Word2Vec model. For both the datasets we use the same model and compute the similarities between all the classes (source + target) and to make it more discriminative, we perform L2 normalization on each row so that it can be directly used in transition probability matrix.

Transfer Learning

To perform cross-class knowledge transfer the similarity based sample transfer method is proposed where we select some examples from the source dataset (which do not belong to any target class) 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 by exploiting the similarities between the source and target domain.

Similarity between source domain samples and target domain classes is obtained by two methods:

1. **Class-Class similarity Graph:**
   Similarity propagation on the class-class similarity graph which is motivated by the graph-based random walk for information propagation. The relationship between classes is given by G and the relationship between a source sample to all source classes is computed by training k_s one-vs-all probability classifiers like Logistic Regression (trained on only source dataset), in which each classifier outputs the probability that the sample belongs to a certain source class and normalizes the probabilities by taking softmax and use these probabilities directly in transition probability matrix for the standard absorbing Markov chain. Instead of using hard assignment (one-hot) of source sample on source classes, we use soft assignment as it captures the relationship between the classes and incorporates intra-class diversity. In the transition probability matrix, the target classes are recurrent nodes whereas all other nodes (source samples) are transient nodes. Based on its theory, in similarity graph, the probability that the random walk starting at sample, stops at each target domain class is computed.

$$p_i^c = r_i \left(\mathrm{I} - \mathrm{G}_{k_s \times k_s}^{s \to s}\right)^{-1} \mathrm{G}_{k_s \times k_t}^{s \to t}$$

   where r_i are the output of source classifiers.

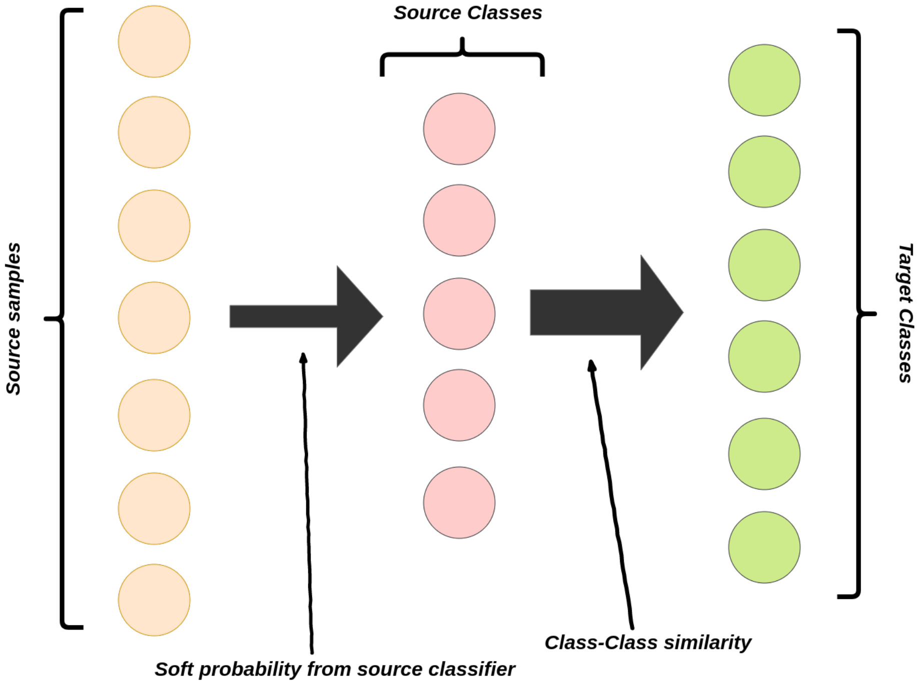


Figure 1: Class-Class similarity graph

2. **Sample-Sample similarity graph:** We also consider the similarity propagation on the sample-sample similarity graph as in active learning, we have some labeled samples of target domain classes as well which can provide some specific information about target domain classes. In this similarity graph, the transition probability matrix between samples is defined by the heat-kernel similarity ($h_{ij} = \exp(-||x_i - x_j||^2/\sigma^2)$)). We first randomly select some source samples n'_s ($n'_s << n_s$) and then compute the heat-kernel similarity between all source samples plus target samples and the target samples plus these randomly selected source samples. And we compute the $sigma$ beforehand by taking the mean Euclidean distance between the feature vectors of all samples in the training dataset (source + target). For target samples of the labeled set in the transition probability matrix, we use their true labels and normalizes the matrix row-wise. Here again the target classes are recurrent nodes whereas all other nodes (source samples + labeled target samples) are transient nodes in transition probability matrix for the standard absorbing Markov chain. Then we perform random walk for similarity propagation on the sample-sample similarity graph as performed in the above method and compute probabilities for source samples to target classes.

$$p_i^s = \left(\mathrm{H}^{x\to s}_{1\times n'_s}, \mathrm{H}^{x\to t}_{1\times n_t}\right)\left(\mathrm{I} - \mathrm{H}^{s,t\to s,t}\right)^{-1}\mathrm{H}^{s,t\to c}$$

where H is a heat-kernel singularity matrix.

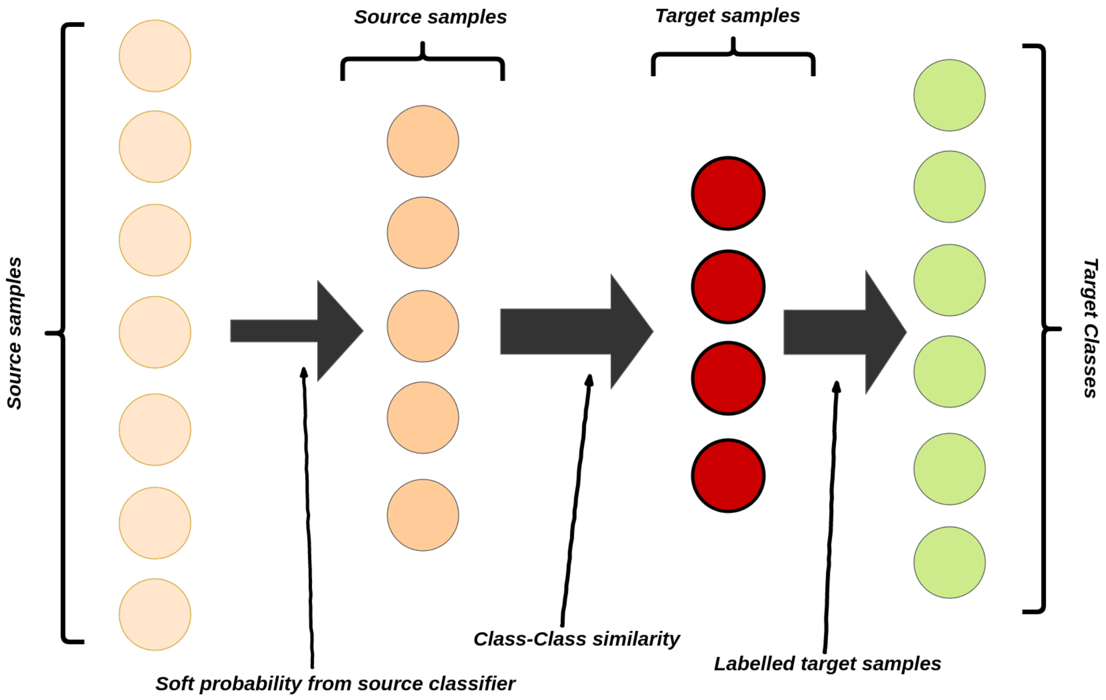


Figure 2: Sample-Sample similarity graph

In this way, we obtain the similarity between a source domain sample and each target domain class from the sample similarity graph and take the linear combination of the similarities from both perspectives.

$$p_i = \lambda p_i^c + (1-\lambda)\, p_i^s$$

We select the top ranked source samples based on these probabilities for each target class and assign pseudo label to them. These samples do not ideally belong to target classes but they are highly similar to target classes and thus capture the characteristics of these classes.

Active Learning

After we have transferred samples for each target classes, the labeled set $\mathcal{L}$ is expanded to $\tilde{\mathcal{L}}$ (which contains the true and pseudo labels). Now we use the expanded labeled set to train a classifier for target domain. But we must have different weights over the data points, since the pseudo labels are not true labels but have different degree of similarity with the target class. Keeping this in mind we assign labels as 1 for all the true label samples and the suppose a transferred sample x_i is assigned by pseudo label c_j^i, we set its weight to be $p_{ic_j^t}$

Then we select samples from $\mathcal{U}$ for expert labeling based on the current target classifier by uncertainty sampling. We first compute the entropy of each sample of unlabeled set as

$$E_i = -\Sigma_{j=1}^{2} p_i^j \log p_i^j where p_i^j = exp(o_i^{c_j})/(\Sigma_{m=1}^{2} exp(o_i^{c_m}))$$

To avoid redundant selection and also consider the information from source domain, we compute ranking score r_i for each sample of unlabeled set. We first compute the heat-kernel similarity between all the samples of unlabeled set. We wish to select only those samples from unlabeled set which are more similar to source samples, in order to avoid that sample to become an outlier transient node in the sample-sample similarity graph in the next iteration which will have little influence in information propagation. So we also compute another similarity matrix between the unlabeled set and randomly selected source samples n_s'' ($n_s'' << n_s$). Then we solves the following optimization which is quadratic in r (ranking score)

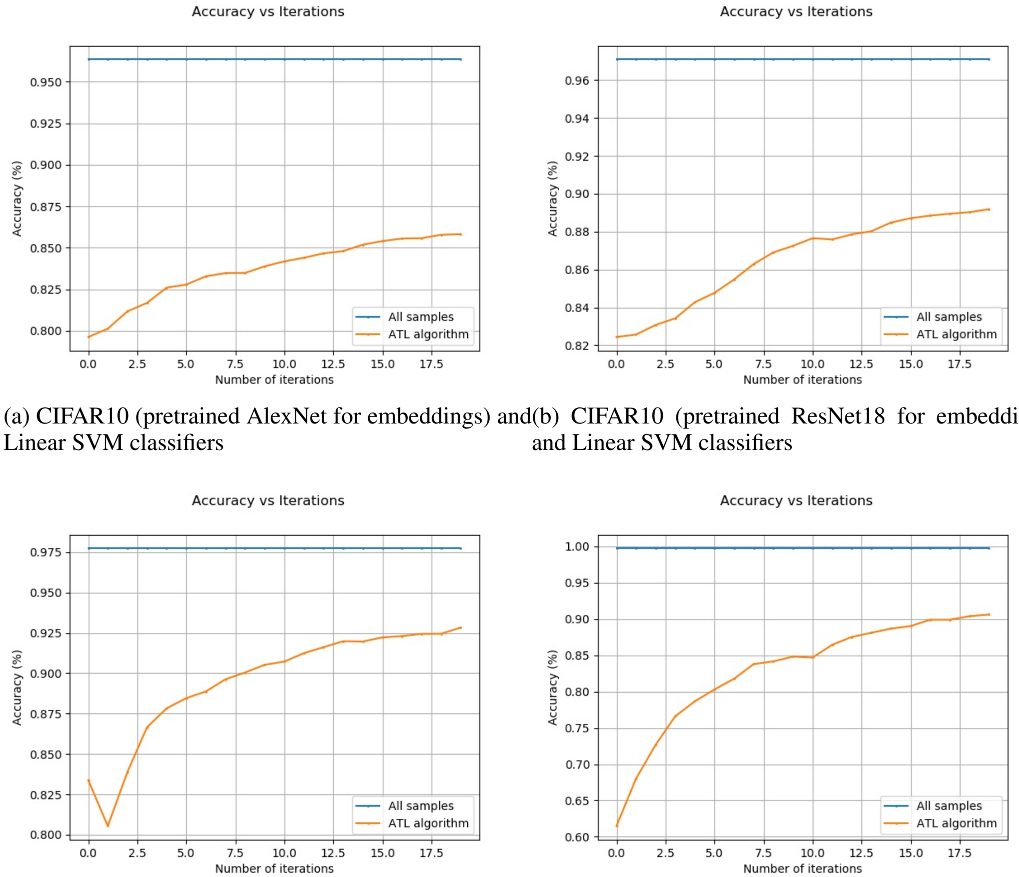
$$\min_{r_i \geq 0, \Sigma_i r_i = 1} - r\mathrm{K}^{uu}\mathrm{E}' - \tau r\mathrm{K}^{us} 1'_{n_s''} + \eta r\mathrm{K}^{uu} r'$$

Here the first term $-r\mathrm{K}^{uu}\mathrm{E}'$ considers the uncertainty which is transferable from one sample to its related samples. The second term $r\mathrm{K}^{us}1'_{n_s'}$ ensures that the selected sample is more similar to source samples. And the final term more-or-less acts like a penalty term which penalizes the above equation to avoid giving high ranking to two similar samples of unlabeled set. So minimizing this term can lead to diverse selection.
We use the ”cvxopt” package of python to solve the above convex optimization problem.
Based on the ranking scores of samples of unlabeled set, we select top ranked samples for expert labeling and transfer to target labeled set.

Results

Model parameters:
We used the following parameters in the algorithm for both datasets- CIFAR-10 and MNIST in all experiments.

- Number of target classes = 2 and number of source classes = 8
- $\lambda = 0.5$ for convex combination of probabilities of source samples computed from both perspectives
- $n_s' = 500$ for computing heat-kernel similarity matrix in sample-sample similarity graph
- $n_s'' = 1000$ for computing heat-kernel similarity matrix in the quadratic equation of ranking scores
- 200 sample for each target class are transferred from source to expand target labeled set
- $\tau = 0.01$ and $\eta = 0.0001$ in the quadratic equation of ranking scores
- 2 samples per iteration are selected from unlabeled target set based on their ranking scores
- Number of maximum iterations = 20



(a) CIFAR10 (pretrained AlexNet for embeddings) and Linear SVM classifiers

(b) CIFAR10 (pretrained ResNet18 for embeddings) and Linear SVM classifiers

(c) CIFAR10 (pretrained ResNet18 for embeddings) and MLP classifiers

(d) MNIST (pretrained ResNet18 for embeddings) and Linear SVM classifiers

Figure 3: Mean classification accuracy (%) for target dataset (test only). The performance curve of ”All samples” denotes the accuracy when all labeled samples of target train dataset are used to build the classifier and ”ATL algorithm” denotes the accuracy curve when target classifiers are trained on samples selected by the active-transfer learning only.

In all our experiments, we divide the dataset based on number of classes in which source classes $= 8$ and target classes $= 2$. So this leads to $\binom{10}{2} = 45$ different splits. The performance curves (mean classification accuracy) are plotted in Figure-3. These plots are the average results on 45 splits.

We note that by just changing the pretrained model for generating embedding of images from AlexNet to Resnet-18, not only it reduces the size of feature vectors from 4096 to 512, but also there is significant increase in mean classification accuracy in same number of iterations where all other hyper-parameters are kept same. Figure-1 (a) and (b)

Further we change the source and target classifiers which are trained on Linear SVM using one-vs-all strategy to Neural net (with one hidden layer of 100 nodes and ReLu activation). We observe that by this change, the performance enhances by significant amount. And another benefit is that we don’t need to create different classifier for each target class, a single MLP classifier is more efficient in terms of space and time complexity. Figure-3 (c)

We also test this algorithm on the MNIST handwritten digits dataset which is not used in the paper. Figure-3 (d)

Link to code

Things Learnt

Learnt about important aspects of machine learning that brought about the need for active learning. Active learning gives us a way of balancing and achieving a trade-off between the accuracy and cost associated with our model. Along with this we learnt the concept of transfer learning which helps model by getting information from a similar but different domain.

With all this we were made to reach a point where we read about combining active learning and transfer learning into a single modeling problem. After this we learnt about the ways to combine them in a single problem and we read some papers that combined them into a single algorithm in their approach. We then learnt about ways in which it was being applied on the same source and target domains.

We then came across this paper [1], which worked on different target and domain classes. This brought the concept of working with the notion of similarity across classes and perform a mixture of transfer and active learning to obtain desirable results.

We got great understanding of how we could relate similar classes and transfer knowledge from one-to-another domain. We then started with implementation, while going through multiple stages of this project we learnt about:

- The theory and applications of Active Learning and Transfer learning
- Active Learning sampling methods- uncertainty sampling and query by committee
- Different settings of transfer learning such as inductive transfer learning, transductive transfer learning, unsupervised transfer learning
- Absorbing Markov Chain and Random walk method of computing similarity
- Word2Vec for computing similarity matrix G

Future Aspects

In current approach, we have to perform cross-validation to estimate the number of samples actively selected from unlabeled set for expert labelling. This modelling can be made more flexible and efficient by making it non-parametric on the number of actively selected samples, it will help us deduce it and come up with better results in significantly reduce number of iterations.

References

[1] Yuchen Guo, Guiguang Ding, Yue Gao, Jungong Han *Active Learning with Cross-Class Similarity Transfer*, AAAI 2017

[2] Alex Krizhevsky *Convolutional Deep Belief Networks on CIFAR-10*

[3] LeCun, Yann, Corinna Cortes, and Christopher JC Burges *MNIST handwritten digit database* AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist 2 (2010).

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

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

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

[7] Aggarwal, Charu C and Kong, Xiangnan and Gu, Quanquan and Han, Jiawei and Yu, Philip S *Active learning: A survey*, 2014

[8] Settles, Burr *Active learning literature survey*, 2010

[9] Yang, Liu and Hanneke, Steve and Carbonell, Jaime *A theory of transfer learning with applications to active learning*, 2013

[10] Ahmed Dawod (Mansoura University) *Active Learning Survey*, 2013

[11] Weiss, Karl and Khoshgoftaar, Taghi M and Wang, DingDing *A survey of transfer learning*, 2016

[12] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. ”Imagenet classification with deep convolutional neural networks.” Advances in neural information processing systems. 2012.

[13] He, Kaiming, et al. ”Deep residual learning for image recognition.” Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.