An Iterative Similarity based Adaptation Technique for Cross Domain Text Classification

Himanshu S. Bhatt **Deepali Semwal** **Shourya Roy**
Xerox Research Center India, Bengaluru, INDIA
{Himanshu.Bhatt,Deepali.Semwal,Shourya.Roy}@xerox.com

Abstract

Supervised machine learning classification algorithms assume both train and test data are sampled from the same domain or distribution. However, performance of the algorithms degrade for test data from different domain. Such cross domain classification is arduous as features in the test domain may be different and absence of labeled data could further exacerbate the problem. This paper proposes an algorithm to adapt classification model by iteratively learning domain specific features from the unlabeled test data. Moreover, this adaptation transpires in a similarity aware manner by integrating similarity between domains in the adaptation setting. Cross-domain classification experiments on different datasets, including a real world dataset, demonstrate efficacy of the proposed algorithm over state-of-the-art.

1 Introduction

A fundamental assumption in supervised statistical learning is that training and test data are independently and identically distributed (i.i.d.) samples drawn from a distribution. Otherwise, good performance on test data cannot be guaranteed even if the training error is low. In real life applications such as business process automation, this assumption is often violated. While researchers develop new techniques and models for machine learning based automation of one or a handful business processes, large scale adoption is hindered owing to poor generalized performance. In our interactions with analytics software development teams, we noticed such pervasive diversity of learning tasks and associated inefficiency. Novel predictive analytics techniques on standard datasets (or limited client data) did not generalize across different domains (new products & services) and has limited applicability. Training models from scratch for every new domain requires human annotated labeled data which is expensive and time consuming, hence, not pragmatic.

On the other hand, transfer learning techniques allow domains, tasks, and distributions used in training and testing to be different, but related. It works in contrast to traditional supervised techniques on the principle of transferring learned knowledge across domains. While transfer learning has generally proved useful in reducing the labelled data requirement, brute force techniques suffer from the problem of *negative transfer* (Pan and Yang, 2010a). One cannot use transfer learning as the proverbial hammer, but needs to gauge when to transfer and also how much to transfer.

To address these issues, this paper proposes a domain adaptation technique for cross-domain text classification. In our setting for cross-domain classification, a classifier trained on one domain with sufficient labelled training data is applied to a different test domain *with no labelled data*. As shown in Figure 1, this paper proposes an iterative similarity based adaptation algorithm which starts with a shared feature representation of source and target domains. To adapt, it iteratively learns domain specific features from the unlabeled target domain data. In this process, similarity between two domains is incorporated in the adaptation setting for similarity-aware transfer. The major contributions of this research are:

- An iterative algorithm for learning domain specific discriminative features from unlabeled data in the target domain starting with an initial shared feature representation.

- Facilitating similarity-aware domain adaptation by seamlessly integrating similarity between two domains in the adaptation settings.

Proceedings of the 19th Conference on Computational Language Learning, pages 52–61,
Beijing, China, July 30-31, 2015. ©2015 Association for Computational Linguistics

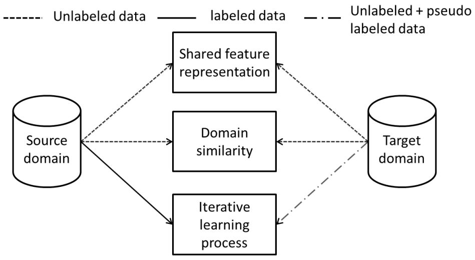
Unlabeled data
labeled data
Unlabeled + pseudo labeled data
Shared feature representation
Source domain
Domain similarity
Target domain
Iterative learning process

Figure 1: Outlines different stages of the proposed algorithm i.e. shared feature representation, domain similarity, and the iterative learning process.

To the best of our knowledge, this is the first-of-its-kind approach in cross-domain text classification which integrates similarity between domains in the adaptation setting to learn domain specific features in an iterative manner. The rest of the paper is organized as follows: Section 2 summarizes the related work, Section 3 presents details about the proposed algorithm. Section 4 presents databases, experimental protocol, and results. Finally, Section 5 concludes the paper.

2 Related Work

Transfer learning in text analysis (domain adaptation) has shown promising results in recent years (Pan and Yang, 2010a). Prior work on domain adaptation for text classification can be broadly classified into instance re-weighing and feature-representation based adaptation approaches.

Instance re-weighing approaches address the difference between the joint distributions of observed instances and class labels in source domain with that of target domain. Towards this direction, Liao et al. (2005) learned mismatch between two domains and used active learning to select instances from the source domain to enhance adaptability of the classifier. Jiang and Zhai (2007) proposed instance weighing scheme for domain adaptation in NLP tasks which exploit independence between feature mapping and instance weighing approaches. Saha et al. (2011) leveraged knowledge from source domain to actively select the most informative samples from the target domain. Xia *et al.* (2013) proposed a hybrid method for sentiment classification task that also addresses the challenge of mutually opposite orientation words.

A number of domain adaptation techniques are based on learning common feature representation (Pan and Yang, 2010b; Blitzer et al., 2006; Ji et al., 2011; Daumé III, 2009) for text classification. The basic idea being identifying a suitable feature space where projected source and target domain data follow similar distributions and hence, a standard supervised learning algorithm can be trained on the former to predict instances from the latter. Among them, Structural Correspondence Learning (SCL) (Blitzer et al., 2007) is the most representative one, explained later. Daumé (2009) proposed a heuristic based non-linear mapping of source and target data to a high dimensional space. Pan et al. (2008) proposed a dimensionality reduction method Maximum Mean Discrepancy Embedding to identify a latent space. Subsequently, Pan et al. (2010) proposed to map domain specific words into unified clusters using spectral clustering algorithm. In another follow up work, Pan *et al.* (2011) proposed a novel feature representation to perform domain adaptation via Reproducing Kernel Hilbert Space using Maximum Mean Discrepancy. A similar approach, based on co-clustering (Dhillon et al., 2003), was proposed in Dai *et al.* (2007) to leverage common words as bridge between two domains. Bollegala et al. (2011) used sentiment sensitive thesaurus to expand features for cross-domain sentiment classification. In a comprehensive evaluation study, it was observed that their approach tends to increase the adaptation performance when multiple source domains were used (Bollegala et al., 2013).

Domain adaptation based on iterative learning has been explored by Chen et al. (2011) and Garcia-Fernandez et al. (2014) and are similar to the philosophy of the proposed approach in appending pseudo-labeled test data to the training set. The first approach uses an expensive feature split to co-train two classifiers while the former presents a single classifier self-training based setting. However, the proposed algorithm offers novel contributions in terms of 1) leveraging two independent feature representations capturing the shared and target specific representations, 2) an ensemble of classifiers that uses labelled source domain and pseudo labelled target domain instances carefully moderated based on similarity between two domains. Ensemble based domain adaptation for text classification was first proposed by Aue and Gammon (2005) though their approach could not achieve significant improvements over baseline. Later, Zhao et al. (2010) proposed online transfer learning (OTL) framework which forms the basis of our ensemble based domain adaptation. However, the proposed algorithm differs in the following ways: 1) an unsupervised approach that transforms unlabeled data into pseudo labeled data unlike OTL which is supervised, and 2) incorporates similarity in the adaptation setting for gradual transfer.

3 Iterative Similarity based Adaptation

The philosophy of our algorithm is gradual transfer of knowledge from the source to the target domain while being cognizant of similarity between two domains. To accomplish this, we have developed a technique based on ensemble of two classifiers. Transfer occurs within the ensemble where a classifier learned on shared representation transforms unlabeled test data into pseudo labeled data to learn domain specific classifier. Before explaining the algorithm, we highlight its salient features:

Common Feature Space Representation: Our objective is to find a *good* feature representation which minimizes divergence between the source and target domains as well as the classification error. There have been several works towards feature-representation-transfer approach such as (Blitzer et al., 2007; Ji et al., 2011) which derives a transformation matrix Q that gives a shared representation between the source and target domains. One of the widely used approaches is Structural Correspondence Learning (SCL) (Blitzer et al., 2006) which aims to learn the co-occurrence between features expressing similar meaning in different domains. Top k Eigenvectors of matrix, W, represent the principal predictors for weight space, Q. Features from both domains are projected on this principal predictor space, Q, to obtain a shared representation. Source domain classifier in our approach is based on this SCL representation. In Section 4, we empirically show how our algorithm generalizes to different shared representations.

Iterative Building of Target Domain Labeled Data: If we have enough labeled data from the target domain then a classifier can be trained without the need for adaptation. Hence, we wanted to explore if and how (*pseudo*) labeled data for the target domain can be created. Our hypothesis is that certain target domain instances are more similar to source domain instances than the rest. Hence a classifier trained on (a suitably chosen transformed representation of) source domain instances will be able to categorize similar target domain instances confidently. Such confidently predicted instances can be considered as pseudo labeled data which are then used to initialize a classifier in target domain.

Only handful of instances in the target domain can be confidently predicted using the shared representation, therefore, we further iterate to create pseudo labeled instances in target domain. In the next round of iterations, remaining unlabeled target domain instances are passed through both the classifiers and their output are suitably combined. Again, confidently labeled instances are added to the pool of pseudo labeled data and the classifier in the target domain is updated. This process is repeated till all unlabeled data is labeled or certain maximum number of iterations is performed. This way we gradually adapt the target domain classifier on pseudo labeled data using the knowledge transferred from source domain. In Section 4, we empirically demonstrate effectiveness of this technique compared to one-shot adaptation approaches.

Domain Similarity-based Aggregation: Performance of domain adaptation is often constrained by the dissimilarity between the source and target domains (Luo et al., 2012; Rosenstein et al., 2005; Chin, 2013; Blitzer et al., 2007). If the two domains are largely similar, the knowledge learned in the source domain can be aggressively transferred to the target domain. On the other hand, if the two domains are less similar, knowledge learned in the source domain should be transferred in a conservative manner so as to mitigate the effects of *negative transfer*. Therefore, it is imperative for domain adaptation techniques to account for similarity between domains and transfer knowledge in a similarity aware manner. While this may sound obvious, we do not see many works in domain adaptation literature that leverage inter-domain similarity for transfer of knowledge. In this work, we use the cosine similarity measure to compute similarity between two domains and based on that gradually transfer knowledge from the source to the target domain. While it would be interesting to compare how different similarity measures compare towards preventing negative transfer but that is not the focus of this work. In Section 4, we empirically show marginal gains of transferring knowledge in a similarity aware manner.

Table 1: Notations used in this research.

Symbol	Description
$\{\mathbf{x}_i^s, y_i^s\}_{i=1:n_s}$; $\mathbf{x}_i^s \in R^d; y_i^s \in \{-1,+1\}$	Labeled source domain instances
$\{\mathbf{x}_i^t\}_{i=1:n_t}$; $\hat{y}_i \in \{-1,+1\}$	Unlabeled target domain instances and predicted label for target domain
Q	Co-occurrence based projection matrix
P_u, P_s	Pool of unlabeled and pseudo-labeled target domain instances respectively
C_s, C_t ; function from $R^d \rightarrow \{-1,+1\}$	Classifier C_s is trained on $\{(Q\mathbf{x}_i^s, y_i^s)\}$; classifier C_t is trained on $\{\mathbf{x}_i^t, \hat{y}_i^t\}$ where $x_i^t \in P_s$ and $\hat{y}$ is the pseudo label predicted labels by Ensemble E
α	confidence of prediction
E	Weighted ensemble of C_s and C_t
θ_1, θ_2	confidence threshold for C_s and ensemble E
w^s, w^t	Weights for C_s and C_t respectively

3.1 Algorithm

Table 1 lists the notations used in this research. Inputs to the algorithm are labeled source domain instances $\{x_i^s, y_i^s\}_{i=1:n_s}$ and a pool of unlabeled target domain instances $\{x_i^t\}_{i=1:n_t}$, denoted by P_u. As shown in Figure 2, the steps of the algorithm are as follows:

1. Learn Q, a shared representation projection matrix from the source and target domains, using any of the existing techniques. SCL is used in this research.

2. Learn C_s on SCL-based representation of labeled source domain instances $\{Q\mathbf{x}_i^s, y_i^s\}$.

3. Use C_s to predict labels, $\hat{y}_i$, for instances in P_u using the SCL-based representation $Q\mathbf{x}_i^t$. Instances which are predicted with confidence greater than a pre-defined threshold, θ_1, are moved from P_u to P_s with pseudo label, $\hat{y}$.

4. Learn C_t from instances in $P_s \in \{\mathbf{x}_i^t, \hat{y}_i^t\}$ to incorporate target specific features. P_s only contains instances added in step-3 and will be growing iteratively (hence the training set here is small).

5. C_s and C_t are combined in an ensemble, E, as a weighted combination with weights as w^s and w^t which are both initialized to 0.5.

6. Ensemble E is applied to all remaining instances in P_u to obtain the label $\hat{y}_i$ as:

$$E(x_i^t) \rightarrow \hat{y}_i \rightarrow w^s C_s(Qx_i^t) + w^t C_t(x_i^t) \tag{1}$$

   (a) If the ensemble classifies an instance with confidence greater than the threshold θ_2, then it is moved from P_u to P_s along with pseudo label $\hat{y}_i$.

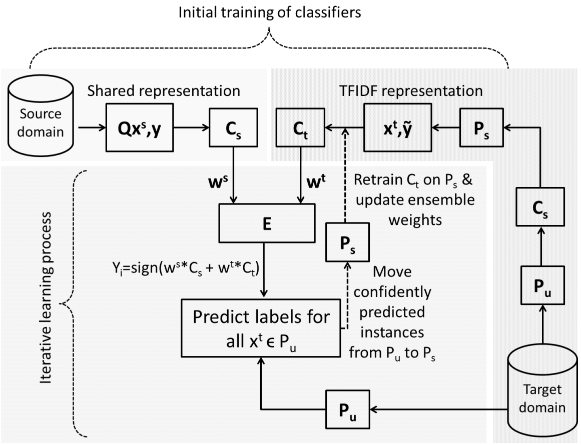
Initial training of classifiers
Shared representation
TFIDF representation
Source domain
Qx^s,y
C_s
C_t
x^t,ỹ
P_s
w^s
w^t
Retrain C_t on P_s & update ensemble weights
C_s
E
P_s
Y_i=sign(w^s*C_s + w^t*C_t)
Move confidently predicted instances from P_u to P_s
P_u
Predict labels for all x^t ∈ P_u
Iterative learning process
P_u
Target domain

Figure 2: Illustrates learning of the initial classifiers and iterative learning process of the proposed similarity-aware domain adaptation algorithm.

   (b) Repeat step-6 for all $x_i^t \in P_u$.

7. Weights w^s and w^t are updated as shown in Eqs. 2 and 3. This update facilitates knowledge transfer within the ensemble guided by the similarity between domains.

$$w^s_{(l+1)} = \frac{(sim * w_l^s * I(C_s))}{(sim * w_l^s * I(C_s) + (1 - sim) * w_l^t * I(C_t))} \tag{2}$$

$$w^t_{(l+1)} = \frac{((1 - sim) * w_l^t * I(C_t))}{(sim * w_l^s * I(C_s) + (1 - sim) * w_l^t * I(C_t))} \tag{3}$$

   where, l is the iteration, sim is the similarity score between domains computed using cosine similarity metric as shown in Eq. 4

$$sim = \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}||||\mathbf{b}||} \tag{4}$$

   where **a** & **b** are normalized vector representations for the two domains. $I(\cdot)$ is the loss function to measure the errors of individual classifiers in each iteration:

$$I(\cdot) = \exp\{-\eta l(C, Y)\} \tag{5}$$

   where, η is learning rate set to 0.1, $l(y, \hat{y}) = (y - \hat{y})^2$ is the square loss function, y is the label predicted by the classifier and $\hat{y}$ is the label predicted by the ensemble.

8. Re-train classifier C_t on P_s.

9. Repeat step $6 - 8$ until P_u is empty or maximum number of iterations is reached.

In this iterative manner, the proposed algorithm transforms unlabeled data in the test domain into pseudo labeled data and progressively learns classifier C_t. Confidence of prediction, α_i for i^{th} instance, is measured as the distance from the decision boundary (Hsu et al., 2003) which is computed as shown in Eq. 6.

$$\alpha = \frac{R}{|v|} \tag{6}$$

where R is the un-normalized output from the support vector machine (SVM) classifier, v is the weight vector for support vectors and $|v| = v^T v$. Weights of individual classifiers in the ensemble are updated with each iteration that gradually shifts emphasis from the classifier learned on shared representation to the classifier learned on target domain. Algorithm 1 illustrates the proposed iterative learning algorithm.

Algorithm 1 Iterative Learning Algorithm

```
Input: $C_s$ trained on shared co-occurrence based representation $Q\mathbf{x}$, $C_t$ initiated on TFIDF representation from $P_s$, $P_u$ remaining unlabeled target domain instances.
Iterate: $l = 0$ : till $P_u = \{\phi\}$ or $l \leq iterMax$
Process: Construct ensemble $E$ as weighted combination of $C_s$ and $C_t$ with initials weights $w_l^s$ and $w_l^t$ as $0.5$ and $sim$ = similarity between domains.
for $i = 1$ to $n$ (size of $P_u$) do
    Predict labels: $E(Q\mathbf{x}_i, \mathbf{x}_i) \rightarrow \hat{y}_i$; calculate $\alpha_i$
    if $\alpha_i > \theta_2$ then
        Remove $i^{th}$ instance from $P_u$ and add to $P_s$ with pseudo label $\hat{y}_i$.
    end if.
end for. Retrain $C_t$ on $P_s$ and update $w_l^s$ and $w_l^t$.
end iterate.
Output: Updated $C_t$, $w_l^s$ and $w_l^t$.
```

4 Experimental Results

The efficacy of the proposed algorithm is evaluated on different datasets for cross-domain text classification (Blitzer et al., 2007), (Dai et al., 2007). In our experiments, performance is evaluated on two-class classification task and reported in terms of classification accuracy.

4.1 Datasets & Experimental Protocol

The first dataset is the Amazon review dataset (Blitzer et al., 2007) which has four different domains, Books, DVDs, Kitchen appliances and Electronics. Each domain comprises 1000 positive and 1000 negative reviews. In all experiments, 1600 labeled reviews from the source and 1600 unlabeled reviews from the target domains are used in training and performance is reported on the non-overlapping 400 reviews from the target domain.

The second dataset is the 20 Newsgroups dataset (Lang, 1995) which is a text collection of approximately $20,000$ documents evenly partitioned across 20 newsgroups. For cross-domain text classification on the 20 Newsgroups dataset, we followed the protocol of Dai et al. (2007) where it is divided into six different datasets and the top two categories in each are picked as the two classes. The data is further segregated based on sub-categories, where each sub-category is considered as a different domain. Table 2 lists how different sub-categories are combined to represent the source and target domains. In our experiments, $4/5^{th}$ of the source and target data is used to learn shared feature representation and results are reported on the remaining $1/5^{th}$ of the target data.

Table 2: Elaborates data segregation on the 20 Newsgroups dataset for cross-domain classification.

dataset	D_s	D_t
comp vs rec	comp.graphics comp.sys.ibm.pc.hardware comp.sys.mac.hardware rec.motorcycles rec.sport.hockey	comp.os.ms-windows.misc comp.windows.x rec.autos rec.sport.baseball
comp vs sci	comp.graphics comp.os.ms-windows.misc sci.crypt sci.electronics	comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x sci.med sci.space
comp vs talk	comp.graphics comp.sys.mac.hardware comp.windows.x talk.politics.mideast talk.religion.misc	comp.os.ms-windows.miscnewline comp.sys.ibm.pc.hardware talk.politics.guns talk.politics.misc
rec vs sci	rec.autos rec.sport.baseball sci.med sci.space	rec.motorcycles rec.sport.hockey sci.crypt sci.electronics
rec vs talk	rec.autos rec.motorcycles talk.politics.guns talk.politics.misc	rec.sport.baseball rec.sport.hockey talk.politics.mideast talk.religion.misc
sci vs talk	sci.electronics sci.med talk.politics.misc talk.religion.misc	sci.crypt sci.space talk.politics.guns talk.politics.mideast

The third dataset is a real world dataset comprising tweets about the products and services in different domains. The dataset comprises tweets/posts from three collections, $Coll1$ about gaming, $Coll2$ about Microsoft products and $Coll3$ about mobile support. Each collection has 218 positive and negative tweets. These tweets are collected based on user-defined keywords captured in a listening engine which then crawls the social media and fetches comments matching the keywords. This dataset being noisy and comprising short-text is more challenging than the previous two datasets.

All datasets are pre-processed by converting to lowercase followed by stemming. Feature selection based on document frequency ($DF = 5$) reduces the number of features as well as speed up the classification task. For Amazon review dataset, TF is used for feature weighing whereas TFIDF is used for feature weighing in other two datasets. In all our experiments, constituent classifiers used in the ensemble are support vector machines (SVMs) with radial basis function kernel. Performance of the proposed algorithm for cross-domain classification task is compared with different techniques[1] including 1) in-domain classifier trained and tested on the same domain data, 2) baseline classifier which is trained on the source and directly tested on the target domain, 3) SCL[2], a widely used domain adaptation technique for cross-domain text classification, 4) 'Proposed w/o sim', removing similarity from Eqs. 2 & 3.

4.2 Results and Analysis

For cross-domain classification, the performance degrades mainly due to 1) feature divergence and 2) negative transfer owing to largely dissimilar domains. Table 3 shows the accuracy of individual classifiers and the ensemble for cross-domain classification on the Amazon review dataset. The ensemble has better accuracy than the individual classifiers, therefore, in our experiments the final reported performance is the accuracy of the ensemble. The combination weights in the ensemble represent the contributions of individual classifiers toward classification accuracy. In our experiments, the maximum number of iterations ($iterMax$) is set to 30. It is observed that at the end of the iterative learning process, the target specific classifier is assigned more weight mass as compared to the classifier trained on the shared representation. On average, the weights for the two classifiers converge to $w^s = 0.22$ and $w^t = 0.78$ at the end of the iterative learning process. This further validates our assertion that the target specific features are more discriminative than the shared features in classifying target domain instances, which are efficiently captured by the proposed algorithm. Key observations and analysis from the experiments on different datasets is summarized below.

Table 3: Comparing the performance of individual classifiers and the ensemble for training on Books domain and test across different domains. C_s and C_t are applied on the test domain data before performing the iterating learning process.

SD → TD	C_s	C_t	Ensemble
B → D	63.1	34.8	72.1
B → E	64.5	39.1	75.8
B → K	68.4	42.3	76.2

Table 4: List some examples of domain specific discriminative features learned by the proposed algorithm on the Amazon review dataset.

Domain	Domain specific features
Books	*pictures_illustrations, more_detail, to_read*
DvDs	*Definite_buy, delivery_prompt*
Kitchen	*invaluable_resource, rust, delicious*
Electronics	*Bargain, Energy_saving, actually_use*

4.2.1 Results on the Amazon Review dataset

To study the effects of different components of the proposed algorithm, comprehensive experiments are performed on the Amazon review dataset[3].

1) Effect of learning target specific features: Results in Figure 3 show that iteratively learning target specific feature representation (slow transfer as opposed to one-shot transfer) yields better performance across different cross-domain classification tasks as compared to SCL, SFA (Pan et al., 2010)[4] and the baseline. Unlike SCL and SFA, the proposed approach uses shared and target specific feature representations for the cross-domain classification task. Table 4 illustrates some examples of the target specific discriminative features learned by the proposed algorithm that leads to enhanced performance. At 95% confidence, parametric t-test suggests that the proposed algorithm and SCL are significantly (statistically) different.

2) Effect of similarity on performance: It is observed that existing domain adaptation techniques enhance the accuracy for cross-domain classification, though, negative transfer exists in camou-

[1] We also compared our performance with sentiment sensitive thesaurus (SST) proposed by (Bollegala et al., 2013) and our algorithm outperformed on our protocol. However, we did not include comparative results because of difference in experimental protocol as SST is tailored for using multiple source domains and our protocol uses single source domain.

[2] Our implementation of SCL is used in this paper.

[3] Due to space restrictions, we show this analysis only on one dataset; however similar conclusions were drawn from other datasets as well.

[4] We directly compared our results with the performance reported in (Pan et al., 2010).

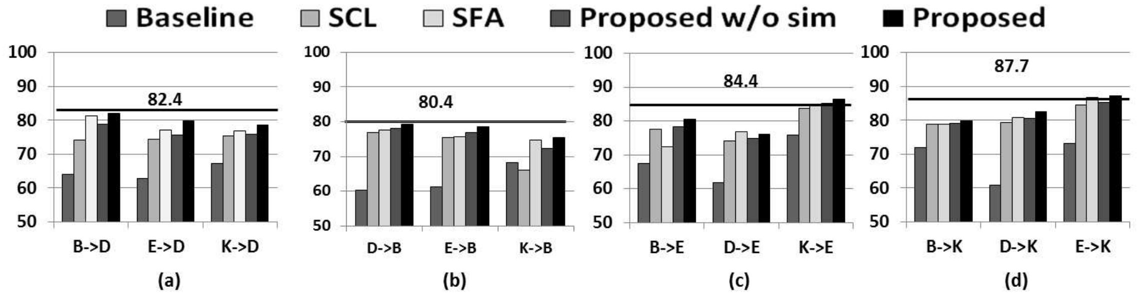

Figure 3: Comparing the performance of the proposed approach with existing techniques for cross-domain classification on Amazon review dataset.

flage. Results in Figure 3(b) (for the case K → B) describes an evident scenario for negative transfer where the adaptation performance with SCL descends lower than the baseline. However, the proposed algorithm still sustains the performance by transferring knowledge proportionate to similarity between the two domains. To further analyze the effect of similarity, we segregated the 12 cross-domain classification cases into two categories based on similarity between two the participating domains i.e. 1) > 0.5 and 2) < 0.5. Table 5 shows that for 6 out of 12 cases that fall in the first category, the average accuracy gain is 10.8% as compared to the baseline. While for the remaining 6 cases that fall in the second category, the average accuracy gain is 15.4% as compared to the baseline. This strongly elucidates that the proposed similarity-based iterative algorithm not only adapts well when the domain similarity is high but also yields gain in the accuracy when the domains are largely dissimilar. Figure 4 also shows how weight for the target domain classifier w_t varies with the number of iterations. It further strengthens our assertion that if domains are similar, algorithm can readily adapt and converges in a few iterations. On the other hand for dissimilar domains, slow iterative transfer, as opposed to one-shot transfer, can achieve similar performance; however, it may take more iterations to converge.While the effect of similarity on domain adaptation performance is evident, this work opens possibilities for further investigations.

Table 5: Effect of similarity on accuracy gain for cross-domain classification on the Amazon review dataset.

Category	SD → TD	Sim	Gain	Avg. (SD)
> 0.5	E → K	0.78	13.1	10.8 (4.9)
	K → E	0.78	10.6	
	B → K	0.54	8.0	
	K → B	0.54	2.9	
	B → E	0.52	13.1	
	E → B	0.52	17.2	
< 0.5	K → D	0.34	8.9	15.4 (4.4)
	D → K	0.34	21.6	
	E → D	0.33	14.5	
	D → E	0.33	14.5	
	B → D	0.29	14.1	
	D → B	0.29	19.1	

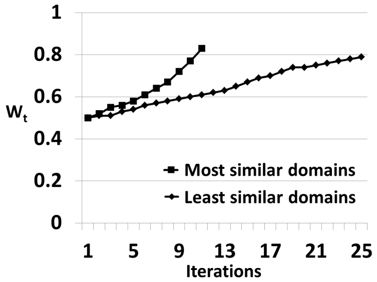

Figure 4: Illustrates how the weight (w_t) for target domain classifiers varies for the most and least similar domains with number of iterations.

3) Effect of varying threshold θ_1 ***&*** θ_2**:** Figure 5(a) explains the effect of varying θ_1 on the final classification accuracy. If θ_1 is low, C_t may get trained on incorrectly predicted pseudo labeled instances; whereas, if θ_1 is high, C_t may be deficient of instances to learn a good decision boundary. On the other hand, θ_2 influences the number of iterations required by the algorithm to reach the stopping criteria. If this threshold is low, the algorithm converges aggressively (in a few iterations) and does not benefit from the iterative nature of learning the target specific features. Whereas a high threshold tends to make the algorithm conservative. It hampers the accuracy because of the unavailability of sufficient instances to update the classifier after each iteration which also leads to large number of iterations to converge (may not even converge).

θ_1 and θ_2 are set empirically on a held-out set, with values ranging from zero to distance of farthest classified instance from the SVM hyperplane (Hsu et al., 2003). The *knee-shaped* curve on the graphs in Figure 5 shows that there exists

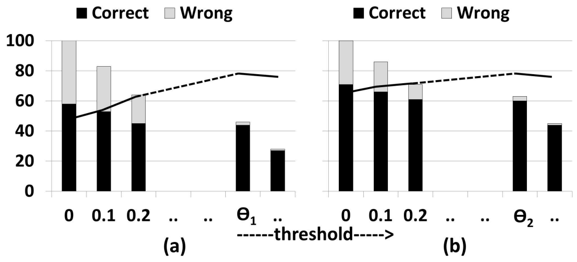

Figure 5: Bar plot shows % of data that crosses confidence threshold, lower and upper part of the bar represents % correctly and wrongly predicted pseudo labels. The black line shows how the final classification accuracy is effected with threshold.

an optimal value for θ_1 and θ_2 which yields the best accuracy. We observed that the best accuracy is obtained when the thresholds are set to the distance between the hyper plane and the farthest support vector in each class.

***4) Effect of using different shared representations in ensemble*:** To study the generalization ability of the proposed algorithm to different shared representations, experiments are performed using three different shared representations on the Amazon review dataset. Apart from using the SCL representation, the accuracy is compared with the proposed algorithm using two other representations, 1) common features between the two domains ("common") and 2) multi-view principal component analysis based representation ("MVPCA") (Ji et al., 2011) as they are previously used for cross-domain sentiment classification on the same dataset. Table 6 shows that the proposed algorithm yields significant gains in cross-domain classification accuracy with all three representations and is not restricted to any specific representation. The final accuracy depends on the initial classifier trained on the shared representation; therefore, if a shared representation sufficiently captures the characteristics of both source and target domains, the proposed algorithm can be built on any such representation for enhanced cross-domain classification accuracy.

4.2.2 Results on 20 Newsgroups data

Results in Figure 6 compares the accuracy of proposed algorithm with existing approaches on the 20 Newsgroups dataset. Since different domain are crafted out from the sub-categories of the same dataset, domains are exceedingly similar and therefore, the baseline accuracy is relatively better than that on the other two datasets. The proposed algorithm still yields an improvement of at least 10.8% over the baseline accuracy. As compared to other existing domain adaptation approaches like SCL(Blitzer et al., 2007) and CoCC (Dai et al., 2007), the proposed algorithm outperforms by at least 4% and 1.9% respectively. This also validates our assertion that generally domain adaptation techniques accomplishes well when the participating domains are largely similar; however, the similarity aggregation and the iterative learning offer the proposed algorithm an edge over one-shot adaptation algorithms.

Table 6: Comparing the accuracy of proposed algorithm built on different shared representations.

SD → TD	**Common**	**MVPCA**	**SCL**
B → D	66.8	76.4	**78.2**
B → E	69.0	79.2	**80.6**
B → K	71.4	79.2	**79.8**
D → B	64.5	78.4	**79.3**
D → E	62.8	**76.4**	76.2
D → K	64.3	80.9	**82.4**
E → B	68.9	77.8	**78.5**
E → D	65.7	77.0	**77.3**
E → K	75.1	85.4	**86.2**
K → B	**71.3**	71.0	71.1
K → D	70.4	75.0	**76.1**
K → E	76.7	85.7	**86.4**

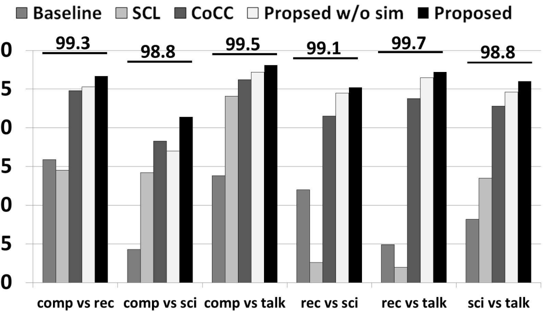

Figure 6: Results comparing the accuracy of proposed approach with existing techniques for cross domain categorization on 20 Newsgroups dataset.

4.2.3 Results on real world data

Results in Figure 7 exhibit challenges associated with real world dataset. The baseline accuracy for cross-domain classification task is severely affected for this dataset. SCL based domain adaptation does not yields generous improvements as selecting the pivot features and computing the co-occurrence statistics with noisy short text is arduous and inept. On the other hand, the proposed algorithm iteratively learns discriminative target specific features from such perplexing data and translates it to an improvement of at least 6.4% and 3.5% over the baseline and the SCL respec-

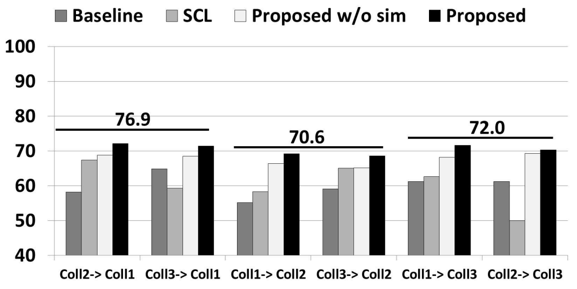

Figure 7: Results comparing the accuracy of the proposed approach with existing techniques for cross domain categorization on the real world dataset.

tively.

5 Conclusion

The paper presents an iterative similarity-aware domain adaptation algorithm that progressively learns domain specific features from the unlabeled test domain data starting with a shared feature representation. In each iteration, the proposed algorithm assigns pseudo labels to the unlabeled data which are then used to update the constituent classifiers and their weights in the ensemble. Updating the target specific classifier in each iteration helps better learn the domain specific features and thus, results in enhanced cross-domain classification accuracy. Similarity between the two domains is aggregated while updating weights of the constituent classifiers which facilitates gradual shift of knowledge from the source to the target domain. Finally, experimental results for cross-domain classification on different datasets show the efficacy of the proposed algorithm as compared to other existing approaches.

References

A. Aue and M. Gamon. 2005. Customizing sentiment classifiers to new domains: A case study. *Technical report, Microsoft Research.*

J. Blitzer, R. McDonald, and F. Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 120–128.

J. Blitzer, M. Dredze, and F. Pereira. 2007. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of Association for Computational Linguistics*, pages 187–205.

D. Bollegala, D. Weir, and J. Carroll. 2011. Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification. In *Proceedings of Association for Computational Linguistics: Human Language Technologies*, pages 132–141.

D. Bollegala, D. Weir, and J. Carroll. 2013. Cross-domain sentiment classification using a sentiment sensitive thesaurus. *IEEE Transactions on Knowledge and Data Engineering*, 25(8):1719–1731.

M. Chen, K. Q Weinberger, and J. Blitzer. 2011. Co-training for domain adaptation. In *Proceedings of Advances in Neural Information Processing Systems*, pages 2456–2464.

Si-Chi Chin. 2013. Knowledge transfer: what, how, and why.

W Dai, G-R Xue, Q Yang, and Y Yu. 2007. Co-clustering based classification for out-of-domain documents. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, pages 210–219.

Hal Daumé III. 2009. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*.

I. S. Dhillon, S. Mallela, and D. S Modha. 2003. Information-theoretic co-clustering. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, pages 89–98.

A. Garcia-Fernandez, O. Ferret, and M. Dinarelli. 2014. Evaluation of different strategies for domain adaptation in opinion mining. In *Proceedings of the International Conference on Language Resources and Evaluation*, pages 26–31.

C-W. Hsu, C.-C. Chang, and C.-J. Lin. 2003. A practical guide to support vector classification. Technical report, Department of Computer Science, National Taiwan University.

Y.-S. Ji, J.-J. Chen, G. Niu, L. Shang, and X.-Y. Dai. 2011. Transfer learning via multi-view principal component analysis. *Journal of Computer Science and Technology*, 26(1):81–98.

J. Jiang and C. Zhai. 2007. Instance weighting for domain adaptation in NLP. In *Proceedings of Association for Computational Linguistics*, volume 7, pages 264–271.

K Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of International Conference on Machine Learning*.

X. Liao, Y. Xue, and L. Carin. 2005. Logistic regression with an auxiliary data source. In *Proceedings of International Conference on Machine Learning*, pages 505–512.

C. Luo, Y. Ji, X. Dai, and J. Chen. 2012. Active learning with transfer learning. In *Proceedings of Association for Computational Linguistics Student Research Workshop*, pages 13–18. Association for Computational Linguistics.

S. J. Pan and Q. Yang. 2010a. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Sinno Jialin Pan and Qiang Yang. 2010b. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359.

Sinno Jialin Pan, James T Kwok, and Qiang Yang. 2008. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, pages 677–682.

S. J. Pan, X. Ni, J-T Sun, Q. Yang, and Z. Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings International Conference on World Wide Web*, pages 751–760. ACM.

S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. 2011. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210.

M. T. Rosenstein, Z. Marx, L. P. Kaelbling, and T. G. Dietterich. 2005. To transfer or not to transfer. In *Proceedings of Advances in Neural Information Processing Systems Workshop, Inductive Transfer: 10 Years Later*.

A. Saha, P. Rai, H. Daumé, S. Venkatasubramanian, and S. L. DuVall. 2011. Active supervised domain adaptation. In *Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 97–112.

R. Xia, C. Zong, X. Hu, and E. Cambria. 2013. Feature ensemble plus sample selection: domain adaptation for sentiment classification. *IEEE Intelligent Systems*, 28(3):10–18.

P. Zhao and S. C. H. Hoi. 2010. OTL: A Framework of Online Transfer Learning. In *Proceeding of International Conference on Machine Learning*.