TL-PLSA: Transfer learning between domains with different classes

Anastasia Krithara and Georgios Paliouras
Software and Knowledge Engineering Laboratory,
Institute of Informatics and Telecommunications,
National Center for Scientific Research (NCSR) "Demokritos"
Athens, Greece
Email: {akrithara,paliourg}@iit.demokritos.gr

***Abstract*—A new transfer learning method is presented in this paper, addressing a particularly hard transfer learning problem: the case where the target domain shares only a subset of its classes with the source domain and only unlabeled data are provided for the target domain. This is a situation that occurs frequently in real-world applications, such as the multiclass document classification problems that motivated our work. The proposed approach is a transfer learning variant of the Probabilistic Latent Semantic Analysis (PLSA) model [1] that we name TL-PLSA. Unlike most approaches in the literature, TL-PLSA captures both the difference of the domains and the commonalities of the class sets, given no labelled data from the target domain. We perform experiments over three different datasets and show the difficulty of the task, as well as the promising results that we obtained with the new method.**

***Keywords*-multiclass classification; transfer learning; PLSA;**

I. Introduction

Machine learning technologies have already achieved significant success in many knowledge engineering areas including classification, regression and clustering. However, many machine learning methods work well only under a common assumption: the training and test data are drawn from the same distribution. When the distribution changes most statistical models need to be rebuilt from scratch using newly collected training data. In many real world applications, it is expensive or impossible to collect the needed training data and rebuild the models. Knowledge transfer would greatly improve the performance of learning by avoiding expensive data-labeling efforts. In recent years, *transfer learning* has emerged as a new learning framework to address this problem. It tries to extract knowledge from previous experience and apply it on new learning domains or tasks.

As an example, we may want to learn a classifier for web blog posts, in order to categorize them into different themes, but we may not have such documents pre-annotated with appropriate categories. However, we may be given plenty of news articles that are categorized as a source. Hand labeling data in the new domain is costly, and one often wishes to be able to leverage the original, "out-of-domain" data when building a model for the new domain.

Existing transfer learning approaches can be categorized into three main types [2], based on the characteristics of the source and target domains and tasks:

1) *Inductive transfer:* The target task is different from the source task and some labeled data in the target domain are required. For document classification, two tasks are considered different if either the label sets are different in the two domains, or the source and target documents are very imbalanced in terms of user-defined classes. Depending on the availability of labeled data in the source domain, we distinguish two subcategories:
   - Labeled data in the source domain are available. This setting is similar to multitask learning.
   - No labeled data in the source domain are available. This setting is similar to self-taught learning.

   Most existing approaches of this type focus on the former subcategory.
2) *Transductive transfer learning setting:* The source and target tasks are the same, while the source and target domains differ. For document classification, two domains are considered different if either the term features are different, or their marginal distributions are different. No labeled data for the target domain are available, while labeled data are available for the source domain.
3) *Unsupervised transfer learning:* Similar to inductive transfer learning, the target task is different from but related to the source task. However, the unsupervised transfer learning focuses on solving unsupervised learning tasks in the target domain, such as clustering. There are no labeled data available in either the source or the target domains.

In this work, we propose a new approach, which lies between the first two categories (Inductive and Transductive transfer learning). In our setting, no labeled data for the target domain are available, as in transductive transfer learning, while the source and target tasks are also not exactly the same, as in inductive transfer learning. In particular, we assume that we have labeled data for the source domain and unlabeled data for the target domain. Additionally, the two tasks are multiclass classification ones and they *share only a subset of common classes*. This is a situation that occurs frequently in real-world applications, such as in document classification problems. In many cases, as we mentioned before, it is very hard to find training data in a particular domain we are interested in. At the same time, it also unlikely to find training data from different domains which are classified to the exact same classification schema we want to train our model to (i.e. exactly the same classes). With the current transfer learning approaches, this problem cannot be faced. The approach is an extension of the PLSA method [1], and it is based on previous work on Dual-PLSA [3].

The rest of the paper is organized as follows: In section II, we present related work, focusing on the types of transfer learning that are most relevant for our work. In section III, we present the preliminaries and the problem formulation and in section IV the proposed approach is presented. Then, in section V we evaluate the proposed approach and present experimental results. Finally, section VI concludes this paper and presents some future directions.

II. Related Work

Despite its importance, the transfer learning problem only gained sufficient attention in the machine learning community recently. There have been a number of studies on solving specific transfer learning problems or addressing the problem from various perspectives. However, transfer learning is not yet completely understood, and there are no dominating methods that are used widely. Below, we present some of the approaches in the literature which fall into the two first types of transfer learning, that are most relevant to our work.

Inductive transfer learning: TrAdaBoost [4] is an extension of the AdaBoost algorithm. TrAdaBoost assumes that the source and target domain data use exactly the same set of features and labels, but the conditional probability distributions between the domains are different. It also assumes that there are labeled data in both source and target domain data. It attempts to iteratively reweight the source domain data to reduce the effect of the "bad" source data while encouraging the "good" source data to contribute more for the target domain. In the same vein, a heuristic method was proposed in [5], in order to remove "misleading" training examples from the source domain based on the difference between conditional probabilities between domains. Some approaches in inductive transfer learning, try to find new feature representations in order to minimize domain divergence. For example, in [6], a convex optimization algorithm for this scope is presented. The idea is to simultaneously learn metapriors and feature weights from an ensemble of related prediction tasks. The metapriors can be transferred among different tasks. Another method [7] uses features from source and target domains to construct an augmented feature space. However, despite its simplicity, a formal theoretical analysis is clearly missing. Some other approaches try to take advantage of the labeled data from the target domain using active learning techniques. To this end, [8] proposed a method where active learning is used for word sense disambiguation in a transfer learning setting. Their active learning setting is pool-based whereas [9], propose a similar method but in a streaming (online) setting, as a result there is not the requirement of an initial pool of labeled target domain. Nevertheless, both methods are applicable when labeled data exist also in the target domain.

Transductive transfer learning: Many approaches of this type are motivated by importance sampling. Their motivation is to add weights to instances, using the probability density ratio (i.e. the difference of the source and target distributions). For example, kernel-mean matching (KMM) algorithm is proposed in [10], to learn directly the density ration, by matching the means between the source and the target domain data in a reproducing-kernel Hilbert space (RKHS). In the same vein, an algorithm known as Kullback-Leibler Importance Estimation Procedure (KLIEP) is proposed in [11], in order to estimate the difference of the source and target distributions directly, based on the minimization of the Kullback-Leibler divergence. In [5], the proposed approach uses instance weighting, by adding instance-dependent weights to the loss function. Another family of transductive transfer learning approaches are the feature-representation-transfer ones. For example, a structural correspondence learning (SCL) algorithm is proposed in [12], to make use of the unlabeled data from the target domain and extract some relevant features that may reduce the difference between the domains. The effect of representation change for domain adaptation is also analyzed in [13]. Also, a co-clustering based approach is presented in [14], aiming to propagate the class information from the target to source domain, by identifying word clusters shared among the two domains. Transfer learning via dimensionality reduction was proposed in [15]. They exploited the Maximum Mean Discrepancy Embedding (MMDE) method, originally designed for dimensionality reduction, to learn a low-dimensional space that reduces the difference of distributions between different domains. However, MMDE has been proved computationally expensive. Thus, in [16], Transfer Component Analysis (TCA) is proposed, which uses an efficient feature exctration algorithm. In [17], an approach based on a mixture model is presented. Their key idea is to assume that source domain data are drawn from a mixture of two distributions: a truly "in-domain" distribution and a "general domain" one. Similarly, the target domain data is treated as if drawn from a mixture of "out-of-domain" distribution and the "general domain" distribution, as the source domain data.

III. PLSA-based Transfer Learning

The works that are most related to ours are presented in [18] and [3]. They both belong to the transductive transfer learning setting and extend the Probabilistic Semantic Analysis (PLSA) model [1]. Before giving more details about them, we first describe the notations used in this paper, and briefly review PLSA.

Notations

A domain D consists of two components: a feature space $\mathcal{F}$ and a marginal probability distribution $P(\mathcal{X})$, where $\mathcal{X} = x_1, x_2, \ldots, x_n \in \mathcal{X}$. As our task is multiclass text classification, $\mathcal{X}$ is the space of all term vectors (i.e. all documents), and x_i is the i^{th} term vector corresponding to some document. In the transfer learning problem, we suppose that we have two domains, namely the source (D_S) and the target (D_T) domains. Respectively, we consider a common set of features $\mathcal{F}$, and two sets of documents, $\mathcal{X}_S$ and $\mathcal{X}_T$, for the source and target domains respectively.

Our focus is on a particularly hard transfer learning problem in which the two tasks are multiclass classification ones and they *share only a subset of classes*. To this end, we assume that each domain (source and target) has a set of classes in which we want to classify the documents. We denote with $\mathcal{C} = c_1, c_2, \ldots, c_j$ the set of shared classes between the two domains, with $\mathcal{C}_S = c_{S_1}, c_{S_2}, \ldots, c_{S_m}$ the classes of the source domain not in C and with $\mathcal{C}_T = c_{T_1}, c_{T_2}, \ldots, c_{T_l}$ the classes of the target domain not in C (fig. 1).

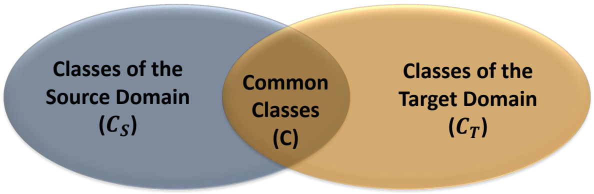
Classes of the Source Domain (C_S)
Common Classes (C)
Classes of the Target Domain (C_T)

Figure 1. Only a subset of classes (C) are shared between the source and the target domain.

A. PLSA

Probabilistic Latent Semantic Analysis (PLSA) [1] is a probabilistic model which characterizes each word in a document as a sample from a mixture model, where mixture components are conditionally-independent multinomial distributions. It has been proposed as a probabilistic version of the Latent Semantic Analysis (LSA) method [19]. This model associates an unobserved latent variable (called aspect, topic or component) $k \in \{k_1, ..., k_K\}$ to each observation corresponding to the occurrence of a word $f \in \mathcal{F}$ within a document $x \in \mathcal{X}$. One component or topic can coincide with one class or, in another setting, a class may be associated with more than one component. PLSA models the co-occurrence matrix, whose elements represent the frequency of word f appearing in document x_i by using a mixture model of latent topics (each topic is denoted by k) as follows:

$$P(f, x) = \sum_{k \in K} P(f \mid k) P(x \mid k) P(k) \qquad (1)$$

Figure 2 shows the graphical model for PLSA. The parameters of $P(f \mid k)$, $P(x \mid k)$, and $P(k)$ over all f, x, k are obtained by EM estimation of the maximum likelihood.

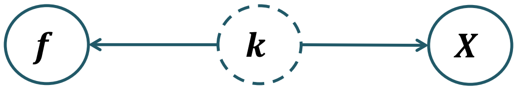

Figure 2. Graphical model representation of PLSA. Latent variables are indicated by dotted circles.

B. Dual-PLSA

In the PLSA model, documents and words share the same latent topic k. However, documents and words usually exhibit different organization and structure. Specifically, they may have different kinds of latent topics, denoted by k for a word topic and z for a document topic. To this end, an extension of PLSA has been proposed in [20]. They presented a framework, namely the probabilistic matrix tri-factorization, where they consider two latent variables for PLSA. Its graphical model is shown in Figure 3.

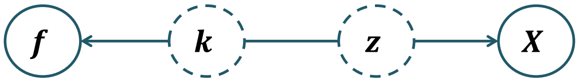

Figure 3. Graphical model representation of Dual-PLSA. Latent variables are indicated by dotted circles.

Given the word-document co-occurrence matrix, we obtain a similar mixture model like in equation 1,

$$P(f, x) = \sum_{k,z} P(f, x, k, z) = \sum_{k,z} P(f \mid k) P(x \mid z) P(k, z) \qquad (2)$$

The parameters of $P(f \mid k)$, $P(x \mid z)$, $P(k, z)$ over all f, x, k, z can also be estimated by EM.

This model was proposed in [20] for the clustering problem. Since the word topic and document topic are separated in this model, the label information can be injected into $P(x|z)$, where x is a labeled instance and z is actually a document class. This way this model can also be used for semi-supervised classification.

C. *Transfer learning approaches*

This approach has been extended in [3], under the name *Collaborative Dual-PLSA*, for document classification in a transfer learning setting. The intuition behind this approach is that the association between word topics and document topics is usually stable across domains. The method tries to capture the commonalities and differences of the topics across the different domains. Their model has two latent variables for word topics and document topics, as Dual-PLSA [20]. It also introduce an additional variable, which represents the different data domains, as their focus is the transfer learning across multiple domains.

A different extension of PLSA for transfer learning, the so-called *topic-bridged PLSA*, has been proposed in [18]. The idea is to exploit the common underlying topics between two domains, and transfer knowledge across them through a topic-bridge. Specifically, they conduct two topic modelings over the source and target domains jointly, and induce the supervision of the labeled source domain data by the pair-wise constraints, similar to the must-link and cannot-link constraints used in semi-supervised clustering. The method has been applied to text classification.

Our approach extends the work in [20], in a significant way. Our model has three latent variables, and our focus is on a particularly hard transfer learning problem which, to our knowledge, has not been studied in the literature so far: the case where the target domain shares only *a subset of its classes* with the source domain and only unlabeled data are provided for the target domain.

IV. TL-PLSA

We propose an extension of the Dual-PLSA [20] towards a different setting: the case where the target domain shares only *a subset of its classes* with the source domain and only unlabeled data are provided for the target domain. This is a particularly hard setting, that, to our knowledge, is not addressed by of the current transfer learning approaches.

Since our model has two latent variables, for word topic and document topic, it can naturally include the supervision from the source domain. In addition, in our approach we assume that we have different document topics in the two domains. As a result, we use two parameters to represent the document topics, one for the source domain and one for the target. Assuming that the document topics correspond to document classes, there are some shared document topics between the two domains (z_{ST}), and some unshared ones (z_S for the source and z_T for the target). In order to determine which are the shared ones, after training the model, we calculate the Kullback-Leibler (KL) divergence [21] between the document topics in each domain. The topics with the smaller divergence are chosen as the shared one.

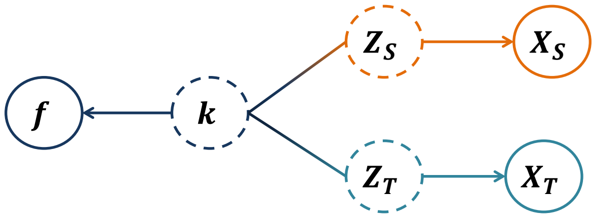

Figure 4. Graphical model representation of the TL-PLSA model. Latent variables are indicated by dotted circles.

At this point, it is worth stressing that the only information the algorithm needs is the *number* of common classes and the *number* of source and target domain classes. For example, the algorithm needs to know that there are 10 common classes, 4 additional classes in the source domain, and 6 in the target domain. The algorithm identifies which of the classes are the shared ones and which are not.

Hence, the model parameters are:

$$\Xi = \{P(x \mid z), P(f \mid k), P(k, z)) : z \in Z, k \in K, x \in \mathcal{X}, f \in \mathcal{F}\} \tag{3}$$

The above model parameters Ξ are obtained by maximizing the complete data log-likelihood, using the EM algorithm:

$$\begin{aligned}\mathcal{L} &= \sum_f \sum_{x_S} n(f, x_S) \log \sum_k \sum_{z_S} P(x_S|z_S)P(f|k)P(k, z_S) \\ &+ \sum_f \sum_{x_T} n(f, x_T) \log \sum_k \sum_{z_T} P(x_T|z_T)P(f|k)P(k, z_T)\end{aligned} \tag{4}$$

where $n(f, x)$ denotes the frequency of the word f in document x. The training procedure of this model using EM is described in algorithm (1).

We can show that the maximum likelihood estimates of the model parameters with respect to 4 are:

$$P^{(j+1)}(x|z) \propto \begin{cases} \sum_f \sum_k n(f, x_S)P^{(j)}(k, z_S|f, x_S), \text{ for } x \in X_S \\ \\ \sum_f \sum_k n(f, x_T)P^{(j)}(k, z_T|f, x_T), \text{ for } x \in X_T \end{cases} \tag{5}$$

Algorithm 1 Training of TL-PLSA for multiclass text classification

```
Input:
  • Data from source and target domains $\mathcal{X}_S$ and $\mathcal{X}_T$,
  • Random initial model parameters $\Xi^{(0)}$.
  • $j \leftarrow 0$
repeat
  • E-step: Estimate the latent class posteriors (equation (8))
  • M-step: Estimate the new model parameters $\Xi^{(j+1)}$
    by maximizing the complete-data log-likelihood (equations
    (5), (6) and (7))
  • $j \leftarrow j + 1$
until convergence of the complete-data log-likelihood (4)
Output: A generative classifier with parameters $\Xi^{(j)}$
```

$$P^{(j+1)}(f|k) \propto \sum_{X_S}\sum_{z_S} n(f,x_S)P^{(j)}(k,z_S|f,x_S) + \sum_{X_T}\sum_{z_T} n(f,x_T)P^{(j)}(k,z_T|f,x_T) \quad (6)$$

$$P^{(j+1)}(k,z) \propto \sum_{f}\sum_{X_S} n(f,x_S)P^{(j)}(k,z_S|f,x_S) + \sum_{f}\sum_{X_T} n(f,x_T)P^{(j)}(k,z_T|f,x_T) \quad (7)$$

where

$$P(k,z|f,x) = \frac{P(x \mid z)P(f \mid k)P(k,z)}{\sum\limits_{k' \in K, z' \in Z} P(x \mid z')P(f \mid k')P(k',z')} \quad (8)$$

At the initialization of the model ($\Xi^{(0)}$), the $P(x_S \mid z_S)$ is fixed as we know the classes of the source domain data. All the rest are initialized with random values.

In order to find the shared classes that we are interested in, we estimate the KL divergence between the resulted document topics z_S and z_T. As the number of shared classes C is known, we choose the C document topics z_T which have the smaller divergence compared with z_S.

In particular:

$$d_{z_T|z_S} = argmin \sum_{z_T}\sum_{z_S} \mathrm{KL}(P(k,z_T) \| P(k,z_S)) \quad (9)$$

where the KL divergence is calculated as follows:

$$\mathrm{KL}[P(k,z_T) \| P(k,z_S)] = \sum_{k} P(k,z_T) \log \frac{P(k,z_T)}{P(k,z_S)} \quad (10)$$

Once the model is trained, we then classify the documents in one of the classes using chain rule:

$$P(z \mid x) \propto P(x \mid z)P(z) = P(x \mid z)\sum_{k} P(k,z) \quad (11)$$

We choose as label for each document, the one with the highest probability, taking into account that there is a one-to-one matching between document topics z and classes:

$$argmax_z P(z \mid x) \quad (12)$$

In addition, as our model is a generative one, we can run TL-PLSA for new documents (x_{new}) from the target domain, using the calculated model (i.e. $P(f|k)$), in order to learn the $P(x_{new}|z)$.

It is worth mentioning, that the equation 12 can be applied also for the non-shared classes, as the produced model, has learned their parameters.

V. Experiments

In order to evaluate the algorithm proposed in the previous section, we performed experiments on three different datasets. We first describe these datasets (section V-A) and the evaluation measures (section V-B), then we present the results we obtained in section V-C.

For all three datasets, we ran four algorithms, in order to verify, how our approach behaves in comparison to non-transfer learning approaches, and to another state-of-the-art transfer learning approach:

- non-transfer learning approaches:
  - PLSA [1]
  - Dual-PLSA [20]
- transfer learning approaches:
  - Collaborative Dual-PLSA [3]
  - TL-PLSA (section IV)

A. Datasets

In our experiments, we used the following three datasets:

***20Newsgroups**:* is a text collection of approximately $20,000$ newsgroup documents, partitioned across 20 different newsgroups nearly evenly. Since this dataset is not originally designed for evaluating cross-domain classification, we pre-processed the original data as follows. First, we observed that the dataset has a hierarchical structure. In particular, it contains six top categories. Under the 6 top categories, there are 20 sub-categories (see table I). We focused on the task of classifying documents into top-level categories. In particular, we conducted experiments, in which only some of the classes were shared (e.g. Computers and Recreation) and the others contained documents from only one of the domains (e.g. Science and Religion contain only source documents, and Politics and Misc only target documents). Different configuration of the classes were tested, and 10-fold cross validation was used, to obtain the average performance of the method.

Computers	**Recreation**	**Science**	**Politics**	**Misc**	**Religion**
comp.graphics	rec.autos	sci.crypt	talk.politics.misc	misc.forsale	alt.atheism
comp.os.ms-windows.misc	rec.motorcycles	sci.electronics	talk.politics.guns		soc.religion.christian
comp.sys.ibm.pc.hardware	rec.sport.baseball	sci.med	talk.politics.mideast		talk.religion.misc
comp.sys.mac.hardware	rec.sport.hockey	sci.space			
comp.windows.x					

Table I
20NEWSGROUPS DATASET: CATEGORIES AND SUB-CATEGORIES

Dataset	20Newsgroups	SYNC3	LSHTC
Source domain size	10161	1089	2297
Target domain size	8613	864	1917
# of shared classes	6	102	1044
Vocabulary size, $\|\mathcal{F}\|$	61188	4092	74082

Table II
CHARACTERISTICS OF THE DATASETS. THE # OF SHARED CLASSES CORRESPONDS TO THE MAXIMUM POSSIBLE (100% OVERLAP) SHARED CLASSES BETWEEN THE DOMAINS. IN THE EXPERIMENTS, THIS VALUE VARIES ACCORDING TO THE PERCENTAGES OF SHARED CLASSES.

SYNC3 gold corpus: is a manually annotated News and Blogs corpus. This corpus was actually the main motivation of our work. The experiments reported here used the version released by the SYNC3 project[1] in October 2011 and contains 864 blogs and 1089 news items, categorized into 102 events. Similar to 20Newsgroups, we performed different experiments, by considering different percentages of shared classes. For example, in one of the experiments we considered that 20% of the classes contain only news items, another 20% only blog posts, and the rest 60% of the classes contain both news items and blog posts. In order to keep the number of blogs and news stable, we added to the dataset irrelevant documents from other classes (different from the 102 ones).

LSHTC: is a dataset provided by the 1st edition of the Large Scale Hierarchical Text Classification (LSHTC) Pascal Challenge[2]. The LSHTC Challenge is a hierarchical text classification competition, using large datasets. The data have been constructed by:

- crawling Web pages that are found in the ODP and translating them into feature vectors (content vectors),
- translating into feature vectors the ODP descriptions of Web pages and ODP categories (Web page and category description vectors).

Two datasets were provided by the challenge: a large one (12294 categories) and a smaller one (1139 categories). As in the case of 20Newsgroups, the dataset was preprocessed in order to create different domains and different percentages of shared classes between the domains.

Table II summarizes the characteristics of these datasets.

B. Evaluation

In order to evaluate the performance of the various algorithms, we used the microaveraged F-score measure. For each classifier, $\mathcal{G}_f$, we first compute its microaverage precision P and recall R by summing over all the individual decisions it made on the test set:

$$r(\mathcal{G}_f) = \frac{\sum_{k=1}^{K} \theta(k, \mathcal{G}_f)}{\sum_{k=1}^{K} (\theta(k, \mathcal{G}_f) + \psi(k, \mathcal{G}_f))}$$

$$p(\mathcal{G}_f) = \frac{\sum_{k=1}^{K} \theta(k, \mathcal{G}_f)}{\sum_{k=1}^{K} (\theta(k, \mathcal{G}_f) + \phi(k, \mathcal{G}_f))}$$

where $\theta(k, \mathcal{G}_f)$, $\phi(k, \mathcal{G}_f)$ and $\psi(k, \mathcal{G}_f)$ respectively denote the true positive, false positive and false negative documents in class k found by $\mathcal{G}_f$. The F-score measure is then defined as [22]:

$$F(\mathcal{G}_f) = \frac{2p(\mathcal{G}_f)r(\mathcal{G}_f)}{p(\mathcal{G}_f) + r(\mathcal{G}_f)}$$

C. Results

We compared the performance of the models on the three datasets by varying the percentage of shared classes between the domains and using 10-fold cross validation (CV). We performed 10 runs for each of the folds and we calculated the average F-score. As we initialize some of the training hyper-parameters at random, we wanted to get representative performance for multiple random initializations. In order to evaluate the significance of the observed differences in performance, we performed a t-test at the 5% significance level.

Please note, that it is the algorithm that identifies which of the classes are the shared ones and which are not. Only the number of unshared classes is known. As a result, in our experiments we are interested in the performance of TL-PLSA for the shared classes only. This decision was also taken due to the problem we are focusing on, which was motivated by the SYNC3 dataset. In other words, we are not evaluating the classification results of all target domain data, but only of the ones that belong to the shared classes. As mentioned in the previous subsection, in order to keep the number of source and target domain data stable as we reduce the number of shared classes, we add to the dataset irrelevant documents from other classes (for which we do not have the labels, and as a result, cannot evaluate).

[1] http://www.sync3.eu/

[2] $http : //lshtc.iit.demokritos.gr/$

20 Newsgroups

% of common classes	PLSA			TL-PLSA			Difference
	Precision	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**	
5	0.6713	0.6713	0.6713	0.6709	0.6709	0.6709	0.0%
4	0.5481	0.6105	0.5776	0.5614	0.6015	0.5808	0.3%
3	0.5012	0.6040	0.5478	0.5150	0.6132	0.5598	1.2%
2	0.4813	0.5821	0.5269	0.4993	0.5921	0.5418	1.5%
1	0.4315	0.5646	0.4892	0.4551	0.5701	0.5062	1.7%

SYNC3

% of common classes	PLSA			TL-PLSA			Difference
	Precision	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**	
100	0.7831	0.7831	0.7831	0.7842	0.7842	0.7842	0.1%
80	0.7228	0.7701	0.7454	0.7332	0.7745	0.7533	0.8%
60	0.6629	0.7823	0.7163	**0.7021**	0.7733	**0.7360**	**2.0%**
40	0.6115	0.7660	0.6801	**0.7055**	0.7402	**0.7224**	**4.2%**
30	0.5822	0.7517	0.6562	**0.6812**	0.7208	**0.7004**	**4.4%**
20	0.5409	0.7475	0.6276	**0.6385**	0.7367	**0.6841**	**5.6%**
10	0.5309	0.7160	0.6097	**0.6031**	0.6921	**0.6445**	**3.5%**

LSHTC

% of common classes	PLSA			TL-PLSA			Difference
	Precision	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**	
100	0.8192	0.8192	0.8192	0.8194	0.8194	0.8194	0.0%
80	0.7458	0.7994	0.7717	0.7580	0.8082	0.7823	1.1%
60	0.6654	0.7728	0.7144	**0.7299**	0.7668	**0.7454**	**3.1%**
40	0.5904	0.7413	0.6573	**0.6522**	0.7367	**0.6919**	**3.5%**
30	0.5457	0.7329	0.6256	**0.5794**	0.7394	**0.6497**	**2.4%**
20	0.4929	0.7461	0.5936	**0.5898**	0.7354	**0.6546**	**6.1%**
10	0.4881	0.7342	0.5632	**0.5343**	0.7411	**0.6209**	**5.8%**

Table III
Precision, Recall and F-Score for PLSA and TL-PLSA, for all three datasets. Dual-PLSA and Collaborative Dual-PLSA give the same results as the simple PLSA

It is worth mentioning though, that our approach is able to categorize all target domain data in all the target classes (shared and unshared).

Table III and figure 5 present the results. As a general comment, we can argue that the fewer shared classes the two domains have, the more difficult the classification problem is. In all three datasets, PLSA, Dual-PLSA and Collaborative Dual-PLSA give similar results, without statistically significant differences. For this reason, we present only the results of PLSA in III. In two out of the three datasets (SYNC3 and LSHTC), the proposed approach outperforms the other three. In the 20Newsgroups dataset there is small improvement compared with simple PLSA, but with no statistical difference.

For 20Newsgroups dataset we varied the number of shared classes between source and target domain, from 5 to 1. As the number of classes is small, the experiment was repeated for different shared classes and averaged. Our algorithm performs similarly to the other three algorithms in this dataset. This is probably due to the fact that the dataset has very few classes, with many documents in each class. Nevertheless, the use of TL-PLSA does not hurt the performance of PLSA.

The results for the SYNC3 dataset and LSHTC dataset show that the fewer classes that are shared between the source and target domains we have, the more our approach outperforms the other three. TL-PLSA outperforms the other three approaches, especially in terms of precision, when there is a large percentage of unshared classes

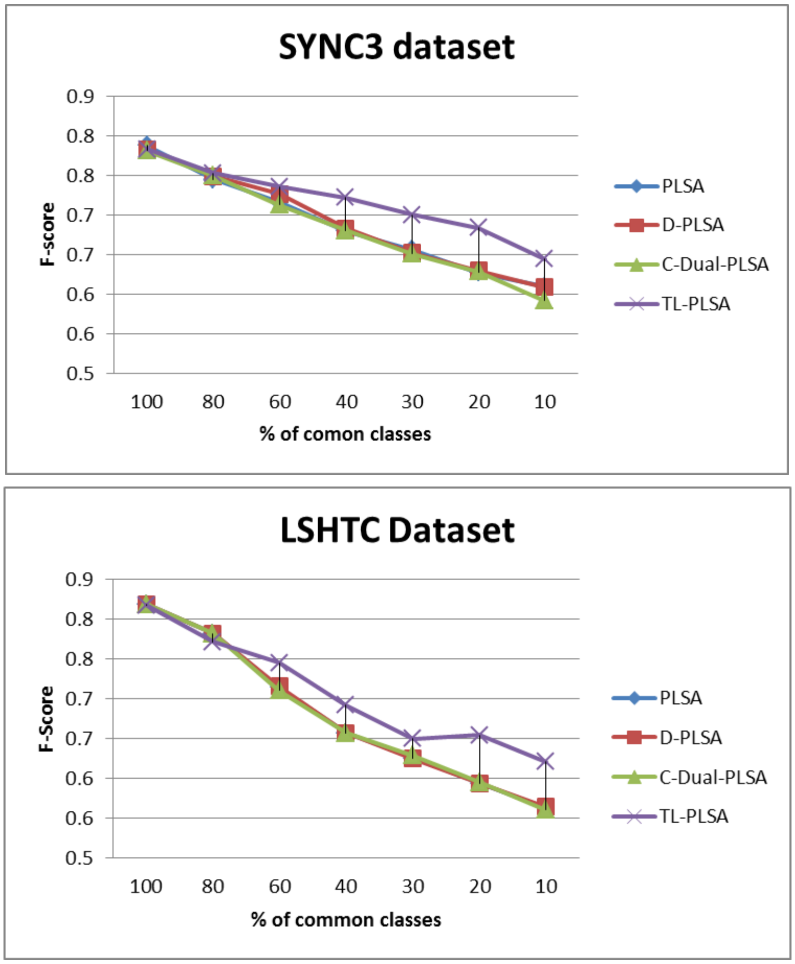

Figure 5. Results for all four algorithms, for different percentages of shared classes, for SYNC3 and LSHTC datasets. We do not provide the same diagram for 20Newsgroups as the results obtained are similar for all 4 algorithms.

between the two domains. This is also confirmed by the fact that our approach significantly outperforms the others (bold values in figure III), for 60% or less of shared classes.

Comparing the obtained results between the three datasets, we can notice that our approach in SYNC3 and LSHTC datasets achieves similar performance when reducing the percentage of shared classes. In the 20Newsgroups however, our approach faces more difficulties. This can be due to the fact that 20Newsgroups categories seem to be closer to each other, and as a result, the classifiers are not affected so much. The latter strengthen also our intuition, that TL-PLSA can learn the shared and unshared classes between domains, when few documents per class exist, given a large number of classes (as in the SYNC3 and LSHTC datasets).

Acknowledgments

The authors would like to acknowledge partial support of this work from the European Community Seventh Framework Programme, as part of the FP7 231854 SYNC3 project.

VI. Conclusions

In this paper, we presented TL-PLSA, a new approach to transfer learning, based on PLSA. The motivation for this work was to use transfer learning, when the source and target domain share only a subset of classes. This is a particularly hard setting that, up to our knowledge, has not yet been studied in the literature. We conducted experiments over three datasets. The evaluation shows the difficulty of the task, as well as the promising results achieved by the new method. Our approach outperforms both the simple PLSA and Dual-PLSA methods, as well as a transfer learning approach (Collaborative Dual-PLSA). TL-PLSA seems particularly effective for multiclass text classification tasks with a large number of classes (more than 100) and few documents per class. The performance of TL-PLSA is higher when the percentage of shared classes of source and target domain is smaller.

Our immediate next target is to extend TL-PLSA with a method for estimating the number of shared classes of the two domains. An idea towards this direction, is the Bayesian Information Criterion (BIC) [23], as it has been shown to give a good approximation of the number of clusters (i.e. classes) in PLSA.

References

[1] T. Hofmann, “Unsupervised learning by probabilistic latent semantic analysis,” *Mach. Learn.*, vol. 42, no. 1-2, pp. 177–196, 2001.

[2] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, October 2010.

[3] F. Zhuang, P. Luo, Z. Shen, Q. He, Y. Xiong, Z. Shi, and H. Xiong, “Collaborative dual-plsa: mining distinction and commonality across multiple domains for text classification,” in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, ser. CIKM ’10, 2010.

[4] W. Dai, Q. Yang, G. Xue, and Y. Yu, “Boosting for transfer learning,” in *Proceedings of the 24th international conference on Machine learning*, ser. ICML ’07, 2007.

[5] J. Jiang and C. Zhai, “Instance weighting for domain adaptation in nlp,” in *In ACL 2007*, 2007, pp. 264–271.

[6] S. Lee, V. Chatalbashev, D. Vickrey, and D. Koller, “Learning a meta-level prior for feature relevance from multiple related tasks,” in *Proceedings of the 24th international conference on Machine learning*, ser. ICML ’07, 2007.

[7] H. Daumé, III, A. Kumar, and A. Saha, “Frustratingly easy semi-supervised domain adaptation,” in *In Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, 2010.

[8] Y. Chan and H. Ng, “Domain adaptation with active learning for word sense disambiguation,” in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, June 2007.

[9] P. Rai, A. Saha, H. Daumé, III, and S. Venkatasubramanian, “Domain adaptation meets active learning,” in *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, ser. ALNLP ’10, 2010.

[10] J. Huang, A. Smola, A. Gretton, K. Borgwardt, and B. Schölkopf, “Correcting sample selection bias by unlabeled data,” in *Advances in Neural Information Processing Systems 19*, 2007.

[11] M. Sugiyama, S. Nakajima, H. Kashima, P. von Bnau, and M. Kawanabe, “Direct importance estimation with model selection and its application to covariate shift adaptation.” in *NIPS*, 2007.

[12] J. Blitzer, R. McDonald, and F. Pereira, “Domain adaptation with structural correspondence learning,” in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP ’06, 2006.

[13] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, “Analysis of representations for domain adaptation,” in *In Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2007.

[14] W. Dai, G. Xue, Q. Yang, and Y. Yu, “Co-clustering based classification for out-of-domain documents,” in *In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007.

[15] S. J. Pan, J. T. Kwok, and Q. Yang, “Transfer learning via dimensionality reduction,” in *Proceedings of the 23rd national conference on Artificial intelligence - Volume 2*, ser. AAAI’08, 2008.

[16] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis,” in *Proceedings of the 21st international jont conference on Artifical intelligence*, ser. IJCAI’09, 2009.

[17] H. Daumé, III and D. Marcu, “Domain adaptation for statistical classifiers,” *Journal of Artificial Intelligence Research*, vol. 26, no. 1, pp. 101–126, 2006.

[18] G.-R. Xue, W. Dai, Q. Yang, and Y. Yu, “Topic-bridged plsa for cross-domain text classification,” in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR’08, 2008.

[19] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm.” *J. Royal Statistical Society, Series B*, vol. 39, no. 1, 1977.

[20] J. Yoo and S. Choi, “Probabilistic matrix tri-factorization,” *IEEE International Conference on Acoustics Speech and Signal Processing (2009)*, no. 3, pp. 1553–1556, 2009.

[21] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[22] D. Lewis and M. Ringuette, “A comparison of two learning algorithms for text categorization,” in *Proceedings of the Symposium on Document Analysis and Information Retrieval (SDAIR)*, 1994, pp. 81–93.

[23] G. Schwarz, “Estimating the dimension of a model,” *The Annals of Statistics*, vol. 6, pp. 461–464, 1978.