Domain Adaptation for Learning from Label Proportions Using Self-Training

Ehsan Mohammady Ardehaly and **Aron Culotta**
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616
emohamm1@hawk.iit.edu, aculotta@iit.edu

Abstract

Learning from Label Proportions (LLP) is a machine learning problem in which the training data consist of bags of instances, and only the class label distribution for each bag is known. In some domains label proportions are readily available; for example, by grouping social media users by location, one can use census statistics to build a classifier for user demographics. However, label proportions are unavailable in many domains, such as product review sites. The goal of this paper is to determine whether an LLP classifier fit in one domain can be modified to classify instances from another domain. To do so, we propose a *domain adaptation* algorithm that uses an LLP model fit on the source domain to generate label proportions for the target domain. A new LLP model is then fit on the target domain, and this self-training process is repeated to adapt the model from source to target. Our experiments on five diverse tasks indicate an 11% average absolute improvement in accuracy as compared to using LLP without domain adaptation. In contrast to existing domain adaptation algorithms, our approach requires only label proportions in the source domain, and the results suggest that the approach is effective even when the target domain is substantially different from the source domain.

1 Introduction

In the typical supervised learning setting, training data take the form $\{(\mathbf{x}_i, y_i)\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \mathbb{N}$ is a class label. Because it is often difficult to collect such data, an alternative setting has been investigated recently in which the training data take the form $\{(X_j, \tilde{p}_j)\}$, where $X_j \in \mathbb{R}^{n_i \times d}$ is a *bag* of n_i feature vectors, and $\tilde{p}_j \in \mathbb{R}^k$ is a distribution over the k class labels in that bag. This setting is called *Learning from Label Proportions* (LLP), and a number of solutions have been proposed [Kück and de Freitas, 2005; Quadrianto *et al.*, 2009; Mann and McCallum, 2010; Rueping, 2010; Yu *et al.*, 2013; Zhu *et al.*, 2014; Patrini *et al.*, 2014].

LLP is particularly attractive in domains for which label proportions are readily available, such as medical records [Wojtusiak *et al.*, 2011], fraud detection [Rueping, 2010], or social media [Ardehaly and Culotta, 2015]. In these domains, it is very easy to collect many observations and join them with label proportions for training purposes.

Since label proportions are not available in all domains, the goal of this paper is to determine whether an LLP classifier fit in one domain (e.g., social media) can be modified to classify instances from another domain (e.g., product review sites). Doing so would allow us to fit classifiers for many tasks without collecting labeled training data.

To do so, we propose a *domain adaptation* [Margolis, 2011] algorithm that uses an LLP model fit on the source domain to generate its own label proportions for the target domain. A new LLP model is then fit on the target domain, and this self-training process is repeated to adapt the model from source to target. This is inspired by self-training approaches to domain adaptation [McClosky *et al.*, 2006; Bhatt *et al.*, 2015], but here rather than generating instance labels, we generate bag label proportions.

Central to our approach is how to create bags in the target domain. As our domain is text classification, we draw inspiration from the idea of *feature labeling* [Druck *et al.*, 2008]. We select bags by identifying terms that are expected to be strongly indicative of a class, then grouping together instances containing such terms. Additionally, we use Latent Dirichlet Allocation (LDA) [Blei *et al.*, 2003] to help select terms that are salient in the target domain.

Across five diverse text classification tasks, we find an 11% average absolute improvement in accuracy as compared to using LLP without domain adaptation. In contrast to existing domain adaptation algorithms, our approach requires only label proportions in the source domain, and the results suggest that the approach is effective even when the target domain is substantially different from the source domain (e.g., the source is Twitter and the target is movie reviews).

2 Related Work

Our approach is a type of *unsupervised domain adaptation*, in which we have labeled data from the source domain but not from the target domain. Four primary approaches to this problem are: instance weighting for covariate shift, changes in feature representation, cluster-based learning, and self training methods [Margolis, 2011]. However, none of these methods have been extended to LLP models.

Because we allow both $P(\mathbf{x})$ and $P(y|\mathbf{x})$ to change from source to target, we investigate self-training methods [McClosky *et al.*, 2006; Axelrod *et al.*, 2011; Chattopadhyay *et al.*, 2012; Bhatt *et al.*, 2015], which have the least restrictive assumptions of how the domains differ. The most similar work we know of is that of Kadar and Iria [2011], who assume labeled features are given in the source domain. Using the generalized expectation framework, they assume an oracle who provides terms and the label proportions for each term [Druck *et al.*, 2008]. One substantial difference is that their work assumes the label proportions for a term are the same in the source and target data. In contrast, in the domains we consider, these label proportions often vary considerably between domains. Furthermore, our work assumes naturally found, possibly noisy label proportions, rather than features labeled by an oracle.

Thus, the primary contributions of this paper are to develop a self-training domain adaptation algorithm for LLP models and to empirically validate it on a challenging set of diverse domains.

3 Approach

Our learning setting is as follows: each observation j is represented by a feature vector $\mathbf{x}_j$; these vectors are grouped into a source domain S and a target domain T. The vectors in the source domain are further grouped into sets of m possibly overlapping bags $B_S = \{X_1 \dots X_m\}$, where each bag i is an observation by term matrix $X_i \in \mathbb{R}^{n_i \times d}$, and n_i is the number of observations in bag i and d is the number of features. Each vector $\mathbf{x}_j \in X_i$ has an unobserved class label y_j. For convenience we will restrict our attention to binary classification tasks; $y \in \{0, 1\}$. We assume an LLP setting in the source domain, so our supervision is a set of values $\tilde{p}_i \in [0, 1]$ indicating the proportion of observations in bag i that have the positive class label; i.e., $\tilde{p}_i = \frac{1}{n_i} \sum_{(\mathbf{x},y)\in X_i} \mathbf{1}[y = 1]$. T is the set of unlabeled observations from the target domain; as in the source, each observation $\mathbf{x}_j \in T$ has an unobserved class label y_j; however, we do not have access to any predefined bags or label proportions for the target domain. The task of unsupervised domain adaptation using LLP is to train a classifier with this information to predict the label of observations from the target domain. The primary difficulty of domain adaptation is that S and the target data T are not independent draws from the same underlying distributions, e.g. $P_S(\mathbf{x}, y) \neq P_T(\mathbf{x}, y)$. Depending on the nature of this mismatch, this problem may be called sample selection bias [Zadrozny, 2004], covariate shift [Bickel *et al.*, 2009], or concept drift [Widmer and Kubat, 1996]. Here, we do not restrict the nature of this mismatch, allowing arbitrary differences between the source and target.

3.1 LLP Baselines

Our domain adaptation approach is agnostic to the specific LLP algorithm used, requiring only a way to access the model parameters for each feature. Thus, for simplicity, we consider two linear LLP models (linear in the sense that each feature has a corresponding model parameter):

Ridge Regression, a linear regression with L2 regularization, has recently been used for LLP [Ardehaly and Culotta, 2014]. In this approach, the average of the feature vectors in bag X_i is used as the independent variable and the label proportion $\tilde{p}_i$ is used as the dependent variable. Let $\mathbf{z}_i \in \mathbb{R}^d$ be the vector of mean feature values in X_i, and θ be the model parameters, which are set by minimizing the penalized mean squared error objective:

$$J(\theta) = \sum_i (\tilde{p}_i - \mathbf{z}_i^T \theta)^2 + \frac{1}{2\sigma^2} ||\theta||^2$$

where σ is the L2 regularization parameter. To classify a new instance $\mathbf{x}$, if $\mathbf{x}^T\theta$ is greater than .5, it is classified as the positive class, otherwise as the negative class.

Label Regularization [Mann and McCallum, 2010] is a classification algorithm based on multinomial logistic regression. It selects parameters to minimize the difference between the label proportions $\tilde{p}_i$ and the estimated posterior distribution of bag labels $\hat{p}_i$, defined as $\hat{p}_i(y) = \frac{1}{n_i} \sum_{\mathbf{x}\in X_i} p_\theta(y|\mathbf{x})$. Label regularization uses KL-divergence as a distance metric with L2 regularization. The model parameters θ_y for class y are computed by minimizing the following cost function:

$$J(\theta) = \sum_i D(\tilde{p}_i||\hat{p}_i) + \frac{1}{2\sigma^2} \sum_y ||\theta_y||^2$$

The classification function is the same as multinomial logistic regression. Following Mann and McCallum [2010], we set a temperature term to 2 to avoid degenerate solutions: $p_\theta(y|\mathbf{x}) = \frac{\exp(\mathbf{x}^T\theta_y/2)}{\sum_y \exp(\mathbf{x}^T\theta_y/2)}$. While the cost function for label regularization is not guaranteed to be convex, we follow prior work in using L-BFGS to minimize the cost function, and to avoid overfitting we use early stopping by setting the maximum number of iterations to 12.

3.2 Self-training for LLP

Self-training is a bootstrapping approach that has been used to support domain adaptation in natural language processing tasks [McClosky *et al.*, 2006; Chattopadhyay *et al.*, 2012]. The general approach is as follows: first, fit a standard supervised classifier on source data S; second, use it to compute class posteriors for each instance in the target domain T; third, select the n most confidently labeled examples in T, and add them to the original training set, using the predicted labels as the true labels; fourth, retrain a new classifier with the augmented data. This process is then repeated until some convergence criterion is met. While the underlying idea is rather simple, it has been found to be empirically effective on tasks such as parsing.

Inspired by this prior work, we develop a self-training algorithm for LLP text classifiers. Whereas traditional self-training adds pseudo-labeled *instances* to the training set, here we add pseudo-labeled *bags*. To do so, we identify features that we estimate to be both strongly indicative of class assignment and also prevalent in the target domain. For each identified feature, we collect instances containing that feature in the target domain and construct a bag with an estimated label proportion. We then refit the model on these bags and repeat this process for a fixed number of iterations.

The success of this approach is dependent on the quality of the features identified to construct bags in the target domain. As mentioned above, a good feature should have two properties: (1) it should be strongly predictive of one class label, to ensure a nearly homogeneous bag; (2) it should appear with some salience in the target domain, to ensure adequate coverage of those observations. To satisfy the first property, we use the θ coefficients learned from an LLP model on the source domain. We assume that the highly weighted coefficients of each class are likely to be predictive of that class in the target domain. However, in the domain adaptation setting, features that satisfy property (1) are not guaranteed to satisfy property (2). For example, certain terms that are predictive in S may be used rarely or not at all in T. To address this problem, we perform topic modeling on the target data using Latent Dirichlet Analysis (LDA) [Blei *et al.*, 2003]. We use LDA to identify N_t topics in the target domain T, then select the top N_f terms from each topic as candidates for selection. (LDA has similarly been used to identify salient features in the context of labeling features [Druck *et al.*, 2008].)

To finally select terms, then, we first identify the top $N_f \times N_t$ terms by running LDA on T; from this set, we identify the N_s terms for each class that have the highest coefficient in the source model θ. For example, if $N_t = 50, N_f = 3, N_s = 10$, then we run LDA to generate 50 topics on T, select the top 3 terms from each topic, then select from these 150 terms the 10 that have the largest coefficient for the positive class and the 10 that have the largest coefficient for the negative class.

Once a term is identified, we collect instances in T containing that term and form a bag. The next question is what the label proportions should be for that bag. While we do not know the true label proportion, by design bag i should be mostly composed of instances with the same class label implied by the term used to construct it. E.g., if the term was strongly predictive of the positive class, then the resulting bag should contain mostly positive instances. In the experiments below, we set this label proportion by assuming that 90% of samples in the bag belong to the class associated with the term. We find that ridge regression is not sensitive to this guess (any probability more than 50% has the same result), but label regularization may be sensitive to it, and low entropy proportions tend to outperform high entropy proportions. In practice, accuracy is reduced by roughly 1% if we reduce the label proportions from 90% to 80%; and by up to 5% if the proportions are 70%. The reason that ridge regression is not sensitive is because different values of the label proportion only shift linearly the resulting coefficients.[1]

Using this approach, we construct $2 \times N_s$ bags on the target data, which we call B_T. We then fit the underlying LLP classifier on these new bags to obtain a model more appropriate for T. As in prior work on self-training [McClosky *et al.*, 2006], we find that better results are achieved by iterating this process to incrementally increase the number of pseudo-bags on the target domain. In each iteration, we increment N_s by 1, then reselect the top features using the latest θ parameters.

[1]We also tried estimating the bag label proportions using the source model, but this was not effective, most likely because the source classifier has low accuracy on the target domain.

Algorithm 1 Self-training for LLP. $B_S = \{(X_i, \tilde{p}_i)\}$ are the source bags, T is the target data, N_s is the number of target bags to construct for each class, N_t is the number of LDA topics, N_f is the number of terms per topic to consider for bag construction, and N_i is the number of iterations. LLP_TRAIN calls the underlying LLP training algorithm (e.g., ridge).

procedure LDA_TRAIN(B_S, T, N_s, N_t, N_f, N_i)
  $\theta \leftarrow$ LLP_TRAIN(B_S)
  Run LDA with N_t topics on T.
  Add top N_f terms for each topic to candidate set F
  for N_i iterations **do**
    $F' \leftarrow$ top N_s terms per class in F sorted by θ.
    $B_T \leftarrow$ the target bags created using F'.
    $\theta \leftarrow$ LLP_TRAIN(B_T)
    $N_s \leftarrow N_s + 1$
  end for
  return θ
end procedure

The final training procedure is outlined in Algorithm 1.

Tuning parameters and ensemble methods

Algorithm 1 has four tuning parameters: N_s, N_t, N_f, N_i. The number of LDA topics (N_t), can be selected by minimizing held-out perplexity. For simplicity, our experiments below set $N_T = 50$ for all tasks. The number of top terms per topic to consider (N_f) places an upper bound on the total number of bags that will be created in the target data. We fix $N_f = 3$ in all experiments below, limiting the model to at most 150 target bags. We did not optimize this value.

The remaining parameters are the number of bags to construct per class (N_s) and the number of iterations of the algorithm (N_i). As the results below will show, our approach can be sensitive to these values. This is in large part due to the varying quality of the terms used to construct bags. For example, if the label proportions for a term's bag is inverted (e.g., it should be 90% positive instead of 90% negative), then this can degrade accuracy. However, we observed that when there are many bags, the majority have the proper proportions, which can offset the errors of the few.

To capitalize on this observation and avoid having to set these values in advance, we implemented an ensemble approach that creates one model for each (N_s, N_i) pair, then uses a majority vote of the resulting models to classify the test instances. In the experiments below, we set the range of $N_s \in [5, 24]$ and $N_i \in [3, 7]$, resulting in 100 total models. For example, the model for ($N_s = 5$, $N_i = 4$) begins with 5 bags per class and iterates up to 8 bags per class. These ranges ignore the small models (e.g., just a few bags and a few iterations) as these have few training data and are likely to be unduly influenced by a single error in bag selection. For example, setting $N_i = 1$ has about 4% lower accuracy on average.

We refer to this ensemble model as **ridge-lda-ens** for ridge regression, and **lr-lda-ens** for label regularization. For comparison, we also report results for the individual models that had the best average accuracy on the test set: **ridge-lda-14** and **lr-lda-14**, which use $N_s = 14$ and use an ensemble of

iteration values $N_i \in [3, 7]$.

4 Data

Our experiments consider several diverse text classification tasks: classifying bloggers by age and political orientation, classifying movie reviews by sentiment, and classifying forum posts by topic. We use standard unigram features with minimum preprocessing. We describe both the source and target data below.

4.1 Twitter datasets

For age, politics, and sentiment, we use Twitter for our source data. Due to the numerous geographical and social constraints available, Twitter provides many natural label proportions for training LLP models, which has been investigated in prior work [Chang *et al.*, 2010; Oktay *et al.*, 2014; Ardehaly and Culotta, 2015]. The three source datasets are described below.

Twitter-age: This dataset contains 18M geolocated tweets from 2.7M users in U.S., posted in July 2014. We construct two types of bags, one based on follower information and one based on the user's first name. For follower bags, we match website traffic data from Quantcast.com [Kamerer, 2013] to corresponding Twitter accounts. For example, according this data, 11% of Twitter users who follow "oprah" are 18-24 years old. We simplify this to a binary prediction: users less than 25 are called "young," and users older than 25 are called "old." For roughly 1,000 Twitter accounts, we identify all users that follow them in our data, and construct a bag with the label proportions from QuantCast.

The first name constraints use baby names from the Social Security Administration, as in Silver and McCanc [2014]. We create bags for the 175 most popular names in our data. For example, 86% of people with name Katherine are estimated to be under 25.

Twitter-politic: Here we classify users as "Democrat" or "Republican." This dataset is the same as the Twitter-age dataset but with county and hashtag constraints. For county constraints, we use the 2012 presidential election results for each county as the label proportions, and create a bag for each county using the user's geolocation. For hashtag constraints, we create bags for users that follow or use one of 18 hashtags found to be strongly affiliated with one party by Ardehaly and Culotta [2015].

Twitter-TV: Here the task is to classify tweets about movies and television shows as expressing positive or negative sentiment. We collect 38M tweets from March - September 2015 by tracking hashtags of 50 TV shows. We create bags for 308 episodes (tweets after episode launch time). Our hypothesis is that the tweets posted after an episode is broadcast reflects the sentiment of users for that episode. To identify label proportions, we match each episode with its IMDB rating. For example, the final episode of season 5 of "The Walking Dead" series has IMDB rating 9.4 (out of 10), so we set the positive label proportion to be 94% for tweets associated with this episode.

Note that for all of the above datasets, the label proportions are only approximations — e.g., the proportion of citizens in a county who voted for Mitt Romney in 2012 is certainly not precisely equal to the proportion of Twitter users from that county in our data who are Republicans. However, as observed in prior work, the LLP model is still able to be effective because of the *relative* differences between bags (e.g., a Twitter user from a county with 80% Romney voters is more likely to be Republican than a user from a county with 10% Romney voters).

4.2 Blog datasets

We use two blog datasets as the target data for classifying users by age and political orientation:

Blog-2004: This corpus consists of 19,320 bloggers collected from blogger.com in August 2004 with around 35 posts per person [Schler *et al.*, 2006]. In our experiment, we use top three latest posts of users, and we define 10s (ages 13-17) as young users and 20s (ages 23-27) as old users and ignore 30s (ages 33-47) users. These brackets differ somewhat from our young definition (below 25) for Twitter, and as a result adapting to this task is particularly challenging.

Blog-2008: This corpus contains a collection of political blogs from 2008 [Eisenstein and Xing, 2010]. The corpus is divided into blogs that support Barack Obama and blogs that support John McCain during the 2008 presidential election. Because our county constraints for Twitter dataset comes from 2012 presidential election, we must adapt not only to a different data source, but also a different time period.

4.3 IMDB reviews

This corpus provides highly polar movie reviews [Maas *et al.*, 2011]. In our experiments, we only use 25K reviews in the testing set. The source data (Twitter-TV) is relevant to this corpus because both relate to movies; however, the Twitter-TV dataset is not filtered to just sentiment-bearing tweets. Thus, the probability of sentiment is different between source and target domains.

4.4 Discussion forums

Finally, we use two standard discussion forum datasets used in prior work [Kadar and Iria, 2011]. However, because we require that the source domain is presented as bags instead of labeled instances, we synthetically create 30 bags on the source domain, such that each bag contains 200 random samples (90% from one class and 10% from other class). Thus, we create 15 bags in which 90% of the documents belong to the first class, and 15 bags for other class. We use two newsgroups datasets:[2]

20 newsgroups: This corpus contains approximately 20,000 documents corresponding to 20 different newsgroups. The task is to distinguish documents about computers from those about science; as in previous research, we use comp.graphics, comp.os.ms-windows.misc, sci.crypt, and sci.electronics for source, and comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x, sci.med, and sci.space for target [Kadar and Iria, 2011]. We refer to this experiment as comp-sci.

[2] http://people.cs.umass.edu/~mccallum/code-data.html

task name	source dataset	source bags	target dataset	target labels
age	Twitter -age	700 followers 175 names	blog-2004	8.2K 10s 8.1K 20s
politic	Twitter -politic	472 counties 18 hashtags	blog-2008	5.7K dem 7.6K rep
sent-iment	Twitter -TV	308 episodes	IMDB review	12.5K + 12.5K -
comp-sci	20 news	30 random	20 news	3K comp 2K sci
real-sim	SRAA	30 random	SRAA	4K real 4K sim

Table 1: Summary of datasets and tasks.

model	age	politic	sent-iment	comp -sci	real -sim
logistic [4]				73.9	65.9
TSVM [5]				74.4	69
ridge-llp	55.1	45.2	52	73.7	60.9
label-reg	58.2	55.1	52.6	73.8	62.4
ridge-pseudo	53.8	59.8	58.7	80.2	64.1
ridge-lf-ens	63.4	60.9	54.7	75	67.4
ridge-lda-ens	66.5	71.5	65.1	78.3	**72.4**
lr-lda-ens	**69.1**	71.3	53.2	77.6	71.9
ridge-lda-14	64.2	70.6	**67.8**	**82**	71.1
lr-lda-14	66.4	**73.5**	55.4	74.8	70.7

Table 2: Accuracy on target domain.

SRAA: This corpus has messages about simulated and auto racing, simulated aviation, real autos, and real aviation from 4 different newsgroups. We use rec.autos.misc and rec.autos.simulators for source and rec.aviation.student and rec.aviation.simulators for target. We refer to this experiment as real-sim.

5 Experiments and results

Table 1 summarizes each task. We compare the models described above with several baselines:
- ridge-llp: Ridge regression for LLP without domain adaptation.
- label-reg: Label regularization without domain adaptation.
- logistic: Logistic regression without domain adaptation, which only can be applied when we have access to labels in the source dataset (for discussion forum tasks only)
- TSVM: Transductive SVMs [Joachims, 1999], a semi-supervised variant of SVMs that requires fully supervised training data in the source domain and unlabeled data in the target. (Thus we can only compare for the discussion form tasks.)
- ridge-pseudo: This is a baseline domain adaptation model for LLP that implements an alternative self-training approach based on pseudo-labeled instances, rather than pseudo-labeled bags. We first use ridge regression for LLP on the source data, then predict labels on the target data. We next select top 10% confident instances from target domain. Then we create 15 bags by sampling 180 positives and 20 negatives samples (from selected top confident samples) in each bag (90% of samples are positive). Similarly, we create 15 bags by sampling 180 negatives and 20 positives samples in each bag. We then fit a ridge LLP model to these new bags.[3]
- ridge-lf-ens: This is the same as **ridge-lda**, but it does not use LDA features to help select terms from the target domain. Instead, it just uses the top features in source domain according to the original LLP model. Like **ridge-lda-ens**, it computes a majority vote for $N_i \in [3, 7]$ and $N_s \in [5, 24]$ (100 models). Comparing with this baseline allows us to quantify the impact of LDA.

Table 2 compares the accuracy of all models. First, we note that the LLP models without domain adaptation (**ridge-llp**, **label-reg**) are comparable to a standard logistic regression (**logistic**), for the datasets which have fully labeled training data. Second, the need for domain adaptation is clear. For all five tasks, LLP models without domain adaptation have very poor accuracy in the target domain. This is perhaps not surprising given the substantial difference between source and target data, particularly for the Twitter datasets.

Comparing the different LLP domain adaptation approaches (**ridge-pseudo**, **ridge-lf-ens**, **ridge-lda-ens**, **lr-lda-ens**), it is apparent that using LDA to help identify terms consistently improves accuracy. For four of five tasks, **ridge-lda-ens** results in higher accuracy than **ridge-lf-ens**, achieving a roughly 10% absolute improvement on politics and sentiment. Additionally, the **ridge-pseudo** model does not perform well, outperforming **ridge-lda-ens** only for the comp-sci task. For that task, **ridge-llp** already has fairly high accuracy, allowing the pseudo labels to be precise enough to create accurate bags on target dataset. This result suggests that using pseudo-labeled features instead of pseudo-labeled instances is more effective when the source and target domains differ substantially.

Our primary finding comes from comparing the original LLP models (**ridge-llp**, **label-reg**) with their domain adaptation enhancements (**ridge-lda-ens**, **lr-lda-ens**). In every case, domain adaptation improves accuracy, sometimes dramatically so. For example, on the politics task, **ridge-llp** improves from 45.2% to 71.5%, and on the sentiment task **ridge-llp** improves from 52% to 65.1%. Thus, it appears that even when the source classifier is very inaccurate on the test data, domain adaptation can help. We speculate that self-training using pseudo-labeled bags is more effective in this setting than using pseudo-labeled instances because identifying a small number of indicative features is less error-prone than identifying many correctly labeled instances.

5.1 Sensitivity to tuning parameters

For an additional comparison, we also report accuracy of the single member of the ensemble methods that performs best on average across all datasets (**ridge-lda-14**, **lr-lda-14**). In both cases, beginning with 14 features per class ($N_s = 14$) resulted in the best performance. The **ridge-lda-14** model has the highest accuracy for sentiment, and has higher accuracy than TSVM model for comp-sci (even though only TSVM has access to labeled instances in the source domain).[5]

[3] We also considered adding pseudo-labeled instances, instead of bags, but this consistently did worse, so we have omitted for space.

[4] This model trains on labeled data in source domain.

[5] We note that our TSVM results are lower than those of Kadar and Iria [2011], perhaps due to differences in features and tuning.

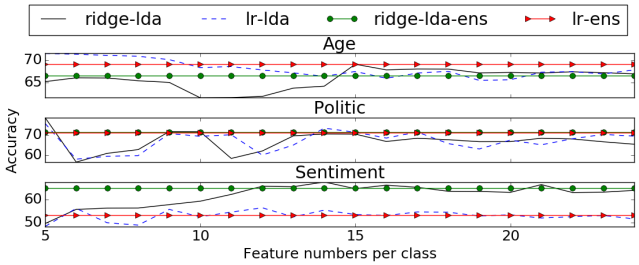

Figure 1: Accuracy per number of selected features.

Task	Incorrect bags	Correct bags
age	farted, listening	just, said, world
politic	financial, iowa	islamist, rocket, moderate
sentiment	tour, alien	burned, jasper, malone
comp-sci	entry, exhausted	grappler, deskjet, whomever
real-sim	wwal, way	fcc, 128, drive

Table 3: Error analysis of ridge-lda-14 model

To further examine how sensitive the results are to the number of initial features (N_s), Figure 1 compares accuracy on three tasks, varying $N_s \in [5, 24]$. Because the ensemble models (**ridge-lda-ens** and **lr-lda-ens**) do not have the N_s hyper parameter, they appear as flat lines in these plots. According to this figure, **lr-lda** is generally poor for sentiment experiment, but has higher accuracy for age (especially for $N < 15$).

We can see that accuracy varies quite a bit for small values of N_s, then begins to plateau. While there are some values that outperform the ensemble method, it does appear that the ensemble effectively finds a high accuracy, comparable to the best single setting of N_s, while avoiding the variance and instability of selecting a single N_s.

5.2 Error analysis

We additionally do an analysis of the errors of the models. The main error that occurs is when a feature is correlated with different classes in the source and target. If such features appear among the top LDA features in the target domain, the model would then infer the wrong sign for it. To measure how often this occurs, we run the **ridge-lda-14** model for four iterations (i.e., $N_c = 14$, $N_i = 4$), and record the bags and label proportions created in the final step. To determine what the correct sign for each bag should be, we fit a logistic regression classifier to the target labeled data and examine the sign for the term for each bag. We can then label a bag as correct if the label proportions align with the logistic regression coefficient, and as incorrect otherwise. Over all datasets, we find that approximately 70% of the bags are correct. Table 3 displays three correct bags and two incorrect bags per dataset. For the politic data, the terms "iowa" and "financial" are indicative of Democrats in the source data, but are indicative of Republicans in the target data. This shift can easily happen in political domains when, e.g., a region or topic becomes talked about more by one party.

Finally, we identified model parameters that were most improved by using domain adaptation. To create Table 4, we first selected the top 100 terms per class for the logistic regression classifier trained on the labeled target data. For each term, we examined its weight in **ridge-llp** and in **ridge-lda-14**, and identified the five terms for which the weight from **ridge-llp** was most improved by **ridge-lda-14**. Improvement here means that the difference from the logistic regression coefficient was reduced.

Task	First class	Second class
age	**10s**: awesome, die theres, wow, thats	**20s**: in, definitely 4th, month, process
politic	**Dem**: ve, am pretty, 06, actually	**Rep**: israel, terrorists sounds, islamic, romney
sentiment	**Pos**: performances, ryan pleasure, strong, great	**Neg**: bother, instead bad, nothing, annoying
comp-sci	**Comp**: chip, phone chips, with, disk	**Sci**: very, who of, med, energy
real-sim	**Real**: flying, ground pilot, wrote, pilots	**Sim**: 3d, card, 3dfx system, screen

Table 4: Most improved features by ridge-lda-14 model

For many of these, domain adaptation has identified terms that occurred infrequently in the source domain, but were salient in the target domain. For example, in politics, "israel" only appears in .4% of documents in the source data, but in 5% of the target documents. Similarly, in comp-sci, the term "med" appears in less than 1% of source documents, but in 6% of target documents. Thus, in addition to correcting the sign of certain coefficients, domain adaptation is able to increase the importance of terms that are rare in the source domain.

6 Conclusions

In this paper, we propose a domain adaptation algorithm for LLP models. The main advantage of our approach is that it can be trained on domains where label proportions exist naturally, then adapted to domains without such knowledge. While having fully labeled data would likely improve accuracy, in some domains such data may not be practical to collect at all (e.g., in health records or sensitive demographics of online users). The results suggests that our approach can be used in such settings.

We find that pseudo-labeling bags is often more effective than pseudo-labeling instances, particularly when the source and target domains differ substantially. Also, using topic modeling to identify salient terms in the target domain appears to be an effective way to guide domain adaptation. Finally, we found that a simple ensemble approach can alleviate some of the burden of tuning the parameters of this approach.

Future work should investigate methods to reduce the noise in the chosen bags, perhaps through outlier detection techniques. Additionally, joint models can be investigated to guide LDA to directly identify topics that contain terms predictive in the target domain.

Acknowledgments

This research was funded in part by the National Science Foundation under grant #IIS-1526674.

References

[Ardehaly and Culotta, 2014] Ehsan Mohammady Ardehaly and Aron Culotta. Using county demographics to infer attributes of twitter users. In *ACL Joint Workshop on Social Dynamics and Personal Attributes in Social Media*, 2014.

[Ardehaly and Culotta, 2015] Ehsan Mohammady Ardehaly and Aron Culotta. Inferring latent attributes of twitter users with label regularization. In *HLT*, pages 185–195, Denver, Colorado, May–June 2015.

[Axelrod *et al.*, 2011] Amittai Axelrod, Xiaodong He, and Jianfeng Gao. Domain adaptation via pseudo in-domain data selection. In *EMNLP*, pages 355–362, 2011.

[Bhatt *et al.*, 2015] Himanshu Sharad Bhatt, Deepali Semwal, and Shourya Roy. An iterative similarity based adaptation technique for cross-domain text classification. In *CoNLL*, pages 52–61, Beijing, China, July 2015.

[Bickel *et al.*, 2009] Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning under covariate shift. *JMLR*, 10:2137–2155, 2009.

[Blei *et al.*, 2003] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003.

[Chang *et al.*, 2010] Jonathan Chang, Itamar Rosenn, Lars Backstrom, and Cameron Marlow. epluribus: Ethnicity on social networks. In *ICWSM*, 2010.

[Chattopadhyay *et al.*, 2012] Rita Chattopadhyay, Qian Sun, Wei Fan, Ian Davidson, Sethuraman Panchanathan, and Jieping Ye. Multisource domain adaptation and its application to early detection of fatigue. *ACM Trans. Knowl. Discov. Data*, 6(4):18:1–18:26, December 2012.

[Druck *et al.*, 2008] Gregory Druck, Gideon Mann, and Andrew McCallum. Learning from labeled features using generalized expectation criteria. In *SIGIR*, pages 595–602, 2008.

[Eisenstein and Xing, 2010] Jacob Eisenstein and Eric Xing. The cmu 2008 political blog corpus. Technical report, Carnegie Mellon University, 2010.

[Joachims, 1999] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *ICML*, pages 200–209, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[Kadar and Iria, 2011] Cristina Kadar and José Iria. Domain adaptation for text categorization by feature labeling. In *ECIR*, pages 424–435, 2011.

[Kamerer, 2013] David Kamerer. Estimating online audiences: Understanding the limitations of competitive intelligence services. *First Monday*, 18(5), 2013.

[Kück and de Freitas, 2005] Hendrik Kück and Nando de Freitas. Learning about individuals from group statistics. In *UAI*, pages 332–339, 2005.

[Maas *et al.*, 2011] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *ACL*, pages 142–150, Portland, Oregon, USA, June 2011.

[Mann and McCallum, 2010] Gideon S. Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *JMLR*, 11:955984, March 2010.

[Margolis, 2011] Anna Margolis. A literature review of domain adaptation with unlabeled data. Technical report, University of Washington, 2011.

[McClosky *et al.*, 2006] David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *HLT/NAACL*, pages 152–159. Association for Computational Linguistics, 2006.

[Oktay *et al.*, 2014] Huseyin Oktay, Aykut Firat, and Zeynep Ertem. Demographic breakdown of twitter users: An analysis based on names. In *Academy of Science and Engineering (ASE)*, 2014.

[Patrini *et al.*, 2014] Giorgio Patrini, Richard Nock, Tiberio Caetano, and Paul Rivera. (almost) no label no cry. In *NIPS*, pages 190–198, 2014.

[Quadrianto *et al.*, 2009] Novi Quadrianto, Alex J. Smola, Tiberio S. Caetano, and Quoc V. Le. Estimating labels from label proportions. *JMLR*, 10:2349–2374, December 2009.

[Rueping, 2010] Stefan Rueping. SVM classifier estimation from group probabilities. In *ICML*, pages 911–918, 2010.

[Schler *et al.*, 2006] Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W. Pennebaker. Effects of age and gender on blogging. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 199–205. AAAI, 2006.

[Silver and McCanc, 2014] Nate Silver and Allison McCanc. How to tell someone's age when all you know is her name. Retrieved from http://fivethirtyeight.com/features/how-to-tell-someones-age-when-all-you-know-is-her-name/, 2014.

[Widmer and Kubat, 1996] Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101, 1996.

[Wojtusiak *et al.*, 2011] Janusz Wojtusiak, Katherine Irvin, Aybike Birerdinc, and Ancha V Baranova. Using published medical results and non-homogenous data in rule learning. In *Machine Learning and Applications and Workshops (ICMLA)*, volume 2, pages 84–89. IEEE, 2011.

[Yu *et al.*, 2013] FX Yu, D Liu, S Kumar, T Jebara, and SF Chang. $\propto$-SVM for learning with label proportions. In *ICML*, 2013.

[Zadrozny, 2004] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *ICML*, page 114. ACM, 2004.

[Zhu *et al.*, 2014] Jun Zhu, Ning Chen, and Eric P Xing. Bayesian inference with posterior regularization and applications to infinite latent SVMs. *JMLR*, 15:1799–1847, 2014.