Active Supervised Domain Adaptation

Avishek Saha[†⋆], Piyush Rai[†⋆], Hal Daumé III[‡],
Suresh Venkatasubramanian[†], and Scott L. DuVall[§]

[†]School of Computing, University of Utah
{avishek,piyush,suresh}@cs.utah.edu
[‡]Department of Computer Science, University of Maryland CP
hal@umiacs.umd.edu
[§]VA SLC Healthcare System & University of Utah
scott.duvall@hsc.utah.edu

Abstract. In this paper, we harness the synergy between two important learning paradigms, namely, active learning and domain adaptation. We show how active learning in a target domain can leverage information from a different but related source domain. Our proposed framework, Active Learning Domain Adapted (ALDA), uses source domain knowledge to transfer information that facilitates active learning in the target domain. We propose two variants of ALDA: a batch B-ALDA and an online O-ALDA. Empirical comparisons with numerous baselines on real-world datasets establish the efficacy of the proposed methods.

Key words: active learning, domain adaptation, batch, online

1 Introduction

We consider the *supervised* [1] domain adaptation setting [9] where we have a large amount of labeled data from some source domain, a large amount of unlabeled data from a target domain, and *additionally* a small budget for acquiring labels in the target domain. We show how, apart from leveraging information in the usual domain adaptation sense, the information from the source domain is *further* leveraged to selectively query for labels in the target domain (instead of choosing them randomly, as is the common practice). We achieve this by first training the best possible classifier in the source without using target labels, for instance, either by simply training a supervised classifier on the source labeled data, or by using some unsupervised adaptation technique using the unlabeled target data as well. Then, we use this learned hypothesis in various ways to leverage the source domain information when we are additionally given some fixed budget for acquiring some extra *labeled* target data (i.e., the active learning setting [12]).

⋆ Authors contributed equally.

[1] We define *supervised domain adaptation* as having labeled data in both *source* and *target*, *unsupervised domain adaptation* as having labeled data in only *source*, and *semi-supervised domain adaptation* as having labeled data in *source* and both labeled and unlabeled data in *target*.

We call this framework Active Learning Domain Adapted (ALDA). Our proposed framework is based on three key components. The first component is *unsupervised* domain adaptation (i.e., without target labeled data). The goal of this step is to suitably adapt the source data representation such that it makes the marginal distributions of both source and target distributions look similar. This enables training any traditional supervised classifier for the target domain using the adapted representation of the source data. The second and the third components improve this classifier even further by using active learning to selectively acquire the labels of target examples, given a budget on the target labels. Moreover, these components leverage the source domain information as well. Specifically, the second step employs a *domain separator hypothesis* that rules out querying labels of those target examples that appear "similar" to examples from the source domain. The domain separator hypothesis is a classifier that distinguishes between source and target domain examples and *is learned using only unlabeled examples from the two domains*. The third component is a hybrid oracle which consists of two oracles: one that provides labels for free but is imperfect (there could be noise), and one expensive (but "perfect") oracle used in the standard active learning settings. The source classifier acts as the free oracle which, although not perfect, can provide correct labels for most of the examples queried (essentially, the ones that appear 'source' like).

The proposed ALDA framework is sufficiently general to allow varied choices of domain adaptation and active learning modules. In addition, ALDA applies to both batch (Section 2) as well as online settings (Section 3). In this paper, we present empirical results (Section 4) for specific choices of the domain adaptation and the active learning schemes. For both batch and online settings, we empirically demonstrate that the proposed approach leads to significant improvement in prediction accuracies for a given target label budget, when compared to other baselines. Moreover, for the online setting, apart from showing empirically better performance, we also show that our approach results in smaller mistake bounds under suitable notions of domain separation. We provide intuitive arguments for smaller label complexity in the target domain when compared to the standard active learning where we do not have access to data from a related distribution.

2 ALDA: Active Learning Domain Adapted

In this section, we propose a principled approach towards active learning in a target domain by leveraging information from a related source domain. In our setting, we are given a small budget for acquiring labels in a target domain, which makes it imperative to use active learning in the target domain. However, our goal is to *additionally* leverage the domain relatedness by exploiting whatever information we might already have from the source domain. At a high level, our proposed approach aims to answer the following questions:

1. given source information, which samples in the *target* are the most informative (in an *active sense*)?

2. among the *informative target samples*, can we use source information to *infer labels* of a few *informative target samples*, such that the actual number of target labels queried (from an oracle) is reduced even further?

In the following, we provide answers to the above questions. We begin by introducing some notations and presenting an overview of the ALDA framework.

2.1 Preliminaries

Let $\mathcal{X} \subset \mathbb{R}^d$ denote the instance space and $\mathcal{Y} = \{-1, +1\}$ denote the label space. Let $\mathcal{D}_s(x, y)$ and $\mathcal{D}_t(x, y)$ be the joint source and target distributions, respectively. We have a set of source labeled examples $L_s(\sim \mathcal{D}_s(x, y))$ and a set of source unlabeled examples $U_s(\sim \mathcal{D}_s(x))$. Additionally, we also have a set of target unlabeled instances $U_t(\sim \mathcal{D}_t(x))$, from which we *actively* acquire labels. Furthermore, $\mathbf{w}_{src}$ denotes a classifier learned from the source labeled data and $\mathbf{w}_{ds}$ denotes the *domain separator* hypothesis. Finally, let ϕ represent an unsupervised domain adaptation algorithm that outputs a classifier $\mathbf{u}_\phi$. Note that learning $\mathbf{u}_\phi$ *does not require* labeled target examples.

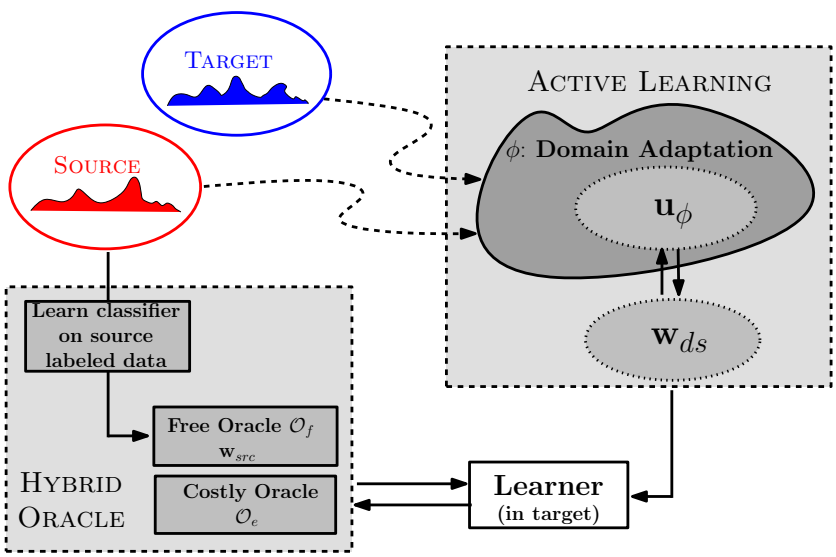

Fig. 1: An illustration of the proposed ALDA framework. Domain adaptation can be performed using any black-box unsupervised domain adaptation approach (e.g., [2, 14]). The active learning block can be any batch or online active learner.

Fig. 1 shows our basic setup for ALDA. The Active Learning (AL) module is a combination of the sub-modules Uncertainty Sampler (US) (that is initialized using the $\mathbf{u}_\phi$ classifier from the unsupervised domain adaptation phase) and Domain Separator (DS) (that uses the $\mathbf{w}_{ds}$ classifier). In addition, the setup

employs a *hybrid oracle* which is a combination of a free oracle $\mathcal{O}_f$ and an expensive oracle $\mathcal{O}_c$. The free oracle $\mathcal{O}_f$ is nothing but the classifier ($\mathbf{w}_{src}$) learned using the source labeled samples L_s. At each step, the learner *actively* selects an informative target sample and gets it labeled by an *appropriate* oracle. This continues in an iterative (for the batch setting) or online fashion until some stopping criterion is met (say, for example, reached prescribed accuracy or exhausted label budget). Next we describe each of these individual modules in more detail.

2.2 Initializing the Uncertainty Sampler

The first phase of ALDA learns an *unsupervised* domain adapted classifier $\mathbf{u}_\phi$ which uses labeled source data, and unlabeled source and target data. Note that this phase does not use any labeled target data (hence the name unsupervised). There are a number of ways to learn the classifier $\mathbf{u}_\phi$. In this paper, we take the approach [14] that is based on estimating the *importance ratio* between the source and the target distribution, without actually estimating these distributions. The source domain examples, with their corresponding importance weights, can then be used to train any classifier which is now readily adapted for the target domain (of course, this can potentially still be improved, given extra labeled target data). Note that the unsupervised domain adaptation step can be performed using a number of other ways as well; for example, Kernel Mean Matching (KMM) can be performed by matching the source and target distributions in some Reproducing Kernel Hilbert Space (RKHS) and computing the importance weights of source domain examples [8]. Another approach (especially for NLP problems), could be to use Structural Correspondence Learning (SCL) to identify invariant ("pivot") features between source and target, and use these features for unsupervised domain adaptation [2]. The unsupervised domain adapted classifier $\mathbf{u}_\phi$ serves as the initializing classifier for the subsequent active learning phase of our approach.

2.3 Leveraging Domain Divergence

It turns out that, in addition to using the source domain information to initialize our active learner in the target domain (Section 2.2), we can further leverage the domain relatedness information to improve the active learning phase in the target.

In this section, we propose the *domain separator* that further leverages the relatedness of source and target domains while performing active learning in the target. Assuming the source and target domains to be related, our proposed technique exploits this relatedness to upfront rule out acquiring labels of those *target domain examples* that "appear" to be close to the source domain.

As an example, Fig. 2 shows a typical domain separator hypothesis (denoted by $\mathbf{w}_{ds}$) that separates the *source* and *target* examples. We note that similar source and target examples are expected to have the same labels since only the marginal distribution of examples changes between the source and target

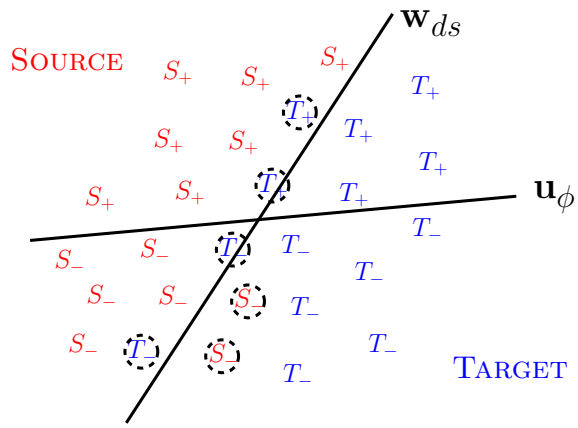

Fig. 2: An illustrative diagram showing the domain separator hypothesis $\mathbf{w}_{ds}$ separating source data from target data and the classifier $\mathbf{u}_\phi$ learned using the unsupervised domain adapted source classifier.

examples (i.e., $\mathcal{D}_s(x) \neq \mathcal{D}_t(x)$) whereas the conditional distribution of labels (given the examples) stays the same (i.e., $\mathcal{D}_s(y|x) = \mathcal{D}_t(y|x)$). Observe that if the source and target distributions are far apart, then the two domains can be perfectly classified by this separator. However, if the domains are similar, it is expected that there will be a reasonable overlap and therefore some of the target (or source) domain examples might lie on the source (or target) side (encircled instances in Fig. 2) and hence will be misclassified by the domain separator hypothesis. Acquiring labels for such target domain examples (that lie on the source side) is not really needed since the initial hypothesis (refer $\mathbf{u}_\phi$ in Fig. 1) of ALDA would already have taken into account such examples. Therefore, such target examples can be effectively ignored from being queried. Thus the domain separator hypothesis, which can be learned using *only* source and target *unlabeled* examples, provides a novel way of performing active sampling in domain adaptation settings.

The domain separator hypothesis avoids querying the labels of all those target examples that lie on the source side of the domain separator and hence are misclassified by it. This number, in turn, depends on the domain divergence between the source and target domains. For reasonably similar domain pairs, the domain divergence is expected to be small which implies that a large number of target examples lies on the source side. We can formalize the label complexity reduction due to the domain separator hypothesis. As earlier, let $\mathcal{D}_s$ and $\mathcal{D}_t$ denote the source and target joint distributions, and let $p_{\mathcal{D}_s}(x)$ and $p_{\mathcal{D}_t}(x)$ be probabilities of an instance x belonging to the source and the target respectively, in the unlabeled pool used to train the domain separator hypothesis. Let Δ denote the Mahalanobis distance between the source and target distributions. The Bayes error rate [15] of the domain separator hypothesis is: $E_{bayes} \leq \frac{2p_{\mathcal{D}_s}(x)p_{\mathcal{D}_t}(x)}{1+p_{\mathcal{D}_s}(x)p_{\mathcal{D}_t}(x)\Delta}$. Thus, the label complexity reduction due to the domain separator hypothesis

is proportional to the number of target examples misclassified by the domain separator hypothesis. This is again proportional to the Bayes error rate, which in turn is inversely related to the distance between the two domains.

2.4 Hybrid Oracle

ALDA additionally exploits the source domain information by using the source learned hypothesis (see, $\mathbf{w}_{src}$ in HYBRID of Fig. 1) as an oracle that provides labels for free. We denote this oracle by $\mathcal{O}_f$. Accordingly to the *Covariate Shift* assumption in domain adaptation, only the marginal distribution changes across domains whereas the conditional distribution remains fixed. If some target example appears to be close to the source domain then it is reasonable to assume that the prediction of the source classifer (which depends on the source conditional distribution) on that target sample should be close to the prediction of a *good* target classifier on that target sample. This explains the use of the source learned classifier as a free oracle for the target domain examples. Moreover, as in the standard active learning setting, we also have an expensive oracle $\mathcal{O}_c$. This leads to a hybrid setting which utilizes one of these two oracles for each actively sampled target example. The hybrid oracle starts with a domain adapted source initialized classifier ($\mathbf{u}_\phi$ in US of Fig. 1) and uses the domain separator hypothesis ($\mathbf{w}_{ds}$ in DS of Fig. 1) to assess which of the uncertain target examples lie on the source side and, for all such examples, it queries the labels from the free oracle $\mathcal{O}_f$. For the remaining uncertain examples that lie on the target side, the hybrid approach queries the expensive oracle $\mathcal{O}_c$. Although the oracle $\mathcal{O}_f$ is not perfect, the hope is that it can still provide correct labels for most of the target examples.

Algorithm 1 presents the final algorithm that combines all aforementioned schemes. This algorithm operates in a batch setting and we call it B-ALDA (for Batch-ALDA). As mentioned earlier (ref. Section 2.2), the importance ratio in line 2 of Algorithm 1 can be obtained by the techniques SCL [2], KMM [8], etc.

3 Online ALDA

In B-ALDA, the active learning module, at each iteration, chooses the data point that lies closest to the decision boundary. However, this approach is prohibitively slow for large or even moderately sized datasets. Hence, we propose Online ALDA (O-ALDA) which performs active learning in an online fashion and for each example decides whether or not to query its label. As in standard active learning, this query decision must be biased by the *informativeness* of the example.

To extend ALDA to the online setting, we adopt the label query strategy proposed in [3]. However, we note that our framework is sufficiently general and allows integration with other *active online sampling strategies*. The sampling scheme in [3] proceeds in rounds and at round i queries the label of the example x^i with probability $\frac{b}{b+|r^i|}$, where $|r^i|$ is the *confidence* (in terms of margin) of the current weight vector on x^i. Parameter b quantifies how aggressively the labels

Algorithm 1 B-ALDA

input $L_s = \{\mathbf{x}_s, y\}$; U_s; U_t; maxCost (label budget K and/or desired accuracy ϵ);
output $\mathbf{v}$ (target classifier);

```
cost := 0;
S := L̃_s (importance weighted L_s learned using L_s, U_s and U_t);
u_Φ := learn a domain adapted source classifier using S;
w_ds := learn a classifier using the data {U_s, +1} and {U_t, −1};
w_src := learn a domain adapted source classifier using L_s;
while (cost < maxCost) do
    x̄_t := US(u_Φ, U_t);                /* choose most informative target point */
    ŷ_ds := DS(w_ds, x̄_t);              /* compute source resemblance */
    if (ŷ_ds == +1) then
        y_t = O_f(w_src, x̄_t);          /* query the free oracle */
    else if (ŷ_ds == −1) then
        y_t = O_c(x̄_t);                 /* query the costly oracle */
        cost ← cost + 1;
    end if
    S = S ∪ {x̄_t, y_t};
    retrain u_Φ using S;
end while
```

are being queried. A large value of b implies that, in expectation, a large number of labels will be queried (aggressive sampling) whereas a small value would lead to a small number of examples being queried (conservative sampling). For each label queried, the algorithm updates the current weight vector if the label was predicted incorrectly. It is easy to see that the total number of labels queried by this algorithm is $\sum_{i=1}^{T} \mathbb{E}[\frac{b}{b+|r^i|}]$, where T is the total number of rounds. At this point we note that the preprocessing stage of O-ALDA assumes the existence of some (maybe, a small amount) of target *unlabeled* data that can be utilized to construct the common representation. The online active learning in the target starts after this preprocessing phase when O-ALDA selectively queries the labels of the target data points that arrive in some random order.

Algorithm 2 presents the online variant of ALDA which we refer to as O-ALDA (for Online-ALDA). As shown in Theorem 1, our proposed O-ALDA yields provable guarantees on mistake bounds and label complexity.

Theorem 1. *Let $S = ((x_1, y_1), \ldots, (x_T, y_T)) \in (\mathbb{R}\times\{-1,+1\})^T$ be any sequence of examples and $\mathcal{UP}_T$ the (random) set of update trials for the algorithm (i.e., the set of trials $i \leq T$ such that $\hat{y}^i \neq y^i$ and $Z^i = 1$). Let $\boldsymbol{v}_0$ be the weight vector with which the base target classifier is initialized and r^i be the margin of* O-ALDA *on example x^i. Then the expected number of mistakes made by the algorithm is upper bounded by*

$$\inf_{\gamma>0} \inf_{\boldsymbol{v}^* \in \mathbb{R}^D} \left(\frac{(2b+1)}{2b} \mathbb{E}\left[\sum_{i \in \mathcal{UP}_T} \frac{1}{\gamma} D_\gamma(\boldsymbol{v}^*; (\hat{x}^i, y^i)) \right] + \frac{(2b+1)^2}{8b} \frac{||\boldsymbol{v}^* - \boldsymbol{v}_0||^2}{\gamma^2} \right)$$

The expected number of labels queried by the algorithm is equal to $\sum_{i=1}^{T} \mathbb{E}[\frac{b}{b+|r^i|}]$.

Algorithm 2 O-ALDA

input $b > 0$; $L_s = \{\mathbf{x}_s, y\}$; U_s; U_t; maxCost (label budget K/desired accuracy ϵ);
output $\mathbf{v}$ (target classifier);

```
cost := 0;
u_Φ := learn a domain adapted source classifier using L_s, U_s and U_t;
w_ds := learn a classifier using the data {U_s, +1} and {U_t, −1};
w_src := learn a domain adapted source classifier using L_s;
while ( (i <= T) & (cost < maxCost) ) do
   r^i := US(u_Φ, x_t^i);                    /* compute margin of i^th target point */
   ŷ_ds^i := DS(w_ds, x_t^i);                /* compute source resemblance */
   sample Z^i ~ Bernoulli(b/(b+|r^i|));
   if (Z^i == 1) then
      if (ŷ_ds^i == +1) then
         y_t^i = O_f(w_src, x_t^i);          /* query the free oracle */
      else if (ŷ_ds^i == −1) then
         y_t^i = O_c(x_t^i);                 /* query the costly oracle */
         cost ← cost + 1;
      end if
      if (y_t^i ≠ u_Φ^T x_t^i) then
         update u_Φ using online update rule (such, as perceptron);
      end if
   end if
end while
```

In the above theorem, γ refers to some margin greater than zero such that the cumulative hinge loss of the optimal target hypothesis $\mathbf{v}^*$ on S is given by $\sum_1^T D_\gamma(\mathbf{v}^*; (x^i, y^i))$, where $D_\gamma(\mathbf{v}^*; (x^i, y^i)) = max\{0, \gamma - y^i \mathbf{v}^{*T} x^i\}$ is the hinge-loss on example i. In Appendix A, we discuss the above theorem and provide a proof sketch for the mistake bound and the label complexity of O-ALDA. In addition, we discuss the conditions on $\mathbf{v}_0$ that lead to improved mistake bounds in domain adaptation settings as compared to the case where there is no access to data from a related source domain.

4 Experiments

In this section, we demonstrate the empirical performance of our algorithms and compare them with a number of baselines.

4.1 Setup

Datasets: We present results for **Sentiment** and **Landmine** datasets. The **Sentiment** dataset consists of user reviews of eight product types (apparel, books, DVD, electronics, kitchen, music, video, and other) from Amazon.com. The sentiment classification task for this dataset is binary classification which corresponds to classifying a review as positive or negative. The dataset consists of

several domain pairs with varying $\mathcal{A}$-distances, akin to a sense described in [1]. Table 1 shows some of the domain pairs used in our experiments and their corresponding domain divergences in terms of the $\mathcal{A}$-distance [1].

Source	**Target**	**$\mathcal{A}$-distance**
Dvd (D)	Books (B)	0.7616
Dvd (D)	Music (M)	0.7314
Books (B)	Apparel (A)	0.5970
Dvd (D)	Apparel (A)	0.5778
Electronics (E)	Apparel (A)	0.1717
Kitchen (K)	Apparel (A)	0.0459

Table 1: Proxy $\mathcal{A}$-distances between some domain pairs in the sentiment data

To compute the $\mathcal{A}$-distance from finite samples of source and target domain, we use a surrogate to the true $\mathcal{A}$-distance (the *proxy* $\mathcal{A}$-distance) in a manner similar to [1]. First, we train a linear classifier to separate the *source* domain from the *target* domain using only unlabeled examples from both. The average per-instance hinge-loss of this classifier subtracted from 1 serves as our estimate of the *proxy* $\mathcal{A}$-distance. A score of 1 means perfectly separable distributions whereas a score of 0 means that the two distributions are essentially the same. The amount of useful information that can be leveraged from the other domain would depend on how similar the two domains are. To this end, we therefore choose two datasets from the sentiment data - one with a domain-pair that is reasonably close (Kitchen→Apparel), and another with a domain-pair that is reasonably far apart (DVD→Books).

Our second dataset (**Landmine**) is the real Landmine Detection data [16] which consists of 29 datasets. The datasets 1 to 10 are collected at foliated regions whereas the datasets 20 to 24 are collected from bare earth or desert regions. We combined datasets $1-5$ as our source domain and treat dataset 24 as the target domain.

Methods: Table 2 summarizes the methods used with a brief description of each. Among the first three (ID, sDA, Feda), Feda [6] is a state-of-the-art *supervised* domain adaptation method but assumes *passively* acquired labels. The first three methods (ID, sDA, Feda) acquire labels *passively*. The last five (Alzi, Alri, Alsi, B-Alda and O-Alda) methods in Table 2 acquire labels in an active fashion. As the description denotes, Alzi and Alri start active learning in *target* with a zero initialized and randomly initialized hypothesis, respectively. It is also important to distinguish between Alsi and Alda (which jointly denotes both B-Alda and O-Alda). While both are products of our proposed Alda framework, Alsi uses an unmodified source classifier learned only from *source* labeled data as the initializer, whereas Alda (i.e., both B-Alda and O-Alda) uses an *unsupervised* domain-adaptation technique (i.e., without using labeled target data) to learn the initializer.

Method	Summary	Active ?
ID	In-domain data	No
sDA	Unsupervised domain adaptation followed by *passively* chosen labeled target data	No
Feda	Frustratingly Easy Domain Adaptation [6]	No
Alzi	Active learning zero initialized	Yes
Alri	Active learning random initialized (with fixed label budget)	Yes
Alsi	Active learning source (hypothesis) initialized	Yes
B-Alda	Batch active learning domain adapted	Yes
O-Alda	Online active learning domain adapted	Yes

Table 2: Description of the methods compared

In our experiments, we use the instance reweighting approach [14] to construct the unsupervised domain adaptated classifier $\mathbf{u}_\phi$. However, we note that this step can also be performed using any other unsupervised domain adaptation technique such as Structural Correspondence Learning (SCL) [2] and Kernel Mean Matching (KMM) [8].

We compare all the approaches based on classification accuracies achieved for a fixed unlabeled pool of target examples with varying label budgets. For B-Alda, we use a margin based classifier (SVM) whereas for O-Alda we use vanilla Perceptron as the base classifier. All online experiments have been averaged over multiple runs with respect to random data order permutations.

4.2 B-Alda results

We present results for B-Alda using a fixed target unlabeled pool and varying target label budgets. Since, domain adaptation is required only when there are small amounts of labeled data in the target, we limit our target label budget to values that are much smaller than the size of the unlabeled target data pool. In addition, due to long running times of our batch Alda (owing to repeated retraining), we report results on relatively smaller target pool sizes. The B-Alda results are presented for a unlabeled target pool size of 2500 data points.

Sentiment Classification: Table 3a and Table 3b present the results for the domain pairs DVD→Books and Kitchen→Apparel, respectively. For these domain pairs, both Alsi and B-Alda substantially outperform all other baselines. For the distant source-target pair (DVD→Books), Alsi performs very well for a small number of target labels (say, 100 and 200). As the number of target labels increases B-Alda consistently improves with increasing number of target labels and finally outperforms Alsi. When the source-target pairs are reasonably close (Kitchen→Apparel), both Alsi and B-Alda have similar prediction accuracies which are in turn are much higher that the baseline accuracies.

Method	Target Label Budget				
	100	**200**	**300**	**400**	**500**
	Acc	**Acc**	**Acc**	**Acc**	**Acc**
ID	50.83	57.86	62.42	55.69	62.68
sDA	62.18	62.78	55.75	52.45	50.49
Feda	63.92	64.27	64.88	65.94	66.19
Alzi	54.40	54.36	54.33	54.33	54.33
Alri	54.99	59.42	61.28	65.81	65.52
Alsi	**63.75**	**66.26**	**68.73**	63.10	62.08
B-Alda	63.40	65.17	67.84	**68.61**	**68.51**
Acc: Accuracy					

(a) DVD→Books

Method	Target Label Budget				
	100	**200**	**300**	**400**	**500**
	Acc	**Acc**	**Acc**	**Acc**	**Acc**
ID	48.40	43.44	44.92	48.40	49.77
sDA	52.78	55.41	57.37	53.60	46.37
Feda	70.47	69.97	70.06	71.83	69.96
Alzi	54.56	54.50	54.44	54.44	54.44
Alri	64.97	66.86	69.01	70.40	71.06
Alsi	**74.91**	70.58	**72.97**	**72.34**	72.29
B-Alda	71.30	**70.90**	71.19	71.73	**73.07**
Acc: Accuracy					

(b) Kitchen→Apparel

Table 3: Classification accuracies and number of labels requested. Note: ID, sDA and Feda are given labels of all examples in the target pool.

Landmine Detection: The **Landmine** dataset has a high class imbalance (only about 5% positive examples), so we report AUC (area under the ROC curve) scores instead of accuracies. We compare our algorithms with other baselines in terms of the AUC score on the entire pool of target data (the pool size was 300; rest of the examples in dataset 24 were treated as test data). As shown in Table 4, our approaches perform better than the other baselines with the domain separator based B-Alda doing the best (in terms of AUC scores).

We do not report any label complexity result for B-Alda as the nature of the algorithm is such that it iterates until the entire label budget is exhausted. Hence, in all the results presented above in Table 3a, Table 3b and Table 4, the number of labels used is equal to the target label budget provided.

Method	Target Budget (300)
	AUC
ID	0.59
sDA	0.60
Feda	0.56
Alzi	0.59
Alri	0.53
Alsi	0.63
B-Alda	**0.65**
AUC: AUC score **Lab:** Labels Requested	

Table 4: AUC scores and labels requested for the **Landmine** dataset.

4.3 O-Alda results

One of the goals to propose an online variant for Alda is to make the proposed approach scale efficiently for larger target pool sizes because batch mode Alda requires repeated retraining. On the other hand, an online active learner is an efficient alternative because it allows incremental update of the learner for each new selected data point. In this section, we present results for O-Alda and demonstrate the scalability of the Alda framework to larger target pool sizes. The results for O-Alda use the entire target unlabeled pool ($\sim$ 7000 for

Sentiment data). As a result, the label budget allocated is also much larger as compared to B-ALDA. We note that ID and sDA and FEDA have been made online by the use of the perceptron classifier. In addition, the same online active strategy as O-ALDA has been used for ALZI, ALRI and ALSI.

Sentiment Classification: The results are shown in Table 5a and Table 5b. As the results indicate, on both datasets, our approaches (ALSI and ALDA) perform consistently better than the baseline approaches (Table 2) which also include one of the state-of-the-art supervised domain adaptation algorithms (FEDA). We note that ALDA outperforms ALSI for KITCHEN→APPAREL as compared to DVD→BOOKS. When the domains are far (DVD→BOOKS), the performance of ALDA depends on the underlying domain adaptation technique. However, when the domains are close (KITCHEN→APPAREL), ALDA performs better than ALSI. This behavior suggests that the performance gains achieved by these variants is significant when the source and target domains are *reasonably close*.

Method	**Target Label Budget**				
	1000	**2000**	**3000**	**4000**	**5000**
	Acc(±Std)	**Acc(±Std)**	**Acc(±Std)**	**Acc(±Std)**	**Acc(±Std)**
ID	65.94(±3.40)	66.66(±3.01)	67.00(±2.40)	65.72(±3.98)	66.25(±3.18)
sDA	66.17(±2.57)	66.45(±2.88)	65.31(±3.13)	66.33(±3.51)	66.22(±3.05)
FEDA	67.31(±3.36)	68.47(±3.15)	68.37(±2.72)	66.95(3.11)	67.13(±3.16)
ALZI	66.24(±3.16)	66.72(±3.30)	63.97(±4.82)	66.28(±3.61)	66.36(±2.82)
ALRI	51.79(±4.36)	53.12(±4.65)	55.01(±4.20)	57.56(±4.18)	58.57(±2.44)
ALSI	**68.22(±2.17)**	**69.65(±1.20)**	**69.95(±1.55)**	70.54(±1.42)	**70.97(±0.97)**
O-ALDA	67.64(±2.35)	68.89(±1.37)	69.49(±1.63)	**70.55(1.15)**	70.65(±0.94)
Acc: Accuracy | **Std:** Standard Deviation					

(a) DVD→BOOKS

Method	**Target Label Budget**				
	1000	**2000**	**3000**	**4000**	**5000**
	Acc(±Std)	**Acc(±Std)**	**Acc(±Std)**	**Acc(±Std)**	**Acc(±Std)**
ID	69.64(±3.14)	69.61(±3.17)	69.36(±3.14)	69.77(±3.58)	70.77(±3.05)
sDA	69.70(±2.57)	70.48(±3.42)	70.29(±2.56)	70.86(±3.16)	70.71(±3.65)
FEDA	70.05(±2.47)	69.34(±3.50)	71.22(±3.00)	71.67(±2.59)	70.80(±3.89)
ALZI	70.09(±3.74)	69.96(±3.27)	68.6 (±3.94)	70.06(±2.84)	69.75(±3.26)
ALRI	52.13(±5.44)	56.83(±5.36)	58.09(±4.09)	59.82(±4.16)	62.03(±2.52)
ALSI	73.82(±1.47)	**74.45(±1.27)**	75.11(±0.98)	75.35(±1.30)	**75.58(±0.85)**
O-ALDA	**73.93(±1.84)**	74.18(±1.85)	**75.13(±1.18)**	**75.88(±1.32)**	**75.58(±0.97)**
Acc: Accuracy | **Std:** Standard Deviation					

(b) KITCHEN→APPAREL

Table 5: Classification accuracies and number of labels requested. Results are averaged over 20 runs (w.r.t. different permutations of the training data). Note: ID, sDA and FEDA are given labels of all examples in the target pool.

Landmine Detection: Similar to B-ALDA results, in this case also we used the entire pool of 300 target data points. The rest of the examples in dataset 24 were treated as test data. As earlier, our approaches perform better than the other baselines with the domain separator based O-ALDA demonstrating slightly better AUC score and slightly lesser label complexity as compared to online ALSI. Table 6 presents the AUC scores and the label complexities of the various methods.

4.4 Remarks

For all datasets considered, both batch and online versions of ALDA demonstrate substantial improvement of prediction accuracy for **Sentiment** data ($\sim$ ($\mathbf{0.4\%} - \mathbf{5.09\%}$)). This improvement is particularly high when the domains are reasonably similar (for example, KITCHEN→APPAREL in Table 3b and Table 5b). In addition, the **Landmine** data reports AUC scores (not accuracies), and 1% increase in AUC score implies substantial improvement.

Method	**Target Budget (300)** AUC±Std (Lab)
ID	0.57±0.03 (-)
SDA	0.60±0.02 (-)
FEDA	0.52±0.04 (-)
ALZI	0.61±0.02 (284)
ALRI	0.56±0.05 (229)
ALSI	0.65±0.02 (244)
O-ALDA	**0.67±0.03 (241)**
AUC: AUC score **Std:** Standard Deviation **Lab:** Labels Requested	

Table 6: AUC scores and labels requested for the **Landmine** dataset. Results are averaged over 20 runs.

For **Sentiment** and **Landmine** datasets, both ALSI and ALDA (i.e., B-ALDA and O-ALDA) demonstrate improvement in *prediction accuracy for a fixed label budget* when compared to other baselines. Apart from the results for DVD→BOOKS in the batch setting (Table 3a), the prediction accuracies obtained by ALSI and ALDA in all other cases are comparable. However, to get a better sense of the robustness of these two approaches, we compare the number of mistakes made by the online variants of these two approaches during the training phase. Table 7 presents the results for **Sentiment** data. As can be seen, in almost all case the number of mistakes made by O-ALDA is much lesser (almost half in many cases) than online ALSI. Hence, irrespective of the nearness or farness of the source-target domain pairs, ALDA is a better choice as compared to ALSI.

	Target Label Budget									
	1000	**2000**	**3000**	**4000**	**5000**	**1000**	**2000**	**3000**	**4000**	**5000**
	Number of Mistakes									
Method	DVD→BOOKS					KITCHEN→APPAREL				
ALSI	**369**	**739**	1117	1460	1816	245	532	810	1097	1088
O-ALDA	384	741	**1000**	**1012**	**1004**	**232**	**478**	**549**	**551**	**556**

Table 7: Number of mistakes made by ALSI and O-ALDA for **Sentiment** data.

5 Related Work

Active learning in a domain adaptation setting has received little attention so far and, to the best of our knowledge, there exists no prior work that presents a principled framework to harness domain adaptation for active learning. One interesting setting was proposed in [4] where the authors apply active learning for word sense disambiguation in a domain adaptation setting. In addition, they also improve vanilla active learning when combined with domain adaptation. However, their approach does not use the notions of domain separator and hybrid oracle. Moreover, unlike our approach, their method only works in a batch setting.

Active learning in an online setting has been discussed in [5] and [3]. The work of [5] assumes input data points uniformly distributed over the surface of an unit sphere. However, we cannot make such distributional assumptions for domain adaptation. As mentioned earlier, [3] provide worst-case analysis which is independent of any input data distribution. However, none of these explicitly consider the case of domain adaptation. Nonetheless, the framework of [3] folds nicely into our proposed ALDA framework. [10] present extensive empirical results to compare the performance of the two aforementioned approaches. However, all these settings are different from our in that these works consider only active learning in an online setting without leveraging inter-domain information.

A combination of transfer learning with active learning has been presented in [13]. One drawback of their approach is the requirement of an initial pool of labeled target domain data which helps train the in-domain classifier. Without this in-domain classifier, no transfer learning is possible in their setting.

6 Discussions and Future Work

In this work, we have considered a domain adaptation setting, and presented a framework that helps leverage inter-domain information transfer while performing active learning in the target. Both the batch and online versions of the proposed ALDA empirically demonstrate the benefits of domain transfer for active learning.

At present, ALDA is oblivious to the feature set used and, as such, does not depend on domain knowledge and feature selection. It takes all features into consideration. Nonetheless, it is possible that in the feature space, not all features contribute equally while transferring information from source to target and without a priori information about the source and target domains, it is difficult to assess which features might maximally benefit the transfer of parameters from source to target. However, if prior domain knowledge about the target is available from related source domains, then one can potentially leverage active learning to selectively choose *only* those features that transfer maximum information between the two domains.

An alternative approach to leverage feature information for ALDA is to perform active learning on features. There exists work in active learning that queries

labels for features [7] and, in some cases, queries labels for both instances and features in tandem [11]. We note that this is different from the above where active learning can essentially be used as a tool for feature selection. In this case, active strategies query labels that exploit both instance and feature informativeness (for e.g., in NLP, consider querying labels for rare words which serve as informative features in the target domain). It would be interesting to extend the proposed ALDA to perform active domain transfer by querying labels of both instances and features.

Acknowledgements

This work was sponsored in part by the NSF grants CCF-0953066 and CCF-0841185 and in part by the Consortium for Healthcare Informatics Research (CHIR), VA HSR HIR 08-374, VA Informatics and Computing Infrastructure (VINCI) and VA HSR HIR 08-204. This work was also partially supported by the NSF grant IIS-0712764. The authors gratefully acknowledge the support of the grants. Any opinions, findings, and conclusion or recommendation expressed in this material are those of the author(s) and do not necessarily reflect the view of the funding agencies or the U.S. government.

References

1. Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *NIPS'06*, Vancouver, Canada, December 2006.
2. John Blitzer, Ryan Mcdonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *EMNLP'06*, Sydney, Australia, July 2006.
3. Nicolò Cesa-Bianchi, Claudio Gentile, and Luca Zaniboni. Worst-case analysis of selective sampling for linear classification. *JMLR*, 7, 2006.
4. Yee Seng Chan and Hwee Tou Ng. Domain adaptation with active learning for word sense disambiguation. In *ACL'07*, Prague, Czech Republic, June 2007.
5. Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. *JMLR*, 10, 2009.
6. Hal Daumé III. Frustratingly easy domain adaptation. In *ACL'07*, Prague, Czech Republic, June 2007.
7. Gregory Druck, Burr Settles, and Andrew McCallum. Active learning by labeling features. In *EMNLP'09*, Singapore, August 2009.
8. Jiayuan Huang, Alexander J. Smola, Arthur Gretton, Karsten M. Borgwardt, and Bernhard Schölkopf. Correcting sample selection bias by unlabeled data. In *NIPS'07*, Vancouver, Canada, 2007.
9. Jing Jiang. A literature survey on domain adaptation of statistical classifiers. 2008.
10. Claire Monteleoni and Maati Kääriäinen. Practical online active learning for classification. In *IEEE CVPR Workshop on Online Learning for Classification*, Minneapolis, USA, June 2007.
11. Hema Raghavan, Omid Madani, and Rosie Jones. Active learning with feedback on features and instances. *JMLR*, 7, December 2006.

12. Burr Settles. Active learning literature survey. In *Computer Sciences Technical Report 1648*, University of Wisconsin-Madison, 2009.
13. Xiaoxiao Shi, Wei Fan, and Jiangtao Ren. Actively transfer domain knowledge. In *ECML/PKDD'08*, Antwerp, Belgium, September 2008.
14. Masashi Sugiyama, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NIPS'07*, Vancouver, Canada, December 2007.
15. Kagan Tumer and Joydeep Ghosh. Estimating the bayes error rate through classifier combining. In *ICPR'96*, volume 2, Vienna, Austria, August 1996.
16. Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *JMLR*, 8, 2007.

A Discussion of Theorem 1

To conserve space, we skip presenting a detailed proof of the mistake bound in Theorem 1. Proceeding in a manner similar to the proof of Theorem 1 of [3], it can be seen that almost all terms in the final expression for the mistake bound cancel out by the telescopic argument. The term that remains is $||\mathbf{v}^* - \mathbf{v}_0||^2$ and the mistake bound follows. It is easy to see that setting $\mathbf{v}_0 = 0$ in Theorem 1 yields mistake bounds for online active learning in traditional single task settings. We note that, the first term in the mistake bound of Theorem 1 is the cumulative hinge loss of the *optimal* target classifier which is the same for both domain adaptation and non-domain adaptation (traditional single task) settings and hence is independent of the initialization used. The second term in the mistake bound, in our case, is smaller than single task settings provided $\theta \leq cos^{-1}\left(\frac{||\mathbf{v}_0||}{2||\mathbf{v}^*||}\right)$, where θ is the angle between the initializing hypothesis $\mathbf{v}_0$ and the target hypothesis $\mathbf{v}^*$. Without loss of generality, assuming the norm of $\mathbf{v}_0$ and $\mathbf{v}^*$ stays fixed (which is true since both the *initial* and the *optimal* hypotheses remain unchanged during learning in target domain), as the value of θ decreases, it causes $||\mathbf{v}^* - \mathbf{v}_0||^2$ to decrease, leading to our claim of reduced mistake bounds. Thus, in our framework, θ incorporates the notion of the domain separation that improves the mistake bounds. For small values of θ, the source and target domains have high proximity such that the *initial target* hypothesis $\mathbf{v}_0$ lies reasonably close to the *optimal target* hypothesis $\mathbf{v}^*$. As a result, is such cases, O-Alda is expected to make a smaller number of mistakes to get to the optimal hypothesis.

Now, we present an intuitive argument for the lower label complexity of O-Alda as compared to single task online active settings. O-Alda is initialized with a *non-zero hypothesis* $\mathbf{v}_0 = \mathbf{w}_{src}$ learned using data from a related source domain. Hence, the sequence of hypotheses O-Alda produces will in expectation have higher confidences margins $|\bar{r}^i|$ as compared to some *zero initialized hypothesis*. Therefore, at each step the sampling probability of O-Alda given by $\frac{b}{b+|\bar{r}^i|}$ will also be smaller, which will lead to a smaller number of queried labels since it is nothing but $\sum_{i=1}^{T} \mathbb{E}[\frac{b}{b+|\bar{r}^i|}]$.