Learning Algorithms for Domain Adaptation

Manas A. Pathak and Eric H. Nyberg

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{manasp, ehn}@cs.cmu.edu

Abstract. A fundamental assumption for any machine learning task is to have training and test data instances drawn from the same distribution while having a sufficiently large number of training instances. In many practical settings, this ideal assumption is invalidated as the labeled training instances are scarce and there is a high cost associated with labeling them. On the other hand, we might have access to plenty of labeled data from a different domain, which can provide useful information for the present domain. In this paper, we discuss adaptive learning techniques to address this specific problem: learning with little training data from the same distribution along with a large pool of data from a different distribution. An underlying theme of our work is to identify situations when the auxiliary data is likely to help in training with the primary data. We propose two algorithms for the domain adaptation task: dataset reweighting and subset selection. We present theoretical analysis of behavior of the algorithms based on the concept of domain similarity, which we use to formulate error bounds for our algorithms. We also present an experimental evaluation of our techniques on data from a real world question answering system.

1 Introduction

In machine learning tasks, it is assumed that the labeled *primary training data* is similar to the test data in order to expect good accuracy on the test data. Further, it is important to have training data of a sizable amount in order to build a reliable model for classification. In practice, while it is usually time-consuming and expensive to acquire labeled instances which are similar to the test data, we might often have plenty of labeled data from an *auxiliary source* which is somewhat different from our test data. Despite this overall difference between the datasets, there may be parts of the auxiliary data that are similar and thus useful. More specifically, here we explore algorithms based on the assumption that the auxiliary data distribution is a mixture between the primary data distribution and a different distribution. In NLP tasks such as named-entity recognition, parsing, text classification, etc. we usually have plenty of data from standard text corpora (e.g. Penn Treebank [1]) but little data for specialized genres of text we might be interested in processing. In user-centric tasks such as spam detection, handwriting/voice

recognition, etc., there is usually little labeled data for each individual user who is using the system while there is a large amount of labeled data for all users combined.

A domain adaptation learning setting involves a *primary domain* $\mathcal{D}_P$ and an *auxiliary domain* $\mathcal{D}_A$. We mostly consider the problem of learning with two domains, although it is possible to generalize to handle multiple domains [2]. We denote the datasets sampled from primary and auxiliary domains as $(x_1^p, y_1^p), ...(x_{N_P}^p, y_{N_P}^p)$ and $(x_1^a, y_1^a), ...(x_{N_A}^a, y_{N_A}^a)$, where N_P and N_A are the sample sizes of the primary and auxiliary datasets respectively. A learning algorithm takes labeled training instances sampled from the two domains and estimates a labeling function for the primary domain. The test dataset for evaluating the learning algorithm is sampled from the primary domain. The central issue in such a learning setting is that we have few primary training instances and relatively large number of auxiliary training instances. The labeled primary instances can be thought of having a high *cost* associated with them as opposed to auxiliary instances, which have a lower cost. If we had large number of primary instances to start with, we could perform learning well enough with standard supervised learning approaches and would not need the auxiliary data. As it is difficult to learn a good labeling function with a small primary training dataset, we aim to make use of auxiliary data for learning. In fact, identifying situations when using auxiliary data with primary data helps in training is a fundamental question we are trying to answer in this paper.

2 Related Work

A major area of domain adaptation research is extending conventional supervised learning algorithms to handle data from multiple domains. Wu and Dietterich [3] proposed an extension to support vector machines called C-SVM for training with inadequate primary data and low quality auxiliary data in an image classification task. This is done by formulating the optimization problem with two separate loss functions and slack variables for each dataset. While they reported a noticeable improvement using auxiliary data, they did not present a quantitative study over varying auxiliary data. Liao et al. [4] improved on this work using a logistic regression based approach "M-Logit", with additional parameters for each auxiliary instance for controlling their contributions in the training process. However, in both of these works, there was limited analysis about identifying parameters which control the contribution of auxiliary data. There were some heuristics presented in [4] based on the size of primary and auxiliary data, but we found them to be unstable in practice. Also, there was no study of the underlying conditions of primary and auxiliary data when the domain adaptation is likely to provide an improvement. These two works were a major influence early on in our research. We modeled our domain adaptation algorithms (see section 3) with ideas drawn from these works. There have been several empirical studies of domain adaptation [5,6,7]. Jiang and Zhai [7] suggest purely empirical algorithms which in part inspire our theoretical analysis. We hope that our analysis can provide deeper understanding of the workings of these algorithms.

While there have been a large number of investigations into domain adaptation from an experimental perspective, theoretical studies have been limited in number. Only re-

cently, Ben-David et al. [8], presented the analysis of structure learning for domain adaptation in specific natural language problems. This was extended Blitzer et al. [9] who theoretically formulated the problem in terms of a domain similarity metric [10] and provided an elegant theory based on VC dimensions for a simple domain adaptation classifier. The work by Blitzer et al. is a major inspiration for this work. Our goal was to develop a similar theory for other domain adaptation algorithms to gain a better understanding about their behavior.

Mansour et al. [11] presented a more general theoretical analysis of the domain adaptation problem using Rademacher complexity [12] for a large class of regression and classification problems. We hope to further explore this direction of research in the future.

3 Domain Adaptation Algorithms

In this section, we propose various different learning algorithms for the domain adaptation task. Although the strategies are applicable generally, we present them in the framework of a logistic regression classifier for concreteness and simplicity. We model the class posterior probabilities with a sigmoid function $\sigma(s) = \frac{1}{1+e^{-s}}$. We assume the training set instances x_i with labels $y_i \in \{-1, 1\}$ to be i.i.d. The data log-likelihood $\ell(w)$ is equal to

$$\ell(w) = \sum_i \log \sigma(y_i w^T x_i). \tag{1}$$

The classification algorithm involves maximizing the data log-likelihood with respect to w and using the maximum likelihood value of $\hat{w}$ to classify the test instances using the sigmoid function $\sigma(\hat{w}^T x_i)$. The data log-likelihood ℓ is convex and can be optimized by gradient ascent.

3.1 Baseline

We treat the simple combination of primary and auxiliary data as a baseline domain adaptation technique. The data log-likelihood of the combined primary and auxiliary training set becomes

$$\ell(w) = \sum_i^{N_P} \log \sigma(y_i^p w^T x_i^p) + \sum_i^{N_A} \log \sigma(y_i^a w^T x_i^a) \tag{2}$$

With $N_A > N_P$, the effect of the auxiliary dataset in the training typically overwhelms that of the primary dataset. When the model is applied to the primary test set, the classification accuracy will be low compared to a model trained with adequate primary data. We use this intuition to develop algorithms which take the differences in domains into account.

3.2 Auxiliary Dataset Reweighting

In this algorithm, we decrease the importance of auxiliary dataset in training by a predefined amount. This will cause the learned model be more aligned towards primary domain distribution while retaining the generalization properties provided by large auxiliary instances. In context of the logistic regression framework defined above, we reduce the contribution of auxiliary instances in training by multiplying its label by a parameter $\alpha \in [0, 1]$.

$$\ell(w, \alpha) = \sum_{i}^{N_P} \log \sigma(y_i^p w^T x_i^p) + \sum_{i}^{N_A} \log \sigma(y_i^a w^T x_i^a \cdot \alpha) \tag{3}$$

When α is close to 0, $\log \sigma(y_i^a w^T x_i^a \cdot \alpha)$ will have a constant value and the auxiliary training instances (x_i^a, y_i^a) are discounted from training. The importance of auxiliary dataset in the training increases with α and when $\alpha = 1$, we get our baseline algorithm.

Even though this algorithm is simplistic, it provides insights into learning with two domains. The algorithm performs quite well in practice when there is little difference between primary and auxiliary datasets. Identifying $\hat{\alpha}$ which minimizes the primary test error raises some interesting questions.

3.3 A1: Auxiliary Subset Selection

While the auxiliary dataset reweighting algorithm works usually well in practice, it makes a simplistic assumption of treating all auxiliary instances equally. This can result in problems when we have a multi-modal auxiliary distribution and only part of it matches the primary data. One strategy is to directly select the auxiliary instances which have lower primary risk. As the information of primary distribution is not available, we approximate this by the primary empirical risk and identify subsets of auxiliary dataset which minimize it. As there are 2^{N_A} possible subsets of instances which can be selected, evaluating the primary empirical risk for each one of them is intractable. Even if we restrict ourselves to subset of a fixed predefined size k, we are left with $\binom{N_A}{k}$ selections which is still very large as N_A is large for most practical situations.

We propose a greedy algorithm called "A1"[1] for selecting the auxiliary subsets in an efficient way. Let S denote the size k set of auxiliary dataset to be selected. We start with initializing $S^{(0)}$ to k randomly selected auxiliary instances. We use the complete primary data and auxiliary data which is in $S^{(t)}$ for training the model $w^{(t+1)}$.

$$\hat{w}^{(t+1)} = \underset{w}{\operatorname{argmax}} \sum_{i}^{N_P} \log \sigma(y_i^p w^T x_i^p) + \sum_{(x_i^a, y_i^a) \in S^{(t)}} \log \sigma(y_i^a w^T x_i^a) \tag{4}$$

We apply this model $w^{(t+1)}$ to the auxiliary dataset and calculate the error between predicted value and auxiliary label for each auxiliary instance. The top k auxiliary instances minimizing this error are assigned to $S^{(t+1)}$.

$$S^{(t+1)} = \underset{top\ `k'\ i's}{\operatorname{argmin}} |y_i^a - 2\sigma(w^T x_i^a) + 1| \tag{5}$$

[1] The choice of name is non-mnemonic.

As the auxiliary labels are in $\{-1, 1\}$, we scale the predicted values given by the sigmoid function which are in $[0, 1]$ to that range. We repeat steps 4 and 5 until the selection into S does not change. In practice we observe that this algorithm converges within a few iterations.

As was the case with α in dataset reweighting, the value of selection parameter plays an important role in the algorithm. If k is too small, it is similar to training on the primary data which is small by itself. If k is too large, it is similar to optimizing the risk function on the combined primary and auxiliary data. In this way, the optimal value of k would be related to how similar the two domains are.

3.4 Soft A1 Algorithm

In the original A1 algorithm, we select k auxiliary instances for training and disregard the rest. One possible improvement is to discount the auxiliary instances partially. Instead of a set S, we consider an auxiliary weight vector $z \in [0, 1]^{N_A}$. The value of z_i indicates how much (x_i^a, y_i^a) is discounted in training. In order to avoid the learned model to discount all of auxiliary data, we fix $\sum z_i = k$. The parameter k has similar properties as in the original A1 algorithm.

We start with a randomly initialized $z^{(0)}$, summing to k and compute the learned model $w^{(t+1)}$.

$$\hat{w}^{(t+1)} = \underset{w}{\operatorname{argmax}} \sum_{i}^{N_P} \log \sigma(y_i^p w^T x_i^p) + \sum_{i}^{N_A} \log \sigma(y_i^a w^T x_i^a \cdot z_i^{(t)}) \tag{6}$$

We apply this model $w^{(t+1)}$ to the auxiliary dataset and calculate the error between predicted value and auxiliary label for each auxiliary instance. We assign $z_i^{(t+1)}$ the following value indicating how close was the predicted value to the true auxiliary label.

$$z_i^{(t+1)} = 1 - |y_i^a - 2\sigma(w^T x_i^a) + 1| \tag{7}$$

In order to maintain the $\sum z_i = k$ constraint, we assign the z_i values starting in the decending order. Once the cumulative sum becomes k, we set the remaining $z_i = 0$ as shown below.

```
sum = 0;
for (j in 1 to Na) {
    sum = sum + jth largest value in z;
    if (sum >  k) break;
}
set all values < jth largest value in z to 0;
```

3.5 Asymptotic Time Complexity Analysis

A unit step of the gradient ascent optimization in the logistic regression algorithm is computation of the log-likelihood function ℓ and its gradient. This involves the dot product $w^T x_i$ and summing over all instances x_i. In terms of the number of instances

N and features d, the time complexity of this step is $O(Nd)$. The gradient ascent optimization terminates only after convergence in either ℓ or w, hence its asymptotic time complexity is data dependent. The time complexity of the unit step in the baseline procedure is $O((N_P + N_A)\ d)$. As the auxiliary dataset reweighting algorithm involves multiplying the auxiliary instance term in the log-likelihood computation by a fixed parameter α, the time complexity of its unit step will also be $O((N_P + N_A)\ d)$. The unit step of computation in the A1 algorithm is executing logistic regression algorithm over k auxiliary instances are used along with N_P primary instances till convergence, which in turn will have a unit step of time complexity $O((N_P + k)\ d)$. Hence, it is much more expensive to execute the A1 algorithm than baseline or auxiliary dataset reweighting algorithm. For the soft A1 algorithm, the unit step of computation is the logistic regression algorithm with N_P primary instances and all N_A auxiliary instances multiplied by the z vector. This in turn will have a unit step time complexity same as the baseline of $O((N_P + N_A)\ d$. Hence, the soft A1 algorithm is more expensive to compute than the original A1 algorithm. To summarize, the order of asymptotic time complexities of the algorithms will be: baseline = auxiliary dataset reweighting $<<$ A1 $<$ Soft A1.

4 Theoretical Analysis

4.1 Domain Similarity

The central issue with domain adaptation is that we have limited primary data and auxiliary data is abundant but different from primary data. We aim to quantify the magnitude of the difference in this section. There are many standard measures of differences between probability distributions like KL-divergence and ℓ_P distance. However, as we do not have the knowledge of the exact distributions of the two domains beforehand, we need to estimate these measures from finite primary and auxiliary training datasets which will always have an associated error. To avoid the inconsistency in estimating the distance between domains from finite data samples, we consider a distance metric defined on hypothesis classes called $d_{\mathcal{H}}$ distance, which was originally proposed by [13] and [10]. It has been recently used by [8,9] as a foundation for investigating theoretical properties of domain adaptation.

Let $\mathcal{H}$ be a hypothesis class having a finite VC dimension of a given instance space $\mathcal{X}$. Let $\mathcal{A}_{\mathcal{H}} \subseteq 2^{\mathcal{X}}$ be the set of subsets of $\mathcal{X}$ such that $\mathcal{A}_{\mathcal{H}} = \{a | \exists\ h \in \mathcal{H}\ s.t.\ a = supp\ h\}$. Intuitively, $\mathcal{A}_{\mathcal{H}}$ contains sets of those $x \in \mathcal{X}$ which are labeled positively by some $h \in \mathcal{H}$. We calculate the absolute difference in probability of each subset $a \in \mathcal{A}_{\mathcal{H}}$, belonging to both the domain distributions $\mathcal{D}_P$ and $\mathcal{D}_A$. The $d_{\mathcal{H}}$ distance is the maximum difference in probability across all such subsets. It indicates that given a set of instances which are both classified in the same way, how different are the chances that would they have been generated from the two distributions.

Definition 1. *Let $\mathcal{A}_{\mathcal{H}} \subseteq 2^{\mathcal{X}}, \{x | x \in \mathcal{X}, h \in \mathcal{H}, h(x) = 1\} \in \mathcal{A}_{\mathcal{H}}$. $d_{\mathcal{H}}$ distance between two distributions $\mathcal{D}_P$ and $\mathcal{D}_A$ is defined as*

$$d_{\mathcal{H}}(\mathcal{D}_P, \mathcal{D}_A) = \sup_{a \in \mathcal{A}_{\mathcal{H}}} |\mathcal{D}_P(a) - \mathcal{D}_A(a)|. \tag{8}$$

In this original formulation, the $d_{\mathcal{H}}$ distance is not directly helpful when understanding the behavior of classifiers. Given the hypothesis space $\mathcal{H}$, we construct a symmetric difference set $\mathcal{H}\Delta\mathcal{H} = \{h(x)\ xor\ h'(x) | h, h' \in \mathcal{H}\}$. For any two hypotheses $h, h' \in \mathcal{H}$ which disagree in their labels for some $x \in \mathcal{X}$ ($h(x) = 1, h'(x) = 0$ or $h(x) = 0, h'(x) = 1$), there exists a hypothesis in $\mathcal{H}\Delta\mathcal{H}$ which labels x as 1. Similarly, if ($h(x) = h'(x) = 0$, $h(x) = h'(x) = 1$), there exists a hypothesis in $\mathcal{H}\Delta\mathcal{H}$ which labels x as 0. As before, the support set $\mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}$ contains all $x \in \mathcal{X}$ such that $h(x) \neq h'(x)$ for some two hypothesis $h, h' \in \mathcal{H}$. However, computing $d_{\mathcal{H}\Delta\mathcal{H}}$ is NP-hard even for hypothesis spaces with finite VC dimension [10]. Instead, we approximate this by training a linear classifier to discriminate between the primary and auxiliary domains. We evaluate the classifier over held-out data from the two domains and use the accuracy of classification as an empirical estimate $\hat{d}_{\mathcal{H}\Delta\mathcal{H}} \in [0, 1]$. For domains that are fully separable by a linear classifier, we will have $\hat{d}_{\mathcal{H}\Delta\mathcal{H}} = 1$. When both domains are completely indistinguishable, we have $\hat{d}_{\mathcal{H}\Delta\mathcal{H}} = 0$.

4.2 Learning Bounds

We begin by establishing a bound between primary and auxiliary risk functions in terms of the distance metric. The following useful theorem follows from using $\mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}$ as the basis set for $d_{\mathcal{H}\Delta\mathcal{H}}$ distance. The proof of all theorems is given in the Appendix.

Theorem 1. *Let h, h' be any two hypothesis belonging to the hypothesis space $\mathcal{H}$. Given the two domains $\mathcal{D}_P$ and $\mathcal{D}_A$, under the 0-1 loss function L_{01} and the corresponding risk functions R_P and R_A defined over the two domains,*

$$|R_A(h(x), h'(x)) - R_P(h(x), h'(x))| \leq d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_P, \mathcal{D}_A) \tag{9}$$

In section 3.1, we discussed the baseline domain adaptation algorithm which involves training with combined primary and auxiliary data. Theoretically, this is equivalent to minimizing the sum of primary and auxiliary risk together. The baseline hypothesis h^* is given by

$$h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}}\ R_P(h) + R_A(h). \tag{10}$$

We denote the risk of the baseline hypothesis by $\lambda = R_P(h^*) + R_A(h^*)$. This quantity is useful in getting an understanding of the domains. If λ is large, we cannot expect to do well on the while minimizing the auxiliary risk.

In both the auxiliary dataset reweighting and A1 algorithms, we reduce the importance of auxiliary dataset in training by introducing the α and k parameters. As both the parameters have the same effect of increasing the contribution of auxiliary data while varying between 0 and 1, we refer to both of them by α. The situation is slightly different in the A1 algorithm as size of the effective auxiliary dataset changes, but the theory generally holds in principle. We are working on developing this theory using additional information about the algorithm. The learning procedure minimizes the following combined true and empirical risk functions which we call true and empirical α risks.

$$R_\alpha(h) = R_P(h) + \alpha R_A(h), \hat{R}_\alpha(h) = \hat{R}_P(h) + \alpha \hat{R}_A(h) \tag{11}$$

We denote the hypothesis minimizing the empirical α risk $\hat{R}_\alpha(h)$ by $\hat{h}_\alpha$. As was the case before, when $\alpha = 0, R_\alpha(h)$ is equivalent to the primary risk and when $\alpha = 1, R_\alpha(h)$ is equivalent to the baseline risk. We now present a concentration of measure analysis to establish uniform convergence of empirical and true α risk. For clarity, let us denote the total number of instances $N_P + N_A$ by N and β as the fraction of primary instances $\frac{N_P}{N}$. The fraction of auxiliary instances will be $1 - \beta$. Using a similar argument and linearity of expectations, we show that $\hat{R}_\alpha$ is unbiased in the following lemma.

Lemma 1. *For a given hypothesis $h \in \mathcal{H}$, $\hat{R}_\alpha(h)$ is an unbiased estimator of $R_\alpha(h)$.*

$$\mathbb{E}\hat{R}_\alpha(h) = R_\alpha(h).$$

We use this result with the Hoeffding's inequality to establish the following error bound between the true and estimated α risk.

Theorem 2. *Let $\mathcal{H}$ be a hypothesis space with $VC(\mathcal{H}) = C$. Let N denote the total training instances $N_P + N_A$. Let β denote the fraction of primary instances $\frac{N_P}{N}$. Then, with probability $1 - \delta$, for every $h \in \mathcal{H}$,*

$$|\hat{R}_\alpha(h) - R_\alpha(h)| \leq \sqrt{\left(\frac{1}{\beta} + \frac{\alpha^2}{1-\beta}\right)\frac{C\log(2N/C) - \log\delta}{2N}}.$$

We use theorem 2 to formulate the following bound between the hypothesis $\hat{h}_\alpha$ which minimizes the α risk and the hypothesis h_P^* which minimizes the true primary risk.

Theorem 3. *Let $\mathcal{H}$ be a hypothesis space with $VC(\mathcal{H}) = C$. Let N denote the total training instances $N_P + N_A$. Let β denote the fraction of primary instances $\frac{N_P}{N}$. Let the true primary risk minimizer be $h_P^* = \operatorname{argmin}_{h\in\mathcal{H}} R_P(h)$ and the empirical α risk minimizer be $\hat{h}_\alpha = \operatorname{argmin}_{h\in\mathcal{H}} \hat{R}_\alpha(h)$. Then, with probability $1 - \delta$,*

$$R_P(\hat{h}_\alpha) \leq R_P(h_P^*) + \frac{2}{1+\alpha}\sqrt{\left(\frac{1}{\beta} + \frac{\alpha^2}{1-\beta}\right)\frac{C\log(2N/C) - \log\delta}{2N}} \quad + \alpha[\lambda + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_P, \mathcal{D}_A)]$$

The auxiliary dataset reweighting and A1 algorithms are equivalent to training on the primary dataset when $\alpha = 0$. If we substitute $\alpha = 0$ in the bound given by Theorem 3, we have the original uniform convergence bound on training with primary data.

h_P^* is the best hypothesis we can have as it will have the minimum true primary risk. This hypothesis will have the minimum possible error on the on the primary test data. As the auxiliary dataset reweighting algorithm minimizes the empirical α risk, the classifier generated by this algorithm will satisfy the above theorem. The key result of Theorem 3 is that we have a bound on how much excess error $\hat{h}_\alpha^*$ can possibly have on the primary test dataset compared to the best possible classifier. It is important to note that this bound contains two terms: $\frac{\alpha^2}{1-\beta}$ can be thought of excess error caused

due to limited primary data ($1-\beta$ factor). In the second term, α interacts with $d_{\mathcal{H}\Delta\mathcal{H}}$ which can be thought of excess error caused due to the difference between primary and auxiliary distribution. Hence, the parameter α effectively tries to make a trade-off between these two factors.

Theorem 3 gives us an upper bound on the excess primary risk of the hypothesis minimizing empirical α risk as a function of α, say, $f(\alpha)$. We denote the α independent expression inside the square root by c_1. To find the value of α minimizing the excess primary risk, we have:

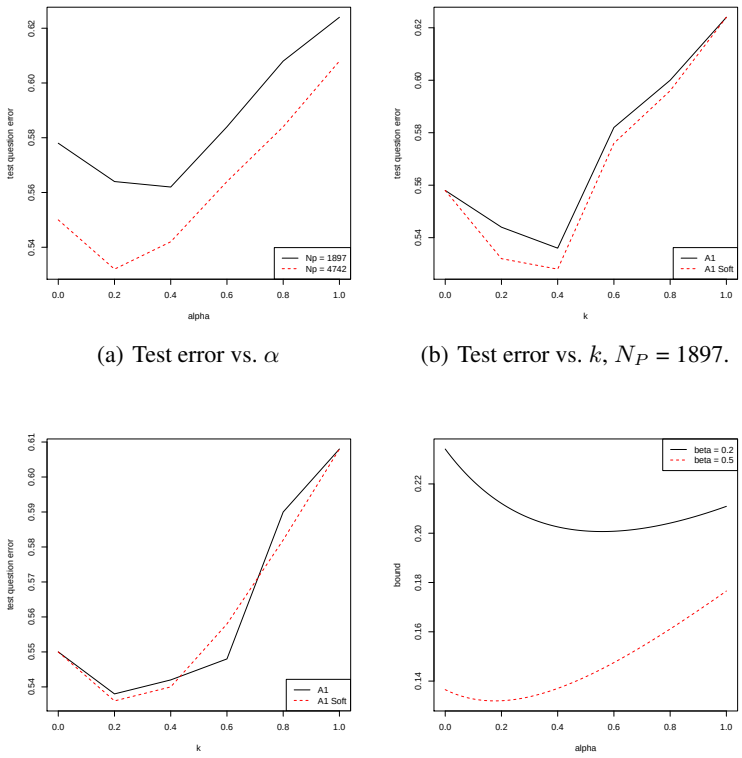
$$\frac{\partial}{\partial\alpha}f(\alpha) = \frac{-2}{(1+\alpha)^2}\sqrt{\left(\frac{1}{\beta}+\frac{\alpha^2}{1-\beta}\right)c_1} + \frac{1}{1+\alpha}\left[\left(\frac{1}{\beta}+\frac{\alpha^2}{1-\beta}\right)c_1\right]^{-1/2}\frac{2\alpha c_1}{1-\beta} + \lambda + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_P,\mathcal{D}_A) = 0. \quad (12)$$

Due to the nature of the equation, there is no simple closed form solution for α. For a given setting, as we do not know the value of true baseline risk λ beforehand, this would only give us a theoretical expression for α. We can get an empirical estimate using an estimate of λ from the primary and auxiliary datasets.

5 Experimental Results

We report experiences with our question answering (QA) system. For each question, our QA system produces answer candidates which are scored with 296 feature scores which are used to produce a ranked list of answers. We use *question error*, given by the percent of questions for which an incorrect answer appears ranked first as a metric to evaluate the ranking quality. We reduce the ranking of answer candidates to a classification problem. In training, manually vetted answer keys are used to label each question-answer pair as correct or incorrect. For testing, the probability of an answer being correct is used to rank the answers per question. For the primary dataset we used factoid questions from the Text REtrieval Conference (TREC) QA track [14]. TREC 8-10, which consist of 1,200 questions with 9,485 question-answer pairs, was used for training while TREC 11, which consists of 500 questions and 4,742 question-answer pairs, was used for test. As an auxiliary dataset we made use of a question set used for internal development which covers a broad range of general reference knowledge topics such as history, geography, arts and entertainment. This auxiliary QA set consists of 2,500 questions and 9,579 question-answer pairs.

We first evaluate the auxiliary data reweighting algorithm on this dataset. The primary test errors for different values of α are shown in Figure 1(a). We compare the classifiers trained on primary datasets of two different sizes N_P = 1897 (solid) and N_P = 4742 (dashed) both with full auxiliary data N_A = 9579. The error for the larger primary dataset is lower than the error for the smaller dataset. The error when only using the primary data $\alpha = 0$ is lower than the error when using the primary and auxiliary data together $\alpha = 1$. This can be attributed to the qualitative differences in the two datasets. We observe that the minimum test error for the algorithm in both cases is observed around $\alpha = 0.4$ and $\alpha = 0.2$ for indicating that we require less contribution from auxiliary data for training as our primary data increases.



(a) Test error vs. α

(b) Test error vs. k, N_P = 1897.

(c) Test error vs. k, N_P = 4742.

(d) Bound values vs. α

Fig. 1. Experimental results for QA data

We evaluate both the original and soft versions of A1 algorithm on this dataset. We compare the primary test question error for the classifiers trained on primary datasets of two different sizes N_P = 1897 shown in Figure 1(b) and N_P = 4742 shown in Figure 1(c), both with full auxiliary data N_A = 9579. We see that the test error is clearly lower when we use larger primary dataset. Similar to the auxiliary dataset reweighting, the test error increases with increasing k up to a point and then starts increasing. This means that as we go on adding auxiliary instances in training, the algorithm is able to identify parts of the auxiliary data which are useful for training. Also, we see that the soft A1 algorithm performs a lot better than the original A1 algorithm. This can be attributed to the relative flexibility of the algorithm in choosing the auxiliary instances. For the larger primary dataset with N_P = 4742, we see that the test error for the A1 algorithm increases beyond a point after adding additional auxiliary instances. This means that the primary data is sufficient for training and adding auxiliary instances introduces noise. However, the soft A1 algorithm does not perform much better than the original A1

algorithm, but it still has the minimum error at k = 0.2. Again, we observe that the value of k which has the minimum error is lower for larger value of N_P.

It is seen that the performance of both the algorithms is varies with the choice of α and k parameters and the changing contribution of the auxiliary data in training. As we have seen before, there is no simple closed form solution for the α parameter minimizing the theoretical excess risk given by Theorem 3. However, we can choose the parameter values empirically by performing cross-validation using different samples of primary data and auxiliary data, and evaluating on remainder of the primary data.

We evaluate the bound given in Theorem 3 for this dataset. For the sake of comparing the primary datasets of two sizes, we assign $\lambda + d_{\mathcal{H}\Delta\mathcal{H}}$ a constant value 0.1. β takes the values 0.2 and 0.5. Figure 1(d) shows the error bounds for this setup. We see that the experimental results are very similar in shape to the bound values. For both the curves, the minimum value $\hat{\alpha}$ is smaller for $\beta = 0.5$ as compared to $\beta = 0.2$ which is confirmant with our theory. Note that the values of the bound are relative to the true primary risk and not be directly compared to test error values. We see that the experimental results are very similar in shape to the bound values. For both the curves, the minimum value $\hat{\alpha}$ is smaller for $\beta = 0.5$ as compared to $\beta = 0.2$ which is confirmant with our theory.

(a) N_P = 1897.

Algorithm	**Parameter**	**Test Error**	**Time**
Primary		0.588	32.52
Baseline		0.624	153.87
Reweighting	$\alpha = 0.4$	0.562	161.07
A1	$k = 0.4$	0.536	325.15
Soft A1	$k = 0.4$	**0.528**	721.38
M-Logit		0.554	448.62

(b) N_P = 4742.

Algorithm	**Parameter**	**Test Error**	**Time**
Primary		0.550	53.67
Baseline		0.608	260.75
Reweighting	$\alpha = 0.2$	**0.532**	274.03
A1	$k = 0.2$	0.538	629.87
Soft A1	$k = 0.2$	0.536	1132.23
M-Logit		0.546	627.52

Table 1. Primary test errors and execution times (in seconds) for QA data.

In Tables 1(a) and 1(b), we report the comparative performance of each algorithm including primary test errors and execution times[2] for the two primary dataset samples of size N_P = 1897 and N_P = 4742. We used our implementation of M-Logit based on [4] as a state of the art method for comparison. The errors were always found to be lower when we train with the larger primary dataset. For N_P = 1897, which is 20% of the primary data, the soft A1 algorithm achieves a substantial 15.39% improvement over the baseline and 10.20% improvement over training with primary data. This error is even lower than the baseline approach with N_P = 4742 which is 50% of the primary data. We have a 13.16% improvement over the baseline and 4% improvement with training with primary data alone. For N_P = 1897, the M-Logit algorithm performs better than baseline and dataset reweighting algorithms, while for N_P = 4742, the reweighting

[2] Execution times are reported for the experiments conducted in the R programming environment running over a 64-bit GNU/Linux machine with dual 2 GHz processors and 3 GB RAM.

algorithm performs better than M-Logit. In both cases, the A1 and soft A1 algorithm have lower primary test errors than M-Logit. The A1 and soft A1 algorithms take much longer to execute than the baseline and reweighting algorithms, which in turn are much slower than training with primary data.

6 Conclusion

In this paper, we investigated the problem of domain adaptation which is learning with little training data from the same distribution along with large amount of data from a different distribution. We introduced our domain adaptation algorithms in the logistic regression classifier framework. The auxiliary dataset reweighting algorithm modifies the contribution of auxiliary data in training with the α parameter. The A1 algorithm efficiently selects the instances from auxiliary data which are likely to minimize error in training with primary data. We also discussed a soft variant of A1 algorithm which partially discounts auxiliary instances from training by a fixed amount. We explored the concept of domain similarity with the hypothesis class based $d_{\mathcal{H}\Delta\mathcal{H}}$ distance metric and used it to develop a theoretical framework for analyzing domain adaptation methods. We presented an error bound for the auxiliary dataset reweighting algorithm which indicated a tradeoff between the domain similarity and size of the primary training data. We presented an experimental analysis of our algorithms over data from a question answering system. The experimental results were found to be closely aligned with our theory.

A few directions for making further enhancements in this research include performing a wider evaluation with other datasets and along with comparisons other techniques. It would be interesting to compare our approach to other paradigms like semi-supervised learning. As we alluded before, we are currently working towards establishing tighter error bounds for the A1 algorithm. We think there is also some real potential for improvement by exploring the problem with other complexity classes like Rademacher complexity. In all of the work here, we considered only simple cost functions where all auxiliary instances and features are equal in training. Extending our approaches to handle more complex cases would also be an interesting direction.

7 Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments.

References

1. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of english: The penn treebank. Computational Linguistics **19**(2) (1994) 313–330
2. Crammer, K., Kearns, M.J., Wortman, J.: Learning from multiple sources. In: Advances in Neural Information Processing Systems 19. (2006) 321–328
3. Wu, P., Dietterich, T.: Improving svm accuracy by training on auxiliary data sources. In: Proceedings of the 21st International Conference on Machine Learning. (2004) 871–878

4. Liao, X., Xue, Y., Carin, L.: Logistic regression with an auxiliary data source. In: Proceedings of the 22nd International Conference on Machine Learning. (2005) 505–512
5. Chelba, C., Acero, A.: Adaptation of maximum entropy capitalizer: Little data can help a lot. In: Proceedings of EMNLP 2004. (2004) 285–292
6. Bickel, S., Bruckner, M., Scheffer, T.: Discrimminative learning for differing training and test distributions. In: Proceedings of the 24th International Conference on Machine Learning. (2007) 81–88
7. Jiang, J., Zhai, C.: Instance weighting for domain adaptation in nlp. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics. (2007) 264–271
8. Ben-david, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: NIPS. (2006)
9. Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Wortman, J.: Learning bounds for domain adaptation. In: Advances in Neural Information Processing Systems 20. (2007) 129–136
10. Kifer, D., Ben-David, S., Gehrke, J.: Detecting change in data streams. In: Proceedings of the Thirtieth international conference on very large databases. (2004) 180–191
11. Mansour, Y., Mohri, M., Rostamizadeh, A.: Domain adaptation: learning bounds and algorithms. arXiv (2009)
12. Bartlett, P.L., Mendelson, S., Long, M.: Rademacher and gaussian complexities: Risk bounds and structural results. Journal of Machine Learning Research **3** (2002)
13. Devroye, L., Gyorfi, L., Lugosi, G. In: A probabilistic theory of pattern recognition. Springer (1996) 271–272
14. Voorhees, E.M., Harman, D.: Overview of the eighth text retrieval conference (trec-8). In: In Proceedings of the Eighth Text REtrieval Conference (TREC-8. (1999)

Appendix

– Proof of Theorem 1.

$$\begin{aligned}
\left|R_A(h(x),h'(x)) - R_P(h(x),h'(x))\right| &= \left|\mathbb{E}_{x\sim A}L_{01}(h(x),h'(x)) - \mathbb{E}_{x\sim P}L_{01}(h(x),h'(x))\right| \\
&= \left|\sum_{x\sim A} L_{01}(h(x),h'(x))\mathcal{D}_A(x) - \sum_{x\sim P} L_{01}(h(x),h'(x))\mathcal{D}_P(x)\right| \\
&= \left|\sum_{x\in\mathcal{A}} L_{01}(h(x),h'(x))\mathcal{D}_A(x) + \sum_{x\notin\mathcal{A}} L_{01}(h(x),h'(x))\mathcal{D}_A(x)\right. \\
&\left. + \sum_{x\in\mathcal{A}} L_{01}(h(x),h'(x))\mathcal{D}_P(x) + \sum_{x\notin\mathcal{A}} L_{01}(h(x),h'(x))\mathcal{D}_P(x)\right| \quad \text{(for some } \mathcal{A}\in\mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}\text{)} \\
&= \left|\sum_{x\in\mathcal{A}}\mathcal{D}_A(x) - \sum_{x\in\mathcal{A}}\mathcal{D}_P(x)\right| = |\mathcal{D}_A(\mathcal{A}) - \mathcal{D}_P(\mathcal{A})| \leq \sup_{\mathcal{A}\in\mathcal{A}_{\mathcal{H}\Delta\mathcal{H}}} |\mathcal{D}_A(\mathcal{A}) - \mathcal{D}_P(\mathcal{A})| = d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_p,\mathcal{D}_A)
\end{aligned}$$

□

– Proof of Lemma 1. We define the random variables $Z_1, ...Z_{N_P}$ to take the values $\frac{1}{\beta}|y-h(x)| \in \left[0, \frac{1}{\beta}\right]$ corresponding to the primary instances $x_1, ..., x_{N_P} \in \mathcal{X}_P \sim \mathcal{D}_P$. Similarly, define the random variables $Z_{N_P+1}, ..., Z_N$ to take the values $\frac{1}{1-\beta}|y-h(x)| \in \left[0, \frac{1}{1-\beta}\right]$ corresponding to the auxiliary instances $x_1, ..., x_{N_A} \in \mathcal{X}_A \sim \mathcal{D}_A$. From the definition of α risk, we have

$$\hat{R}_\alpha(h) = \hat{R}_P(h) + \alpha\hat{R}_A(h) = \frac{1}{N_P}\sum_{x\in\mathcal{X}_P}|y-h(x)| + \frac{\alpha}{N_A}\sum_{x\in\mathcal{X}_A}|y-h(x)|$$

$$= \frac{1}{N}\left[\frac{N}{N_P}\sum_{x\in\mathcal{X}_P}|y-h(x)| + \frac{N\alpha}{N_A}\sum_{x\in\mathcal{X}_A}|y-h(x)|\right] = \frac{1}{N}\left[\frac{1}{\beta}\sum_{x\in\mathcal{X}_P}|y-h(x)| + \frac{\alpha}{1-\beta}\sum_{x\in\mathcal{X}_A}|y-h(x)|\right]$$

$$= \frac{1}{N}\left[\sum_{i=1}^{N_P} Z_i + \sum_{i=N_P+1}^{N_A} Z_i\right] = \frac{1}{N}\sum_{i=1}^{N} Z_i = \bar{Z}.$$

$$\mathbb{E}\hat{R}_\alpha(h) = \mathbb{E}\left[\hat{R}_P(h) + \alpha\hat{R}_A(h)\right] = \mathbb{E}\left[\frac{1}{N_P}\sum_{x\in\mathcal{X}_P}|y-h(x)| + \frac{\alpha}{N_A}\sum_{x\in\mathcal{X}_A}|y-h(x)|\right]$$

$$= \frac{1}{N_P}\sum_{x\in\mathcal{X}_P}\mathbb{E}|y-h(x)| + \frac{\alpha}{N_A}\sum_{x\in\mathcal{X}_A}\mathbb{E}|y-h(x)| = \frac{1}{N_P}\sum_{x\in\mathcal{X}_P}R_P(h) + \frac{\alpha}{N_A}\sum_{x\in\mathcal{X}_A}R_A(h)$$

$$= R_P(h) + \alpha R_A(h) = R_\alpha(h).$$

□

– Proof of Theorem 2.

$$\mathbb{P}\left[|\hat{R}_\alpha(h) - R_\alpha(h)| > t\right] = \mathbb{P}\left[|\bar{Z} - \mathbb{E}\bar{Z}| > t\right] \leq 2exp\left[\frac{-2N^2t^2}{\sum_{i=1}^{N}(\max(Z_i) - \min(Z_i))^2}\right]$$

$$= 2exp\left[\frac{-2N^2t^2}{\sum_{i=1}^{N_P}\frac{1}{\beta^2} + \sum_{i=1}^{N_P}\frac{\alpha^2}{(1-\beta)^2}}\right] = 2exp\left[\frac{-2N^2t^2}{\frac{N_P}{\beta^2} + \frac{N_A\alpha^2}{(1-\beta)^2}}\right]$$

$$= 2exp\left[\frac{-2N^2t^2}{\frac{N}{\beta} + \frac{N\alpha^2}{1-\beta}}\right] = 2exp\left[\frac{-2Nt^2}{\frac{1}{\beta} + \frac{\alpha^2}{1-\beta}}\right].$$

The above result holds true for a single $h \in \mathcal{H}$. To generalize for the whole hypothesis class, we proceed with the standard uniform bound argument with VC dimensions as our growth function. Let $VC(\mathcal{H}) = C$.

$$\forall h \in \mathcal{H}, \mathbb{P}\left[|\hat{R}_\alpha(h) - R_\alpha(h)| > t\right] \leq \left(\frac{2N}{C}\right)^C exp\left[\frac{-2Nt^2}{\frac{1}{\beta} + \frac{\alpha^2}{1-\beta}}\right].$$

It should be noted that as $N \to \infty$, $\hat{R}_\alpha(h) \to R_\alpha(h)$ in probability. Hence, $\hat{R}_\alpha(h)$ is a consistent estimator of $R_\alpha(h)$.
We equate the RHS by δ and solve for t.

$$t = \sqrt{\left(\frac{1}{\beta} + \frac{\alpha^2}{1-\beta}\right)\frac{C\log(2N/C) - \log\delta}{2N}}.$$

With this, we get the following relationship.

$$\forall h \in \mathcal{H}, \mathbb{P}\left[|\hat{R}_\alpha(h) - R_\alpha(h)| > \sqrt{\left(\frac{1}{\beta} + \frac{\alpha^2}{1-\beta}\right)\frac{C\log(2N/C) - \log\delta}{2N}}\right] = \delta.$$

□

– Proof of Theorem 3. For any $h \in \mathcal{H}$,

$$|R_\alpha(h) - R_P(h)| = \alpha |R_A(h)|$$

$$= \alpha \left[|R_A(h) - R_A(h, h^*)| + |R_A(h, h^*) - R_P(h, h^*)| + |R_P(h, h^*) - R_P(h)| + R_P(h) \right]$$

$$\leq \alpha \left[|R_A(h^*)| + |R_A(h, h^*) - R_P(h, h^*)| + |R_P(h^*)| + R_P(h) \right] \quad \text{(triangle inequality)}$$

$$\leq \alpha \left[(R_A(h^*) + R_P(h^*)) + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_P, \mathcal{D}_A) + R_P(h) \right] \quad \text{(theorem 1)}$$

$$= \alpha \left[\lambda + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_P, \mathcal{D}_A) + R_P(h) \right].$$

$$\Rightarrow R_P(h) \leq R_\alpha(h) - \alpha[\lambda + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_P, \mathcal{D}_A) + R_P(h)]$$

$$(1 + \alpha) R_P(h) \leq R_\alpha(h) - \alpha[\lambda + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_P, \mathcal{D}_A)]$$

$$R_P(h) \leq \frac{1}{1+\alpha} R_\alpha(h) - \frac{\alpha}{1+\alpha}[\lambda + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_P, \mathcal{D}_A)] \tag{13}$$

We apply this inequality for the empirical α risk minimizer $\hat{h}_\alpha$.

$$R_P(\hat{h}_\alpha) \leq \frac{1}{1+\alpha} R_\alpha(\hat{h}_\alpha) - \frac{\alpha}{1+\alpha}[\lambda + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_P, \mathcal{D}_A)]$$

$$\leq \frac{1}{1+\alpha} \hat{R}_\alpha(\hat{h}_\alpha) + \frac{1}{1+\alpha} \sqrt{\left(\frac{1}{\beta} + \frac{\alpha^2}{1-\beta} \right) \frac{C \log(2N/C) - \log \delta}{2N}}$$

$$- \frac{\alpha}{1+\alpha}[\lambda + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_P, \mathcal{D}_A)] \quad \text{(theorem 2)}$$

$$\leq \frac{1}{1+\alpha} \hat{R}_\alpha(h_P^*) + \frac{1}{1+\alpha} \sqrt{\left(\frac{1}{\beta} + \frac{\alpha^2}{1-\beta} \right) \frac{C \log(2N/C) - \log \delta}{2N}}$$

$$- \frac{\alpha}{1+\alpha}[\lambda + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_P, \mathcal{D}_A)] \quad (\hat{h} = \operatorname*{argmin}_{h \in \mathcal{H}} \hat{R}_\alpha(h))$$

$$\leq \frac{1}{1+\alpha} R_\alpha(h_P^*) + \frac{2}{1+\alpha} \sqrt{\left(\frac{1}{\beta} + \frac{\alpha^2}{1-\beta} \right) \frac{C \log(2N/C) - \log \delta}{2N}}$$

$$- \frac{\alpha}{1+\alpha}[\lambda + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_P, \mathcal{D}_A)] \quad \text{(theorem 2)}$$

$$\leq R_P(h_P^*) + \frac{2}{1+\alpha} \sqrt{\left(\frac{1}{\beta} + \frac{\alpha^2}{1-\beta} \right) \frac{C \log(2N/C) - \log \delta}{2N}}$$

$$+ \alpha[\lambda + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_P, \mathcal{D}_A)].$$

□