Domain Adaptation with Ensemble of Feature Groups

Rajhans Samdani[*]
University of Illinois at Urbana-Champaign
Urbana, IL, USA
rsamdan2@illinois.edu

Wen-tau Yih
Microsoft Research
Redmond, WA, USA
scottyih@microsoft.com

Abstract

We present a novel approach for domain adaptation based on feature grouping and re-weighting. Our algorithm operates by creating an ensemble of multiple classifiers, where each classifier is trained on one particular feature group. Faced with the distribution change involved in domain change, different feature groups exhibit different cross-domain prediction abilities. Herein, ensemble models provide us the flexibility of tuning the weights of corresponding classifiers in order to adapt to the new domain. Our approach is supported by a solid theoretical analysis based on the expressiveness of ensemble classifiers, which allows trading-off errors across source and target domains. Moreover, experimental results on sentiment classification and spam detection show that our approach not only outperforms the baseline method, but is also superior to other state-of-the-art methods.

1 Introduction

Discriminative learning models work effectively when the training and testing examples are drawn from the same distribution. However, in several real-world applications, it is often highly desirable to train a classifier from one *source* domain, and apply it to a similar but different *target* domain, where the annotated data is unavailable or expensive to create. One example of this scenario is to learn a text categorizer from a large collection of labeled newswire articles, but use it to process regular Web documents. In this *domain adaptation* setting, the goal is to leverage the data available in the source domain to improve the accuracy of the model when testing on *target* domain examples.

As observed by previous studies, unfortunately, naïvely applying classifiers to a different domain often leads to considerable performance degradation [Daumé III, 2007; Jiang and Zhai, 2007]. Consequently, a number of approaches have been proposed recently to address the problem of domain adaptation. Some methods focus on re-weighting training instances from different domains [Jiang and Zhai, 2007; Bickel *et al.*, 2009], while others modify the feature space to capture domain-specific and domain-invariant aspects [Blitzer *et al.*, 2006; Daumé III, 2007; Finkel and Manning, 2009; Jiang and Zhai, 2006]. Despite the fact that these methods adapt seemingly very different strategies, the shared rationale behind is to bring the empirical source distribution closer to the target domain, and thus increase the accuracy of the classifier when evaluated on the target domain data.

In this paper, we present a novel ensemble-based approach for domain adaptation, based on *feature re-weighting*. Our method builds on the observation that when moving from the source domain to the target domain, different features may have different levels of distributional change. For instance, in email spam detection, the distribution of content-based features changes heavily over time as it is very easy for spammers to revise the subject and body of a spam message. In contrast, the distribution of features based on user-preference or sender-IP does not change as much. This phenomenon can be easily observed empirically, which we demonstrate in Sec. 4.2. Intuitively, a classifier trained on distributionally stable features is likely to behave more consistently than a classifier based on an unstable feature group. Therefore, in addition to the expressiveness of its feature space in predicting the label, the importance of each classifier should also depend on the distributional stability of its features. Given a set of feature groups that capture this notion, where the grouping can be decided by domain knowledge or statistics derived from unlabeled data, we first train individual classifiers separately using only the corresponding group of features. The final model is a weighted ensemble of individual classifiers, where the weights are tuned based on the performance of the ensemble on a small amount of labeled target data. Compared to the existing approaches, our method is unique in that it considers the cross-domain behavior of different feature groups directly in terms of their classification accuracy. Afterwards, instead of creating an instance distribution close to the target domain, it adjusts the influence of features on the final classifier by tuning their weights.

Our approach is supported both by a solid theoretical analysis and a strong empirical validation. We present a generalization bound for the style of ensembles mentioned above based on the d_H-distance framework proposed by Ben-David *et al.* [2007]. We show that ensembles provide an additional degree of freedom in the form of tunable weights of individ-

[*] This research was conducted during author's internship at Microsoft Research.

ual classifiers which can be leveraged to adapt to the target domain. Empirically, when applied to the problems of *sentiment analysis* and *spam detection*, our approach outperforms several state-of-the-art methods, as well as the strong baseline that is trained simply on combined source and target data.

The rest of the paper is organized as follows. We first formalize the problem setting in Sec. 2. Then we show our algorithm along with the theoretical analysis in Sec. 3, followed by the experimental results in Sec. 4. Closely related work is surveyed in Sec. 5 and Sec. 6 concludes the paper with some discussion and future work.

2 Problem Setting

Let $\mathcal{X} = \{0,1\}^p$ be the feature space from which a p-dimensional feature $X = \{X_1, \dots, X_p\}$ is sampled and let $\mathcal{Y} = \{-1, 1\}$ be the output space. Suppose $D_S(\mathbf{x}, y)$ and $D_T(\mathbf{x}, y)$ are two distributions over $\mathcal{X} \times \mathcal{Y}$ — the former is called the *source* distribution and the latter, the *target* distribution. Let L_S and L_T be labeled examples sampled from D_S and D_T, respectively. In the setting of domain adaption, usually $|L_S| \gg |L_T|$. Moreover, we assume that some unlabeled data from the target domain, $U_T \sim D_T(\mathbf{x})$, is available.

Let $\Delta : \Re \times \Re \to \{0,1\}$ be the standard zero-one loss function: $\Delta(z_1, z_2)$ is zero if $z_1 z_2 \geq 0$, one otherwise. The idea behind domain adaptation is to leverage L_S and U_T, in addition to L_T, to learn a hypothesis $h_t : \mathcal{X} \to \mathcal{Y}$ with low expected error on target, defined by $\Delta_{D_T}(h_t) = E_{(x,y)\sim D_T}[\Delta(h_t(x), y)]$. Of course, L_S can help only if D_S and D_T are not "too different".

Although the feature space defined here is binary (i.e., $\mathcal{X} \subseteq \{0,1\}^p$), our analysis and technique, by no means, are restricted to discrete feature spaces. In the text applications considered in this paper, such a setting is very common; for example, the occurrences of unigrams or bigrams in the text are encoded as binary features. Also, the number of features, p, is typically much larger than the number of training examples.

Last, we assume that we have a decomposition of feature vector X into $r+1$ (possibly overlapping) feature vectors or groups: $X^1, X^2, \dots, X^r \subset X$ with $X^0 = X$; that is, X^i is a subset of features contained in X. Essentially, our approach considers ensembles of classifiers where each classifier is trained on one particular feature group. The nature and purpose of these groupings would become clear in section 3.

3 Feature Ensembles for Domain Adaptation

In this section, we propose FEAD, a Feature Ensembles approach for Domain Adaptation, which learns a classifier for the target domain in the form of an ensemble of feature group classifiers. The design of FEAD is motivated by a VC-style generalization bound analysis for ensemble classifiers in the context of domain adaptation. Compared to existing domain adaption methods, FEAD provides an additional degree of freedom to adjust the trade-off between the generalization error and domain distribution change of *individual* classifiers trained on the corresponding feature groups. Although the selection of the underlying learning algorithm is not restricted, we give an interesting insight into FEAD via a concrete instantiation of our ensemble approach when using logistic regression.

Algorithm 1 Algorithm for Feature Ensembles for Domain Adaptation (FEAD)

```
1: Given: Data: $L_S$ and $L_T$; feature groups: $X_0, \dots, X_r$; hypotheses spaces: $H_0, \dots, H_r$; convex loss function: $\Delta^c$
2: for $i = 0$ to $r$ do
3:     learn: $h_S^i \leftarrow \arg\min_{h \in H_i} \Delta_{L_S}^c(h)$
4: end for
5: $\overline{\alpha} \leftarrow \arg\min_{\overline{\alpha}' \geq 0} \sum_{(\mathbf{x},y)\in L_T} \Delta(\sum_i \alpha_i' h_S^i(\mathbf{x}), y)$
6: return $\mathbf{w_t} = \sum_{i=0}^r \alpha_i h_S^i$
```

3.1 Algorithm

Before introducing our algorithm, we first give some notations that are useful throughout the whole section. Let $\Delta^c : \Re \times \Re \to \Re_+$ be a convex loss function upper bounding Δ. For a set of labeled examples L, $\Delta_L^c(f) = \frac{1}{|L|}\sum_{(x,y)\in L} \Delta^c(f(x), y)$ denotes the empirical loss of the hypothesis f on L. Also, let $\Delta_D(f, g) = E_{(x,y)\sim D}[\Delta(f(x), g(x))]$ be the expected zero-one loss between hypotheses f and g on a distribution D. Let H be the hypothesis space being considered. For example, H can be the space of all linear classifiers.

For a feature group $X^i \subseteq X$: let $D_S^i(x, y) = \frac{D_S(X^i = x^i, y)}{Z_i}$ be the marginal source distribution over (X^i, y) where Z_i is chosen to making D_S^i a valid distribution; let H_i be the space of hypotheses defined over X^i (clearly $H_0 = H$ and for several popular, including linear, hypothesis spaces, $H_i \subseteq H$); and let $h_i^S = \arg\min_{h \in H_i} \Delta_{L_S}^c(h)$ be a classifier in H_i minimizing the loss on L_S with $\Delta_{L_S}^c(h_i^S) = \epsilon_i^S$.

We express the final classifier as an ensemble of classifiers given by $h(\overline{\alpha}) = \sum_{i=0}^r \alpha_i h_i^S$, where $\overline{\alpha} = (\alpha_0, \dots, \alpha_r) \geq 0$. $\overline{\alpha}$ is learned by minimizing the empirical target error. The whole learning procedure is delineated in Algorithm 1. On line 5, we can use grid search to find a good $\overline{\alpha}$, given the small search space. Empirically, we found that applying a linear classification technique like logistic regression is more efficient and performs well, which is used in our experiments.

Note that $\overline{\alpha} \geq 0$ (Line 5) is not a restrictive assumption as in the event some h_i^S is rendered ineffectual on the target domain due to drastic distribution change of X^i, α_i can be simply set to zero — it is highly unlikely for classification decisions to be reversed for any feature group. Since Δ is invariant to positive re-scaling, we assume that $\sum_i \alpha_i = 1$.

3.2 Theoretical Analysis

In order to present generalization results for feature groups-based ensembles, we use a classifier-induced distance for distributions called the d_H-distance — related to the *generalized Kolmogorov-Smirnov distance* [Devroye *et al.*, 1996] — introduced for domain adaptation by Ben-David *et al.* [2007].[1]

[1] We slightly modify the notation to suit our analysis.

Definition 1 ($\mathbf{d_H}$-distance [Ben-David *et al.*, 2007]). *Let H be a set of hypotheses, mapping $\mathcal{X}$ to $\Re$. The d_H-distance between two distributions Q_1 and Q_2 over $\mathcal{X}$ is defined as*

$$d_H(Q_1, Q_2) = \max_{h,h' \in H} |\Delta_{Q_1}(h, h') - \Delta_{Q_2}(h, h')| \ .$$

Intuitively, $d_H(Q_1, Q_2)$ bounds how closely we can predict the loss between two hypotheses on Q_2 if we know their loss on Q_1. We first present two propositions regarding d_H.

Proposition 1 (Convexity). *For any distributions D, D_1, and D_2 with $D_\lambda = \lambda D_1 + (1 - \lambda)D_2, \lambda \geq 0$, we have*

$$d_H(D, D_\lambda) \leq \lambda d_H(D, D_1) + (1 - \lambda)d_H(D, D_2) \ .$$

Proof.

$$\begin{aligned} d_H(D, D_\lambda) &= \max_{h,h' \in H} |\Delta_D(h, h') - \Delta_{D_\lambda}(h, h')| \\ &= \max_{h,h' \in H} |\left(\lambda + (1 - \lambda)\right) \Delta_D(h, h') \\ &\quad - (\lambda \Delta_{D_1}(h, h') + (1 - \lambda)\Delta_{D_2}(h, h'))\,| \\ &\leq \max_{h,h' \in H} \big(\lambda|\Delta_D(h, h') - \Delta_{D_1}(h, h')| \\ &\quad +(1 - \lambda)|\Delta_D(h, h') - \Delta_{D_2}(h, h')|\big) \\ &\leq \max_{h,h' \in H} \lambda|\Delta_D(h, h') - \Delta_{D_1}(h, h')| \\ &\quad +(1 - \lambda) \max_{h,h' \in H} |\Delta_D(h, h') - \Delta_{D_2}(h, h')| \\ &= \lambda d_H(D, D_1) + (1 - \lambda)d_H(D, D_2) \ . \end{aligned}$$

□

Proposition 2 (Triangle inequality). *For any distributions D_1, D_2 and D' we have*

$$d_H(D_1, D_2) \leq d_H(D_1, D') + d_H(D', D_2) \ .$$

Proof. Similar to the proof of proposition 1. □

We now have the following bound, which is similar to Theorem 3 in [Ben-David *et al.*, 2010].

Theorem 1. *For all $h = \sum_i \alpha_i h_i^S$ with $\alpha_i \geq 0$ $\forall i$ and $\sum_i \alpha_i = 1$, the following bound holds with probability at least $1 - \delta$ for $\delta > 0$ and for all $\beta \in [0, 1]$*

$$\begin{aligned} \Delta_{D_T}(h) &\leq (1 - \beta)\Big(\epsilon^* + \sum_i \alpha_i \big(\epsilon_i^S + d_H(D_S, D_S^i) \\ &\quad +d_H(D_S^i, D_T)\big)\Big) + \epsilon(p_H, \beta, \delta) \qquad (1) \\ &\quad +\beta \Delta_{L_T}(h) \qquad (2) \end{aligned}$$

where $D_S^i(x, y) = \frac{D_S(X^i = x^i, y)}{Z_i}$ is the normalized marginal source distribution over (X^i, y), $\epsilon^* = \arg\min_{h' \in H}(\Delta_{D_T}(h') + \Delta_{D_S}(h'))$ and $\epsilon(p_H, \beta, \delta) = 2\sqrt{\frac{\beta^2}{\eta} + \frac{(1-\beta)^2}{1-\eta}} \sqrt{\frac{2p_H \log(2(m+1)) + 2\log(\frac{8}{\delta})}{m}}$ where $m = |L_S| + |L_T|, \eta = \frac{|L_T|}{m}$, and p_H is the VC Dimension of H.

Proof. Using $D' = \sum_i \alpha_i D_S^i$ in Proposition 2 and then using Proposition 1 yields

$$d_H(D_S, D_T) \leq \sum_i \alpha_i \left(d_H(D_S, D_S^i) + d_H(D_S^i, D_T)\right) \ .$$

Substituting the above in the second step of the proof of Theorem 3 in [Ben-David *et al.*, 2010], along with the fact that $\Delta \leq \Delta^c$ and the convexity of Δ^c yields the result. □

Theorem 1 gives a bound on the expected error ($\Delta_{D_T}(h)$) on target in terms of empirical source errors (ϵ_i^S) of individual feature group classifiers, empirical error ($\Delta_{L_T}(h)$) of the ensemble on target, and the d_H-distance of each feature group distribution with the source $\left(d_H(D_S, D_S^i)\right)$ and the target $\left(d_H(D_S^i, D_T)\right)$. In addition, it also provides several interesting theoretical insights, which we discuss below.

By using the triangle inequality for d_H-distance (Proposition 2), we have not loosened the bound in Theorem 1 in the sense that by setting $\alpha_0 = 1$, we can recover the bound for a single classifier based on X and trained on L_S. Furthermore, the above bound provides free parameter $\overline{\alpha} = (\alpha_0, \ldots, \alpha_r)$, in addition to β, which can be tuned to obtain a lower target error. While most domain adaptation algorithms (e.g., [Ben-David *et al.*, 2010]) operate by tuning β, which controls the weight of the error on L_T vs. the error on L_S through instance weighting, FEAD focuses on minimizing the target error by tuning $\overline{\alpha}$ instead. Simultaneously, terms in Eq. (1) can be kept bounded by considering feature groups that allow a low error on source domain data *and* have a low d_H-distance from D_S and D_T (i.e. they generalize well across domains).

This analysis naturally motivates an approach that is even simpler than FEAD — picking a single "best" feature group classifier (BFG) for the target domain, where the notion of "best" can be decided by the user. Although the ensemble approach is clearly more expressive than the BFG approach, as the former's output space is a superset of the latter, it is valid to ask whether the additional expressiveness indeed earns FEAD a lower error than BFG. In general, there is no reason to expect that the expected target error or its upper bound given by Theorem 1 is minimized when only one of the classifiers is picked (in this case the result will be the same as the BFG approach.)

We give a simple artificial example which elucidates this point. Consider the following setting: We have two features $(x_1, x_2) \in \Re^2$ and a label $y \in \{-1, 1\}$, which we wish to predict. The features are conditionally independent given the label i.e. $P(x_1, x_2|y) = P(x_1|y)P(x_2|y)$. Moreover, we are given the following description of the distribution: (1) $P(y = 1) = P(y = 0) = \frac{1}{2}$; (2) $D_S(x_1|y) = D_S(x_2|y) = \mathcal{N}(y, 1)$; (3) $D_T(x_1|y) = \mathcal{N}(y, 1)$; 4) $D_T(x_2|y) = \mathcal{N}(y, 2)$, where D_S and D_T describe distributions on source and target, respectively, and $\mathcal{N}(\mu, \sigma^2)$ represents a Gaussian distribution with mean μ and variance σ^2.

It is easy to see that the asymptotically optimal classifiers for source domain are given by x_1 and $x_1 + x_2$ when considering feature groups x_1 and (x_1, x_2), respectively; similarly, the bayes optimal classifier for the target domain and the feature group (x_1, x_2) is given by $x_1 + \frac{1}{2}x_2$.

If we consider two feature groups: (x_1, x_2) and x_1, the BFG approach would, asymptotically, dictate picking the best of $x_1 + x_2$ and x_1 for the target domain; clearly, both of these classifiers give higher error on the target domain than the optimal classifier: $x_1 + \frac{1}{2}x_2$. However, the WPoE approach would be able to produce the optimal classifier $x_1 + \frac{1}{2}x_2$ as $x_1 + \frac{1}{2}x_2 = \frac{1}{2}(x_1 + x_2) + \frac{1}{2}(x_1)$. This example illustrates how additional expressiveness of feature group ensembles can help outperform the BFG approach.

Overall, our theoretical analysis shows that FEAD is a flexible algorithm that directly tackles domain change — it allows for minimizing error on L_T while keeping the error on L_S bounded. Although the focus of this paper is not on techniques for deciding the "best" feature groups, we demonstrate in Sec. 4 that by adapting simple heuristics to determine the feature grouping, based on statistical measures or domain knowledge, FEAD can yield strong empirical results in a number of real-world domain adaption tasks.

3.3 FEAD as Product of Experts

When considering a specific setting where H is the space of all linear classifiers and Δ^c is the log-loss given by $\Delta^c(h(x), y) = \log(1 + \exp(-yh(x)))$, our analysis in this section provides an interesting probabilistic perspective of FEAD. Let $\mathbf{w_S^i}$ be a weight vector learned for feature group X^i via logistic regression. Given a feature vector $\mathbf{x}$, the probability of output y, as per $\mathbf{w_S^i}$, is given by

$$Pr(y|\mathbf{x}, \mathbf{w_S^i}) = \frac{\exp(y\mathbf{w_S^i} \cdot \mathbf{x})}{1 + \exp(y\mathbf{w_S^i} \cdot \mathbf{x})} ,$$

which implies that the odds ratio of $y = 1$ and $y = -1$ for this instance is given by

$$Odds(\mathbf{x}|\mathbf{w_S^i}) = \frac{Pr(y = 1|\mathbf{x}, \mathbf{w_S^i})}{Pr(y = -1|\mathbf{x}, \mathbf{w_S^i})} = \exp(2\mathbf{w_S^i} \cdot \mathbf{x}) .$$

FEAD expresses the final linear classifier for the target domain, $\mathbf{w_t}$, as $\sum_i \alpha_i \mathbf{w_s^i}$, which is equivalent to expressing the odds for any instance as a weighted product of odds given by individual classifiers. In this sense, our approach is similar to the Product-of-Experts framework proposed by Hinton [1999], wherein classifiers based on feature groups can be thought of as *experts*.

4 Experiments

We evaluate our approach empirically on two domain adaption tasks — *sentiment analysis* and *email spam detection* — and compare with other state-of-the-art approaches.

4.1 Sentiment Classification

Sentiment analysis is the task of classifying textual reviews into positive or negative based on the expressed "sentiment" [Pang *et al.*, 2002; Blitzer *et al.*, 2007]. Reviews for different categories of products tend to be different in terms of vocabulary and writing style. Consequently, a sentiment classifier trained on one domain (e.g., books) suffers from considerable performance degradation when applied to another domain (e.g., electronics), thus necessitating a need for domain adaption methods.

In this set of experiments, we use the benchmark dataset released by Blitzer *et al.* [2007], which consists of reviews of four different product categories: *books*, *DVDs*, *electronics*, and *kitchen* appliances, collected from Amazon.com. Along with the IMDB movie reviews released by Pang *et al.* [2002], we generate all 20 possible source–target pairs for the empirical evaluation. For all the methods we tested, we follow the same data split and experimental procedure. Each domain contains 2000 instances and is randomly split into a training/testing set of 1600/400 examples. Furthermore, we randomly select 160 instances from the training set of each domain to serve as a validation set. For each source–target adaptation setting, we use the training set of source as L_S (labeled source training data), the validation set of target as L_T (labeled target training data), and the remaining training examples of target as U_T (unlabeled target data.) Features extracted from each review are the unigrams and bigrams occurring at least twice in $L_S \cup L_T \cup U_T$, encoded as binary features. We perform 30 such random splits for each setting and report the averaged accuracy when evaluating the model on the target testing data.

We compare four approaches – logistic regression (LR), multiview transfer learning (Multi-T), easy adapt (EA) and our feature ensemble approach (FEAD). As the baseline approach, LR is a model trained over all the features. We first find the best Gaussian smoothing prior by training the model on the source training set (L_S) and testing it on the validation set (L_T.) The final model is learned on the entire labeled data ($L_S \cup L_T$) using the selected prior. Proposed recently by Blitzer *et al.* [2009], Mutlti-T is an enhanced version of structural correspondence learning (SCL) [Blitzer *et al.*, 2007], which has been tested previously on the sentiment dataset[2]. We conduct the experiments using the exact setting as reported in their paper. Another state-of-the-art method, EA [Daumé III, 2007] projects the source and target examples to a new expanded feature space first, and then trains the classifier on the new labeled examples in a way similar to the baseline LR approach. Finally, FEAD uses two groups of features. The first one is the trivial group that contains all the features — call this feature group *total*. For the second feature group, we aim to have features that lead to low d_H-distance and empirical error. We adapt a heuristic similar to the one used in [Blitzer *et al.*, 2007] based on mutual information. For each feature appearing in L_S, we first compute its mutual information compared with the label. For a feature to be included in this group, its mutual information needs to be higher than at least half of the features in L_S and it needs to occur at least once in the target data ($L_T \cup U_T$). We call this feature group *common*. Notice that this by no means provides the *best* feature grouping for our approach. Instead, we demonstrate that even with simple groups of features selected following the principle suggested by our theoretical analysis, a significant improvement can be observed empirically.

Table 1 shows the averaged accuracy of different methods for each source–target adaptation setting. As can be seen

[2] We use the software package provided by the authors. Personal communication confirms that the results of Multi-T and SCL-MI are comparable.

Setting	Algorithms			
Src-Tgt	LR	Multi-T	EA	FEAD
B-D	81.10	$80.01^{\dagger}$	$78.63^{\dagger}$	**81.80**
B-E	78.88	78.82	**79.38**	79.34
B-K	**82.29**	$79.60^{\dagger}$	81.66	82.26
B-M	80.23	$77.91^{\dagger}$	$79.25^{\dagger}$	**80.70**
D-B	$81.60^{\dagger}$	$79.91^{\dagger}$	$80.14^{\dagger}$	**82.46**
D-E	81.27	81.09	80.34	**81.54**
D-K	**82.95**	$81.83^{\dagger}$	$82.08^{\dagger}$	82.81
D-M	82.53	$79.50^{\dagger}$	$81.53^{\dagger}$	**82.53**
E-B	75.34	$72.74^{\dagger}$	**75.85**	*75.60*
E-D	$75.85^{\dagger}$	75.86	$74.78^{\dagger}$	**76.76**
E-K	$86.50^{\dagger}$	$83.77^{\dagger}$	$85.33^{\dagger}$	**87.59**
E-M	$72.60^{\dagger}$	$70.86^{\dagger}$	72.63	**73.54**
K-B	$74.74^{\dagger}$	75.47	$74.78^{\dagger}$	**75.75**
K-D	75.93	**76.97**	$75.21^{\dagger}$	76.88
K-E	84.90	84.57	$83.81^{\dagger}$	**85.24**
K-M	72.38	$71.02^{\dagger}$	$70.45^{\dagger}$	**72.62**
M-B	$77.11^{\dagger}$	$77.06^{\dagger}$	$76.07^{\dagger}$	**78.88**
M-D	$77.76^{\dagger}$	$77.94^{\dagger}$	$76.20^{\dagger}$	**79.52**
M-E	$76.45^{\dagger}$	**80.18**	$76.50^{\dagger}$	$77.62^{\dagger}$
M-K	$76.72^{\dagger}$	**79.41**	$76.48^{\dagger}$	$77.59^{\dagger}$
Avg.	78.86	78.23	78.06	**79.55**

Table 1: Results on sentiment classification of the baseline (LR), multiview transfer (Multi-T), easy adapt (EA), and our ensemble approach (FEAD). Numbers in bold-face font are the best performing method in their source–target setting. Numbers with † are statistically significantly worse than the best performing method. Statistical significance is based on paired-t test with the p-value less than 0.05.

clearly, FEAD yields the best results for most of the settings. Among them, FEAD performs statistically significantly better than both EA and Multi-T in 8 cases, and better than all competing approaches in 4 cases. Even when FEAD does not achieve the highest accuracy, the difference compared with the best performing method is not statistically significant, except for pairs M-E and M-K. Table 2 shows the normalized (i.e. weights for total and common features groups sum to 1) values of the weight assigned to the *common* feature group. From the table, it can be infered that the *common* feature group plays a significant role in the final classifier. Notice that the baseline (LR), when provided with some amount of labeled target data, is notoriously hard to defeat [Daumé III, 2007; Chang *et al.*, 2010]; our experiments corroborate this. Overall, the experiments on the sentiment analysis empirically show that FEAD is a simple yet powerful algorithm for domain adaptation.

4.2 Email Spam Detection

Spam detection is a problem of classifying email messages to either spam or good. Compared with the typical binary classification setting, the main challenge of spam detection comes from its adversarial nature. As spammers often quickly react to a newly deployed filter by modifying the spam messages [Lowd and Meek, 2005b; 2005a], the sample distribution from the testing period often dramatically changes. As argued previously by Jiang and Zhai [2007], such phenomenon can be treated as a domain adaptation scenario due to the temporal difference between the source and target domains, and thus is a good application for our approach.

Src \ Tgt	Book	DVD	Elec.	Kitchen	Movie
Book	-	0.45	0.18	0.20	0.41
DVD	0.35	-	0.29	0.27	0.22
Elec.	0.26	0.24	-	0.47	0.20
Kitchen	0.33	0.34	0.54	-	0.22
Movie	0.55	0.52	0.37	0.42	-

Table 2: The weights assigned to frequent feature groups by FEAD in different sentiment classification settings.

We use real-world email spam data in the experiment. The dataset contains real email messages received by users of Hotmail. The sampling process randomly picks emails sent to the volunteer users and asks them to label the messages as either *spam* or *good*. Among the total 915,000 labeled messages, we treat the 765,000 messages received from July-01-2005 to Nov-30-2005 as the source training set (L_S). The target set consists of 150,000 messages received from Dec-01-2005 and Dec-15-2005, where we keep the first 30,000 examples for training and tuning (L_T) and the rest for testing. Note that we are making a distinction between source and target data by cutting off a continuous timeline at an arbitrarily picked point. We assume that this distinction is a good enough approximation to the 'source-target' setting of domain adaptation.

Although spam detection is primarily treated as a text-classification problem, several non-textual types of features have been proven useful in previous work, such as the sender information [Leiba *et al.*, 2005] or the distribution of email received by specific users [Chang *et al.*, 2008]. These features not only capture very distinctive information, but also behave differently in the domain adaptation setting. For example, while it is easy for spammers to change email content, switching to a new server farm to send spam is relatively difficult. As a result, each specific type of features has nonidentical distribution change and leads to classifiers with different changes in expected errors.

In order to show how different feature groups undergo different changes in their discriminative power, we compare the performance of two classifiers: one trained only on features from email-content (*content features*) and the other trained only on features based on sender information (*sender features*.) For each set of features, we used two different settings: *random* and *time-shifted*. The random setting is the control group that tests the scenario when the training and testing data come from the same distribution. In this setting, we train the classifiers on randomly chosen 765,000 instances, tune on randomly chosen 50,000, and test on the remaining 100,000. In contrast, the time-shifted setting simulates the domain adaptation scenario, where we train the classifiers on first (chronologically) 765,000 instances, tune on the next 50,000, and test on the final 100,000. Fig. 1 shows the comparisons of the content and sender based classifiers

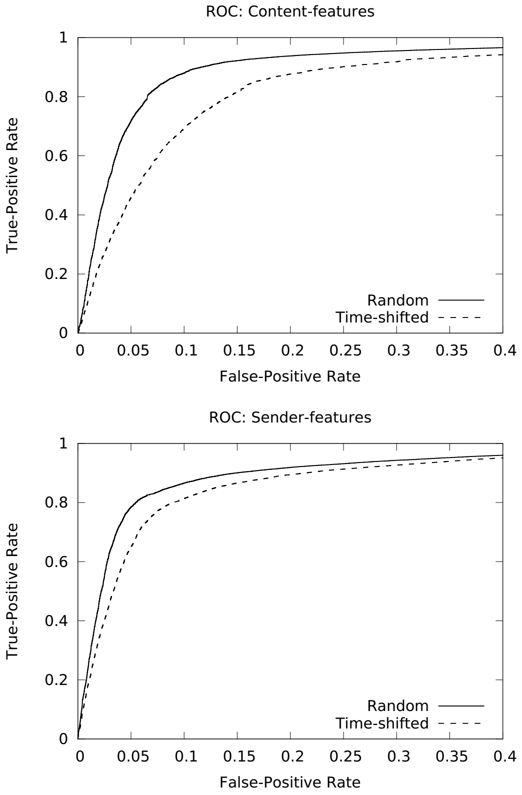

Figure 1: ROC curves of classifiers based on content-features and sender-features in the random and time-shifted settings. It can be observed that the performance degradation of the content-based filter is greater than that of the sender-based filter, which could imply that the distribution of content-features changes more drastically over time.

in these two settings, using the ROC analysis. A large performance drop in the time-shifted setting compared to the random setting can be explained by a drastic distribution or domain change. The performance degradation is more significant for content-features compared to sender-features, which empirically shows they undergo different distribution change.

In the experiments, we leverage this background knowledge to form the feature grouping used by FEAD. In addition to content and sender features, we have two other feature groups: *user* features and a trivial feature group that contains all the features. The content features are words in the subject and body of the email that occur at least three time in the source set. The sender features include the first 16 bits, the first 24 bits, and the entire IP address of the sender. Finally, the user features are simply the recipient ids.

We compare the performance of FEAD with two other approaches, logistic regression (LR) and easy adapt (EA)[3]. LR and EA do not take advantage of the background knowledge of feature grouping and are trained on all (i.e. content, sender, and user) features put together. The experimental settings are the same as described in the experiments on sentiment analysis (Sec. 4.1). To compare performance, we report the ROC curves and the normalized AUC values in a low false-positive rate (FPR) region, as the cost of losing good email is much higher than receiving spam. Normalized AUC for FPR $\leq \delta$ is defined as $\frac{1}{\delta}\text{AUC}(\delta)$ where $\text{AUC}(\delta)$ is the area under the curve for FPR between 0 and δ. As can be observed from Fig. 2, FEAD performs better than LR and EA consistently in this region, and has a much higher normalized AUC score. In comparison, although EA indeed performs better than the baseline LR, the improvement gain is relatively small and its ROC curve in fact crosses that of LR. This empirical result suggests that FEAD can effectively leverage the domain knowledge of the feature grouping, which is valuable in improving the model performance for domain adaptation.

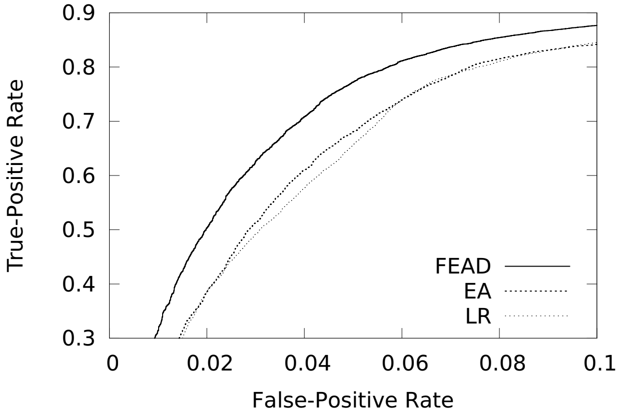

Figure 2: The ROC curves of the baseline (LR), easy adapt (EA) and our ensemble approach (FEAD) when the false-positive rate (FPR) is less than 0.1. The normalized AUC scores of LR, EA and FEAD in this region are 0.592, 0.602 and 0.677, respectively.

5 Related Work

The problem of domain adaptation has been studied extensively from both the theoretical perspective and algorithmic sides. To understand formally the characteristics of the problem, Ben-David *et al.* [2007; 2010] proposed the d_H-distance framework, which was later extended to a wider variety of loss functions by Mansour *et al.* [2009]. Although the idea of feature grouping was never discussed previously, their work provides a solid foundation for the theoretical analysis of our ensemble approach.

A variety of algorithm-centric domain adaption methods have also been proposed. For example, EasyAdapt [Daumé III, 2007] uses feature-replication to implicitly capture the domain-invariant and domain-specific aspects of features. Blitzer *et al.* [2006] proposed Structural Correspondence Learning (SCL), which projects features to a latent space such that these latent features exhibit roughly the same distribution across source and target domains. Alternatively, Finkel

[3] We attempted to include multiview transfer learning (Multi-T). Unfortunately, it does not seem to scale well to a large amount of training data, and failed to produce the final model due to memory issues, even after increasing the heap size to 20GB.

and Manning [2009] and Jiang and Zhai [2006] propose approaches based on imposing priors over individual features, where the priors can be decided directly based on the *generalizability* of the features. FEAD is related to these works as it massages the weights of different features in terms of their contribution to the final prediction function. However, unlike these approaches, the feature weights in FEAD are tweaked after creating the classifiers.

The use of feature grouping in FEAD is analogous to adaptation from multiple sources [Mansour *et al.*, 2008; Ben-David *et al.*, 2010], where feature groups can be thought of as different sources created artificially. However, adaptation from multiple sources typically assumes that the instances from different sources are independent, which is clearly not the case in our setting. Finally, we share the idea of using ensemble-like learning for domain adaptation with Dai *et al.* [2007], who use boosting for the same purpose. However, our approach considers ensembles over different feature groups, whereas they create ensembles by re-weighting instances.

6 Conclusion

In this paper, we introduced a technique that leverages the expressiveness of ensembles to adapt to distribution change between domains. Building on this notion, we presented a simple and easy-to-implement method based on re-weighting classifiers learned on different feature groups. Our approach provides the flexibility to re-adjust the influence of different feature groups on the target domain classifier based on their cross-domain distributional stability. This design is motivated by our theoretical analysis, and its empirical effectiveness is demonstrated through experiments on the benchmark sentiment analysis task and spam detection, where our approach outperforms several state-of-the-art methods.

Although in this work we showed that by using simple statistical measures or domain knowledge to generate the feature grouping, our approach can lead to significant improvement, it is nevertheless interesting to find automatically the best feature grouping for the task. We leave that as future work.

Acknowledgment We would like to thank John Blitzer for providing the Multiview-T software, Michael Gamon for preparing sentiment analysis data, Sumit Basu and Chris Meek for their comments and the anonymous reviewers for their useful suggestions. The work was partly supported by Army Research Laboratory (ARL) under agreement W911NF-09-2-0053 and by an ONR award on "Guiding Learning and Decision Making in the Presence of Multiple Forms of Information."

References

[Ben-David *et al.*, 2007] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *NIPS*. MIT Press, 2007.

[Ben-David *et al.*, 2010] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Vaughan. A theory of learning from different domains. *Machine Learning Journal*, 2010.

[Bickel *et al.*, 2009] Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning under covariate shift. *Journal of Machine Learning Research*, 10, 2009.

[Blitzer *et al.*, 2006] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proc. of EMNLP*, 2006.

[Blitzer *et al.*, 2007] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *ACL*, 2007.

[Blitzer *et al.*, 2009] J. Blitzer, D. Foster, and S. Kakade. Zero-shot domain adaptation: A multi-view approach. Technical Report TTI-TR-2009-1, Toyota Technological Institute Chicago, 2009.

[Chang *et al.*, 2008] M. Chang, W. Yih, and C. Meek. Partitioned logistic regression for spam filtering. In *Proc. of ACM SIGKDD*, 2008.

[Chang *et al.*, 2010] M. Chang, M. Connor, and D. Roth. The necessity of combining adaptation methods. In *Proc. of EMNLP*, 2010.

[Dai *et al.*, 2007] Wenyuan Dai, Qiang Yang, Gui rong Xue, and Yong Yu. Boosting for transfer learning. In *ICML*, 2007.

[Daumé III, 2007] Hal Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2007.

[Devroye *et al.*, 1996] L. Devroye, L. Gyorfi, and G. Lugosi. *A probabilistic theory of pattern recognition*. Springer, 1996.

[Finkel and Manning, 2009] J. Finkel and C. Manning. Hierarchical bayesian domain adaptation. In *Proc. of NAACL*, 2009.

[Hinton, 1999] G.E. Hinton. Products of experts. In *Proc. of ICANN*, pages 1–6, 1999.

[Jiang and Zhai, 2006] J. Jiang and C. Zhai. Exploiting domain structure for named entity recognition. In *Proc. of NAACL*, 2006.

[Jiang and Zhai, 2007] J. Jiang and C. Zhai. Instance weighting for domain adaptation in NLP. In *Proc. of ACL*, 2007.

[Leiba *et al.*, 2005] B. Leiba, J. Ossher, V. Rajan, R. Segal, and M. Wegman. SMTP path analysis. In *CEAS-2005*, 2005.

[Lowd and Meek, 2005a] D. Lowd and C. Meek. Adversarial learning. In *Proc. of KDD*, 2005.

[Lowd and Meek, 2005b] D. Lowd and C. Meek. Good word attacks on statistical spam filters. In *CEAS-2005*, 2005.

[Mansour *et al.*, 2008] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *NIPS*, 2008.

[Mansour *et al.*, 2009] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *COLT*, 2009.

[Pang *et al.*, 2002] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proc. of EMNLP*, 2002.