Cross-Domain Text Understanding in Online Social Data

Qian Lin, Shenxiu Liu, Zhao Yang, Aditya Jami, Ashutosh Saxena

December 11, 2014

1 Introduction

The text classification is a long standing problem, and fruitful results have been achieved in various situations. However it is still a tough problem to train a text classifier with training data and test data from different source. The rationale behind such classification task is that, on the one hand the data in the target domain (data source we care about) have only few data labelled, meaning that the training set is too small to train a reasonable text classifier; while on the other hand, there is another data source (source domain) which shares the similar feature space with our test data, but have massive labelled data. Thus it is fair to ask whether it is possible to train a classifier using source domain data while apply such classifier on the target domain. So is our goal for the cross-domain text understanding.

To be more specific, the data we deal with are from Amazon, eBay and Twitter. Our goal is to train the text classifier on Amazon (source domain), whose review has inherent label indicating the type of goods it is describing, and adopt the model to classify reviews from eBay and twits (target domain). We present the baseline models in section 2, which simply trains classifiers on one data source and test it on another. In section 3, we present our first cross domain model, instance adaptation[2], which tries to understand the similarity between different data source and assign larger weight on the data from source domain which resemble the target domain data. However, such approach does not work well in our situation. We provide an explanation for the poor performance. In section 4, we present another strategy, feature replication[1]. Intuitively, this approach expand the feature space and arrange target domain data and source domain data into different hypersurface while the intersection of the two hypersurfaces is the original feature space. Such approach provides a good performance, especially in the small training data set limit. We talk about the future research in section 5.

2 Baseline Model

We have 2M reviews from Amazon, 34k reviews from eBay and 63k twits from Twitter all with manually labeled Amazon categories. We construct the vocabulary

for Amazon, eBay and Twitter of size 45k, 9k, 11k respectively after stemming. We use tf-idf to obtain the feature vector for each review. We use SVM with linear kernel to construct the baseline model. The classifier aims to decides whether the given review is describing a book or not. The baseline model is simply to train the model on one data source and to test it on the other.

Tab.2 shows the error rate of the baseline model. In our report, all error rates are test error and is defined as the average of false positive rate and false negative rate. In Tab.1, each row represents the one training data source and each column stands for one testing data source. It is obvious that if the training data and testing data are from the same source, the binary classification performs reasonably, while the naive cross domain yields a much worse result.

error (%)	A	T	E
A	2.87	28.27	13.49
T	20.73	3.45	N/A
E	18.12	N/A	7.67

Table 1: Test error rate of baseline model

3 Instance Adaptation

Our first cross domain algorithm [2] is the instance adaptation algorithm. Intuitively, such algorithm aims to first decide the similarity between the source domain and the target domain, and to weight samples in the source domain based such similarity measure when training.

Mathematically, we assume $p_S(y|\vec{x}) \approx p_t(y|\vec{x})$, where s means source domain and t means target domain. Thus in the generative model, in order to maximize the joint probability in the target domain $p_t(\vec{x}, y)$, we perform following transformation

$$p_t(\vec{x}, y) = p_t(y|\vec{x})p_t(\vec{x}) \approx p_s(y|\vec{x})p_s(\vec{x})\frac{p_t(\vec{x})}{p_s(\vec{x})} = p_s(\vec{x}, y)\frac{p_t(\vec{x})}{p_s(\vec{x})} \tag{1}$$

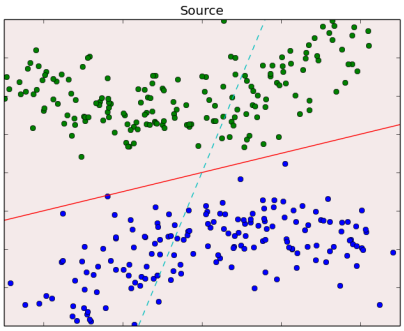
Thus we assign samples in the source domain with proper weight

$$\omega(x) = \frac{p_t(\vec{x})}{p_s(\vec{x})} = \frac{p(label = target|\vec{x})}{p(label = source|\vec{x})} \tag{2}$$

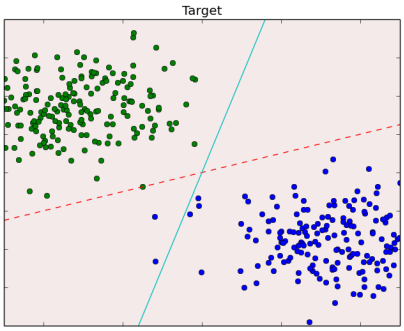
thus $\omega(x)$ can be obtained by a logistic regression (SVM) in the feature space that decides whether a given sample belongs to the target domain or source domain. An illustration of such algorithm is given in Fig.1.

Tab.2 shows our results using instance adaptation. We have following observations.

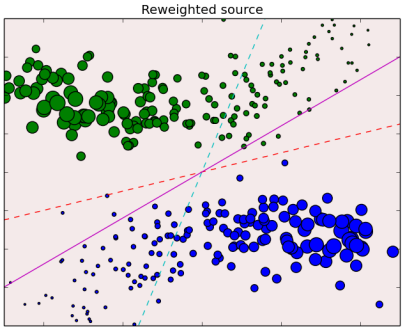
- We find in general the instance adaption do not help much.
- We also notice that when we use the source domain idf for the feature extraction in the target domain, the error rates reduces.



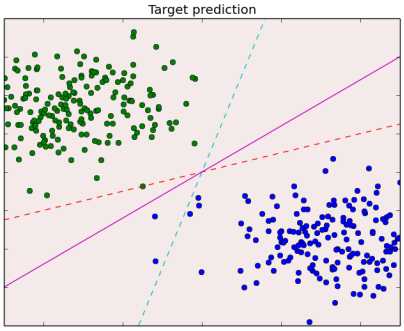
(a) Original source data



(b) Original target data



(c) Weighted source data



(d) Target prediction with weighted source

Figure 1: Illustration of weighting of source instances to match target density [3]

The reason for the limited improvement of performance is that, ideally if the samples of the target domain is a subset of the source domain samples in the feature space, then by properly weighting the source domain sample, we should see a big improvement of the classification performance. However in our case, the features from source domain and target domain do not have much overlap. This conclusion is drawn from the fact that when we use SVM(logistic regression) to calculate $\omega(x)$, the classifier is able to make a clear decision (within 2% error rate) whether a sample belongs to the target or the source domain. Thus in the feature space, the reviews from the target and the source domain form clusters with themselves.

For the second observation, it can be understood that if we use the source domain idf for the feature extraction in the target domain, we effectively improve the accuracy of the assumption $p_S(y|\vec{x}) \approx p_t(y|\vec{x})$, thus instance adaption provides a better performance.

error(%)	BL	BL(IDF-S)	SWSVM	SWSVM(IDF-S)	SWLR	SWLR(IDF-S)
A to T	28.27	26.68	29.14	24.04	30.05	24.28
A to E	13.49	13.33	12.90	12.70	14.97	14.99

Table 2: Test error rate of instance adaption algorithm. IDF-S means we use the source idf for the feature extraction of the target domain. SWSVM means we calculate $\omega(x)$ using SVM. SWLR means we calculate $\omega(x)$ using logistic regression.

4 Feature Replication

In this section, we introduce another cross domain algorithm, feature replication [1]. We first talk about the intuition of this algorithm. Geometrically, this approach expand the feature space and arrange target domain data and source domain data into different hypersurface while the intersection of the two hypersurfaces is the original feature space. To be more specific, Let $\mathcal{X} = \mathcal{R}^F$ be the original input feature space. Define an augmented input space $\tilde{\mathcal{X}} = \mathcal{R}^{3F}$. Define mapping $\Phi^s, \Phi^t : \mathcal{X} \to \tilde{\mathcal{X}}$, which are the mappings from the original feature space to augmented feature space of source domain sample and target domain sample respectively.

$$\Phi^s(\mathbf{x}) \quad = \quad \langle \mathbf{x}, \mathbf{x}, \mathbf{0} \rangle, \quad \Phi^t(\mathbf{x}) = \langle \mathbf{x}, \mathbf{0}, \mathbf{x} \rangle \tag{3}$$

This approach on the one hand admits the resemblance between the target and source domain as they all share the first $\mathcal{R}^F$ dimensions of features. While the target data and source data manifest their difference in the rest of the dimensions.

Then the immediate question is what is the size of source data we should use. Because if the size of the source data is too small, it can not resolve the lack of training samples problem, while if the size of the source data is too big, they will flood the target training data and introduce too much noise. To answer such question, we show the learning curves (test error) of Amazon to Twitter and Amazon to eBay cross domain classifier with different size of the Amazon data.

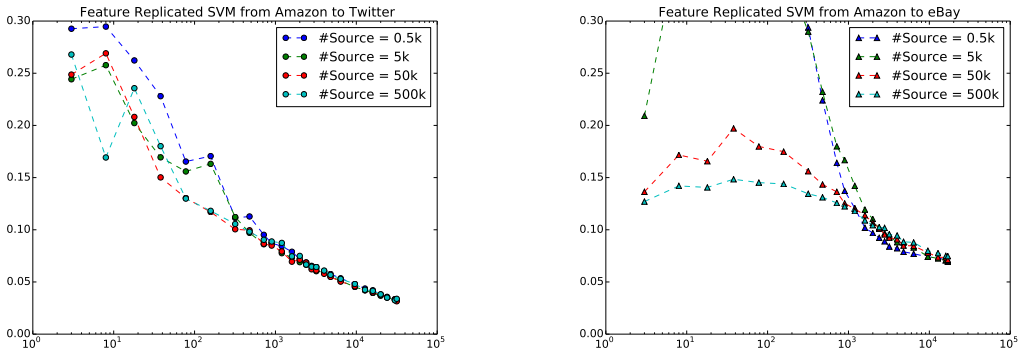


Figure 2: Learning curves of feature replicated SVM from Amazon to Twitter and eBay. The x axis is the size of the target domain data (Twitter or eBay) involved in training. The y axis is the test error. Different curves represent different size of source domain data (Amazon) included in training. The test error rate beyond 0.3 is not shown in the figure.

In our data, Fig.2, we do not see much difference for the Twitter case, but for the eBay case, the competition between insufficient training data and the large

source data corpus noise manifests. When the source data size is too small, the test error goes beyond 0.3, while if the source data size is too big, it injects too much noise when the target data size is big enough, thus affects the performance. To compromise the competition, we choose the source data size to be 50k for further study.

Finally, we want to test whether feature replicate approach performs better than the others. Fig.3 shows the test error among no cross domain training, standard cross domain SVM and feature replicate SVM. No cross domain training means we only use target domain data to train the classifier. The standard cross domain SVM means we train our classifier using 50k Amazon samples along with the target domain data. The feature replicate SVM also uses 50k Amazon data along with target domain data, but the features are manufactured using above method.

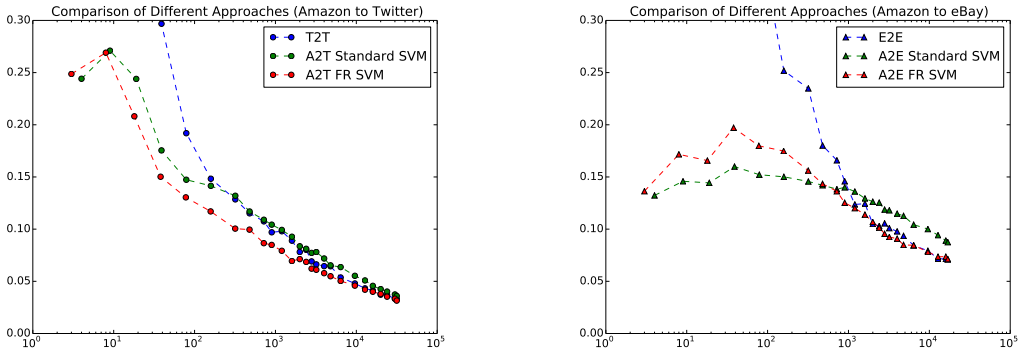


Figure 3: Learning curves of no cross domain training, standard cross domain SVM and feature replicate SVM. No cross domain training means we only use target domain data to train the classifier. The standard cross domain SVM means we train our classifier using 50k Amazon samples along with the target domain data. The feature replicate SVM also uses 50k Amazon data along with target domain data, but the features are manufactured using FR method. The x axis is the size of the target domain data (Twitter or eBay) involved in training. The y axis is the test error.

We draw following conclusions from Fig.3.

- From Amazon to Twitter, the FR SVM always performs better than other methods.
- From Amazon to eBay, the FR SVM performs always better than the no cross domain approach, but only outperforms the Standard SVM when the target domain data size is large.

In total, FR SVM introduces less noise than the standard SVM and take advantages of the similarity between the two domains to reduce the test error rate when the target domain training samples are insufficient.

5 Conclusion and Future Research

Our cross domain text classification work aims to take the advantage of the similarity between the source domain and the target domain, so that given insufficient target domain training samples, we can also achieves a good classification job. We have mainly tried two cross domain algorithms, the Instance Adaptation and the Feature Replicate. The improvement of IA approach is tiny, for which we present the possible reasons, while the FR approach provides a reasonably good performance.

In the feature, we will try the following thing. Since Twitter texts are usually quite short, its distribution space is well-distinguished from the Amazon data (linear separator error is only $< 1\%$). Extending the Twitter vocabulary/text by additional information from twits sharing the same tag, or simply using word-net may improve performance.

References

[1] Hal Daumé III, Abhishek Kumar, and Avishek Saha. Frustratingly easy semi-supervised domain adaptation. pages 53–59, 2010.

[2] Jing Jiang and ChengXiang Zhai. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271, Prague, Czech Republic, June 2007. ACL.

[3] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer joint matching for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1410–1417. IEEE, 2014.