Opinion Mining by Transformation-Based Domain Adaptation

Róbert Ormándi, István Hegedűs, and Richárd Farkas

University of Szeged, Hungary
{ormandi,ihegedus,rfarkas}@inf.u-szeged.hu

Abstract. Here we propose a novel approach for the task of domain adaptation for Natural Language Processing. Our approach captures relations between the source and target domains by applying a model transformation mechanism which can be learnt by using labeled data of limited size taken from the target domain. Experimental results on several Opinion Mining datasets show that our approach significantly outperforms baselines and published systems when the amount of labeled data is extremely small.

1 Introduction

The generalization properties of most statistical machine learning approaches are based on the assumption that the samples of the training dataset come from the same underlying probability distribution than those that are used in the prediction phase of the model. Unfortunately – mainly in real-world applications – this assumption often fails. There are numerous Natural Language Processing tasks where plentiful labeled training databases are available from a certain domain, but we have to solve the same task using data taken from a different domain where we have only a small dataset. Manually labeling the data in the new domain is costly and inefficient. However, if an accurate statistical model from the source domain is present we can adapt it to the target domain [1].

Opinion Mining aims to automatically extract emotional cues from texts [2]. For instance it can classify product reviews according to the customers positive or negative polarity. Opinion Mining is a typical problem where the requirement for domain adaptation is straightforward as there exits numerous slightly different domains (e.g. different products are different domains) and the construction of manually labeled training data for each of them would be costly.

Here, we will define a general framework to directly capture the relations between domains. In order to experimentally evaluate our approach, Support Vector Machine (SVM) [3] was plugged into the framework and the approach was compared to a number of baseline algorithms and published results on Opinion Mining datasets.

2 Related Work

Numerous preliminary algorithms have been developed in the field of domain adaptation which roughly can be categorised into two mainstreams.

One of these types of methods tries to *model the differences between the distributions* of the source and target domains empirically. In [4] the parameters of the maximum entropy model learned from the source domain as the means of a Gaussian prior was used during training a new model on target data. A different technique proposed in [1] defines a general domain distribution that is shared between source and target domains. In this way, each source (target) example can be considered a mixture of source (target) and general distributions. Using these assumptions, their method was based on maximum entropy model and used the EM algorithm for training. Another approach was proposed in [5] where a heuristic nonlinear mapping function is used to map the data into a high dimensional feature space where a standard supervised learner can be employed in the area of domain adaptation.

The newer generation of domain adaptation algorithms are based on *defining new features for capturing the correspondence* between source and target domains [6, 7]. In this way, the two domains appear to have very similar distributions, which enable effective domain adaptation. A more specific subtype of the above described algorithm family learns a *joint feature representation* for the source and the target domain where the corresponding marginal distributions are close to each other [8].

Theoretical results on domain adaptation have been also proposed [9, 10]. For instance [10] considered the problem of multiple source domain adaptation and gave theoretical results of the expected loss of combined hypotheses on the target domain.

3 Transformation-Based Domain Adaptation Approach

In this section we shall give a more precise formalism of the domain adaptation task and we will describe our approach in detail.

3.1 Domain Adaptation Task

In the current context of domain adaptation, we will assume that there are two feature spaces given – $\mathcal{D}_S$ and $\mathcal{D}_T$ – the "source domain" and the "target domain" feature spaces, respectively. We have two sets of labeled training samples, $S \subseteq \mathcal{D}_S$ and $T \subseteq \mathcal{D}_T$ as well ($|T| \ll |S|$). In addition we will assume that both the source domain and the target domain use the same label set. The labels[1] in both domains come from the $C = \{C_1, \ldots, C_l\}$ set and the $t : \mathcal{D}_S \cup \mathcal{D}_T \to C$ function assigns the *correct* class label to each sample from $\mathcal{D}_S$ and $\mathcal{D}_T$. The learning problem of the domain adaptation task is to find a $p_{\mathcal{D}_T} : \mathcal{D}_T \to C$ prediction function that achieves a high accuracy on the target domain.

[1] Our approach will focus on classification problems, but it can easily be extended to regression problems as well.

3.2 Transformation-based Approach

One of the main assumptions of the domain adaptation task is that there exists some kind of relation between the source domain and the target domain. Our idea is to try to model this relation, i.e. try to find a $\phi : \mathcal{D}_T \to \mathcal{D}_S$ transformation or target-source domain transformation. This transformation maps the samples from $\mathcal{D}_T$ into the feature space of $\mathcal{D}_S$.

More precisely, we look for a $\phi : \mathcal{D}_T \to \mathcal{D}_S$ transformation which minimizes the prediction error of each transformed sample taken from the training database of the target domain. Our idea is to utilize the $p_{\mathcal{D}_S} : \mathcal{D}_S \to C$ model (a prediction function on the source domain with a high prediction accuracy) directly for this task. Hence the following optimization problem was formed: $\min_\phi \quad E_{T,p_{\mathcal{D}_S}}(\phi) + Q\sum_{x\in T}\|\phi(x)\|$. Here $E_{T,p_{\mathcal{D}_S}}(\phi)$ is an error function which just depends on ϕ. If we can solve this optimization problem, we will get the prediction function of the target domain in the form $p_{\mathcal{D}_T}(x_0) = p_{\mathcal{D}_S}(\phi^*(x_0))$. Here the $\phi^* : \mathcal{D}_T \to \mathcal{D}_S$ mapping is the transformation which is the solution for the above-defined minimization task and $x_0 \in \mathcal{D}_T$ is an arbitrary sample from the target domain.

In this paper, we shall apply the following constraints on target-source domain mapping and on the two domains: $\mathcal{D}_S := \mathbb{R}^n$, $\mathcal{D}_T := \mathbb{R}^m$ and $\phi : \mathbb{R}^m \to \mathbb{R}^n$, $\mathbb{R}^m \ni x \mapsto Wx \in \mathbb{R}^n$, where $W \in \mathbb{R}^{n\times m}$. With these constraints, we will get the following specialized optimization task: $\min_{W,\ \|W\|=1} \quad E_{T,p_{\mathcal{D}_S}}(W)$. Here the regularization term is not necessary since it is replaced by the $\|W\| = 1$ constraint on the transformation matrix. This modification can be interpreted as the regularization term in the original form without weighting. To solve this optimization problem we can simply use a gradient descent-based optimization algorithm [11].

3.3 Support Vector Machine as source model

Here, the widely used SVM [12] classification method will be intorduced as the base prediction method [2] and the following error function:

$$E_{T,p_{\mathcal{D}_S}}(W) = \frac{1}{2}\sum_{x\in T}\left(t(x) - p_{\mathcal{D}_S}(Wx)\right)^2. \tag{1}$$

We assume that both the source domain and the target domain are labeled with the following labels: $C = \{-1, +1\}$ (binary classification). In this case the prediction function of the SVM classifier in our formalism is:

$$p_{\mathcal{D}_S}(Wx) = \sum_{s_k\in SV_S} \alpha_k t(s_k) K(s_x, Wx) + b. \tag{2}$$

[2] We derived and implemented Logistic Regression as source model with Cross Entropy error function as well. The description of this learner and the results achieved by it are available at `http://www.inf.u-szeged.hu/rgai/~ormandi/DA2010_TSD_sup.pdf` as supplementary materials.

Here SV_S denotes the set of support vectors that is the subset of the training database of the source domain, i.e. $\|SV_S\| \leq \|S\|$, s_k denotes the kth support vector, α_k is the learnt coefficient corresponding to s_k, the $b \in \mathbb{R}$ value is a learnt parameter of SVM as well and $K : \mathcal{D}_S \times \mathcal{D}_S \to \mathbb{R}$ is the kernel function over the source domain. The argument of the prediction function is Wx, which is the product of the transformation matrix W and an arbitrary sample from the target domain $x \in \mathcal{D}_T$. Here the multiplying with W means the target-source domain mapping.

We decided to apply two commonly used kernel functions to compute the necessary gradient: the Polynomial kernel and the RBF kernel [3, 12]. In Eq. 3 we can see the gradient of the error function applying the polynomial kernel. The form of the kernel is shown in this equation as well. The degree of the polynomial is denoted by d.

$$\begin{aligned} K_d(s_k, Wx) &= (s_k W x)^d, \\ \nabla E_{T,p_{\mathcal{D}_S},K_d}(W) &= -d \sum_{s_k \in SV_S} \alpha_k t(s_k) \cdot \\ \sum_{x \in T} (t(x) - p_{\mathcal{D}_S}(Wx)) &K_{d-1}(s_k, Wx)\, s_k x^T \end{aligned} \tag{3}$$

Similarly, in Eq. 4 we show the RBF kernel and the gradient of the error function using the RBF kernel. Here γ is a parameter of the RBF kernel.

$$\begin{aligned} K_\gamma(s_k, Wx) &= \exp\left(-\gamma \|s_k - Wx\|^2\right), \\ \nabla E_{T,p_{\mathcal{D}_S},K_\gamma}(W) &= -2\gamma \sum_{s_k \in SV_S} \alpha_k t(s_k) \cdot \\ \sum_{x \in T} (t(x) - p_{\mathcal{D}_S}(Wx)) &K_\gamma(s_k, Wx)\,(s_k - Wx)\, x^T \end{aligned} \tag{4}$$

These gradients can be employed in the gradient descent-based algorithm. The whole learning systems will be denoted by PolyDML (using the Polynomial Kernel) and RBFDML (using the RBF Kernel and its gradient).

4 Experimental Results

In this section, the experimental results achieved on a synthetic dataset and real-world Opinion Mining tasks will be presented.

4.1 Evaluation methodology

We hypothetised that domain adaptation is especially required when target training dataset is small, thus experiments using target training data with various sizes were carried out. In the case of extremely small datasets one evaluation

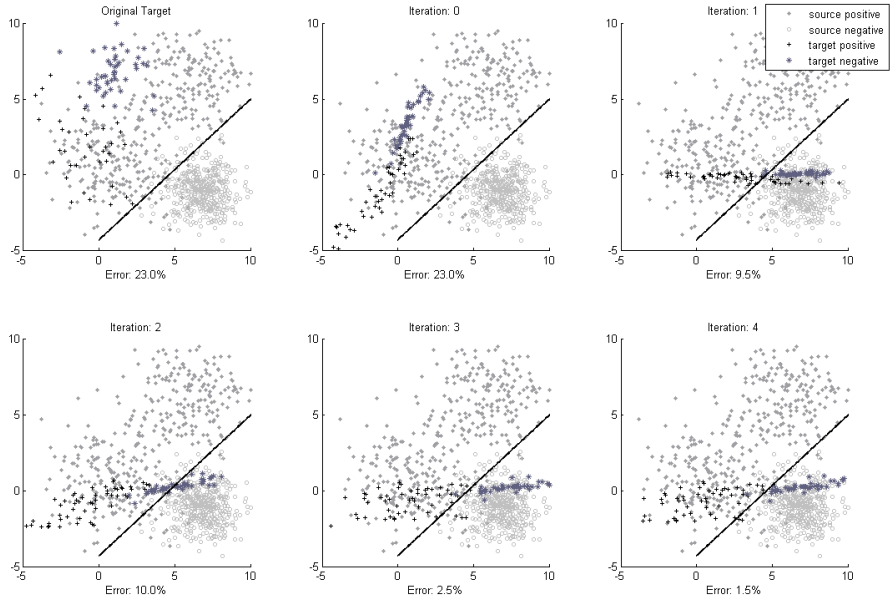

Fig. 1. The first 4 iterations of DML algorithm on Synthetic Database

per target domain size could not be trusted, thus for each size of the target domain we performed 10 runs and computed the average value of the elementary accuracy scores along with their variances.

Supervised SVMs trained on the target training data were employed as a baseline method (the usual choice in domain adaptation settings [5]). and SVM-Light [13] was used as an SVM implementation.

4.2 Synthetic Database

To gain insight into the behavour of our Transformation-based approach, we considered synthetically generated source and target domains. In order to visualize it, both domains were two dimensional. The positive samples of the source domain were generated based on the sum of two Gaussian distributions and the negative ones similarly, but using just one Gaussian distribution. We generated 1000 samples and used only the first 800 of them as the training database of the source domain. The training and evaluation sets of the target domain were generated from the previously generated 1000 samples by rotating them by 90 degrees and the same train-test split was employed.

In Fig. 1 we can see a sample run of the PolyDML algorithm on the synthetic database. We applied the Polynomial kernel with $d = 1$ (i.e. the Linear

kernel) and set the C value of SVM to 1. The figure shows six different states of the algorithms. In each state we can see the data samples of the source domain and the classification boundary, which are constants. The first state shows the position of the original training samples of the target domain based on the samples taken from the source domain. The second state called "Iteration 0" shows the position of samples of the target domain which were transformed by applying a $W^{(0)}$ random transformation from the gradient descent-based algorithm proposed in section 3.2. The next four states show the first four iterations of the DML algorithm. For each state we also included the error measured on the target train dataset. As one can see, in the initial states (i.e. in the first two states) the error rate is quite high, but in the first four iterations the error rate decreases fast and almost monotonically. PolyDML significantly outperforms the supervised baseline as well ($Error = 17.0\%$).

4.3 Results on Multi-Domain Sentiment Dataset

Our Transformation-based method was evaluated on Opinion Mining datasets [14] as well. These datasets contains product reviews taken from Amazon.com for four product types (domains), namely: books, DVDs, electronics and kitchen appliances. Originally the attributes of instances of each dataset were the term frequencies of the words of the corresponding review texts, and the labels of the instances were generated from the rating of the reviews. More precisely, reviews with rating ≥ 3 were considered as positive, while those with rating < 3 were labeled negative (binary classification problem). The datasets of each domain were balanced, all of them having 1000 positive and 1000 negative samples with a very high dimension (about 5000 dimensions), because each different word in a review generates a dimension in the database.

We split the datasets of each domain into two parts in a random way (80% training set and 20% evaluation set). Then we performed a feature selection step, selecting the attributes where the InfoGain score was positive on the train set and performed a Principle Component Analysis (PCA) on each training dataset. The feature dimensionality reduction steps found on the training sets were then applied to the evaluation sets.

Since we had four different domains, we investigated all the possible 12 domain adaptation tasks. The results of this are summarized in Fig. 2. Each subfigure shows the results of RBFDML and the corresponding supervised methods (baselines) and – with a horizontal line – the result of the direct method applying the SVM source model which uses the *full* training dataset of the target domain. This is independent of the values of the x axis and can be viewed as the "limit values" of the corresponding results of direct methods. At each point in the sub-figures we can see average accuracy scores of 10.

As can be seen in Fig. 2, when we use limited-sized datasets from the target domain, the proposed methods can achieve a significantly higher accuracy than the baseline methods.The reason for this phenomenon might be that the baseline could not made valid generalization from the small number of samples – since the database of the target domain might not contain enough information to

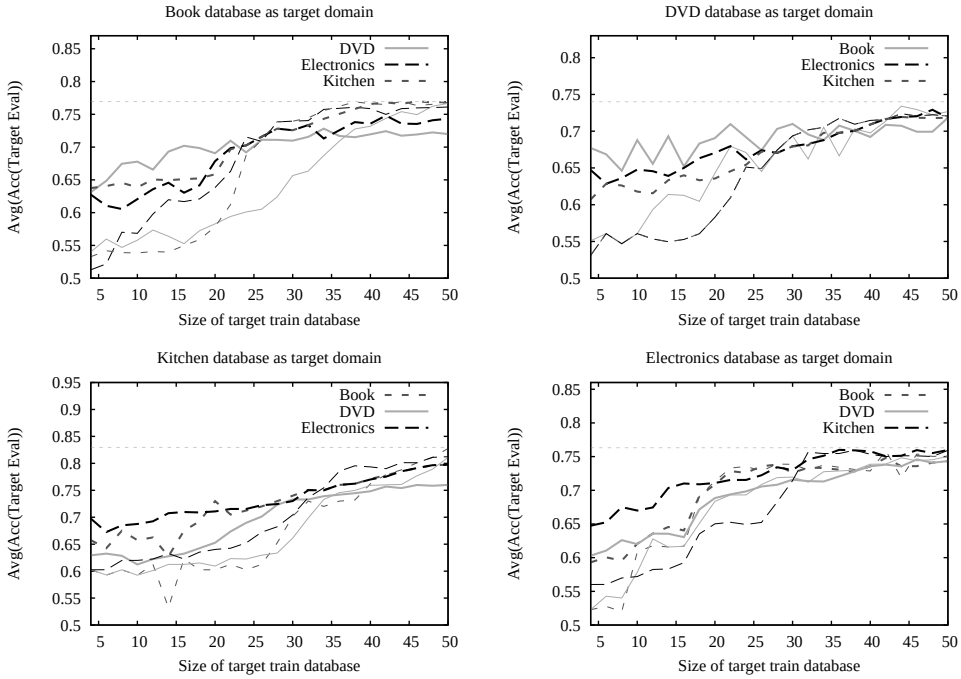

Fig. 2. The average accuracies of RBFDML algorithm using different sizes of subsets of the target domains of Multi-Domain Sentiment Dataset. (In each subfigure each of thinner lines denotes the corresponding baseline result, and the result denoting by a horizontal line accords to the full sized target train dataset.)

build a well-generalizing model – but the transformation-based approach uses the well-generalized source model which helps the generalization of the final transformation-based model.

Structural Correspondence Learning (SCL) is a domain adaptation approach [14] which has published results on the Opinion Mining datasets we used. In comparison with its results, our approach achived better accuracy scores 10 times compared to the base SCL, and 7 times compared its extended version (SCL-MI).

5 Conclusions

In this paper, we presented our novel, transformation-based approach for handling the task of domain adaption. We have described two instances of our main algorithm and experimentally showed that – applying them to a real world dataset in 12 different scenarios – our methods outperform the baseline approaches (direct methods) and published results of the same dataset.

Our experimental results proved that the approach it is possible to train models for the target domain that uses a very limited number of labeled samples taken from the target domain. This is true as well in those cases when there are enough samples, but baseline methods cannot generalize well using such samples. On the other hand, our approach has a key advantage against other domain adaptation procedures as it does not require access to the source data just to a trained source model which can be crucial in several cases (e.g. privacy issues).

In the near future we would like to investigate our general approach with other learning models.

References

1. Daumé, III, H., Marcu, D.: Domain adaptation for statistical classifiers. J. Artif. Int. Res. **26**(1) (2006) 101–126
2. Kobayashi, N., Inui, K., Matsumoto, Y.: Extracting aspect-evaluation and aspect-of relations in opinion mining. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Prague, Czech Republic, Association for Computational Linguistics (June 2007) 1065–1074
3. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery **2**(2) (1998) 121–167
4. Chelba, C., Acero, A.: Adaptation of maximum entropy capitalizer: Little data can help a lot. Computer Speech & Language **20**(4) (2006) 382–399
5. Daumé, III, H.: Frustratingly easy domain adaptation. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Association for Computational Linguistics (2007) 256–263
6. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: Advances in Neural Information Processing Systems 20, Cambridge, MA, MIT Press (2007)
7. Gupta, R., Sarawagi, S.: Domain adaptation of information extraction models. SIGMOD Rec. **37**(4) (2008) 35–40
8. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. In: IJCAI. (2009) 1187–1192
9. Mansour, Y., Mohri, M., Rostamizadeh, A.: Domain adaptation: Learning bounds and algorithms. CoRR **abs/0902.3430** (2009)
10. Mansour, Y., Mohri, M., Rostamizadeh, A.: Domain adaptation with multiple sources. In: NIPS. (2008) 1041–1048
11. Snyman, J.A.: Practical Mathematical Optimization: An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms (Applied Optimization). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2005)
12. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press (2000)
13. Joachims, T.: Making large-scale support vector machine learning practical. (1999) 169–184
14. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, Prague, Czech Republic, Association for Computational Linguistics (2007) 440–447