Cross Domain Active Learning

Christian Poelitz
TU Dortmund University, Dortmund 44227, Germany
christian.poelitz@cs.uni-dortmund.de

Abstract

In this paper, we propose a solution to reduce the labeling costs by applying domain adaption methods coupled with active learning to reduce the number labels needed to train a classifier. We assume to have only one task but different domains in the sense that we have texts that come from different distributions. Our approach uses multi domain learning together with active learning to find a minimum number of texts to label from as few domains as possible to train a classifier with a certain confidence in its predictions.

1 Introduction

A large cost factor in computer linguistic rises from the labeling of texts. For example, we want to investigate the hypothesis that certain statements occur always in positive or negative context in a large set of texts. A usual approach in computer linguistic would be to go through the texts and label parts of it as positive or negative to use them as examples for a classifier. This can be quite expensive with respect to the texts and the task.

The problem gets an additional twist, in case we have a large corpus or many corpora of texts from different domains. To investigate the texts, we would start to label the texts from a certain domain. Now, it is easier to keep labeling only texts from this domain instead of switching to another.

To reduce labeling costs, we propose to use active learning techniques to support us on what to label. We base our decision on what to label on a trained classifier and the confidence of the classification of unlabeled texts. First, we need to know when we can stop labeling in the current domain. This will happen when further labels will not increase the quality of the classifier anymore. Next, we want to find out if we actually need labels from the other domains when there are not further texts in the current domain or the quality of the classifier saturates. In case we have large confidence in predicting the texts in all domains we do not need further labels. This means, we train a classifier on only some domains and expect a generalization on all domains. Unfortunately, this is usually only possible under strong assumptions on the distributions of the texts in the different domains. By distributions of the texts we mean the probability distribution of a stochastic process that generates the text.

Further, we assume a nonlinear cost model with respect to the number of labels. We expect that at the beginning the labeling of the texts demands largest effort but decreases with more and more labels. This is intuitively clear since we will grow accustom to the texts. Further, when we start labeling texts from an other domain, we must consider that the effort will be again higher at the beginning and smaller after a while.

The paper is organized as the following. First, we explain how we can statistically model the distribution of texts in the different domains and what classifier we use in our training. Then, we describe how we use domain adaptation and active learning to train a classifier for all domains. Finally, we report results on our propose method on a benchmark data set.

2 Related Work

We leverage methods from active learning, domain adaptation and multi domain learning.

Active learning tries to direct the labeling process considering intermediate results. A classifier that is trained on a small amount of labeled texts is used to estimate which further texts should be labeled to increase the quality of the classifier when trained also on these labeled texts. As candidates for further labeling we use the texts that are classified with least confidence. This strategy is called

uncertainty sampling (LC94). There are different sampling strategies in the literature. A general overview is given by (Set09).

We assume that the texts have different distributions in the domains but the labels have the same distribution given a text. In this case, instance weights can be used. In (JZ07), a classifier is trained on examples with labels and weights for each example. The weights are chosen such that the mass distribution of the examples from one domain adapts to the mass distributions of an other domain. By this, they train a classifier using examples and labels from one domain that generalizes to an other domain. An other approach is to model the commonalities of different domains as proposed by (BMP06) or (DM06) for instance.

In multi domain learning, a classifier is learned over several domains. A classifier shall be generated that performs best over all domains while using only a small amount of training data that comes from very few domains. In best case we need only to training the classifier on a single domain. An overview on existing multi learning methods is given by (JCDR12).

3 Statistical modeling

Since we pose assumptions on the distributions of texts, we need to model these distributions based on text examples from the domains. There are many approaches to model the probability distribution of texts. Here, we use the language model (PC98).

The probability of a word w or a sequence of words (a text) can be estimated by the frequency of the occurrences of the word. Formally, we note $p(w_1 \cdots w_n)$ as the probability of the event to see (or to read) the sequence $w_1 \cdots w_n$ in the domain. Further, $p(w_n|w_1 \cdots w_{n-1})$ is the probability of seeing word w_n, after we have already seen the words $w_1 \cdots w_{n-1}$. By assuming independence of words that are farther away from each other than a given context size, we can estimate the probability by frequencies easily. Using a context of only one word we also speak of a bigram model in contrast to a unigram model when we assume all words are independent, hence: $p(w_1 \cdots w_n) = \prod_{i=1}^{n} p(w_i)$. This naturally generalizes to ngram models, when we consider a context of $n-1$ words.

The concrete probabilities for unigrams can be estimated for a given domain by the maximum likelihood estimate of the Multinomial distribution, hence $p(w) = \frac{N_w}{N}$ for N_w number of occurrences of word w among the N words in the domain.

4 Classifier

As classifier we use support vector machines that have proven to be efficient in text classification, see (Joa02) for example. Given a set of texts with labels, we find a separating hyperplane in a Reproducing Kernel Hilbert space. In this paper we use the bag of word representation. Each text is mapped to a large vector (a word vector) such that each component tells how many times a certain word occurs in the text.

During SVM training we minimize a regularized loss, formally $min_f \frac{1}{N} \sum_{i=1}^{N} [(1 - y_i \cdot f(x_i))_+] + \lambda \cdot ||f||$ using the hinge loss $()_+$, y_i the labels and x_i the texts. We use an adaptation that integrates weights on the texts. This means we solve the following minimization problem: $min_f \frac{1}{N} \sum_{i=1}^{N} \beta_i \cdot [(1 - y_i \cdot f(x_i))_+] + \lambda \cdot ||f||$ See (LLW02) for further details.

In order to retrieve confidence in the prediction of our classifier we use the approach by (Pla99) to derive posterior probabilities using the outcome of an SVM. The probability of a prediction given an example (here a text) is modeled as sigmoid function: $P(y = 1|f(x)) = \frac{1}{1+exp(A \cdot f(x)+B)}$. The parameters A and B are estimated using the labeled texts. Using this method, we get the confidence of the prediction of a text as.

5 Domain Adaptation

Assuming that the texts are differently distributed in different domains we use the SVM with weighted examples as described above. The weights are estimated based on the difference of the distributions of texts using importance sampling based on language models.

5.1 Importance sampling

If P_s and P_t are the text distributions from domain s and domain t with the same support, we can estimate the expected loss under the domain t using texts from domain s, using importance sampling. In importance sampling we sample from P_s but weight the examples by $\beta(x)$ such that $\beta(x) \cdot x$ has approximately the distribution P_t. For further reading we refer to (OZ00). We integrate these weights into the risk minimization framework for

the SVM using the hinge loss L. This results to the following:

$$E(L(x,y,f)) = \int L(x,y,f) \cdot P_t(x,y) \cdot dx$$

$$= \int \frac{P_t(x,y)}{P_s(x,y)} \cdot L(x,y,f) \cdot P_s(x,y) \cdot dx$$

$$\simeq \frac{1}{N} \sum_{i=1}^{N} \frac{P_t(x,y)}{P_s(x,y)} \cdot L(x,y,f)$$

$$= \frac{1}{N} \sum_{i=1}^{N} \beta_i \cdot L(x,y,f)$$

In this paper we concentrate on covariate shifts. This means, we expect that conditional probabilities of the labels, given an observation, are the same over two different domains. This means, $P_s(y|x) = P_t(y|x)$. Hence, we can write $\frac{P_t(x,y)}{P_s(x,y)} = \frac{P_t(x) \cdot P_t(y|x)}{P_s(x) \cdot P_s(y|x)} = \frac{P_t(x)}{P_s(x)}$.

5.2 Multi Domain Classifier

We want to train a classifier that can be applied on different domains but the training is only done on texts from a single domain or a small amount of domains. Using the language model we estimate the probability distributions of the texts from each domain i, noted as $P_i(x)$. Further, we define an ensemble of classifiers $f_i(x)$. Each classifier f_i is trained with respect to the distribution of domain i using importance sampling on an other domain. Given a trained set of classifiers f_i we perform the prediction on a given text - from any domain - as: $F(x) = f_{i(x)}(x)$ with $i(x) = argmax_i\{P_i(x)\}$.

6 Active Learning across different domains

In this section we describe how we use active learning and domain adaptation in order to reduce the labeling effort over different domains in a classification task. We generally assume that the distribution of the texts differ among different domains. Formally this means $P_i(x) \neq P_j(x)$, for two different domains i and j and a text x. Further, we assume that the distributions of the labels for a given example are the same among the domains, hence $P_k(y = i|x) = P_l(y = i|y)$.

The goal is to train the classifier only on few domains and examples but apply it to all domains. To achieve this goal, we use an active learning technique to ask for labels in a certain domain such that the number of overall labels are minimized while maximizing the expected quality over all domains. Therefore, we train an SVM with probabilistic outputs to estimate the confidence in the predictions. This means, for each domain i we train an SVM on the texts from a single domain j, but weight them as described above by $\frac{P_i(x)}{P_j(x)}$. These classifiers f_i are then combined to the multi domain classifier F. The multi domain classifier is applied to all unlabeled examples from all domains. In case all the resulting predictions have at least a certain level of confidence we can stop here and use F as final classifier. When there are still predictions with less confidence we need further labels.

There are two possibilities to continue. First, we can ask for more labels from the current domain. Second, we ask for labels from any different domain. We propose to suggest to switch to a different domain only when there are no further unlabeled examples in the current domain. Then, the next domain can be any domain that still contains unlabeled examples. This is a valid approach since we expect that the texts samples in the different domains are independent identical distributed. Then, we only need to start asking for labels from a next domain in case we have no further examples in the current domain. In general, with enough examples in one domain we expect the same quality of the classifier when we train only on this domain as when we train on all domains. This holds because we expect only a covariance shift.

When we continue - maybe with a new current domain - we apply each classifier f_i on all unlabeled examples from the current domain, but each time we weight the examples before applying the classifier. By this we adapt the mass distribution to the corresponding domain. Then, the examples that have the least confident predictions among all classifiers should be used. Hence, among the least confident predictions of the unlabeled data we sample k examples and ask for their labels. Afterwards, we train the classifiers again using also the newly labeled samples, build the multi domain classifier and test if we have enough confident predictions now. If we still have some examples with low confidence, we simply perform the steps again.

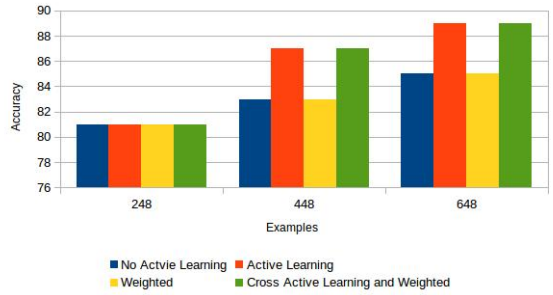

Figure 1: Results on Reuters People.

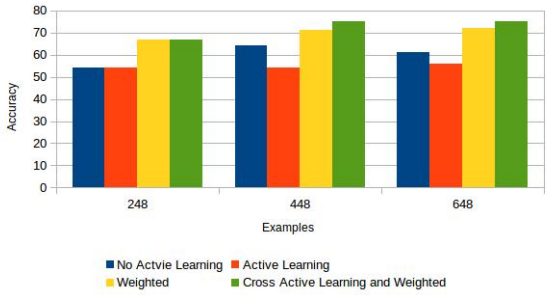

Figure 2: Results on Reuters Organizations.

7 Experiments

We test our proposed method on a standard benchmark data set that is commonly used in NLP. We use the Reuters-21578[1] data set with the topics people and organizations. For both domains we estimate a language model to model $P_i(d)$ the probability that document d was generated in the domain i for $i = s$ the source domain containing texts talking about people and $i = t$ the target domain containing texts talking about organizations. We use the weighted SVM as classifier and weighted texts from the source domain for training. The weights are chosen with respect to the probability of the texts on target domain and source domain as explained above.

We split the data from the source domain into 3 parts each having $1/5, 1/5, 3/5$ of the original data. One split is used for the first training without active learning. One split is used for testing the classifier and the final split is used for active learning as described above. We use always batches of 200 examples for which we ask labels. In iteration k these 200 are the examples that are classified with the least confidence by the classifier trained with all labeled examples so far.

To investigate the behavior of our proposed model we conducted several experiments. First, we tested how good we perform on the source domain when we use an active learning approach. Then, we investigate how our trained classifier performs on the target domain. We are specially interested in how much benefit we get from weightening the examples. Finally, we test our proposed active learning strategy across the two domains.

Figure 1 and Figure 2 shows the accuracy on the source domain respectively target domain for different experiments. For the first batch, we cannot perform an active learning strategy since we need a trained classifier first. Only after we already have trained a classifier we can perform an active learning strategy. The first two bars show the results on the accuracy when we perform an active learning strategy and when not. We get faster a better accuracy on the source domain when we actively ask for the next labels. Next, we test the classifier on the target domain. The first two bars show the accuracy when the classifier is only trained on the source domain without weights. We see that the accuracy is low and more train data increases the quality only slightly. Furthermore, we see that the active learning strategy, which considers only the source domain here, performs worse on the target domain. Next, we investigate the domain adaption by weighting the texts with respect to the language models. The third bar shows that the accuracy on the target domain increases when we use importance sampling. Finally, we test our active learning strategy across the two domains. The last bar shows that when we actively ask for labels with respect to both domains, we get the best overall accuracy.

8 Conclusion and Future Work

We explained an approach to perform active learning across different domains. We used importance sampling and statistical language models to adapt an SVM trained on a certain source domain to a different target domain. Our proposed active learning strategy that considers both domains shows good results on a benchmark data set. In the future we want to investigate how other probability models for the texts can be used. Further, we plan to extend our approach to multi task learning.

[1] http://www.daviddlewis.com/resources/testcollections/

References

John Blitzer, Ryan McDonald, and Fernando Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 120–128, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

Hal Daumé, III and Daniel Marcu. Domain adaptation for statistical classifiers. *J. Artif. Int. Res.*, 26(1):101–126, May 2006.

Mahesh Joshi, William W. Cohen, Mark Dredze, and Carolyn P. Rosé. Multi-domain learning: when do domains matter? In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1302–1312, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

Thorsten Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

Jing Jiang and Chengxiang Zhai. Instance weighting for domain adaptation in nlp. In *In ACL 2007*, pages 264–271, 2007.

David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *In Proceedings of the Eleventh International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1994.

Yi Lin, Yoonkyung Lee, and Grace Wahba. Support vector machines for classification in nonstandard situations. *Mach. Learn.*, 46(1-3):191–202, March 2002.

Art Owen and Yi Zhou. Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):pp. 135–143, 2000.

Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

Burr Settles. Active Learning Literature Survey. Technical Report 1648, University of Wisconsin–Madison, 2009.