Part-of-speech tagging with antagonistic adversaries

Anders Søgaard
Center for Language Technology
University of Copenhagen
DK-2300 Copenhagen S
soegaard@hum.ku.dk

Abstract

Supervised NLP tools and on-line services are often used on data that is very different from the manually annotated data used during development. The performance loss observed in such cross-domain applications is often attributed to covariate shifts, with out-of-vocabulary effects as an important subclass. Many discriminative learning algorithms are sensitive to such shifts because highly indicative features may swamp other indicative features. Regularized and adversarial learning algorithms have been proposed to be more robust against covariate shifts. We present a new perceptron learning algorithm using antagonistic adversaries and compare it to previous proposals on 12 multilingual cross-domain part-of-speech tagging datasets. While previous approaches do not improve on our supervised baseline, our approach is better across the board with an average 4% error reduction.

1 Introduction

Most learning algorithms assume that training and test data are governed by identical distributions; and more specifically, in the case of part-of-speech (POS) tagging, that training and test sentences were sampled at random and that they are identically and independently distributed. Significance is usually tested across data points in standard NLP test sets. Such datasets typically contain running text rather than independently sampled sentences, thereby violating the assumption that data points are independently distributed and sampled at random. More importantly, significance across data points only says something about the likelyhood of observing the same effect on *more data sampled the same way*, but says nothing about likely performance on sentences sampled from different sources or different domains.

This paper considers the POS tagging problem, i.e. where we have training and test data consisting of sentences in which all words are assigned a label y chosen from a finite set of class labels $\{$NOUN, VERB, DET,$\ldots\}$. We assume that we are interested in *performance across data sets or domains* rather than just performance across data points, but that we do not know the target domain in advance. This is often the case when we develop NLP tools and on-line services. We will do cross-domain experiments using several target domains in order to compute significance across domains, enabling us to say something about likely performance on new domains.

Several authors have noted how POS tagging performance is sensitive to cross-domain shifts (Blitzer et al., 2006; Daume III, 2007; Jiang and Zhai, 2007), and while most authors have assumed known target distributions and pool unlabeled target data in order to automatically correct cross-domain bias (Jiang and Zhai, 2007; Foster et al., 2010), methods such as feature bagging (Sutton et al., 2006), learning with random adversaries (Globerson and Roweis, 2006) and L_∞-regularization (Dekel and Shamir, 2008) have been proposed to improve performance on unknown target distributions. These methods explicitly or implicitly try to minimize average or worst-case expected error across a set of possible test distributions in various ways. These algorithms are related because of the intimate relationship between adversarial corruption and regularization (Ghaoui and Lebret, 1997; Xu et al.,

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 640–644,
Sofia, Bulgaria, August 4-9 2013. ©2013 Association for Computational Linguistics

2009; Hinton et al., 2012). This paper presents a new method based on learning with antagonistic adversaries.

Outline. Section 2 introduces previous work on robust perceptron learning, as well as the methods dicussed in the paper. Section 3 motivates and introduces learning with antagonistic adversaries. Section 4 presents experiments on POS tagging and discusses how to evaluate cross-domain performance. Learning with antagonistic adversaries is superior to the other approaches across 10/12 datasets with an average error reduction of 4% over a supervised baseline.

Motivating example. The problem with out-of-vocabulary effects can be illustrated using a small labeled data set: $\{\mathbf{x}_1 = \langle 1, \langle 0,1,0 \rangle \rangle, \mathbf{x}_2 = \langle 1, \langle 0,1,1 \rangle \rangle, \mathbf{x}_3 = \langle 0, \langle 0,0,0 \rangle \rangle, \mathbf{x}_4 = \langle 1, \langle 0,0,1 \rangle \rangle\}$. Say we train our model on $\mathbf{x}_{1-3}$ and evaluate it on the fourth data point. Most discriminate learning algorithms only update parameters when training examples are misclassified. In this example, a model initialized by zero weights would misclassify $\mathbf{x}_1$, update the parameter associated with feature x_2 at a fixed rate α, and the returned model would then classify all data points correctly. Hence the parameter associated with feature x_3 would never be updated, although this feature is also correlated with class. If x_2 is missing in our test data (out-of-vocabulary), we end up classifying all data points as negative. In this case, we would wrongly predict that $\mathbf{x}_4$ is negative.

2 Robust perceptron learning

Our framework will be averaged perceptron learning (Freund and Schapire, 1999; Collins, 2002). We use an additive update algorithm and average parameters to prevent over-fitting. In adversarial learning, adversaries corrupt the data point by applying transformations to data points. Antagonistic adversaries choose transformations informed by the current model parameters $\mathbf{w}$, but random adversaries randomly select transformations from a predefined set of possible transformations, e.g. deletions of at most k features (Globerson and Roweis, 2006).

Feature bagging. In feature bagging (Sutton et al., 2006), the data is represented by different bags of features or different views, and the models learned using different feature bags are combined by averaging. We can reformulate feature bagging as an adversarial learning problem. For each pass, the adversary chooses a deleting transformation corresponding to one of the feature bags. In Sutton et al. (2006), the feature bags simply divide the features into two or more representations. In an online setting feature bagging can be modelled as a game between a learner and an adversary, in which (a) the adversary can only choose between deleting transformations, (b) the adversary cannot see model parameters when choosing a transformation, and in which (c) the adversary only moves in between passes over the data.[1]

Learning with random adversaries (LRA). Globerson and Roweis (2006) let an adversary corrupt labeled data during training to learn better models of test data with missing features. They assume that missing features are randomly distributed and show that the optimization problem is a second-order cone program. LRA is an adversarial game in which the two players are unaware of the other player's current move, and in particular, where the adversary does not see model parameters and only randomly corrupts the data points. Globerson and Roweis (2006) formulate LRA as a batch learning problem of minimizing worst case loss under deleting transformations deleting at most k features. This is related to regularization in the following way: If model parameters are chosen to minimize expected error in the absence of any k features, we explicitly prevent under-weighting more than $n - k$ features, i.e. the model must be able to classify data well in the absence of any k features. The sparsest possible model would thus assign weights to $k + 1$ parameters.

$\mathbf{L}_\infty$-regularization hedges its bets even more than adversarial learning by minimizing expected error with $\max ||\mathbf{w}|| < C$. In the online setting, this corresponds to playing against an adversary that clips any weight above a certain threshold C, whether positive or negative (Dekel and Shamir, 2008). In geometric terms the weights are projected back onto the hyper-cube C. A related approach, which is not explored in the experiments below, is to regularize linear models toward weights with low variance (Bergsma et al., 2010).

[1] Note that the batch version of feature bagging is an instance of group L_1 regularization (Jacob et al., 2009; Schmidt and Murphy, 2010; Martins et al., 2011). Often group regularization is about finding sparser models rather than robust models. Sparse models can be obtained by grouping correlated features; non-sparse models can be obtained by using independent, exhaustive views.

```
$X = \{\langle y_i, \mathbf{x}_i \rangle\}_{i=1}^N$, $\delta$ deletion rate
$\mathbf{w}^0 = 0, \mathbf{v} = 0, i = 0$
for $k \in K$ do
    for $n \in N$ do
        $\xi^1 \leftarrow \mathbf{random.sample}(P(1) = 1 - \delta)$
        $\xi^2 \leftarrow ||\mathbf{w}|| < \mu_{||\mathbf{w}||} + \sigma_{||\mathbf{w}||}$
        $\xi \leftarrow (\xi^1 + \xi^2)_{(0,1)}$
        if sign$(\mathbf{w} \cdot \mathbf{x}_n \circ \xi) \neq y_n$ then
            $\mathbf{w}^{i+1} \leftarrow \mathbf{update}(\mathbf{w}^i)$
            $i \leftarrow i + 1$
        end if
        $\mathbf{v} \leftarrow \mathbf{v} + \mathbf{w}^i$
    end for
end for
return $\mathbf{w} = \mathbf{v}/(N \times K)$
```

Figure 1: Learning with antagonistic adversaries

3 Learning with antagonistic adversaries

The intuition behind learning with antagonistic adversaries is that the adversary should focus on the most predictive features. In the prediction game, this would allow the adversary to inflict more damage, corrupting data points by removing good features (rather than random ones). If the adversary focuses on the most predictive features, she is implicitly regularizing the model to obtain a more equal distribution of weights.

We draw random binary vectors with $P(1) = 1 - \delta$ as in adversarial learning, but deletions are only effective if $\xi_j = 0$ *and* the weight $\mathbf{w}_j$ is more than a standard deviation $(\sigma_{||\mathbf{w}}||)$ from the mean of the current absolute weight distribution $(\mu_{||\mathbf{w}}||)$. In other words, we only delete the predictive features, with predictivity being relative to the current mean weight.

The algorithm is presented in Figure 1. For each data point, we draw a random binary vector ξ_1 with δ chance of zeros. ξ_2 is a vector with the ith scalar zero if and only if the absolute value of the weight w_i in $\mathbf{w}$ is more than a standard deviation higher than the current mean. The ith scalar in ξ is only zero if the ith scalars in both ξ_1 and ξ_2 are zero. The corresponding features are a random subset of the predictive features.[2]

4 Experiments

We consider part-of-speech (POS) tagging, i.e. the problem of assigning syntactic categories to word tokens in running text. POS tagging accuracy is known to be very sensitive to domain shifts. Foster et al. (2011) report a POS tagging accuracy on social media data of 84% using a tagger that acchieves an accuracy of about 97% on newspaper data. In the case of social media data, many errors occur due to different spelling and capitalization conventions. The main source of error, though, is the increased out-of-vocabulary rate, i.e. the many unknown words. While POS taggers can often recover the part of speech of a previously unseen word from the context it occurs in, this is harder than for previously seen words.

We use the LXMLS toolkit[3] as our baseline with the default feature model, but use the PTB tagset rather than the Google tagset (Petrov et al., 2011) used by default in the LXMLS toolkit. We use four groups of datasets. The first group comes from the English Web Treebank (EWT),[4] also used in the Parsing the Web shared task (Petrov and McDonald, 2012). We train our tagger on Sections 2–21 of the WSJ data in the Penn-III Treebank (PTB), Ontonotes 4.0 release. The EWT contains development and test data for five domains: answers (from Yahoo!), emails (from the Enron corpus), BBC newsgroups, Amazon reviews, and weblogs. We use the emails development section for development and test on the remaining four test sets. We also do experiments with additional data from PTB. For these experiments we use the 0th even split of the biomedical section (PTB-biomedical) as development data, the 9th split and the chemistry section (PTB-chemistry) as test data, and the remaining biomedical data (splits 1–8) as training data. This data was also used for developing and testing in the CoNLL 2007 Shared Task (Nivre et al., 2007).

Our third group of datasets also comes from Ontonotes 4.0.[5] We use the Chinese Ontonotes (CHO) data, covering five different domains. We use newswire for training data and randomly sampled broadcasted news for development. Finally we do experiments with the Danish section of the Copenhagen Dependency Treebank (CDT). For CDT we rely on the treebank meta-data and sin-

[2] The approach taken is similar in spirit to confidence-weighted learning (Dredze et al., 2008). The intuition behind confidence-weighted learning is to more agressively update rare features or features that we are less confident about. In learning with antagonistic adversaries the adversaries delete predictive features; that is, features that we are confident about. When these features are deleted, we do not update the corresponding weights. In relative terms, we therefore update rare features more aggressively than common ones. Note also that by doing so we regularize toward weights with low variance (Bergsma et al., 2010).

[3] https://github.com/gracaninja/lxmls-toolkit

[4] LDC Catalog No.: LDC2012T13.

[5] LDC Catalog No.: LDC2011T03.

	SP	Our	L_∞	LRA
EWT-answers	86.04	**86.06**	85.90	**86.06**
EWT-newsgroups	87.70	**87.92**	87.78	87.66
EWT-reviews	85.96	**86.10**	85.80	86.00
EWT-weblogs	87.59	**87.89**	87.60	87.54
PTB-biomedical	95.05	95.26	**95.46**	94.43
PTB-chemistry	90.32	**90.60**	90.56	90.58
CHO-broadcast	78.38	**78.42**	78.27	78.28
CHO-magazines	78.50	**78.57**	76.80	78.29
CHO-weblogs	79.64	**79.76**	79.24	79.37
CDT-law	93.96	**95.64**	93.91	94.25
CDT-literature	93.93	**94.19**	94.15	94.15
CDT-magazines	94.95	**95.06**	94.71	95.04
Wilcoxon p		<0.01		
macro-av. err.red		4.0	-1.2	-0.2

Table 1: Results (in %).

gle out the newspaper section as training data and use held-out newspaper data for development.

We observe two characteristics about our datasets: (a) The class distributions are relatively stable across domains. For CDT, for example, we see almost identical distributions of parts of speech, except literature has more prepositions. (b) The OOV rate is significantly higher across domains than within domains. This holds even for the PTB datasets, where the OOV rate is 14.6% on the biomedical test data, but 43.3% on the chemistry test data. These two observations confirm that cross-domain data is primarily biased by covariate shifts.

All learning algorithms do the same number of passes over each training data set. The number of iterations was set optimizing baseline system performance on development data. For EWT and CHO, we do 10 passes over the data. For PTB, we do 15 passes over the data, and for CDT, we do 25 passes over the data. The deletion rate in adversarial learning was fixed to 0.1% (optimized on the EWT emails data; not optimized on PTB, CHO or CDT). In L_∞-regularization, the parameter C was optimized the same way and set to 20. Results are averages over five runs.

4.1 Results

The results are presented in Table 1. Learning with antagonistic adversaries performs significantly better than structured perceptron (SP) learning, L_∞-regularization and LRA across the board. We follow Demsar (2006) in computing significance across datasets using a Wilcoxon signed rank test. This is a strong result given that our algorithm is as computationally efficient as SP and does not pool unlabeled data to adapt to a specific target distribution. What we see is that letting an antagonistic adversary corrupt our labeled data - somewhat surprisingly, maybe - leads to better cross-domain performance. L_∞-regularization leads to worse performance, and LRA performs very similar to SP on average. Improvements to LRA have also been explored in Trafalis and Gilbert (2007) and Dekel and Shamir (2008). We note that on the in-domain dataset (PTB-biomedical), L_∞-regularization performs best, but our approach also performs better than the structured perceptron baseline on this dataset.

4.2 Analysis

The number of zero weights or very small weights is significantly lower for learning with antagonistic adversaries than for the baseline structured perceptron. So our models become less sparse. On the other hand, we have more parameters with average weights in our models. Weights are in other words better distributed. We also observe that parameters are updated slightly more with antagonistic adversaries. In our PTB experiments, for example, the mean weight is 14.2 in structured perceptron learning, but 14.5 with antagonistic adversaries. On the other hand, weight variance is slightly lower; recall the connection to variance regularization (Bergsma et al., 2010). Note that L_∞-regularization with $C = 20$ corresponds to clipping all weights above 20, i.e. roughly a third of the weights in this case. To validate our intuitions about what is going on, we also tried to increase the deletion rate. If δ is increased to 1%, the mean weight goes up to 19.2. The adversarial model is less sparse than the baseline model.

A last observation is that the structured perceptron baseline model expectedly fits the training data better than the robust models. On CDT, the structured perceptron has an accuracy of 98.26% on held-out training data, whereas our model has an accuracy of only 97.85%. The L_∞-regularized has an accuracy of 97.82%, whereas LRA has an accuracy of 98.18%.

5 Conclusion

We presented a discriminative learning algorithms for cross-domain structured prediction that seems more robust to covariate shifts than previous approaches. Our approach was superior to previous approaches across 12 multilingual cross-domain POS tagging datasets, with an average error reduction of 4% over a structured perceptron baseline.

References

Shane Bergsma, Dekang Lin, and Dale Schuurmans. 2010. Improved natural language learning via variance-regularization support vector machines. In *CoNLL*.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *EMNLP*.

Michael Collins. 2002. Discriminative training methods for Hidden Markov Models. In *EMNLP*.

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *ACL*.

Ofer Dekel and Ohad Shamir. 2008. Learning to classify with missing and corrupted features. In *ICML*.

Janez Demsar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.

Mark Dredze, Koby Crammer, and Fernando Pereira. 2008. Confidence-weighted linear classification. In *ICML*.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *EMNLP*.

Jennifer Foster, Ozlem Cetinoglu, Joachim Wagner, Josef Le Roux, Joakim Nivre, Deirde Hogan, and Josef van Genabith. 2011. From news to comments: Resources and benchmarks for parsing the language of Web 2.0. In *IJCNLP*.

Yoav Freund and Robert Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning*, 37:277–296.

Laurent El Ghaoui and Herve Lebret. 1997. Robust solutions to least-squares problems with uncertain data. In *SIAM Journal of Matrix Analysis and Applications*.

Amir Globerson and Sam Roweis. 2006. Nightmare at test time: robust learning by feature deletion. In *ICML*.

Geoffrey Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. http://arxiv.org/abs/1207.0580.

Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. 2009. Group lasso with overlap and graph lasso. In *ICML*.

Jing Jiang and ChengXiang Zhai. 2007. Instance weighting for domain adaptation in NLP. In *ACL*.

Andre Martins, Noah Smith, Pedro Aguiar, and Mario Figueiredo. 2011. Structured sparsity in structured prediction. In *EMNLP*.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 Shared Task on Dependency Parsing. In *EMNLP-CoNLL*.

Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 Shared Task on Parsing the Web. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. CoRR abs/1104.2086.

Mark Schmidt and Kevin Murphy. 2010. Convex structure learning in log-linear models: beyond pairwise potentials. In *AISTATS*.

Charles Sutton, Michael Sindelar, and Andrew McCallum. 2006. Reducing weight undertraining in structured discriminative learning. In *NAACL*.

T Trafalis and R Gilbert. 2007. Robust support vector machines for classification and computational issues. *Optimization Methods and Software*, 22:187–198.

Huan Xu, Constantine Caramanis, and Shie Mannor. 2009. Robustness and regularization of support vector machines. In *JMLR*.