Selection Bias, Label Bias, and Bias in Ground Truth

Anders Søgaard, Barbara Plank, and Dirk Hovy
Center for Language Technology
University of Copenhagen
soegaard@hum.ku.dk, {bplank|dhovy}@cst.dk

Introduction

Language technology is biased toward English newswire. In POS tagging, we get 97–98 words right out of a 100 in English newswire, but results drop to about 8 out of 10 when running the same technology on Twitter data. In dependency parsing, we are able to identify the syntactic head of 9 out of 10 words in English newswire, but only 6–7 out of 10 in tweets. Replace references to Twitter with references to a low-resource language of your choice, and the above sentence is still likely to hold true.

The reason for this bias is obviously that mainstream language technology is data-driven, based on supervised statistical learning techniques, and annotated data resources are widely available for English newswire. The situation that arises when applying off-the-shelf language technology, induced from annotated newswire corpora, to something like Twitter, is a bit like when trying to predict elections from Xbox surveys (Wang et al., 2013). Our induced models suffer from a data *selection bias*.

This is actually not the only way our data is biased. The available resources for English newswire are the result of human annotators following specific guidelines. Humans err, leading to *label bias*, but more importantly, annotation guidelines typically make debatable linguistic choices. Linguistics is not an exact science, and we call the influence of annotation guidelines *bias in ground truth*.

In the tutorial, we present various case studies for each kind of bias, and show several methods that can be used to deal with bias. This results in improved performance of NLP systems.

Selection Bias

The situation that arises when applying off-the-shelf language technology, induced from annotated newswire corpora, to something like Twitter, is, as mentioned, a bit like when trying to predict elections from Xbox surveys. In the case of elections, however, we can correct most of the selection bias by post-stratification or instance weighting (Wang et al., 2013). In language technology, the bias correction problem is harder.

In the case of elections, you have a single output variable and various demographic observed variables. All values taken by discrete variables at test time can be assumed to have been observed, and all values observed at training time can be assumed to be seen at test time. In language technology, we typically have several features only seen in training data and several features only seen in test data.

The latter observation has led to interest in bridging unseen words to known ones (Blitzer et al., 2006; Turian et al., 2010), while the former has led to the development of learning algorithms that prevent feature swamping (Sutton et al., 2006), i.e., that very predictive features prevents weights associated with less predictive, correlated features from being updated. Note, however, that post-stratification (Smith, 1988) may prevent feature swamping, and that predictive approaches to bias correction (Royall, 1988) may solve both problems. Instance weighting (Shimodaira, 2000), which is a generalization of post-stratificiation, has received some interest in language technology (Jiang and Zhai, 2007; Foster et al., 2011), but most work on domain adaptation in language technology has focused on predictive approaches, i.e., semi-supervised learning (Reichart and Rappoport, 2007; Sagae and Tsujii, 2007; McClosky et al., 2010; Chen et al., 2011).

Selection bias introduces a bias in $P(X)$. Note that, in theory, this should not hurt discriminative algorithms trying to estimate $P(Y|X)$, without estimating $P(X)$, but in practice it still does. The inductive bias of our algorithms and the size our samples make our models sensitive to selection bias (Zadrozny, 2004). Predictive approaches try to correct this bias by adding more (pseudo-labeled) data to the training sample, while post-stratification and instance weighting reweigh the data to make $P(X)$ similar to the distribution observed in the population. As mentioned, this will never solve the problem with unseen features, since you cannot up-weigh a null feature.

Semi-supervised learning can correct modest selection bias, but if the domain gap is too wide, our initial predictions in the target domain will be poor, and semi-supervised learning is likely to increase bias rather than decrease it. However, recent work has shown that semi-supervised learning can be combined with distant supervision and correct bias in cases where semi-supervised learning algorithms typically fail (Plank et al., 2014).

In the tutorial we illustrate these different approaches to selection bias correction, with discriminative learning of POS taggers for English Twitter as our running example.

Label Bias

In most annotation projects, there is an initial stage, where the project managers compare annotators' performance, compute agreement scores, select reliable annotators, adjudicate, and elaborate on annotation guidelines, if necessary. Such procedures are considered necessary to correct for the individual biases of the annotators (*label bias*). However, this is typically only for the first batches of data, and it is well-known that even some of the most widely used annotated corpora (such as the Penn Treebank) contain many errors (Dickinson and Meurers, 2003) in the form of inconsistent annotations of the same n-grams.

Obviously, using non-expert annotators, e.g., through crowd-sourcing platforms, increase the label bias considerably. One way to reduce this bias involves collecting several annotations for each datapoint and averaging over them, which is often feasible because of the low cost of non-expert annotation. This is called majority voting and is analogous to using ensembles of models to obtain more robust systems.

In the tutorial we discuss alternatives to averaging over annotators, incl., using EM to estimate annotator confidence (Hovy et al., 2013), and joint learning of annotator competence and model parameters (Raykar and Yu, 2012).

Bias in Ground Truth

In annotation projects, we use inter-annotator agreement measures and annotation guidelines to ensure consistent annotations. However, annotation guidelines often make linguistically debatable and even somewhat arbitrary decisions, and inter-annotator agreement is often less than perfect. Some annotators, for example, may annotate *social*in *social media* as a noun, others may annotate it as an adjective. In this part of the tutorial, we discuss how to correct for the bias introduced by annotation guidelines. For both label bias and bias in ground truth, we, again, use POS tagging for English Twitter as our running example.

Evaluation

Once we accept our data is biased in different ways, we need to reconsider model evaluation. If our data was selected in a biased way, say from a few editions of the Wall Street Journal, does significance over data points make much sense? If our annotators have individual biases, can we no longer evaluate our models on the data of one or two annotators? If the annotation guidelines introduce biases in ground truth, can we somehow correct for that? In practice we typically do not have hundreds of datasets annotated by different annotators using different annotation guidelines, but in the tutorial we present various ways of, nevertheless, correcting for some of these biases.

Acknowledgements

This research is funded by the ERC Starting Grant LOWLANDS No. 313695.

References

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

Minmin Chen, Killiang Weinberger, and John Blitzer. 2011. Co-training for domain adaptation. In *NIPS*.

Markus Dickinson and Detmar Meurers. 2003. Detecting errors in part-of-speech annotation. In *EACL*.

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*.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *NAACL*.

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

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *NAACL-HLT*.

Barbara Plank, Dirk Hovy, Ryan McDonald, and Anders Søgaard. 2014. Adapting taggers to Twitter with not-so-distant supervision. In *COLING*.

Vikas C. Raykar and Shipeng Yu. 2012. Eliminating Spammers and Ranking Annotators for Crowdsourced Labeling Tasks. *Journal of Machine Learning Research*, 13:491–518.

Roi Reichart and Ari Rappoport. 2007. Self-training for enhancement and domain adaptation of statistical parsers trained on small datasets. In *ACL*.

R Royall. 1988. The prediction approach to sampling theory. In Rao Krishnaiah, editor, *Handbook of Statistics*. North-Holland.

Kenji Sagae and Jun'ichi Tsujii. 2007. Dependency parsing and domain adaptation with LR models and parser ensembles. In *EMNLP-CoNLL*.

Hidetoshi Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244.

T Smith. 1988. Post-stratification. *The Statistician*, 40.

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

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *ACL*.

Wei Wang, David Rotschild, Sharad Goel, and Andrew Gelman. 2013. Forecasting elections with non-representative polls. *Forthcoming in International Journal of Forecasting*.

Bianca Zadrozny. 2004. Learning and evaluating classifiers under sample selection bias. In *ICML*.