Reducing Label Cost by Combining Feature Labels and Crowdsourcing

Jay Pujara JAY@CS.UMD.EDU
Dept. of Computer Science, University of Maryland, College Park, MD, USA 20742

Ben London BLONDON@CS.UMD.EDU
Dept. of Computer Science, University of Maryland, College Park, MD, USA 20742

Lise Getoor GETOOR@CS.UMD.EDU
Dept. of Computer Science, University of Maryland, College Park, MD, USA 20742

Abstract

Decreasing technology costs, increasing computational power and ubiquitous network connectivity are contributing to an unprecedented increase in the amount of publicly available data. Yet this surge of data has not been accompanied by a complementary increase in annotation. This lack of annotated data complicates data mining tasks in which supervised learning is preferred or required. In response, researchers have proposed many approaches to cheaply construct training sets. One approach, referred to as *feature labels* (McCallum & Nigam, 1999), chooses features that strongly correlate with the label space and uses instances containing those features as labeled data for training a classifier. These high precision examples help bootstrap the learning process. Another technique, *crowdsourcing*, exploits our ever-increasing connectivity to request annotation from a broader community (who may or may not be domain experts), thereby refining and expanding the labeled data. Combining these techniques provides a means to obtain supervision from large, unlabeled data sources. In this paper, we investigate using active learning to combine these approaches in a unified framework which we call *active bootstrapping*. We show that this technique produces more reliable labels than either approach individually, resulting in a better classifier at minimal cost. We demonstrate the efficacy of our approach through a sentiment analysis task on data collected from the Twitter microblog service.

1. Introduction

A longstanding problem in supervised learning is finding labeled data. Data is produced in high volumes from sources as varied as sensor networks to mobile phone users. Each dataset can be used for many possible applications from intrusion detection to sentiment analysis. Even if labor is expended to meticulously label data, the training set may not adequately represent the distribution of test instances. For all these reasons, methods to produce training data cheaply are an important component of machine learning research. Many researchers have considered the problem of scarce training data. Approaches can be broadly divided between those that find cheaper ways of acquiring training labels and those that design algorithms that benefit from unlabeled data, with a large body of research that combines both approaches.

1.1. Acquiring Labels Cheaply

Data annotation can expensive and time-consuming, requiring hours of manual inspection by a domain expert. To reduce this overhead, researchers have developed clever strategies for acquiring labeled data at minimal cost. One such strategy is to employ a domain-specific heuristic to produce labels. A common heuristic, feature labels (McCallum & Nigam, 1999), chooses a set of features that are strongly associated with the label, and labels instances containing these features accordingly. For instance, in the con-

Appearing in *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, 2011.

text of sentiment analysis, the keyword "overjoyed" is strongly correlated with the class Happy; as such, all instances containing the keyword "overjoyed" could be labeled as happy with high confidence in the accuracy.

Another recent innovation is crowdsourcing, which brings the labeling task to a broader community of willing, motivated participants. This is often accomplished by rewarding volunteers for their service, either monetarily or by designing a game around the task at hand. Not only is this incredibly cost-effective, it is also highly parallelizable; given the multitudes of potential participants all over the world (connected via the internet), annotation can be orders of magnitude faster than would be possible with just a handful of domain experts.

An interesting distinction between these approaches is the precision and recall of the acquired labels. Generally, features are chosen that have high precision to strongly correlate with the label, but often these features have low recall, applying to a small fraction of instances. On the other hand, crowdsourcing labels can be acquired for all instances, or arbitrary instances, allowing high recall, but with no guarantee on the reliability of the labels, the precision can be low.

1.2. Leveraging Unlabeled Data

There are essentially two approaches for leveraging unlabeled data in supervised learning: incorporating the unlabeled examples directly into the model (i.e. semi-supervised learning) or actively acquiring more labels. For the latter, we explore two popular strategies: *bootstrapping* and *active learning*.

Bootstrapping is an iterative process of training and evaluation. First, a highly selective, high precision training set is used to train a model. This model is then used to predict the labels of the unlabeled set. The intermediate predictions with the highest confidence are then used to supplement the existing training set in the next iteration. While the theoretical underpinnings of bootstrapping are not well-explored (Daumé III, 2007), a possible advantage is that the training set is augmented with the most polarized instances, allowing a classifier to use a more robust set of features.

In active learning, the learner is able to influence the distribution of training examples. Starting with a completely (or partially) unlabeled instance space, the learner iteratively requests the labels of a chosen sequence of examples. Intuitively, this allows the learner to focus on the examples it finds most ambiguous. Typically, the most uncertain instances are those that lie close to the decision boundary. Querying for these labels helps to better define the optimal decision boundary, resulting in a better model. Furthermore, by focusing less on the obvious examples (those further from the boundary), it reduces the sample complexity.

1.3. Combining Approaches

In practice, it is not uncommon to explore cheap, effective annotation strategies, while also leveraging unlabeled data. In linguistics tasks, such as semantic analysis, feature labels are often combined with bootstrapping. A set of keywords strongly associated with a type of document are defined, and these labels are used as the seed training set for a classifier trained through bootstrapping (McCallum & Nigam, 1999). Active learning and crowdsourcing have a similar synergy, where uncertain predictions on unlabeled data are converted to queries in crowdsourcing, which are labeled by participants on the Internet.

1.4. Drawbacks

One should note that the aforementioned techniques are not without some risk. For bootstrapping and feature labels, the choice of high precision examples often results in high inductive bias and poor generalization. As the number of bootstrapping iterations increases, the potential for overfitting to a small portion of the instance space also rises. Both methods are based on some human intuition about the distinguishing features, and while human reasoning has very high precision, it has significantly lower recall. Other active approaches in this setting have used unsupervised approaches to suggest labels(Liu et al., 2004). Our hypothesis is that active learning via crowdsourcing will improve the recall by introducing labeled data that bootstrapping and feature labels failed to produce.

In crowdsourcing, the primary concern is the quality of the acquired data. Participants may not be familiar with the data or rigorously trained in the labeling procedure, resulting in noisy labels. In a realistic setting, the noise increases as the examples approach the decision boundary, which is precisely the region of the instance space active learning explores. As such, we require that an active learning algorithm be robust to moderate levels of *random classification noise* (RCN). Recent theoretical results (Castro & Nowak, 2006) (Castro & Nowak, 2007) prove that it is indeed possible to learn an ϵ-optimal classifier in the presence of unbounded RCN, with an exponential dependence on the noise margin. Further results (Balcan et al., 2006) have shown that an exponential increase in sample complexity is realizable when the noise rate is sufficiently low or a high constant – though no improvement can be made under certain high-noise conditions (Kääriäinen, 2006).

2. Active Bootstrapping

Let $\mathcal{D}$ denote a distribution over an instance space $\mathcal{X} \subseteq \mathbb{R}^d$. We assume the existence of a deterministic mapping $c : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{Y}$ is a finite set of labels. In the context of sentiment analysis, $\mathcal{Y} = \{1, -1\}$. The goal is to train a classifier $h : \mathcal{X} \to \mathcal{Y}$ that minimizes the expected error $\Pr_{\mathbf{x} \sim \mathcal{D}}[c(\mathbf{x}) \neq h(\mathbf{x})]$.

Algorithm 1 illustrates our proposed technique, which we refer to as *active bootstrapping.* We are given a training set $\mathcal{U} \subseteq \mathbb{R}^d$, sampled independently and identically according to $\mathcal{D}$, as well as a heuristic $\mathcal{F} : \mathcal{X} \to \mathcal{Y}$ mapping certain features to labels. More precisely, if $\mathbf{x} \in \mathcal{X}$ contains a certain feature that is strongly correlated with a $y \in \mathcal{Y}$, then $\mathcal{F}(\mathbf{x}) = y$; otherwise, $\mathcal{F}(\mathbf{x})$ outputs a null value. We thus begin by invoking $\mathcal{F}(\mathbf{x})$, for all $\mathbf{x} \in \mathcal{U}$. Let S denote the set of instances for which $\mathcal{F}(\mathbf{x})$ returned a value. We then update $\mathcal{U}$ by removing all instances also found in S, leaving only unlabeled examples in $\mathcal{U}$.

The algorithm then iterates over the following steps. A classifier h is trained on S. Consequently, h is used to predict labels for the remaining unlabeled examples in $\mathcal{U}$. From this result, the top-k most confident predictions from each class are added to S. Similarly, the top-(αk) more uncertain predictions are crowdsourced to obtain (possibly noisy) labels. $\mathcal{U}$ is then updated by removing all instances from S. The algorithm terminates when the maximum number of iterations, MaxIters, is reached.

2.1. Our Contribution

As mentioned earlier, label acquisition strategies have differing strengths, particularly in terms of the precision and recall of the labels. Feature labels are assumed to be very high precision indicators of class membership, but suffer from poor recall; crowdsourcing can potentially expand the recall, but suffers from RCN, which hampers its precision. Our hypothesis is that by combining both strategies, we can balance precision and recall. Selecting the top-(αk) uncertain predictions (across the entire sample) explores the instance space, thereby improving recall; whereas selecting the top-k most confident predictions exploits the obvious examples to improve precision. Active bootstrapping performs both exploration and exploitation simultaneously, thus generating higher quality training data without the costs associated with domain-expert annotation.

Algorithm 1 Active-Bootstrapping Algorithm: Augments training data with active learning and bootstrapping

```
Require: Unlabeled data, $\mathcal{U} \subseteq \mathbb{R}^d$
Require: Heuristic mapping $\mathcal{F} : \mathcal{X} \to \mathcal{Y}$
Require: Constants $k$, $\alpha$ and MaxIters
  $S \leftarrow$ instances of $\mathcal{U}$ with features from $\mathcal{F}$ and their labels
  $\mathcal{U} \leftarrow \mathcal{U} - S$
  for $i = 0$ to MaxIters do
    Train a classifier, $h$ on $S$
    Predict labels on $\mathcal{U}$ using $h$
    $S \leftarrow S \cup \{$top-$k$ positive instances$\}$
    $S \leftarrow S \cup \{$top-$k$ negative instances$\}$
    $S \leftarrow S \cup \{$crowdsourced responses of top-$(\alpha k)$ uncertain predictions$\}$
    $\mathcal{U} \leftarrow \mathcal{U} - S$
  end for
```

3. Evaluation

The following section outlines the experimental procedure used to validate our hypothesis; namely, that a classifier trained with labels generated from both features and crowdsourced labels will outperform a classifier using either strategy alone. This was accomplished by training a classifier to detect happy and sad emotional content in the Twitter microblogging network, and using *emoticons* to generate labels from features and getting crowdsourced judgments from Amazon's Mechanical Turk system.

3.1. Dataset

We acquired Twitter data from a corpus included in the Stanford Large Network Dataset Collection as reported in (Yang & Leskovec, 2011). This dataset is believed to contain approximately 20% of all publicly visible tweets (476M tweets) from a period spanning June-December 2009. Since our goal was to predict emotional content and many tweets are objective statements, we filtered the data using a set of emotional indicators, specifically emoticons. Filtering for emoticons resulted in 41M tweets, from which we sampled 1% to obtain a more tractable dataset. Each instance in this set consisted of a user ID, a time-stamp and a tweet. We balanced the number of positive and negative instances, and applied the normalization steps described in subsection 3.2 to yield a total of 77,920 instances, (39033 negative and 38887 positive) used for unlabeled data in our experiments. For evaluation, we manually labeled a set of 500 tweets with the same processing described earlier.

3.2. Normalization

Since Twitter is used for a variety of purposes, it was first necessary to separate messages that communicated a personal state of being from those that were likely focused on just sharing information. To this end, we removed those that contained URLs, under the assumption that they were less likely to be emotional. We also removed tweets that were shorter than 40 characters total. To reduce the lexicon even further, we removed all punctuation, emoticons, username mentions, hashtags, HTML escape sequences, and non-ASCII characters. We then lowercased all text to provide a uniform view. This data was used for classification as well as for crowdsourcing.

Learning a lexicon from Twitter data can be difficult. The tweet length limitation has given rise to a whole new language of acronyms, abbreviations and phonetic abbreviations. Informal context makes users less attentive to proper spelling and "morphological emphasis" is commonly used — e.g. "yay" becomes "yaaaay", or "yes" becomes "yessss". To remove infrequent terms and misspellings, we removed all terms with frequency below the mean (the two lower quartiles). Since it is rare for the same character to appear more than twice consecutively in English words, we replaced three or more occurrences of a character with a single character.

3.3. Feature Labels

We assume that certain lexical features — in this case, emoticons — serve as a proxy for emotional content in tweets, and can thus be used in lieu of manually-assigned labels. By generalizing slightly from the emoticons found on Wikipedia (Wikipedia, 2010), we produced a regular expression mapping to a comprehensive list of emoticons indicating happiness and sadness. Because we have not manually confirmed the labels, we refer to them as feature labels.

3.4. Crowdsourcing

To acquire crowdsourced labels, we created a "Human Intelligence Task" on Amazon's Mechanical Turk. Users on the service received compensation between five and ten cents for labeling a series of ten tweets with labels Happy, Sad or Neither. One of the ten tweets shown to the users was a tweet that had already been labeled by the authors; those responses which had an incorrect label for this tweet were discarded and the user did not receive payment for them. This acted as a simple quality control for filtering out bad data from disinterested or exploitive users. Furthermore, we required that each crowdsourced tweet used in training received the same label by a minimum of two users, thereby providing more certainty in the acquired label.

3.5. Experiments

We compared an active bootstrapping approach with baseline results from bootstrapping, with seed sets generated using feature labels or crowdsourced labels. When bootstrapping with feature labels, we used initial seed sets of 1,000, 2,000, and 10,000 feature-labeled instances. When using crowdsourced labels, we requested a total of 2,000 labels on 1,000 instances that were randomly chosen from the training set. Approximately 1,600 labels were acquired through crowdsourcing, and after employing the validation steps described in subsection 3.4, a total of 670 labels remained. At each iteration, the training set was augmented with a number of instances equal to 10% of the seed set size. These instances were drawn from the most confident predicted labels on unlabeled data, sampled equally from the positive and negative classes. Bootstrapping was run for 7 iterations.

Evaluating active bootstrapping, we used the same 1,000 feature-labeled instances from the baseline classifier as a seed training set. Following algorithm 1, we augmented our training set with 100 instances from predicted labels on unlabeled data. We also augmented the training set with approximately 100 instances that were queried through crowdsourcing. These queries were generated from instances with uncertain predictions in unlabeled data. Each instance was labeled by two users, yielding 200 total labels. Since we employed a data quality test, the number of crowdsourced instances actually added could vary each iteration.

3.6. Results

Through the experiments described, we were able to demonstrate that active bootstrapping provides an advantage over bootstrapping, using the same potentially noisy or biased seed sets. Table 1 shows the test error on our manually labeled evaluation set for the different approaches. Active bootstrapping achieves the lowest error in this experiment, despite starting out with a high initial error. Conventional bootstrapping using the same training set achieves an error of .390 vs .297 for active bootstrapping. Active bootstrapping adds as many instances each bootstrapping iteration as the

Method	Err, I=0	Err, I=7
Feature Labels, s=1000	.332	.390
Feature Labels, s=2000	.302	.343
Feature Labels, s=10000	.295	.346
Crowdsourcing, s=670	.374	.456
Active Bootstrapping, s=1000	.332	.297

Table 1. Error on a Twitter sentiment analysis task with just a seed set and after 7 iterations of the bootstrapping algorithm. The error is reported on a manually labeled test set using different bootstrapping methods. In all cases, bootstrapping causes the test error to increase. Active bootstrapping maintains constant performance.

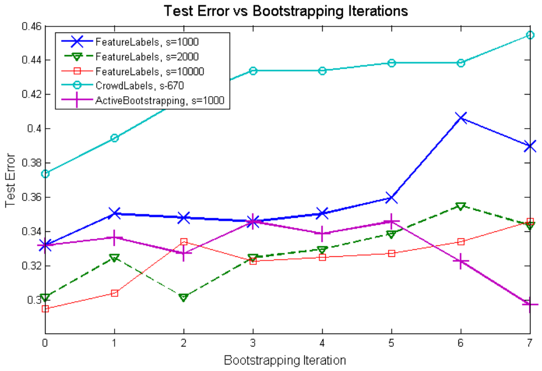

Figure 1. Graph of test error over bootstrapping iterations. Without active labeling, bootstrapping error increases. By using crowdsourced labels, the error remains in a tighter range

conventional bootstrapping classifier with a seed set of 2,000. Despite starting with a larger seed set, this conventional bootstrapping approach also underperforms active bootstrapping. Even with a very large seed set, conventional bootstrapping results in increased error. Using a more diverse seed set from crowdsourced labels does not improve the results of bootstrapping either, but getting crowdsourced labels actively in our approach yields good results. The progress of active bootstrapping and conventional bootstrapping over iterations of the bootstrapping algorithm is shown in Figure 1. While conventional bootstrapping increases test error over iterations, active bootstrapping shows decrease test error.

4. Conclusion

We have presented a framework for acquiring labeled data from inexpensive sources, as well as leveraging unlabeled data. This method balances the precision and recall advantages of both techniques, thereby improving the overall quality of the training data at minimal expense. We have tested our hypothesis using a real-world data set, showing a marked improvement over baseline methods.

For future work, we would like to provide a more rigorous theoretical justification for the benefits of active bootstrapping. It would be interesting to construct a simple abstraction that illustrates analytically how bootstrapping with feature labels compliments active learning with a potentially noisy oracle. We would also like to experiment with iteratively adaptive feature labels, i.e. updating the feature labels based on the results of each crowdsourced query.

References

Balcan, Maria-Florina, Beygelzimer, Alina, and Langford, John. Agnostic active learning. In *In ICML*, pp. 65–72. ICML, 2006.

Castro, Rui M. and Nowak, Robert. Upper and lower bounds for active learning. In *Proceedings of the 44th Annual Allerton Conference on Communication, Control and Computing*. Allerton House, University of Illinois, 2006.

Castro, Rui M. and Nowak, Robert D. Minimax bounds for active learning. In *In COLT*, pp. 151–156. Verlag, 2007.

Daumé III, Hal. Bootstrapping, 2007. URL http://nlpers.blogspot.com/2007/09/bootstrapping.html.

Kääriäinen, Matti. Active learning in the non-realizable case. In *NIPS Workshop on Foundations of Active Learning*, 2006.

Liu, Bing, Li, Xiaoli, Lee, Wee Sun, and Yu, Philip S. Text classification by labeling words. In *In AAAI-2004*, pp. 425–430, 2004.

McCallum, Andrew and Nigam, Kamal. Text classification by bootstrapping with keywords, em and shrinkage. In *ACL99 - Workshop for Unsupervised Learning in Natural Language Processing*, pp. 52–58, 1999.

Wikipedia. List of emoticons, 2010. URL http://en.wikipedia.org/wiki/List_of_emoticons.

Yang, Jaewon and Leskovec, Jure. Patterns of temporal variation in online media. In *ACM International Conference on Web Search and Data Minig (WSDM)*. Stanford InfoLab, 2011. URL http://ilpubs.stanford.edu:8090/984/.