Semi-supervised Learning for Sentiment Classification

Bishan Yang
Department of Computer Science
Cornell University
Ithaca, NY 14850
bishan@cs.cornell.edu

Abstract

With the growing need of identifying opinions and sentiments automatically from online text data, sentiment classification tasks have received considerable attention recently. One can treat sentiment classification as a text classification problem, however, it is very time-consuming and somewhat impractical to acquire enough labeled data to train a good sentiment classifier. This paper investigates a semi-supervised learning method for sentiment classification which can take advantage of large amounts of unlabeled data. Specifically, we learn some structural information from both labeled and unlabeled data to form a good feature mapping for the sentiment classification tasks, and then bootstrap the learner to improve the classification performance. We present an empirical study on two different sentiment classification tasks which indicates the proposed method can make good used of unlabeled data and improve the classification performance.

1 Introduction

With the growing ability of the online resources such as Amazon, Youtube and online forums, a tremendous need arises as people want to know about what other people think and want to seek out opinions in order to make informative decisions. Sentiment classification, which deals with identifying opinions from online text, has received considerable attention recently. The task can be viewed as a text classification problem. Given a instance of text, the goal is to classify it as subjective (opinionated) or objective (non-opinionated). If the text is opinionated, we can further determine whether the opinion is positive or negative.

Sentiment classification is a very challenging task. On one hand, traditional text classification techniques usually do not work well on this task, since they tend to view frequent-occurring words as good indicators of the class labels, while in opinionated text, sentiment words are usually ambiguous and infrequent. On the other hand, acquiring human-labeled data for sentiment classification is very difficult. Opinions are hidden in a huge amount of online resources like forums and blogs. Manual annotation is very expensive and time-consuming.

The goal of this project is to design a semi-supervised learning method for sentiment classification, which can take advantage of unlabeled data and improve the classification performance. The approach consists of two components. One is to learn a good feature mapping for the target sentiment classification task using both labeled and unlabeled data. The other component is a bootstrapping mechanism which can enhance the learner with unlabeled data. In this way, we both amplify the features and the size of the training data. We empirically evaluate the effectiveness of the approach on two different sentiment classification tasks. The results demonstrate that our method can make good use of unlabeled data and improve the classification results.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 briefly describes sentiment classification problems and supervised classification models. Section 4 introduces our semi-supervised learning approach for sentiment classification. Section 5 shows experimental results on two real-word data sets compared with other baseline methods. Section 6 presents conclusions and future work.

2 Related Work

Most prior work in sentiment classification takes a supervised learning approach to learn sentiment classifiers. [1] studied several machine learning methods, e.g., Naive Bayesian and Support Vector Machine (SVM) for classifying movie reviews into positive and negative. [2] built a Naive Bayesian classifier to classify sentences into subjective and objective. These studies dose not take into account the limited availability of labeled data.

Semi-supervised learning approach is an approach to reduce the need for labeled data by taking advantage of unlabeled data. In general, there are two kinds of semi-supervised learning approaches. One is to bootstrap class labels using techniques like self-training[5], Expectation Maximization (EM) [4] and co-training[6]. Self-training trains a classifier and use it to classify unlabeled data, and then add the most confident data to the training data and repeat the process. EM approach can be viewed as a special case of ”soft” self-training. It assumes the data is generated according to some known parametric models, and then iteratively estimates the expectation of hidden class variables and update the model parameters. Co-training splits features into two sets and trains two classifiers. Each classifier picks its most confident data and retrains with the additional labeled data provided by each other. One can imagine that classification mistakes can reinforce itself by using this kind of methods[3]. Another category is structural learning methods which learn good functional structures using unlabeled data. [7] proposed a graph-based method which constructs a graph with labeled and unlabeled examples as nodes and their similarity relationships as edges. It makes the assumption of label smoothness over the graph. [8] proposed a framework to learn predictive structures on hypothesis spaces using unlabeled data, and then use these structures to enhance learning. The performance of such methods is influenced by how much the structure characterize the underlying hypothesis.

Our approach differs from the earlier work. It can benefit from the advantages of both kinds of semi-supervised learning approaches. First, it learns structural information from both labeled and unlabeled data to form a good feature mapping for the classification task. Secondly, it performs bootstrapping with two feature views to enhance learning. The benefits of our approach are confirmed by experiments on two different sentiment classification problems.

3 Sentiment Classification Models

Sentiment classification is the task of automatically classifying text (can be documents or sentences) into a predefined sentiment classes (e.g. positive or negative, subjective or objective). An instance of text is typically represented as a bag-of-words feature vector $x \in R^n$, where n is the size of a pre-specified vocabulary. The value of each entry in the vector usually specify the presence of the corresponding word. Given training examples $X = \{x_1, ..., x_m\}$, we can build a binary classifier $f : X- > Y, Y = \{0, 1\}$ and use it to predict labels for an unseen instance x by computing $f(x)$.

There are many machine learning models which can be used for binary classification. Generally they fall into two categories: discriminative models and generative models. Discriminative models, such as Logistic Regression and SVMs, directly estimate posterior probabilities $p(y|x)$ without modeling the underlying probability distributions over the data. Generative models, such as Naive Bayes and Mixtures of Gaussians make assumptions about the distribution over data, and estimate class-conditional probability $p(x|y)$ and prior probability $p(x)$. In this paper, we consider discriminative models for sentiment classification, since in text classification, the modeling assumptions of generative models may be too strong, while discriminative models are more reflexible.

Specifically, we use the logistic regression model. Given l labeled data $\{x_i, y_i\}_{i=1}^{l}$, $y_i \in \{0, 1\}$, the predictor is in the form of $p(y = 1|x) = 1/(1 + exp(-w^T x))$, where $w \in R^V$ is a weight vector. w can be learned by maximizing log likelihood, which equals to solving an optimization problem of the form

$$w^* = \arg\min_w \sum_{i=1}^{l} -\log p(y_i|x_i, w) + \lambda||w||^2$$

, where λ is a regularization parameter, and the term $\lambda||w||^2$ is introduced to penalize large value of w in order to reduce overfitting.

4 Semi-supervised Sentiment Classification

In this section, we will describe our semi-supervised learning approach for sentiment classification. We will first introduce a structural learning method to learn a good feature mapping by using both labeled and unlabeled data, and then introduce a bootstrapping mechanism which uses the learned mappings and the original input features to enhance the classifier.

4.1 Structural Learning with Labeled and Unlabeled Data

For the sentiment classification tasks, features which are highly related to the sentiment classes (e.g. positive or negative) are usually good indicators of the predicted labels. We refer to such features as informative features for the sentiment classifier. Intuitively, if we strengthen the informative features, the model will have more discriminative capability over data. The goal of structural learning is to learn a good feature mapping which can map data to some important dimensions where the major information of these informative features are captured.

Although we don't have any label information about the unlabeled data, we can use its input features for learning. The way to do that is to create some auxiliary learning tasks, each of which is a binary classification problem of the form ”Whether this instance contains a informative feature?” We can define the informative features based on our prior knowledge or obtain them by calculating the mutual information using the labeled data. In this way, we can get ”labels” for the unlabeled data and use it as training data. For each auxiliary task, we can learn a predictive function for a specific informative feature. As the informative features are highly related to the sentiment classes, we can view the predictive functions obtained from the auxiliary tasks as a sample of predictors in the hypothesis space of our target classification problem. Good predictors are closed to each other in the predictor space. If we can find some common structure shared by all the predictors, it will be helpful for training the target classifier.

To formularize it, given input data $X = \{x^1, ..., x^{l+u}\}$ (suppose there are l labeled samples and u unlabeled samples) and K informative features, create K different learning tasks. Each task k ($1 \le k \le K$) is to predict the presence of an informative feature t_k by using a set of training data $(x^i_{-t_k}, y^i_{t_k})$, where $x^i_{-t_k}$ means the feature representation without feature t_k (let the value on t_k be 0), $y^i_{t_k} = 1$ if x^i contains feature t_k, otherwise $y^i_{t_k} = 0$. In other words, we can create K binary classification tasks, each producing a predictor

$$f_k(x_{-t_k}) = sgn(w_k \cdot x_{-t_k}), k = 1, ..., K$$

(it can be obtained by calculating $p(y = 1|x)/p(y = 0|x)$ using the logistic regression definition). The weight vector w_k encodes the covariance of the informative features with the other features. For informative feature t_k, if the weight associated with feature z is positive, then z is positively correlated with t_k. Also, if feature z_1 and z_2 have similar weight, then they have similar correlation with feature t_k. w_k can be seen as a linear projection of the original feature space onto R.

Denote the common structure shared by the K tasks as Θ. We consider Θx as a low-dimensional parameterized feature mapping (of size $p \times m$ matrix where $p \le K$ is a parameter specified by users) where the importance of the informative features are captured. The linear predictor for each task can be written in the form of

$$f(x) = sgn(w^T x + v^T \Theta x)$$

, where w and v are weighted vectors learned from each task.

Followed by ASO [8], we can learn parameters $\{w_i, v_i\}_{i=1}^{k}, \Theta$ by a alternating optimization procedure. For each task f_k, denote $L(f_k(x, w, v, \Theta))$ as the loss function. For logistic regression classifier, $L(f_k(x, w, v, \Theta)) = -logP(y|x, w, v, \Theta)$.

We minimize the empirical risk minimization.

$$[\{w_k, v_k\}_{k=1}^K, \Theta] = \arg \min_{\{w_k,v_k\}_{k=1}^K,\Theta} \sum_{k=1}^{K} (\frac{1}{l+u} \sum_{i=1}^{l+u} L(f_k(w_k, v_k, x^i, \Theta)) + \lambda_k ||w_k||^2), s.t. \Theta\Theta^T = I$$

($\Theta\Theta^T = I$ is for regularization purpose)

Introduce a new variable $z_k = w_k + \Theta^T v_k$, then the optimization objective becomes

$$[\{z_k, v_k\}_{k=1}^K, \Theta] = \arg \min_{\{z_k,v_k\}_{k=1}^K,\Theta} \sum_{k=1}^{K} (\frac{1}{l+u} \sum_{i=1}^{l+u} L(z_k^T x^i, y^i) + \lambda_k ||z_k - \Theta^T v_k||^2), s.t. \Theta\Theta^T = I$$

Then Θ, z, v can be learned alternatively using the following procedure.

1. Fix Θ, v, and find the optimum z.
2. Fix z, and find the optimum Θ, v.
3. Repeat this process until converge.

For step 1,since $L(z_k^T x^i, y^i) = -\log P(y^i|x^i; z_k)$ is convex, we can use gradient descent to find the optimal value of z_k.

For step 2, it equals to minimize $\sum_{k=1}^K \lambda_k ||z_k - \Theta^T v_k||^2, s.t. \Theta\Theta^T = I$. For a fixed Θ, we can obtain an optimal v_k by setting the gradient to be 0. Since $||z_k - \Theta^T v_k||^2 = z_k^T z_k - 2z_k^T \Theta^T v_k + v_k^T v_k$, we have $\nabla_{v_k} = 2v_k - 2\Theta z_k = 0$, so

$$v_k = \Theta z_k$$

. Replace v_k into the objective function, we can get $\lambda_k ||z_k - \Theta^T v_k||^2 = -\lambda_k ||z_k \Theta^T||^2 + \lambda_k z_k^T z_k$. So the optimization problem becomes maximizing $\sum_{k=1}^K \lambda_k ||z_k \Theta||^2 s.t. \Theta\Theta^T = I$. Let $Z = [\sqrt{\lambda_1} z_1, ..., \sqrt{\lambda_K} z_K]$ be a $m \times K$ matrix, then it becomes

$$\Theta^* = \arg \max_{\Theta} ||\Theta Z||^2, s.t. \Theta\Theta^T = I$$

. Let $Z = UDV^T$ be the singular value decomposition of Z where the diagonal elements of D are arranged in decreasing order, then $||\Theta Z||^2 = tr(\Theta Z Z^T \Theta^T) = tr(\Theta U D^2 U^T \Theta^T) = tr(D U^T \Theta^T \Theta U D) = \sum_i D_{ii}^2 ||\Theta U_i||^2$. Note that we have $0 \leq ||\Theta U_i||^2 \leq 1$ and D has descending diagonal entries, so the solution are $||\Theta U_i||^2 = 1$ for the eigenvectors U_i which corresponds to the first p diagonal entries in D, and we can get $\Theta = U_{[1:p,:]}^T$, whose rows are the first p rows of U^T.

Now we can learn the parameters $\{w_k, v_k\}_{k=1}^K, \Theta$ alternatively using the above procedure. However, it is usually sufficient to use the Θ obtained from the SVD in the first iteration. The small perturbation of Θ in the rest iterations doesn't affect much on the performance. [8]

4.2 Bootstrapping with Two Feature Views

An common technique for semi-supervised learning is to enlarge training set by adding the most confident unlabeled points together with their predicted labels. However, simply using the prediction results of the current classifier may reinforce the classification mistakes. Co-training [6] is an effective technique to reduce mistake propagation. It splits features into two sets, each of which is sufficient to train a good classifier, and the two sets are conditionally independent given the class. Then it trains two separate classifiers and choose the unlabeled data with the most confident prediction and help each other with these additional training data.

From Section 4.1, we can see that there is a natural feature split in our data. One set is from the original feature space x, and the other is from the learned low-dimensional feature space Θx. Each set is sufficiently for learning the sentiment classifiers. However, they are not conditionally independent given the sentiment labels. To relax this strong assumption of feature independence, we introduce a new bootstrapping mechanism to incorporate unlabeled data.

Denote the two separate sets of features as $A = x$ and $B = \Theta x$, and the two classifiers as f_A and f_B. When f_A chooses an unlabeled example with high confidence, it will be automatically added for learning f_B. Intuitively, f_A should only consider choosing unlabeled example x if f_A's confidence in predicting x and f_B's confidence in predicting x are both high. Similar for f_B, its chosen examples should have high confidence in the both views of f_A and f_B.

Since A and B are highly correlated, we can expect the added training data to be helpful for both f_A and f_B. So we consider choosing the same set of unlabeled examples to label for f_A and f_B. The selection criterion is based on combining prediction probability $((p_A(y|x)+p_B(y|x))/2$. Unlabeled examples with the highest combining probability will be chosen with its predicted labels.

The bootstrapping algorithm works as follows. Given a set of labeled data X_l and unlabeled data X_u, the algorithm first creates a sample pool P containing L unlabeled examples. It then iterates the following procedure. First use X_l to train two different classifiers f_A and f_B, and then make predictions on the unlabeled pool P. Rank the L unlabeled examples in decreasing order according to the score $(p_A(y|x) + p_B(y|x))/2$, where $p_A(y = 1|x) = 1/(1 + exp(-f_A(x)))$, and disregard the examples on which f_A and f_B disagree on the predicted labels. According to the prediction, select *cp* positive examples with highest score and *cn* negative examples with highest score, and add them into X_l along with their predicted labels. Finally fill the pool P with new $cn + cp$ examples which are randomly drawn from X_u.

The whole algorithm is described in Algorithm 1.

Algorithm 1 Semi-supervised Learning for Sentiment Classification

```
Input: $X_l = \{x^i, y^i\}_{i=1}^{l}$, $X_u = \{x^j\}_{j=l+1}^{l+u}$, $K$, $p$, $L$, $cp$, $cn$
Output: predictor $f$

 1: Select $K$ informative features $t_1, ..., t_K$
 2: Create $K$ binary classification tasks $f_k$ using training data $(x^i_{-t_k}, y^i_{t_k}), i = 1, ..., l + u$, where
    $y^i_{t_k} = 1$ if $x^i$ contains feature $t_k$, otherwise $y^i_{t_k} = 0$.
 3: for $k = 1$ to $K$ do
 4:   $w_k = \arg\min_{w_k}\{-\sum_{i=1}^{l+u} \log P(y^i_{t_k}|x^i_{-t_k}; w_k) + \lambda||w_k||^2\}$
 5: Let $W = [w_1, ..., w_K]$, compute $SVD(W) = UDV^T$
 6: Let $\Theta = U^T_{[1:p,:]}$
 7: Create a pool $P$ by drawing $L$ unlabeled examples randomly from $X_u$
 8: while Performance does not decrease do
 9:   Train a classifier $f_A$ on labeled data $X_l$ using feature $X$
10:   Train a classifier $f_B$ on labeled data $X_l$ using feature $\Theta X$
11:   for all $x$ in $P$ do
12:     if $f_A$ and $f_B$ agree on the predicted label of $x$ then
13:       Compute $Score(x) = (p_A(y|x) + p_B(y|x))/2$
14:   Rank all $x$ in $P$ in decreasing order according to $Score(x)$
15:   Select the first $cp$ examples in $P$ with positive labels and first $cn$ examples in $P$ with negative
      labels
16:   Replenish $P$ with $cp + cn$ examples drawing randomly from $X_u$
17:   Add the selected examples along with their predicted labels to $X_l$
18:   Train $f$ on $X_l$ with features $(X; \Theta X)$
```

5 Experiments

To evaluate the effectiveness of our approach, we experimented on two different sentiment classification tasks. One is to classify reviews as positive and negative using a corpus of Amazon product reviews[10]. There are four different types of product: Book, DVDs, Electronics and Kitchens. Each domain contains 1000 positive reviews, 1000 negative reviews, and a large number of unlabeled reviews. The other task is to classify sentences of news articles to be subjective or objective using a news data sets MPQA[11]. It consists of 535 news articles, covering different topics from a variety of countries. There are 11112 sentences in total. All sentences are manually labeled with respect

Table 1: Statistics on Evaluation Data Sets

Data set	Labeled(Positive)	Labeled(Negative)	Unlabeled	Features
Book	1,000	1,000	6,000	40,834
DVDs	1,000	1,000	34,741	83,514
Electronics	1,000	1,000	13,153	31,101
Kitchen	1,000	1,000	16,785	27,300
MPQA	6,212	4,900	–	120,146

Table 2: Accuracy on Amazon data sets(%)

Method	Book	DVD	Electronics	Kitchen
LR	73.2	74.06	77.46	79.4
NB-EM	73.13	59.73	71.33	68.4
SSL	77.26	76.26	80.86	84.26
Bi-SSL	78	76.4	81	84.73

to subjectivity. The details of these data sets can be found in Table 1. The MPQA corpus does not contain any unlabeled data, and in Table 1 the number of positive(negative) instances instances corresponds to the number of subjective(objective) instances.

5.1 Methodology

We compared four different baselines in the experiments.

- LR: The standard logistic regression classifier using only labeled data.
- NB-EM[4]: A semi-supervised learning algorithm which learns from labeled and unlabeled data using EM. It first trains a Naive Bayes classifier and labels the unlabeled documents with probabilities. It then retrains the classifier using the labels for both labeled and unlabeled data, and iterates the process to convergence.
- SSL: The structural learning algorithm discussed in Section 4.1, which creates auxiliary tasks to predict informative words with respect to the sentiment classes using both labeled and unlabeled data, and then learn a common feature mapping using the alterative optimization procedure. It is an extension of the work in [8]. (The difference is that [8] create auxiliary tasks to predict the frequent-occurring words and it doesn't consider labeled data in the auxiliary tasks.)
- Bi-SSL: The algorithm described in Algorithm 1. Since it amplifies both features and labeled data, we refer to it as Bi-SSL.

All algorithms were implemented using C++. EM was implemented according to the paper [4]. The alterative optimization procedure in Section 4.1 was implemented according to the paper [8] (SVDLIBC library[12] is used for the computation of SVD). LR was implemented using the package of OWL-QN [9] with L-2 optimization.

Table 3: Performance on MPQA data set

Method	Accuracy (%)
LR	63.24
NB-EM	73.70
SSL	69.70
Bi-SSL	71.65

Method	NB (NB-EM)	LR (Bi-SSL)
Supervised	71.56	63.24
Semi-supervised	73.70 (+2.99%)	71.65 (+13.30%)

5.2 Experimental Results with Amazon Data set

We randomly split the labeled data into a training set of 500 instances and a test set of 1,500 instances. Figure 1 show the experimental results of accuracy averaging over 10 random trials. We set the parameters as $K = 100, p = 50, L = 500, cp = cn = 5$.

Table 2 presents the results. We can see that SSL and Bi-SSL significantly outperform the supervised method LR in all domains, and Bi-SSL performs slightly better than SSL. This indicates that making use of unlabeled data by performing structural learning can significantly improve the performance of sentiment classification, and bootstrapping can contribute a bit more to the classification performance. We can also see that NB-EM performs even worse than the supervised baseline, which suggests that propagating labels using EM doesn't help learning on this data set.

5.3 Experimental Results with MPQA Dataset

For the MPQA corpus, we randomly split the whole data set into a training set of 500 instances, a test set of 2,000 instances and a unlabeled set of 8,612 instances. The setting of parameters is the same as that in Section 5.1.

Table 3 presents the results. We can see that SSL and Bi-SSL significantly outperform the supervised method LR. And Bi-SSL achieves much better performance than SSL on this data set, which implies that bootstrapping did help improve the classification performance. Notice that NB-EM is performing the best on this data set. One reason is that Naive Bayes classifier performs much better than logistic regression on the this data set. If we look at the performance improvement of semi-supervised learning over supervised learning, which shows in Table 4, we can see that Bi-SSL can obtain 13.30% improvement by using unlabeled data while NB-EM can only obtain 2.99% improvement.

6 Conclusion

This paper investigates a semi-supervised learning method for sentiment classification. It can make use of the unlabeled data to learn a good mapping which is helpful for the classification task, and it can use bootstrapping to improve the classification performance. We present an empirical study on two different sentiment classification tasks which indicates the proposed method can make good used of unlabeled data and improve the classification performance.

Acknowledgments

We thank Prof. Saxena for advice and useful comments on this work.

References

[1] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In EMNLP, 2002.

[2] J. Wiebe and E. Riloff. Creating subjective and objective sentence classifiers from unannotated texts. In CICLing-2005.

[3] X. Zhu, "Semi-Supervised Learning Literature Survey," Technical Report 1530, Univ. of Wisconsin-Madison, 2008.

[4] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. Machine Learning, Special issue on information retrieval:103 C134, 2000.

[5] Yarowsky, D. Unsupervised word sense disambiguation rivaling supervised methods. Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL), pp. 189-196, 1995

[6] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pp. 92-100, 1998.

[7] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In ICML 2003.

[8] Ando, R. K. and Zhang, T. A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. J. Mach. Learn. Res. 6, pp. 1817-1853, 2005.

[9] G. Andrew and J. Gao. Scalable training of L1-regularized log-linear models. In ICML 2007.

[10] `http://www.seas.upenn.edu/mdredze/datasets/sentiment`

[11] `http://www.cs.pitt.edu/mpqa/databaserelease/`

[12] `http://tedlab.mit.edu/~dr/SVDLIBC/`