Proposal: Investigating the Effects of Random Negative Sampling and Clustering on Positive and Unlabeled Data

Stephen Dove

October 2018

1 Introduction

Classifiers are required for many tasks when only positive and unlabeled data are available for training. These types of tasks use positive and unlabeled (PU) learning to train the classifier. Tasks requiring PU learning are prevalent across industries from text classification to estimating click traffic to identifying fraudulent charges.

This problem is difficult since previous semi-supervised techniques using an Expectation-Maximization (EM) [2] or Support Vector Machine (SVM) [1] framework fail when they are not presented with negative examples. Intuitively, we can imagine that classifying documents into two buckets (topic A and not-topic A) would be difficult if we do not know what a 'not-topic A' document looks like. We may, however, be able to reliably estimate negative examples from unlabeled examples that appear to be not positive. In PU learning, the model needs to estimate this set of examples that it believes are negative examples before we can use the aforementioned semi-supervised approaches.

Currently, there are several models that use a two step approach to solve this problem.

- **Step 1:** Split the unlabeled dataset (U) into two separate datasets of reliable negative examples (RN) and the other unlabeled data ($Q = U - RN$).
- **Step 2:** Using the positive dataset (P), RN, and Q, iteratively create a classifier for the data.

There are several techniques in literature for both step 1 and step 2 (see section 3). [3]

1.1 Research Plan

We propose to investigate two new approaches to creating the reliable negative set and to creating an accurate classifier for PU scenarios. We also propose to evaluate how important creating the reliable negative set is in PU learning.

Our first approach will be implementing a classifier to classify reliable negative examples using a clustering algorithm. If we are able to cluster the examples, we hope to find some apparent reliable negatives using some confidence score. If the proposed model is not confident in an example, we will leave it unlabeled. From here, we will send P, RN, and Q to an SVM to create the final classifier for evaluation. In the second approach, we will similarly cluster the examples, but will not refrain from labeling an example. Once all examples are labeled, we can evaluate. For our third idea, we will randomly sample from the unlabeled data set and mark them as negative examples. These random negative samples will comprise the RN dataset. Like the first approach, we will pass these groupings into an SVM for consistency.

To evaluate these models, we will use the **20 Newsgroup** [4] dataset similarly to the methods in Liu et al. [3] For our positive dataset, we will use one of the twenty newsgroup categories. The other nineteen categories will belong to the unlabeled dataset. This gives us

twenty distinct datasets. As a benchmark for our new approaches, we will recreate the Naive Bayes model for identifying reliable negatives and use an SVM to create a classifier for evaluation as in Liu et al. [3] Given that we know the true labels of each of the newsgroup examples, we can measure the accuracy that each model attains in correctly identifying the binary labels.

Timeline

- Week 1 (2018-10-29): Write a script to take create the 20 datasets using the 20 Newsgroup [4] dataset and create feature vectors for each document.
- Week 2 (2018-11-05): Implement a semi-supervised SVM classification model using positive, negative, and unlabeled examples to be used as step 2 for most experiments. Implement accuracy metrics for evaluation purposes.
- Week 3 (2018-11-12): Implement Naive Bayes + SVM and Random Negative Sampling (RNS) + SVM models to have a control model and our model for experiment 3, respectively.
- Week 4 (2018-11-19): Implement a clustering algorithm to produce clusters of unlabeled documents using an unsupervised method.
- Week 5 (2018-11-26): Adjust the above clustering algorithm to add labels to each labeled example and extract a set of reliable negatives for experiment 1 and extract labels for all unlabeled examples for experiment 2.
- Week 6 (2018-12-03): Evaluate our three experiments. This will use the evaluation metrics to compare accuracy for all four models (NB + SVM, Clustering + SVM, Clustering, and RNS + SVM).
- Week 7 (2018-12-10): Finish evaluations and prepare final report.

2 Background

There are many real world applications for machine learning that do not have negative examples, but rather they have positive and unlabeled examples. To give some perspective of these applications, we look back at the click traffic scenario mentioned in the introduction. In these datasets, we may have information on the positive data like clicks on an advertisement to gauge a user's interest, but we do not know what a user's interest would be if they did not click on the advertisement. If a user does not click on an advertisement, does this mean that they are not interested in the content? These unlabeled examples can be either positive or negative which leads us to the primary question of Positive and Unlabeled (PU) learning: how do we classify a negative examples when we do not know what a negative example looks like?

There are several techniques that try to create a set of reliable negatives (i.e. examples that we are sufficiently confident are negative) in the unlabeled data (see section 3). Our proposed approach to create these reliable negatives uses a clustering algorithm. The assumption that we are making is that the positive examples in the unlabeled dataset will be similar in some dimensional space to the positive examples in our labeled dataset since we assume they are independent and identically distributed. Using a clustering algorithm should give us distinct groups of examples. The groups that are strongly positive can be ignored in this case since we are only looking for reliable negatives. The mostly unlabeled clusters can turn into our reliable negatives. In our second experiment, we will iteratively label our highest confidence data until all of our initial labeled set has a label.

In our experiments, we are specifically looking at PU learning as a method for text classification. As such, we need a way to encode each document (example) into a feature vector. We propose to use a Bag-of-Words (BOW) vector:

$$\boldsymbol{d}_j = [w_{1j} \;\; w_{2j} \;\; \ldots \;\; w_{mj}] \tag{1}$$

where $\boldsymbol{d}_j \in \boldsymbol{D}$ is the document vector j and w_{ij} is the count of word i in document j. However, this allows common words (i.e. "the", "and", etc.) across document categories to skew each vector towards similarity. A common solution is to give each word a weight based on its frequency among documents. This solution is called *TF.IDF* which is Term Frequency times Inverse Document Frequency. The Term Frequency for word i in document j is defined as:

$$TF_{ij} = \frac{w_{ij}}{\text{MAX}_k \; w_{kj}} \tag{2}$$

The inverse document frequency for word i is defined as:

$$IDF_i = \log_2 \frac{|\boldsymbol{D}|}{n_i} \tag{3}$$

where n_i is the number of documents that contain word i.[5, 6] Using this solution, our feature vectors will look like:

$$\boldsymbol{d}_j = [TF_{1j} \times IDF_1 \; , \; TF_{2j} \times IDF_2 \; , \; \ldots \; , \; TF_{nj} \times IDF_m] \tag{4}$$

In our third experiment, we are looking at the effectiveness of finding the set of reliable negatives. Our hypothesis is that the difference in accuracy between random sampling and modeling RN will correlate with how unbalanced the unlabeled dataset is. However, if we determine that negative random sampling is just as effective as these techniques used in PU learning, then previous 2 step approaches defined in [3] are unnecessary and all we need is an SVM or some other semi-supervised method.

3 Related Work

Learning with positive and unlabeled data has been an area of research that has gained additional traction since the early 2000s, but has been relevant in information retrieval applications for much longer. [7] In 2003, Lee and Liu [8] introduced a weighted logistic regression model using linear functions to learn from the PU data. Another approach to PU learning is to task a model with creating a set of reliable negative examples (RN) from the set of unlabeled examples (U). This RN can then be coupled with our positive set (P) and the remainder of the unlabeled set ($Q = U - RN$) to train a classifier using semi-supervised methods. Biu et al.[3] go into these methods in more detail, but we will outline methods used in our experiments.

Our control experiment uses a Naive Bayes classifier to create the RN since this method (coupled with an SVM) performs comparably better than other proposed methods.[3] The algorithm, as stated in Liu et al.[3], is as follows:

1. Give a label of 1 and -1 to each example in P and U, respectively.
2. Train a Naive Bayes classifier using these labels.
3. Using this classifier, classify the unlabeled examples. Those that are classified as negative examples will comprise RN.

We will also use an SVM for our experiments that require a semi-supervised method for a final evaluation. For this application a soft margin SVM is proposed to combat the error from noisy labels.[1, 3] This would look like the linear program:

$$\text{MINIMIZE: } \frac{1}{2}\boldsymbol{W}^T\boldsymbol{W} + C\sum_{j=1}^{n}\xi_j \tag{5}$$

$$\text{SUBJECT TO: } y_j(\boldsymbol{W}^T\boldsymbol{d}_j + b) \geq 1 - \xi_j, \;\; j = 1, 2, \ldots, n$$

where $\boldsymbol{W}$ is a weight matrix, C is a parameter to control the amount of allowed error, and y_j is the label for document j.

There is previous work on how well clustering algorithms perform at these classification tasks like text classification. Kyriakopolou and Kalamboukis [6] propose a supervised algorithm using clustering. This algorithm clusters the documents into k clusters and extracts *meta-features* from each of these clusters. In this case, meta-features are an additional feature that says whether or not the document was in cluster l. The meta-features are then added similarly to word counts to a document vector. They showed that we can use this information to augment the IDF in the sense that we can look at n_{m+l} as being how many documents belong to cluster l. An SVM is then used to build the final classifier using these features. Further, they found that a transductive SVM [9] yields more accurate results than a normal SVM.

In many of the applications of PU learning, such as fraud detection, there is a large class imbalance between the positive and negative samples. There are very few fraudulent charges compared to the total number of charges. Akbani, Kwek, and Japkowicz [10] suggest that undersampling may not be the most effective method since we are losing data and adding a possible bias into the data. Instead, they propose adding synthetic oversampling to augment the positive training set. The inherent issue with this problem is that we would need to know ahead of the experiment that there is a class imbalance. For common tasks like fraud detection, this may be obvious, but for our purposes, we will not make this assumption.

References

[1] K. Bennett and A. Demiriz (1999). Semi-Supervised Support Vector Machines. *Advances in Neural Information Processing Systems*, 11, 368–374.

[2] K. Nigam, A. McCallum, and T. Mitchell (2006). Semi-supervised Text Classification Using EM. *Semi-Supervised Learning*. MIT Press.

[3] B. Liu, Y. Dai, X. Li, W. S. Lee and and P. Yu (2003). Building Text Classifiers Using Positive and Unlabeled Examples. *Proceedings of the Third IEEE International Conference on Data Mining*.

[4] K. Lang (1995). *Newsweeder: Learning to filter netnews*. ICML-95.

[5] A. Rajaraman and J.D. Ullman (2011). Data Mining. *Mining of Massive Datasets*. 1–17.

[6] A. Kyriakopoulou and T. Kalamboukis (2006). Text classification using clustering. *In ECML-PKDD. Discovery Challenge Workshop Proceedings*.

[7] J. Rocchio (1971). Relevance Feedback in Information Retrieval. *The Smart Retrieval System - Experiments in Automatic Document Processing*.

[8] W. S. Lee and B. Liu (2003). Learning with Positive and Unlabeled Examples using Weighted Logistic Regression. *Proceedings of the Twentieth International Conference on Machine Learning* (ICML-2003).

[9] T. Joachims (1999). Transductive Inference for Text Classification Using Support Vector Machines. *Proceedings of 16th International Conference on Machine Learning*. 200-209.

[10] R. Akbani, S. Kwek, and N. Japkowicz(2004). Applying Support Vector Machines to Imbalanced Data Sets. *in Proceedings of 15th ECML*. 39–50.