Painless Labeling with Application to Text Mining

Sajib Dasgupta
Chittagong Indepedent University
Chittagong, Bangladesh
sdgnew@gmail.com

Abstract

Labeled data is not readily available for many natural language domains, and it typically requires expensive human effort with considerable domain knowledge to produce a set of labeled data. In this paper, we propose a simple unsupervised system that helps us create a labeled resource for categorical data (e.g., a document set) using only fifteen minutes of human input. We utilize the labeled resources to discover important insights about the data. The entire process is domain independent, and demands no prior annotation samples, or rules specific to an annotation.

1 Introduction

Consider the following two scenarios:

Scenario 1: We start processing a new language and we want to get an initial idea of the language before embarking on the expensive process of creating hand annotated resources. For instance, we may want to know how people express opinion in a language of interest, what characterizes the subjective content of the language and how expressions of opinion differ along opinion types. The question is how to acquire such first-hand insights of an unknown language in quick time and with minimal human effort?

Scenario 2: We have a set of blog articles and we are interested in learning how blogging differs across gender. In particular, we seek to learn the writing styles or other indicative patterns – topics of interest, word choices etc. – that can potentially distinguish writings across gender. A traditional NLP approach would be to collect a set of articles that are tagged with gender information, which we can then input to a learning system to learn patterns that can differentiate gender. What if no such annotation is available, as the bloggers don't reveal their gender information? Could we arrange a human annotation task to annotate the articles along gender? Often the articles contain explicit patterns (e.g., "my boyfriend", "as a woman" etc.) which help the annotators to annotate the articles. Often there are no indicative patterns in the written text, and it becomes impossible to annotate the articles reliably.

The above scenarios depict the cases when we are resource constrained and creating a new resource is nontrivial and time consuming. Given such difficulties, it would be helpful if we could design a system that requires less human input to create a labeled resource. In this paper, we present a simple unsupervised system that helps us create a labeled resource with minimal human effort. The key to our method is that instead of labeling the entire set of unlabeled instances the system labels a subset of data instances for which it is confident to achieve high level of accuracy. We experiment with several document labeling tasks and show that a high-quality labeled resource can be produced by a clustering-based labeling system that requires a mere fifteen minutes of human input. It achieves 85% and 78% accuracy for the task of sentiment and gender classification, showing its effectiveness on two nontrivial labeling tasks with distinct characteristics (see Section 3).

We also utilize the labeled resources created by our system to learn discriminative patterns that help us gain insights into a dataset. For instance, we learn how users generally express opinion in a language of interest, and how writing varies across gender. The next section describes the details of our main algorithm. We present experimental results in Section 3 and 4.

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics
and the 7th International Joint Conference on Natural Language Processing (Short Papers), pages 402–407,
Beijing, China, July 26-31, 2015.

The sound from my system did seem to be a little better
(the CD's were not skipping as much).
But the bottom line is it didn't fix the problem as
the CDs are still skipping noticeably,
although not as bad as before. ..

Table 1: Snippet of an ambiguous CD Player review.

2 Problem Formulation

We consider a general classification framework. Let $X = \{x_1, \ldots, x_n\}$ represents a categorical dataset with n data points where $x_i \in \Re^d$. Let $c_x \in$ {1,-1} is the true label of x^1. Our goal is label a subset of the data, $X' = \{C_1, C_2\} \subseteq X$, where C_1 and C_2 comprise data points of positive and negative class respectively. Note that, X' represents the subset of datapoints that are confidently labeled by the system.

To illustrate, we show a snippet of a CD player review taken from Amazon in Table 1. As you can see this review is highly ambiguous, as it describes both the positive and negative aspects of the product: while the phrases *a little better*, *not skipping*, and *not as bad* conveys a positive sentiment, the phrases *didn't fix* and *skipping noticeably* are negative sentiment-bearing. Any automated system would find it *hard* to correctly label this review, as the review is highly ambiguous. Our goal is to remove such ambiguous data points from the data space and label the remaining unambiguous data points. The fact that unambiguous data instances are easier to label allows us to use an automated system to label them quickly with minimal human effort (see the next section).

Now how could we set apart unambiguous data points from the ambiguous ones from a set of unlabeled data points? Note that we desire the system to be unsupervised. We also desire the system to be generic i.e., applicable to any application domain. Next we show how we extend spectral clustering to achieve this goal.

2.1 Ambiguity Resolution with Iterative Spectral Clustering

In spectral clustering, a set of n data points is represented as an undirected graph, where each node corresponds to a data point and the edge weight between two nodes is their similarity as defined by S. The goal is to induce a clustering, or equivalently, a *partitioning function* f, which is typically represented as a vector of length n such that $f(i) \in \{1, -1\}$ indicates which of the two clusters data point i should be assigned to.

In spectral clustering, the normalized cut partition of a similarity graph, S is derived from the solution of the following constrained optimization problem: $\mathrm{argmin}_{f \in \Re^n} \quad \sum_{i,j} S_{i,j}(\frac{f(i)}{\sqrt{d_i}} - \frac{f(j)}{\sqrt{d_j}})^2$ subject to $f^T D f = 1$ and $Df \perp \mathbf{1}$, where D is a diagonal matrix with $D_{i,i} = \sum_j S_{i,j}$ and $d_i = D_{i,i}$. The closed form solution to this optimization problem is the eigenvector corresponding to the second smallest eigenvalue of the Laplacian matrix, $L = D^{-1/2}(D - S)D^{-1/2}$ (Shi and Malik (2000)). Clustering using the second eigenvector, is trivial: since we have a linearization of the points, all we need to do is to determine a threshold for partitioning the data points.

Second eigenvector reveals useful information regarding the ambiguity of the individual data points. In the computation of eigenvectors each data point factors out orthogonal projections of each of the neighboring data points. Ambiguous data points factor out orthogonal projections from both the positive and negative data instances, and hence they have near zero values in the pivot eigenvectors. We exploit this important information. The basic idea is that the data points with near zero values in the second eigenvector are more ambiguous than those with large absolute values. Hence, to cluster only the unambiguous datapoints, we can therefore sort the data points according to second eigenvector, and keep only the top and bottom $m(m < n)$ datapoints. Finally, instead of removing $(n - m)$ datapoints at once, we remove them in iteration.

Here is our final algorithm:

1. Let $s : X \times X \rightarrow \Re$ be a similarity function defined over data X. Construct a similarity matrix S such that $S_{ij} = s(x_i, x_j)$.

2. Construct the Laplacian matrix $L = D^{-1/2}(D - S)D^{-1/2}$, where D is a diagonal matrix with $D_{i,i} = \sum_j S_{i,j}$.

3. Find *eigenvector* e_2 corresponding to second smallest eigenvalue of L.

4. Sort X according to e_2 and remove α points indexed from $(|X|/2 - \alpha/2 + 1)$ to $(|X|/2 + \alpha/2)$.

5. If $|X|$ = m, goto Step 6; else goto Step 1.

[1] We present our system for binary classification task. It can be extended fairly easily to multi-way classification tasks.

Dataset	System	$m = \frac{1}{5}n$	$m = \frac{2}{5}n$	$m = \frac{3}{5}n$	$m = \frac{4}{5}n$	$m = n$	Fully Supervised
Gender	Kmeans++	52.3%	51.6%	52.3%	51.7%	51.2%	-
	TSVM	53.1%	53.6%	52.7%	52.6%	52.0%	**80.4%**
	OUR	**78.5%**	**73.7%**	**69.3%**	**66.8%**	**64.4%**	-
Spam	Kmeans++	67.6%	58.6%	54.9%	53.8%	52.4%	-
	TSVM	**87.8%**	**85.0%**	**82.7%**	**80.7%**	**78.9%**	**96.9%**
	OUR	83.8%	82.9%	80.4%	79.8%	78.4%	-
Sentiment	Kmeans++	64.5%	61.4%	60.5%	57.8%	56.5%	-
	TSVM	70.2%	65.1%	61.5%	61.8%	60.4%	**86.4%**
	OUR	**90.3%**	**85.4%**	**79.9%**	**74.9%**	**71.2%**	-

Table 2: Accuracy of automatically labeled data for each dataset. We also report 5-fold supervised classification result for each dataset.

6. Sort X according to e_2 and put top $\frac{m}{2}$ data points in cluster C_1 and bottom $\frac{m}{2}$ data points in cluster C_2.

In the algorithm stated above, we start with an initial clustering of all of the data points, and then iteratively remove the α most ambiguous points from the data space. We iterate the process of removing ambiguous data points and re-clustering until we have m data points remaining. It should not be difficult to see the advantage of removing the data points in an iterative fashion (as opposed to removing them in a single iteration): the clusters produced in a given iteration are supposed to be better than those in the previous iterations, as subsequent clusterings are generated from less ambiguous points. In all our experiments, we set α to 100. Finally, we label the clusters by inspecting 10 randomly sampled points from each cluster. We use the cluster labels to assign labels to the m unambiguous data points. Note that labeling the clusters is the only form of human input we require in our system.

3 Experiments

We use three text classification tasks for evaluation:

Gender Classification: Here we classify blog articles according to whether an article is written by a male or female. We employ the blog dataset as introduced by Schler et al. (2006) for this task. The dataset contains 19320 blog articles, out of which we randomly selected 5000 blog articles as our dataset.

Spam Classification: Here the goal is to determine whether an email is Spam or Ham (i.e., not spam). We use the Enron spam dataset as introduced by Metris et al. (Metsis et al. (2006)). We join together the BG section of Spam emails and kaminski section of Ham emails, and randomly selected 5000 emails as our dataset.

Sentiment Classification: Here the goal is to determine whether the sentiment expressed in a product review is positive or negative. We use Pang et al.'s movie review dataset for this task (Pang et al. (2002)). The dataset contains 2000 reviews annotated with the positive and negative sentiment label.

To preprocess a document, we first tokenize and downcase it, remove stop words, and represent it as a vector of unigrams, using frequency as presence. For spectral clustering, we use dot product as a measure of similarity between two documents vectors.

Dataset	Data points	Features	Pos:Neg
Gender	5000	75188	2751:2249
Spam	5000	23760	2492:2508
Sentiment	2000	24531	1000:1000

Table 3: Description of the datasets.

3.1 Accuracy of Automatically Labeled Data

For each dataset, given n unlabeled data points, we apply our system to label $m(m <= n)$ least ambiguous data points. We check the quality of labeled data by comparing the assigned (cluster) labels of m datapoints against their true labels, and show the accuracy. Table 2 shows the accuracy of automatically labeled data for five different values of m for each dataset. For example, when $m = n/5$, our system labels 1000 out of available 5000 data points with 78.5% accuracy for the gender dataset. These 1000 data points are the most unambiguous out of the 5000 data points, as selected by the algorithm. For $m = n$ the system labels the entire dataset.

As you can see, for all three datasets, the accuracy of labeling unambiguous data instances is much higher than the accuracy of labeling the entire dataset. For instance, the accuracy of top $n/5$ unambiguous labeled instances of the sentiment dataset is 90.3%, whereas the accuracy of labeling the entire dataset is 71.2%. The more unambiguous the data instances are the higher is the quality of labeled data (as shown by the fact that the accuracy of labeled instances increases as we increase m). Notice that our system labels 60% of the data points of the spam dataset with 80.4% accuracy; 40% of the data points of the sentiment dataset with 85.4% accuracy; and 20% of the data points of the gender dataset with 78.5% accuracy.

We also report 5-fold supervised classification result for each dataset. We used linear SVM for classification with all parameters set to their default values. As you can see, when $m = n/5$ our system achieves near supervised labeling performance for the gender and sentiment dataset. One of the reviewers asked how SVM performed when trained with unambiguous data instances alone. We refer to Dasgupta and Ng (2009) where the authors report that training SVM with unambiguous data alone produces rather inferior result. They, however, work on a small data sample. It would be interesting to know whether large number of unambiguous (or, semi-ambiguous) data instances could offset the need for ambiguous data in a general classification setting. Given that unlabeled data are abundantly available in many NLP tasks, one can employ our method to create decent size labeled data quickly from unlabeled data, and utilize them later in the process to build an independent classifier or augment the performance of an existing classifier (Fuxman et al. (2009)).

We employed two baseline algorithms, i.e., kmeans++ and a semi-supervised learning system, Transductive SVM. For kmeans++ we used the following as a measure of ambiguity for each data point: $1-\frac{(\mathbf{x}-\mu_\mathbf{i})^\mathbf{2}}{\sum_i^k(\mathbf{x}-\mu_\mathbf{i})^\mathbf{2}}$, where $\mathbf{x}$ is a data vector and $\mu_\mathbf{i}$, $i = 1 : k$ are k mean vectors. It ranges from 0 to 1. Ambiguity score near 0.5 suggests that the data point is ambiguous. Following common practice in document clustering, we reduced the dimensions of the data to 100 using SVD before we apply kmeans++. For transductive SVM, we randomly selected 20 labeled data points as seeds. Table 2 shows the result for each baseline.

Notice that our system beat the baselines (one of them is a semisupervised system) by a big margin for the Gender and Sentiment dataset, whereas Transductive SVM performs the best for the Spam dataset. Interesting to point that our method of removing ambiguous data instances to get a qualitatively stronger clustering contrasts with the max-margin methods which use the ambiguous data instances to acquire the margin. Also important to mention that spectral clustering is a graph-based clustering algorithm, where similarity measure employed to construct the graph plays a crucial role in performance (Maier et al. (2013)). In fact, “right” construction of the feature space and a right similarity measure can considerably change the performance of a graph-based clustering algorithm. We have not tried different similarity measures in this initial study, but it provides us room for improvement for a dataset like Spam.

Implementation Details: On a machine with 3GHz of Intel Quad Core Processor and 4GB of RAM, the iterative spectral clustering algorithm takes less than 2 minutes in Matlab for a dataset comprising 5000 data points and 75188 features. This along with the fact that human labelers take on average 12 minutes to label the clusters suggests that the entire labeling process requires less than 15 minutes to complete.

4 Mining Patterns and Insights

In this section, we show that we can utilize the labeled resources created by our system to learn discriminative patterns that help us gain insights into a dataset (Don et al. (2007), Larsen and Aone (1999), Cheng et al. (2007), Maiya et al. (2013)). We utilize the top $n/5$ unambiguous labeled instances for this task, where n is size of the dataset. Note that the quality of unambiguous labeled instances is much higher than the entire set of labeled instances (see Section 3.1), so the statistics we collect from the unambiguous labeled instances to identify discriminative patterns are supposedly more reliable.

We learn our first category of discriminative patterns the following way: for each cluster, we rank all unigrams in the vocabulary by their weighted log-likelihood ratio:

$$P(w_t \mid c_j) \cdot \log \frac{P(w_t \mid c_j)}{P(w_t \mid \neg c_j)}$$

where w_t and c_j denote the t-th word in the vocabulary and the j-th cluster, respectively, and each conditional probability is add one smoothed. Informally, a unigram w will have a high rank with respect to a cluster c if it appears frequently in c and infrequently in $\neg c$. The higher the score the more discriminative the pattern is. We also learn the discriminative bigrams similarly: for each cluster, we rank all bigrams by their weighted

Dataset	Class	Top Discriminative Unigrams
Gender	**Female**	*haha, wanna, sooo, lol, ppl, omg, hahaha, ur, yay, soo, cuz, bye, soooo, hehe, ate, hurts, sucks.*
	Male	*provide, reported, policies, administration, companies, development, policy, services, nations.*
Spam	**Spam**	*vicodin, goodbye, utf, rolex, watches, loading, promotion, reproductions, nepel, fw, fwd, click.*
	Ham	*risk, securities, statements, exchange, terms, third, events, act, investing, objectives, assumptions.*
Sentiment	**Positive**	*relationship, husband, effective, mother, strong, perfect, tale, novel, fascinating, outstanding.*
	Negative	*stupid, worst, jokes, bunch, sequel, lame, guess, dumb, boring, maybe, guys, video, flick, oh.*

Table 4: Top discriminative unigram patterns identified by our system.

Dataset	Class	Top Discriminative Bigrams
Gender	**Female**	*wanna go, im so, im gonna, at like, don't wanna, was sooo, was gonna, soo much, so yeah.*
	Male	*to provide, york times, the issue, understanding of, the political, bush admin, the democratic.*
Spam	**Spam**	*promotional material, adobe photoshop, name it, choose from, you name, stop getting, office xp.*
	Ham	*investment advice, this report, respect to, current price, risks and, information provided.*
Sentiment	**Positive**	*story of, her husband, relationship with, begins to, love and, life of, the central, the perfect.*
	Negative	*the worst, bad movie, bunch of, got to, too bad, action sequences, waste of, than this, the bad.*

Table 5: Top discriminative bigram patterns identified by our system.

log-likelihood ratio score and select the top scoring bigrams as the most discriminative bigrams.

Table 4 and 5 show the most discriminative unigrams and bigrams learned by our system. Notice that the learned patterns are quite informative. For instance, in the case of blog dataset we learn that certain word usages (e.g., sooo, cuz etc.) are more common in women's writings, whereas men's writings often contain discussion of politics, news and technology. For sentiment data, the patterns correspond well to the generic sentiment lexicon manually created by the sentiment experts. The ability of our system to learn top sentiment features could be handy for a resource-scarce language, which may not have a general purpose sentiment lexicon. Note that the system is not limited to unigram and bigram patterns only. The labeled instances can be utilized similarly to gather statistics for other form of usage patterns including syntactic and semantic patterns for document collections.

5 Related Work

Automatic extraction of labeled data has gained momentum in recent years (Durme and Pasca (2008), Nakov and Hearst (2005), Fuxman et al. (2009)). Traditionally, researchers use task-specific heuristics to generate labeled data, e.g., searching for a specific pattern in the web to collect data instances of a particular category (Hearst (1992), Go et al. (2009), Hu et al. (2013)). Another line of research follows semi-supervised information extraction task, where given a list of seed instances of a particular category, a bootstrapping algorithm is applied to mine new instances from large corpora (Riloff and Jones (1999), Etzioni et al. (2005), Durme and Pasca (2008)).

There has also been a surge of interests in unsupervised approaches which primarily rely on clustering to induce psuedo labels from large amount of text (Clark (2000), Slonim and Tishby (2000), Sahoo et al. (2006), Christodoulopoulos et al. (2010)). We differ from existing unsupervised clustering algorithms in a way that we uncomplicate spectral clustering by forcing it to cluster unambiguous data points only, which ensures that the system makes less mistakes during clustering and the clustered data are qualitatively strong.

6 Conclusion

We have presented a system that helps us create a labeled resource for a given dataset with minimal human effort. We also utilize the labeled resources to discover important insights about the data. The ability of our system to learn and visualize top discriminative patterns facilitates exploratory data analysis for a dataset that might be unknown to us. Even if we have some knowledge of the data, the system may unveil additional characterisitcs that are unknown to us. The top features induced for each classification task can also be interpreted as our system's ability to discover new feature spaces, which can be utilized independently or along with a simpler feature space (e.g., *bag of words*) to learn a better classification model. Additional research is needed to further explore this idea.

Acknowledgements

We acknowledge three anonymous reviewers and Vincent Ng for their valuable feedback on an earlier draft of the paper.

References

H. Cheng, X. Yan, J. Han, and C. Hsu. 2007. Discriminative frequent pattern analysis for effective classification. In *International Conference on Data Engineering (ICDE)*.

C. Christodoulopoulos, S. Goldwater, and M. Steedman. 2010. Two decades of unsupervised pos induction: How far have we come? In *Empirical Methods in Natural Language Processing (EMNLP)*.

Alexander Clark. 2000. Inducing syntactic categories by context distributional clustering. In *the Conference on Natural Language Learning (CoNLL)*.

S. Dasgupta and V. Ng. 2009. Mine the easy, classify the hard: A semi-supervised approach to automatic sentiment classification. In *ACL-IJCNLP 2009: Proceedings of the Main Conference*.

A. Don, E. Zheleva, M. Gregory, S. Tarkan, L. Auvil, T. Clement, B. Shneiderman, and C. Plaisant. 2007. Discovering interesting usage patterns in text collections: integrating text mining with visualization. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*.

Benjamin Van Durme and Marius Pasca. 2008. Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction. In *the AAAI Conference on Artificial Intelligence (AAAI)*.

O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. 2005. Unsupervised named-entity extraction from the web: an experimental study. In *Artificial Intelligence*.

A. Fuxman, A. Kannan, A. Goldberg, R. Agrawal, P. Tsaparas, and J. Shafer. 2009. Improving classification accuracy using automatically extracted training data. In *15th ACM Conference on Knowledge Discovery and Data Mining (SIGKDD)*.

A Go, R Bhayani, and L Huang. 2009. Twitter sentiment classification using distant supervision. In *Project Report, Stanford University*.

M. A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *the International Conference on Computational Linguistics (COLING)*.

X. Hu, J. Tang, H. Gao, and H. Liu. 2013. Unsupervised sentiment analysis with emotional signals. In *In the Proceedings of the International World Wide Web Conference (WWW)*.

B. Larsen and C. Aone. 1999. Fast and effective text mining using linear-time document clustering. In *the Conference on Knowledge Discovery and Data Mining (SIGKDD)*.

M. Maier, U. von Luxburg, and M. Hein. 2013. How the result of graph clustering methods depends on the construction of the graph. In *ESAIM: Probability and Statistics, vol. 17*.

A. S. Maiya, J. P. Thompson, F. Loaiza-Lemos, and R. M. Rolfe. 2013. Exploratory analysis of highly heterogeneous document collections. In *the Conference on Knowledge Discovery and Data Mining (SIGKDD)*.

V. Metsis, I. Androutsopoulos, and G. Paliouras. 2006. Spam filtering with naive bayes - which naive bayes? In *3rd Conference on Email and Anti-Spam (CEAS)*.

Preslav Nakov and Marti Hearst. 2005. Using the web as an implicit training set: Application to structural ambiguity resolution. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

E. Riloff and R. Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*.

N. Sahoo, J. Callan, R. Krishnan, G. Duncan, and R. Padman. 2006. Incremental hierarchical clustering of text documents. In *the International Conference on Information and Knowledge Management (CIKM)*.

J. Schler, M. Koppel, S. Argamon, and J. Pennebaker. 2006. Effects of age and gender in blogging. In *AAAI Symposium on Computational Approaches for Analyzing Weblogs*.

Jianbo Shi and Jitendra Malik. 2000. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Noam Slonim and Naftali Tishby. 2000. Document clustering using word clusters via the information bottleneck method. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*.