Sprinkling Topics for Weakly Supervised Text Classification

Swapnil Hingmire[1,2]
swapnil.hingmire@tcs.com

Sutanu Chakraborti[2]
sutanuc@cse.iitm.ac.in

[1]Systems Research Lab, Tata Research Development and Design Center, Pune, India
[2]Department of Computer Science and Engineering,
Indian Institute of Technology Madras, Chennai, India

Abstract

Supervised text classification algorithms require a large number of documents labeled by humans, that involve a labor-intensive and time consuming process. In this paper, we propose a weakly supervised algorithm in which supervision comes in the form of labeling of Latent Dirichlet Allocation (LDA) topics. We then use this weak supervision to "sprinkle" artificial words to the training documents to identify topics in accordance with the underlying class structure of the corpus based on the higher order word associations. We evaluate this approach to improve performance of text classification on three real world datasets.

1 Introduction

In supervised text classification learning algorithms, the learner (a program) takes human labeled documents as input and learns a decision function that can classify a previously unseen document to one of the predefined classes. Usually a large number of documents labeled by humans are used by the learner to classify unseen documents with adequate accuracy. Unfortunately, labeling a large number of documents is a labor-intensive and time consuming process.

In this paper, we propose a text classification algorithm based on Latent Dirichlet Allocation (LDA) (Blei et al., 2003) which does not need labeled documents. LDA is an unsupervised probabilistic topic model and it is widely used to discover latent semantic structure of a document collection by modeling words in the documents. Blei et al. (Blei et al., 2003) used LDA topics as features in text classification, but they use labeled documents while learning a classifier. sLDA (Blei and McAuliffe, 2007), DiscLDA (Lacoste-Julien et al., 2008) and MedLDA (Zhu et al., 2009) are few extensions of LDA which model both class labels and words in the documents. These models can be used for text classification, but they need expensive labeled documents.

An approach that is less demanding in terms of knowledge engineering is ClassifyLDA (Hingmire et al., 2013). In this approach, a topic model on a given set of unlabeled training documents is constructed using LDA, then an annotator assigns a class label to some topics based on their most probable words. These labeled topics are used to create a new topic model such that in the new model topics are better aligned to class labels. A class label is assigned to a test document on the basis of its most prominent topics. We extend ClassifyLDA algorithm by "sprinkling" topics to unlabeled documents.

Sprinkling (Chakraborti et al., 2007) integrates class labels of documents into Latent Semantic Indexing (LSI)(Deerwester et al., 1990). The basic idea involves encoding of class labels as artificial words which are "sprinkled" (appended) to training documents. As LSI uses higher order word associations (Kontostathis and Pottenger, 2006), sprinkling of artificial words gives better and class-enriched latent semantic structure. However, Sprinkled LSI is a supervised technique and hence it requires expensive labeled documents. The paper revolves around the idea of labeling topics (which are far fewer in number compared to documents) as in ClassifyLDA, and using these labeled topic for sprinkling.

As in ClassifyLDA, we ask an annotator to assign class labels to a set of topics inferred on the unlabeled training documents. We use the labeled topics to find probability distribution of each training document over the class labels. We create a set of artificial words corresponding to a class label and add (or sprinkle) them to the document. The number of such artificial terms is propor-

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 55–60,
Baltimore, Maryland, USA, June 23-25 2014. ©2014 Association for Computational Linguistics

tional to the probability of generating the document by the class label. We then infer a set of topics on the sprinkled training documents. As LDA uses higher order word associations (Lee et al., 2010) while discovering topics, we hypothesize that sprinkling will improve text classification performance of ClassifyLDA. We experimentally verify this hypothesis on three real world datasets.

2 Related Work

Several researchers have proposed semi-supervised text classification algorithms with the aim of reducing the time, effort and cost involved in labeling documents. These algorithms can be broadly categorized into three categories depending on how supervision is provided. In the first category, a small set of labeled documents and a large set of unlabeled documents is used while learning a classifier. Semi-supervised text classification algorithms proposed in (Nigam et al., 2000), (Joachims, 1999), (Zhu and Ghahramani, 2002) and (Blum and Mitchell, 1998) are a few examples of this type. However, these algorithms are sensitive to initial labeled documents and hyper-parameters of the algorithm.

In the second category, supervision comes in the form of labeled words (features). (Liu et al., 2004) and (Druck et al., 2008) are a few examples of this type. An important limitation of these algorithms is coming up with a small set of words that should be presented to the annotators for labeling. Also a human annotator may discard or mislabel a polysemous word, which may affect the performance of a text classifier.

The third type of semi-supervised text classification algorithms is based on active learning. In active learning, particular unlabeled documents or features are selected and queried to an oracle (e.g. human annotator).(Godbole et al., 2004), (Raghavan et al., 2006), (Druck et al., 2009) are a few examples of active learning based text classification algorithms. However, these algorithms are sensitive to the sampling strategy used to query documents or features.

In our approach, an annotator does not label documents or words, rather she labels a small set of interpretable topics which are inferred in an unsupervised manner. These topics are very few, when compared to the number of documents. As the most probable words of topics are representative of the dataset, there is no need for the annotator to search for the right set of features for each class. As LDA topics are semantically more meaningful than individual words and can be acquired easily, our approach overcomes limitations of the semi-supervised methods discussed above.

3 Background

3.1 LDA

LDA is an unsupervised probabilistic generative model for collections of discrete data such as text documents. The generative process of LDA can be described as follows:

1. for each topic t, draw a distribution over words: $\phi_t \sim \text{Dirichlet}(\beta_w)$
2. for each document $d \in D$
   a. Draw a vector of topic proportions: $\theta_d \sim \text{Dirichlet}(\alpha_t)$
   b. for each word w at position n in d
      i. Draw a topic assignment: $z_{d,n} \sim \text{Multinomial}(\theta_d)$
      ii. Draw a word: $w_{d,n} \sim \text{Multinomial}(z_{d,n})$

Where, T is the number of topics, ϕ_t is the word probabilities for topic t, θ_d is the topic probability distribution, $z_{d,n}$ is topic assignment and $w_{d,n}$ is word assignment for nth word position in document d respectively. α_t and β_w are topic and word Dirichlet priors.

The key problem in LDA is posterior inference. The posterior inference involves the inference of the hidden topic structure given the observed documents. However, computing the exact posterior inference is intractable. In this paper we estimate approximate posterior inference using collapsed Gibbs sampling (Griffiths and Steyvers, 2004).

The Gibbs sampling equation used to update the assignment of a topic *t* to the word $w \in W$ at the position *n* in document *d*, conditioned on α_t, β_w is:

$$P(z_{d,n} = t | z_{d,\neg n}, w_{d,n} = w, \alpha_t, \beta_w) \propto \frac{\psi_{w,t} + \beta_w - 1}{\sum_{v \in W} \psi_{v,t} + \beta_v - 1} \times (\Omega_{t,d} + \alpha_t - 1) \quad (1)$$

where $\psi_{w,c}$ is the count of the word w assigned to the topic c, $\Omega_{c,d}$ is the count of the topic c assigned to words in the document d and W is the vocabulary of the corpus. We use a subscript $d, \neg n$ to denote the current token, $z_{d,n}$ is ignored in the Gibbs sampling update. After performing collapsed Gibbs sampling using equation 1, we use word topic assignments to compute a point estimate of the distribution over words $\phi_{w,c}$ and a point estimate of the posterior distribution over topics for each document *d* (θ_d) is:

$$\phi_{w,t} = \frac{\psi_{w,t} + \beta_w}{\left[\sum\limits_{v \in W} \psi_{v,t} + \beta_v\right]} \quad (2) \qquad \theta_{t,d} = \frac{\Omega_{t,d} + \alpha_t}{\left[\sum\limits_{i=1}^{T} \Omega_{i,d} + \alpha_i\right]} \quad (3)$$

Let $M_D = < Z, \Phi, \Theta >$ be the hidden topic structure, where *Z* is per word per document topic assignment, $\Phi = \{\phi_t\}$ and $\Theta = \{\theta_d\}$.

3.2 Sprinkling

(Chakraborti et al., 2007) propose a simple approach called "sprinkling" to incorporate class labels of documents into LSI. In sprinkling, a set of artificial words are appended to a training document which are specific to the class label of the document. Consider a case of binary classification with classes c_1 and c_2. If a document *d* belongs to the class c_1 then a set of artificial words which represent the class c_1 are appended into the document *d*, otherwise a set of artificial words which represent the class c_2 are appended.

Singular Value Decomposition (SVD) is then performed on the sprinkled training documents and a lower rank approximation is constructed by ignoring dimensions corresponding to lower singular values. Then, the sprinkled terms are removed from the lower rank approximation. (Chakraborti et al., 2007) empirically show that sprinkled words boost higher order word associations and projects documents with same class labels close to each other in latent semantic space.

4 Topic Sprinkling in LDA

In our text classification algorithm, we first infer a set of topics on the given unlabeled document corpus. We then ask a human annotator to assign one or more class labels to the topics based on their most probable words. We use these labeled topics to create a new LDA model as follows. If the topic assigned to the word *w* at the position *n* in document *d* is *t*, then we replace it by the class label assigned to the topic *t*. If more than one class labels are assigned to the topic *t*, then we randomly select one of the class labels assigned to the topic *t*. If the annotator is unable to label a topic then we randomly select a class label from the set of all class labels. We then update the new LDA model using collapsed Gibbs sampling.

We use this new model to infer the probability distribution of each unlabeled training document over the class labels. Let, $\theta_{c,d}$ be the probability of generating document *d* by class *c*. We then sprinkle *s* artificial words of class label *c* to document *d*, such that $s = K * \theta_{c,d}$ for some constant *K*.

We then infer a set of $|C|$ number of topics on the sprinkled dataset using collapsed Gibbs sampling, where *C* is the set of class labels of the training documents. We modify collapsed Gibbs sampling update in Equation 1 to carry class label information while inferring topics. If a word in a document is a sprinkled word then while sampling a class label for it, we sample the class label associated with the sprinkled word, otherwise we sample a class label for the word using Gibbs update in Equation 1.

We name this model as Topic Sprinkled LDA (TS-LDA). While classifying a test document, its probability distribution over class labels is inferred using TS-LDA model and it is classified to its most probable class label. Algorithm for TS-LDA is summarized in Table 1.

5 Experimental Evaluation

We determine the effectiveness of our algorithm in relation to ClassifyLDA algorithm proposed in (Hingmire et al., 2013). We evaluate and compare our text classification algorithm by computing Macro averaged F1. As the inference of LDA is approximate, we repeat all the experiments for each dataset ten times and report average Macro-F1. Similar to (Blei et al., 2003) we also learn supervised SVM classifier (LDA-SVM) for each dataset using topics as features and report average Macro-F1.

5.1 Datasets

We use the following datasets in our experiments.

1. **20 Newsgroups:** This dataset contains messages across twenty newsgroups. In our experiments, we use *bydate* version of the 20Newsgroup dataset[1]. This version of the dataset is divided into training (60%) and test (40%) datasets. We construct classifiers on training datasets and evaluate them on test datasets.

2. **SRAA: Simulated/Real/Aviation/Auto UseNet data[2]:** This dataset contains 73,218

[1] http://qwone.com/~jason/20Newsgroups/

[2] http://people.cs.umass.edu/~mccallum/data.html

```
• Input: unlabeled document corpus-$D$, number of topics-$T$ and number of sprinkled terms-$K$

1. Infer $T$ number of topics on $D$ for LDA using collapsed Gibbs sampling. Let $M_D$ be the hidden topic structure of this model.

2. Ask an annotator to assign one or more class labels $c_i \in C$ to a topic based on its 30 most probable words.

3. Initialization: For $n$th word in document $d \in D$
   if $z_{d,n} = t$ and the annotator has labeled topic $t$
   with $c_i$ then, $z_{d,n} = c_i$

4. Update $M_D$ using collapsed Gibbs sampling update in Equation 1.

5. Sprinkling: For each document $d \in D$:
   (a) Infer a probability distribution $\theta_d$ over class labels using $M_D$ using Equation 3.
   (b) Let, $\theta_{c,d}$ be probability of generating document $d$ by class $c$.
   (c) Insert $K * \theta_{c,d}$ distinct words associated with the class $c$ to the document $d$.

6. Infer $|C|$ number of topics on the sprinkled document corpus $D$ using collapsed Gibbs sampling update.

7. Let $M'_D$ be the new hidden topic structure. Let us call this hidden structure as TS-LDA.

8. Classification of an unlabled document $d$
   (a) Infer $\theta'_d$ for document $d$ using $M'_D$.
   (b) $k = \text{argmax}_i \, \theta'_{i,d}$
   (c) $y_d = c_k$
```

Table 1: Algorithm for sprinkling LDA topics for text classification

UseNet articles from four discussion groups, for simulated auto racing (sim_auto), simulated aviation (sim_aviation), real autos (real_auto), real aviation (real_aviation). Following are the three classification tasks associated with this dataset.
1. sim_auto vs sim_aviation vs real_auto vs real_aviation
2. auto (sim_auto + real_auto) vs aviation (sim_aviation + real_aviation)
3. simulated (sim_auto + sim_aviation) vs real (real_auto + real_aviation)
We randomly split SRAA dataset such that 80% is used as training data and remaining is used as test data.
3. **WebKB:** The WebKB dataset[3] contains 8145 web pages gathered from university computer science departments. The task is to classify the webpages as *student, course, faculty* or *project*. We randomly split this dataset such that 80% is used as training and 20% is used as test data.

We preprocess these datasets by removing HTML tags and stop-words.

For various subsets of the 20Newsgroups and WebKB datasets discussed above, we choose number of topics as twice the number of classes. For SRAA dataset we infer 8 topics on the training dataset and label these 8 topics for all the three classification tasks. While labeling a topic, we show its 30 most probable words to the human annotator.

Similar to (Griffiths and Steyvers, 2004), we set symmetric Dirichlet word prior (β_w) for each topic to 0.01 and symmetric Dirichlet topic prior (α_t) for each document to 50/*T*, where T is number of topics. We set *K* i.e. maximum number of words sprinkled per class to 10.

5.2 Results

Table 2 shows experimental results. We can observe that, TS-LDA performs better than ClassifyLDA in 5 of the total 9 subsets. For the *comp-religion-sci* dataset TS-LDA and ClassifyLDA have the same performance. However, ClassifyLDA performs better than TS-LDA for the three classification tasks of SRAA dataset. We can also observe that, performance of TS-LDA is close to supervised LDA-SVM. We should note here that in TS-LDA, the annotator only labels a few topics and not a single document. Hence, our approach exerts a low cognitive load on the annotator, at the same time achieves text classification performance close to LDA-SVM which needs labeled documents.

5.3 Example

Table 3 shows most prominent words of four topics inferred on the *med-space* subset of the 20Newsgroup dataset. We can observe here that most prominent words of the first topic do not represent a single class, while other topics represent either *med (medical)* or *space* class. We can say here that, these topics are not "coherent".

We use these labeled topics and create a TS-LDA model using the algorithm described in Table 1. Table 4 shows words corresponding to the top two topics of the TS-LDA model. We can observe here that these two topics are more coherent than the topics in Table 3.

[3] `http://www.cs.cmu.edu/~webkb/`

		Text Classification (Macro-F1)		
Dataset	# Topics	ClassifyLDA	TS-LDA	LDA-SVM
20Newsgroups				
med-space	4	0.892	0.938	0.933
politics-religion	4	0.836	0.897	0.901
politics-sci	4	0.887	0.901	0.910
comp-religion-sci	6	0.853	0.853	0.872
politics-rec-religion-sci	8	0.842	0.858	0.862
SRAA				
real_auto-real_aviation-sim_auto-sim_aviation	8	0.766	0.741	0.820
auto-aviation	8	0.926	0.910	0.934
real-sim	8	0.918	0.902	0.923
WebKB				
WebKB	8	0.627	0.672	0.730

Table 2: Experimental results of text classification on various datasets.

ID	**Most prominent words in the topic**	**Class (med / space)**
0	science scientific idea large theory bit pat thought problem isn	**med + space**
1	information **health** research **medical** water **cancer hiv aids** children institute newsletter	**med**
2	msg **food doctor disease pain** day **treatment blood** steve dyer **medicine symptoms**	**med**
3	**space nasa launch earth orbit moon shuttle** data **lunar satellite**	**space**

Table 3: Topic labeling on the *med-space* subset of the 20Newsgroup dataset

ID	**Most prominent words in the topic**	**Class (med / space)**
0	msg **medical health food disease** years problem information **doctor pain cancer**	**med**
1	**space launch earth** data **orbit moon** program **shuttle lunar satellite**	**space**

Table 4: Topics inferred on the *med-space* subset of the 20Newsgroup dataset after sprinkling labeled topics from Table 3.

Hence, we can say here that, in addition to text classification, sprinkling improves coherence of topics.

We should note here that, in ClassifyLDA, the annotator is able to assign a single class label to a topic. If the annotator assigns a wrong class label to a topic representing multiple classes (e.g. first topic in Table 3), then it may affect the performance of the resulting classifier. However, in our approach the annotator can assign multiple class labels to a topic, hence our approach is more flexible for the annotator to encode her domain knowledge efficiently.

6 Conclusions and Future Work

In this paper we propose a novel algorithm that classifies documents based on class labels over few topics. This reduces the need to label a large collection of documents. We have used the idea of sprinkling originally proposed in the context of supervised Latent Semantic Analysis, but the setting here is quite different. Unlike the work in (Chakraborti et al., 2007), we do not assume that we have class labels over the set of training documents. Instead, to realize our goal of reducing knowledge acquisition overhead, we propose a way of propagating knowledge of few topic labels to the words and inducing a new topic distribution that has its topics more closely aligned to the class labels. The results show that the approach can yield performance comparable to entirely supervised settings. In future work, we also envision the possibility of sprinkling knowledge from background knowledge sources like Wikipedia (Gabrilovich and Markovitch, 2007) to realize an alignment of topics to Wikipedia concepts. We would like to study effect of change in number of topics on the text classification performance. We will also explore techniques which will help annotators to encode their domain knowledge efficiently when the topics are not well aligned to the class labels.

References

David M. Blei and Jon D. McAuliffe. 2007. Supervised Topic Models. In *NIPS*.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3:993–1022, March.

Avrim Blum and Tom Mitchell. 1998. Combining Labeled and Unlabeled Data with Co-Training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100.

Sutanu Chakraborti, Rahman Mukras, Robert Lothian, Nirmalie Wiratunga, Stuart N. K. Watt, David J. Harper. 2007. Supervised Latent Semantic Indexing Using Adaptive Sprinkling. In *IJCAI*, pages 1582-1587.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by Latent Semantic Analysis. *JASIS*, 41(6):391–407.

Gregory Druck, Gideon Mann, and Andrew McCallum. 2008. Learning from Labeled Features using Generalized Expectation criteria. In *SIGIR*, pages 595–602.

Gregory Druck, Burr Settles, and Andrew McCallum. 2009. Active Learning by Labeling Features. In *EMNLP*, pages 81–90.

Shantanu Godbole, Abhay Harpale, Sunita Sarawagi, and Soumen Chakrabarti. 2004. Document Classification through Interactive Supervision of Document and Term Labels. In *PKDD*, pages 185–196.

Thomas L. Griffiths and Mark Steyvers. 2004. Finding Scientific Topics. *PNAS*, 101(suppl. 1):5228–5235, April.

Swapnil Hingmire, Sandeep Chougule, Girish K. Palshikar, and Sutanu Chakraborti. 2013. Document Classification by Topic Labeling. In *SIGIR*, pages 877–880.

Thorsten Joachims. 1999. Transductive Inference for Text Classification using Support Vector Machines. In *ICML*, pages 200–209.

April Kontostathis and William M. Pottenger. 2006. A Framework for Understanding Latent Semantic Indexing (LSI) Performance. *Inf. Process. Manage.*, 42(1):56–73, January.

Simon Lacoste-Julien, Fei Sha, and Michael I. Jordan. 2008. DiscLDA: Discriminative Learning for Dimensionality Reduction and Classification. In *NIPS*.

Sangno Lee, Jeff Baker, Jaeki Song, and James C. Wetherbe. 2010. An Empirical Comparison of Four Text Mining Methods. In *Proceedings of the 2010 43rd Hawaii International Conference on System Sciences*, pages 1–10.

Bing Liu, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. 2004. Text Classification by Labeling Words. In *Proceedings of the 19th national conference on Artifical intelligence*, pages 425–430.

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning - Special issue on information retrieval*, 39(2-3), May-June.

Hema Raghavan, Omid Madani, and Rosie Jones. 2006. Active Learning with Feedback on Features and Instances. *JMLR*, 7:1655–1686, December.

Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from Labeled and Unlabeled Data with Label Propagation. Technical report, Carnegie Mellon University.

Jun Zhu, Amr Ahmed, and Eric P. Xing. 2009. MedLDA: Maximum Margin Supervised Topic Models for Regression and Classification. In *ICML*, pages 1257–1264.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *IJCAI*, pages 1606–1611.