Document Classification by Topic Labeling

Swapnil Hingmire[*†] swapnil.hingmire@tcs.com
Sandeep Chougule[*] sandeep.chougule@tcs.com
Girish K. Palshikar[*] gk.palshikar@tcs.com
Sutanu Chakraborti[†] sutanuc@cse.iitm.ac.in

[*]Systems Research Lab
Tata Research Development and Design Center
Tata Consultancy Services
Pune-13, India

[†]Department of Computer Science and Engineering
IIT Madras
Chennai-36, India

ABSTRACT

In this paper, we propose Latent Dirichlet Allocation (LDA) [1] based document classification algorithm which does not require any labeled dataset. In our algorithm, we construct a topic model using LDA, assign one topic to one of the class labels, aggregate all the same class label topics into a single topic using the aggregation property of the Dirichlet distribution and then automatically assign a class label to each unlabeled document depending on its "closeness" to one of the aggregated topics.
We present an extension to our algorithm based on the combination of Expectation-Maximization (EM) algorithm and a naive Bayes classifier. We show effectiveness of our algorithm on three real world datasets.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*

General Terms

Experimentation, Performance, Theory, Verification

Keywords

Expectation-Maximization, Text classification, Topic Modelling

1. INTRODUCTION

With the advent of cheap and fast storage, there is an explosive growth in the size and number of documents available in electronic format. Document classification is a technique which helps users to make effective use of the knowledge hidden in the documents.
Traditional supervised document classifiers require a large number of labeled dataset. Many times obtaining such a labeled dataset is expensive. Nigam et al. [5] proposed semi-supervised approaches for document classification based on labeled and unlabeled datasets. McCallum and Nigam [4] proposed a semi-supervised approach based on labeling of keywords. In keyword based approaches, finding right set of keywords is a challenge.
In this paper, we propose Latent Dirichlet Allocation (LDA) [1] based document classification algorithm. Our algorithm does not require any labeled dataset. In our algorithm, we construct a topic model using LDA, assign one topic to one of the class labels, aggregate all the same class label topics into a single topic using the aggregation property of the Dirichlet distribution and then automatically assign a class label to each unlabeled document depending on its "closeness" to one of the aggregated topics.
In our algorithm an expert assigns one topic to one of the class labels, also as LDA topics correlate with human assigned class labels [6], our algorithm exerts a low cognitive load on the expert.
Class labels predicted by our algorithm may be approximate or noisy. In order to reduce the influence of such an approximate or noisily labeled documents, we present an extension to our algorithm based on the combination of the Expectation-Maximization (EM) algorithm and a naive Bayes classifier. We show effectiveness of our algorithm on three real world datasets.
The paper is organized as follows: In section 2 we give brief introduction to LDA and the Dirichlet distribution. Section 3 contains our document classification algorithm. Section 4 demonstrates effectiveness of our algorithm with experiments on three real world datasets. We end our paper with conclusions and future prospects of our work in section 5.

2. LATENT DIRICHLET ALLOCATION (LDA)

LDA is an unsupervised generative probabilistic model for collections of discrete data such as text documents. In LDA, each document is generated by choosing a distribution over topics and then choosing each word in the document from a topic selected according to the distribution [3]. Generative process of LDA can be described as follows:

1. for $t = 1...T$
   (a) $\phi_t \sim \text{Dirichlet}(\beta)$

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

2. for each document $d \in D$

   (a) $\theta_d \sim$ Dirichlet(α)

   (b) for each word w at position n in d

      i. $z_d^n \sim$ Multinomial(θ_d)

      ii. $w_d^n \sim$ 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 and β are topic and word Dirichlet priors respectively.

Training an LDA model is estimation of the word-topic distributions and the topic distributions for all documents in the corpus. Direct and exact estimation of these parameters is intractable. Collapsed Gibbs sampling is one of the techniques used for the parameter estimation of LDA [3]. After performing collapsed Gibbs sampling, probability of the word w assigned to the topic t ($\phi_{w,t}$) and the probability of the topic t assigned to document the d ($\theta_{t,d}$) is estimated as:

$$\phi_{w,t} = \frac{\psi_{w,t}+\beta_w}{\left[\sum_{v \in W} \psi_{v,t}+\beta_v\right]} \quad \theta_{t,d} = \frac{\Omega_{t,d}+\alpha_t}{\left[\sum_{i=1}^{T} \Omega_{i,d}+\alpha_i\right]} \tag{1}$$

Where $\psi_{w,t}$ is the count of the word w assigned to the topic t, $\Omega_{t,d}$ is the count of the topic t assigned to words in the document d and W is the vocabulary of the corpus.

LDA discovers a set of topics present in the documents and gives probabilities of observing each word in each topic. Most prominent words in a topic frequently co-occur with each other in the documents so one can infer context of the words in a topic. Using the word probabilities one can interpret meaning of topics and find major themes in the documents. The topic probabilities of a document provide its explicit representation and these probabilities can be embedded in more complex model.

2.1 The Dirichlet distribution

The Dirichlet distribution is defined as:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{t=1}^{T} \alpha_t)}{\prod_{t=1}^{T} \Gamma(\alpha_t)} \prod_{t=1}^{T} \theta_t^{\alpha_t - 1} \tag{2}$$

where, $\theta = \{\theta_1, ..., \theta_t, ..., \theta_T\}$ is a point on the $(T-1)$ simplex (i.e. $0 < \theta_t < 1$ and $\sum_{t=1}^{T} \theta_t = 1$) and $\alpha = (\alpha_1, ..., \alpha_t, ..., \alpha_T)$ is a set of parameters with $\alpha_t > 0$. So,

$$\theta = (\theta_1, ..., \theta_t, ..., \theta_T) \sim \text{Dirichlet}(\alpha_1, ..., \alpha_t, ..., \alpha_T) \tag{3}$$

Aggregation property of the Dirichlet distribution: The Dirichlet distribution has a fractal like aggregation property [2]. It is defined as the aggregation of any subset of Dirichlet distribution variable yields a Dirichlet distribution, with corresponding aggregation of the parameters. If $\{A_1, A_2, ..., A_r\}$ is a partition of $\{1, 2, ..., T\}$ then,

$$\theta' = (\sum_{t \in A_1} \theta_t, \sum_{t \in A_2} \theta_t, ..., \sum_{t \in A_r} \theta_t) \sim \text{Dirichlet}(\sum_{t \in A_1} \alpha_t, \sum_{t \in A_2} \alpha_t, ..., \sum_{t \in A_r} \alpha_t) \tag{4}$$

3. DOCUMENT CLASSIFICATION

In this section, we propose our document classification algorithm based on LDA (ClassifyLDA) and an extension of the algorithm based on the combination of EM algorithm and a naive Bayes classifier (ClassifyLDA-EM).

3.1 ClassifyLDA

Our algorithm is based on generative property of LDA and the aggregation property of the Dirichlet distribution. Let us assume, we want to classify each document to one of the class labels from $C = \{1, 2, ..., m\}$. Using Collapsed Gibbs sampling for LDA, $Z = \{z_1, z_2, ..., z_T\}$ topics are learnt on the document corpus D. Now an expert will assign a class label, $i \in C$ to each topic $z_t \in Z$ based on its most prominent words. Create $Z' = \bigcup_{i=1}^{m} Z^i$, the partition of Z such that $Z^i = \{z_t | z_t \in Z$ and class label of z_t is $i\}$.

If for a document d in the corpus D, $\theta_d = (\theta_{1,d}, \theta_{2,d}, ..., \theta_{T,d}) \sim \text{Dirichlet}(\alpha_1, \alpha_2, ..., \alpha_T)$ then using the aggregation property of the Dirichlet distribution define θ'_d as:

$$\theta'_d = (\sum_{z_t \in Z^1} \theta_{t,d}, \sum_{z_t \in Z^2} \theta_{t,d}, ..., \sum_{z_t \in Z^m} \theta_{t,d}) \sim \text{Dirichlet}(\sum_{z_t \in Z^1} \alpha_t, \sum_{z_t \in Z^2} \alpha_t, ..., \sum_{z_t \in Z^m} \alpha_t) \tag{5}$$

Initialize ϕ'_t and θ'_d using following equations:

$$\psi_{w,Z^i} = \sum_{t \in Z^i} [\psi_{w,t} + \beta_w] \quad \phi'_{w,Z^i} = \frac{\psi_{w,Z^i}}{\sum_{v \in W} \psi_{v,Z^i}} \tag{6}$$

$$\Omega_{Z^i,d} = \sum_{t \in Z^i} [\Omega_{t,d} + \alpha_t] \quad \theta'_{Z^i,d} = \frac{\Omega_{Z^i,d}}{\sum_{k \in C} \Omega_{Z^k,d}} \tag{7}$$

Using Collapsed Gibbs sampling for LDA, update ϕ'_t and θ'_d. A class label $c \in C$ is assigned to document $d \in D$ such that: $c = \arg\max_i \theta'_{Z^i,d}$.

Algorithm 1 describes our for document classification algorithm .

Algorithm 1: ClassifyLDA

input : D = $\{d\}$: Document corpus
output: Class label (d_c) from $C = \{1, 2, ..., m\}$ for each document $d \in D$

```
begin
    Use LDA to learn Z = {z_1, ..., z_T} topics on D;
    Compute φ_t and θ_d using equations 1 ;
    Expert will assign a class label, i ∈ C to each
    topic z_t ∈ Z based on its most prominent words;
    Create a partition Z' = ⋃_{i=1}^{m} Z^i such that:
    Z^i = {z_t | z_t ∈ Z and class label of z_t is i};
    Initialize φ'_t and θ'_d using equations 6 and 7;
    Update φ'_t and θ'_d using Collapsed Gibbs
    sampling;
    for d ∈ D do
        Infer θ'_{Z^i,d}; ∀i ∈ C using equation 7 ;
        d_c = arg max_i θ'_{Z^i,d} ;
    end
end
```

3.2 ClassifyLDA-EM

In ClassifyLDA-EM algorithm, we build a classifier using the combination of EM and a naive Bayes classifier. In this algorithm we use EM iterations along with the relation between word co-occurrence knowledge and class labels to improve the parameters of a naive Bayes classifier.

Initially, we label all the unlabeled documents in the corpus using ClassifyLDA algorithm described in algorithm 1. Then, we build a naive Bayes classifier using these labeled documents and estimate class probabilities for each document. Using these estimated class probabilities we reassign a class label to each document and rebuild a new naive Bayes classifier.
We iterate this process of reassigning class labels to the documents and rebuilding a naive Bayes classifier until it converges to a stable classifier. We say a classier is stable when the change in log likelihood of the parameters of the classier is below a threshold. ClassifyLDA-EM can be described as:

- **Input:** D = $\{d\}$: Unlabeled document corpus
- **Initialization:** Let $\hat{C}$ be an initial classifier, built using ClassifyLDA algorithm. Assign a class label to each unlabeled document in D using ClassifyLDA.
- Loop while $\hat{C}$ converges:
  - **E-step:** Use the current classifier, $\hat{C}$, to estimate the probability of a document belonging to each class.
  - **M-step:** Re-estimate the classifier, $\hat{C}$ using naive Bayes model based on the document-class probabilities computed in E-step
- Use $\hat{C}$ to classify an unlabeled document.

4. EXPERIMENTAL EVALUATION

We determine the effectiveness of our algorithm in relation to semi-supervised text classification algorithm proposed in [5] (NB-EM). We report the minimum number of labeled documents at which the performance of ClassifyLDA-EM and NB-EM are almost similar.

4.1 Datasets

We evaluate the effectiveness of ClassifyLDA and ClassifyLDA-EM on following three real world text classification datasets.
1. 20Newsgroup: This dataset contains messages across twenty newsgroups. In our experiments, we use *bydate* version of the 20Newsgroup dataset[1]. This version contains separate train and test datasets of 20 newsgroups which are grouped into 6 major categories. We selected 4 major categories: comp, politics, rec, and religion. Following are the newsgroups in each selected category.

1. **comp:** comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x
2. **politics:** talk.politics.misc, talk.politics.guns, talk.politics.mideast
3. **rec:** rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey
4. **religion:** talk.religion.misc, alt.atheism, soc.religion.christian

We experimented with all possible combinations of these major categories.
2. SRAA: Simulated/Real/Aviation/Auto UseNet data[2]: This dataset contains 73,218 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)

3. WebKB [3]: This dataset contains 4199 university webpages. The task is to classify the webpages as *student, course, faculty* or *project*.
We randomly split SRAA and WebKB datasets such that 80% is used as training data and remaining 20% is used as test data.

4.2 Experimental settings

We did preprocessing on the dataset by removing headers and stopwords. We evaluated effectiveness of our algorithm by computing the Macro F-measure (F_1).
For classification tasks of 20Newsgroup related dataset we choose number of topics (T) equal to two times number of classes. For SRAA dataset we learnt 10 topics on the complete dataset and labeled these 10 topics for all the three classification tasks. For WebKB dataset we learnt 10 topics. The Dirichlet parameter β was chosen to be 0.01 and α was 50/T. We used Mallet[4] to run LDA on the documents.
In NB-EM algorithm, we do 10 trails per number of labeled documents and report average Macro F1.

4.3 Results

Table 1 shows experimental results. We can observe that, ClassifyLDA-EM algorithm can achieve almost similar performance in relation to NB-EM with significant reduction in labeling efforts and for most of the datasets performance of our algorithm is above 0.9. In table 1, we can also observe improvement in the performance of ClassifyLDA-EM over ClassifyLDA which proves that the combination of EM and a naive Bayes classifier reduces the influence of approximate or noisily labeled documents.
We observed that, performance of NB-EM depends on initial labeled documents.

4.4 Example

Table 2 shows topics learnt and classification of the topics on "politics vs rec" dataset. With the help of most prominent words in a topic an expert can assign a class label to the topic. Due to generative property of LDA, topics labeled with class label "politics" will generate politics related documents with high probability. Now we will create the partition $Z' = \{\{z_0, z_1\}, \{z_2, z_3\}\}$ for the topics $Z = \{z_0, z_1, z_2, z_3\}$. Using the aggregation property of the Dirichlet distribution, all the same class label topics are aggregated into a single topic. Now, we can use algorithm 1 and the combination of EM algorithm and a naive Bayes classifier to estimate class labels for unseen documents.
We also explored how well a topic correlates with the class

[1] http://qwone.com/~jason/20Newsgroups/
[2] http://people.cs.umass.edu/~mccallum/data.html
[3] http://www.cs.cmu.edu/~webkb/
[4] http://mallet.cs.umass.edu/

Data set	ClassifyLDA (Macro-F1)	ClassifyLDA-EM (Macro-F1)	# Topics	NB-EM (Macro-F1)	# Labeled documents for NB-EM
20Newsgroup					
comp vs politics	0.960	0.976	4	0.974	20
comp vs rec	0.903	0.949	4	0.947	25
comp vs religion	0.953	0.979	4	0.981	25
politics vs rec	0.957	0.980	4	0.978	70
politics vs religion	0.872	0.929	4	0.927	65
rec vs religion	0.959	0.988	4	0.986	105
comp vs politics vs rec	0.932	0.960	6	0.960	125
comp vs politics vs religion	0.896	0.932	6	0.929	115
comp vs rec vs religion	0.936	0.965	6	0.964	105
politics vs rec vs religion	0.889	0.937	6	0.935	190
comp vs politics vs rec vs religion	0.891	0.936	8	0.934	480
SRAA					
sim_auto vs sim_aviation vs real_auto vs real_aviation	0.732	0.770	10	0.786	10250
auto vs aviation	0.908	0.929	10	0.927	300
simulated vs real	0.917	0.933	10	0.931	5250
WebKB					
student vs course vs faculty vs project	0.711	0.719	10	0.730	1150

Table 1: Experimental results (Macro-F1) of document classification on 20Newsgroup, SRAA and WebKB datasets

ID	Most prominent words in the topic	Class
0	gun armenian turkish didn guns killed file weapons armenia	politics
1	israel government president jews american fact question law case rights	politics
2	team game play season hockey players win league baseball	rec
3	car bike front road buy drive speed engine	rec

Table 2: Topic labeling on the politics vs rec dataset

assigned to it. We represented each class as probability distribution over words. We computed $P(w|c_j)$, the probability of the word w belonging to the class c_j as the fraction of the number of times word w appears among all the words in documents of class c_j. Then we computed Kullback-Leibler (K-L) divergence between each class and a topic. Table 3 shows K-L divergence between each class and a topic for the same dataset. We can observe that the K-L divergence is least for the class assigned to a topic by the expert.

Topic-class mapping		Class labels	
ID	Expert assigned class label	politics	rec
0	politics	**3.89**	6.12
1	politics	**3.57**	6.12
2	rec	6.64	**3.73**
3	rec	6.13	**4.32**

Table 3: K-L Divergence between each class and a topic for the politics vs rec dataset

5. CONCLUSIONS

In this paper, we propose a novel, inexpensive document classification algorithm which requires minimal supervision. Our algorithm is based on the generative property of LDA and the aggregation property of the Dirichlet distribution. We also show effectiveness of our algorithm with the help of experiments. Our approach is specifically suited for domains where establishing a mapping from topics to class labels is easier than acquiring a labeled collection of documents. In future we would like to carry out experiments on datasets like Reuters-21578 and a more detailed investigation on how the topic-class mapping influences the classification effectiveness. We will also explore tools that help experts arrive at the most appropriate topic-class mapping.

6. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, March 2003.

[2] B. A. Frigyik, A. Kapila, and M. R. Gupta. Introduction to the dirichlet distribution and related processes. Technical report. University of Washington, Seattle, 2012. `https://www.ee.washington.edu/techsite/papers/documents/UWEETR-2010-0006.pdf`

[3] T. L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 101(suppl. 1):5228–5235, April 2004.

[4] A. Mccallum and K. Nigam. Text classification by bootstrapping with keywords, EM and shrinkage. In *ACL-99 Workshop for Unsupervised Learning in Natural Language Processing*, pages 52–58, 1999.

[5] 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*, 39(2-3), May-June 2000.

[6] A. Chanen and J. Patrick. Measuring Correlation Between Linguists' Judgments and Latent Dirichlet Allocation Topics. *Proceedings of the Australasian Language Technology Workshop*, pages 13–20, 2007.