

A Robust Learning Approach for Text Classification

Viet Ha-Thuc^{*}

Padmini Srinivasan^{†*}

Abstract

Previous learning approaches often assume that every part of a positive training document of a class is relevant to that class. However, in practice, it is often the case that only one or a few parts in the training document are really relevant to the class. To overcome this limitation, we propose another learning approach based on relevance-based topic model, an extension of well-known Latent Dirichlet Allocation. In this approach, the real relevant parts in each document are automatically determined by its statistical correlation to the rest of the positive training set. And only these parts contribute to the final results. Therefore, the approach is robust to “impurities” in the training sets. In addition, the approach exploits the “bag-of-words” assumption to re-arrange words in an appropriate order that could reduce the computational complexity of learning algorithm.

Keywords

Probabilistic topic models, LDA, Gibbs sampling, text classification

1 Introduction

Machine learning techniques are popular in text classification. However, most of previous approaches, including both supervised ([9][7]) and semi-supervised learning ([11]), assume that every part of a positive training document of a class is relevant to that class. This assumption is, nonetheless, not true in many cases. For instance, when one builds a training set for topic “*machine learning*”, a positive example could be a paper about *speech recognition* that uses some machine learning technique and another positive document could be an overview article about *artificial intelligence* that contains the term “machine learning”. Therefore, in such cases only one or a few parts in the document are really about machine learning.

On the other hand, probabilistic Latent Semantic Indexing (pLSI) ([6]) and more recent probabilistic topic models ([2][15]), also known as Latent Dirichlet Allocation (LDA), provide a powerful theoretical

framework supporting the idea that documents are mixtures of topics. Here topics represented by multinomial distributions over words are discovered from the corpus. However, the resulting topics are synthetic and might not match well with the prior classification knowledge of human beings. Therefore, the approach could not be applied directly for text classification, where topics (classes) are provided upfront.

In this paper, we propose a relevance-based topic model, an extension of probabilistic topic models, in which the relevance between a class and each of its positive training documents is taken into account. In this approach, we assume that a positive training document d of a class t is generated by a mixture of three topics: topic t corresponding to the class itself, a background topic, b , which captures words that are common in general and common in the particular domain, and $t_o(d)$ (other topics in the document) capturing non-relevant parts in the document. Only the parts generated by t (the really relevant parts) contribute to the resulting distribution over words for topic t . And this distribution is the key parameter that needs to be learned in generative text classification methods ([9]). Therefore, the proposed approach is robust common words and noise in the training documents.

The contributions of topics in documents are automatically determined by intra- and inter-document statistics. Intuitively, the words frequently appearing in the whole corpus are likely generated by the background topic, the words frequently appearing in the training set of a topic but not the whole corpus are likely generated by the topic, and the words appearing in only a particular document d but not in the rest of the training set are likely generated by $t_o(d)$.

Like previous work in probabilistic topic models ([5][13][15]), we also use Gibbs sampling, a form of Markov chain Monte Carlo technique, to learn the topic distributions and mixture weights. In our learning approach, we propose the idea of exploiting the assumption of exchangeability for the tokens in a document (“bag-of-words” assumption) in order to reduce the computational complexity of the learning algorithm. Since token order in a document is not considered, we could re-arrange the tokens in any order that is convenient for the learning algorithm. In

^{*} Computer Science Department – The University of Iowa, Email: hviet@cs.uiowa.edu

[†] School of Library and Information Science – The University of Iowa, Email: padmini-srinivasan@uiowa.edu

our case, we group the tokens having the same stem into continuous segments because the topics of the tokens are sampled from the same distribution. That helps reduce the running time of sampling process.

The rest of the paper is organized as follows. In Section 2, we introduce our relevance-based topic model, whose purpose is learning topic-word multinomial distributions. In Section 3, we describe in detail the inference algorithm for the model. Section 4 presents an empirical result illustrating some features of the proposed model. Section 5 shows the efficiency of the model in text classification. Then, the related work is reviewed in Section 6. Finally, Section 7 is our concluding remarks and future directions.

2 Relevance-based Topic Model

We first introduce some basic notation. Then, we describe our relevance-based topic model

2.1 Notation

- Set of K topics: $T = \{1, 2 \dots K\}$ including a set of original topics T_0 to which we need to classify, a background topic b , and topics $t_o(d)$ covering non-relevant parts in each document d

- Vocabulary set (dictionary) containing W words: $V = \{\text{word}_1, \text{word}_2 \dots \text{word}_w\}$

- A *token* is a specific occurrence of one of the W words in a document. A *document* d is a sequence of N_d tokens. A *corpus* C is a concatenation of training sets of topics in T_0 . It contains a collection of D documents denoted by $C = \{(w_1, d_1), (w_2, d_2) \dots (w_N, d_N)\}$, where $N = \sum_{d=1}^D N_d$, and for every t ($1 \leq t \leq N$), w_t is the word index of the t th token, d_t is the index of document in which the token appears.

For example, assume that there are two documents in the corpus: $doc_1 = \{\text{"apple"}, \text{"orange"}\}$ and $doc_2 = \{\text{"fruit"}, \text{"apple"}\}$, then

$$V = \{\text{"apple"}, \text{"orange"}, \text{"fruit"}\}.$$

$C = \{(1, 1), (2, 1), (3, 2), (1, 2)\}$, where $w_1=1, d_1=1, w_2=2, d_2=1, w_3=3, d_3=2, w_4=1, d_4=2$.

It is worth noting that in the case topics in T_0 are overlapping, a document could appear in several training sets. For instance, a paper could be relevant to both of topics “machine learning” and “text classification”. In such case, the document has several duplicates in the corpus. However, each of the duplicates plays a different role contributing different parts to the topic to which it belongs.

In this paper, vectors are denoted by bold letters such as \mathbf{x} , and single variables are denoted by un-bold letters such as x .

2.2 Model description

Relevance-based topic model is a probabilistic generative model describing the process of generating training documents in the corpus as described bellows.

- 1) Pick a multinomial distribution Φ_b for background topic from W -dimensional Dirichlet distribution with parameter β ($\text{Dir}(\beta)$)
- 2) For each topic t in T_0 :
 - a) Pick a multinomial distribution Φ_t for t from W -dimensional $\text{Dir}(\beta)$
 - b) For each document d relevant to t :
 - i) Pick a multinomial distribution $\Phi_{t_o(d)}$ for topics other than t that are also covered in d from W -dimensional $\text{Dir}(\beta)$
 - ii) Pick a multinomial distribution θ_d from 3-dimensional $\text{Dir}(\alpha)^\ddagger$, each element of θ_d corresponds to a topic in $\mathbf{x}_d = \{b, t, t_o(d)\}$.
 - iii) For each token in document d , pick a topic z among the three topics in \mathbf{x}_d from multinomial θ_d . Then, pick a word from multinomial distribution Φ_z .

This process is described by the graphical model using plate notation in Fig. 1. In the Figure, w and \mathbf{x}_d are observable variables and denoted by shaded circles; z , θ and Φ are hidden variables and denoted by un-shaded variables, where $\theta_{i,j} = p(\text{topic}=j \mid \text{doc}_i)$

($1 \leq i \leq D, 1 \leq j \leq K, \sum_{j=1}^K \theta_{i,j} = 1$) is probability of topic j in

the i th document, $\Phi_{j,k} = p(\text{word}_k \mid \text{topic}=j)$ ($1 \leq j \leq K,$

$1 \leq k \leq W, \sum_{k=1}^W \Phi_{j,k} = 1$) is probability of the k th word

given topic j ; α, β are parameters of Dirichlet distributions, and like most of previous work in probabilistic models, we also treat α, β as constants. The plates (boxes) indicate repetitions of generation process. The number at lower right corner of each plate is the number iterations.

In the context of learning for text classification, the variables of direct interest are topic-word distributions Φ_j ($1 \leq j \leq K$). Therefore, a key purpose for the training phase is to infer posterior distributions of

[‡] For mathematical convenience, we represent θ_d as K -dimensional vector in which elements corresponding to topics that are not in \mathbf{x}_d are set to zero.

each Φ_j , given the observable variables and computing the expected value Φ_j^* , w.r.t. the posterior distributions. The details of the inference algorithm are described in next section.

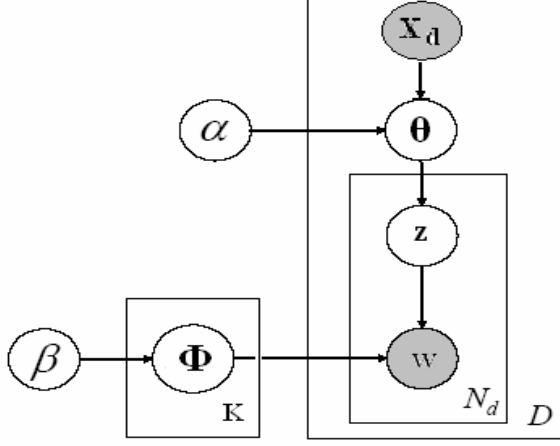


Figure 1 Relevance-based topic model

3 Inference Algorithm

As in recent work on probabilistic models ([4][5][13][15][19]), we also apply Gibbs sampling for inference. In the next subsection, we introduce Gibbs sampling for inference in general Bayesian networks (directed acyclic graphs). Then, we present our adapted version for relevance-based topic model.

3.1 Gibbs sampling

Gibbs sampling is particularly suitable for inference problem in Bayesian networks (Andrieu et al. [1]). Assume $\mathbf{x}=\{x_1, x_2 \dots x_n\}$ is a set of hidden variables and $\mathbf{e}=\{e_1, e_2 \dots e_m\}$ is the set of evidence (observable) variables of a Bayesian network. The inference problem is to compute posterior distribution $p(\mathbf{x}|\mathbf{e})$ where \mathbf{x}' is any subset of \mathbf{x} . Instead of computing the posterior distribution directly, Gibbs sampling approach efficiently generates a sequence of L samples $(x_1^{(l)}, x_2^{(l)} \dots x_n^{(l)})$ ($1 \leq l \leq L$), then uses the sequence to estimate the posterior distribution ([3]). Gibbs sampling algorithm is formally described in Fig. 2.

In Bayesian networks, the conditional probability in Fig. 2 is factorized as follows:

$$p(x_j | \mathbf{x}_{-j}) \propto p(x_j | x_{pa(j)}) \prod_{k \in ch(j)} p(x_k | x_{pa(k)}) \quad (3.1)$$

In (3.1), x_{-j} denotes other nodes except x_j ; $x_{pa(j)}$ denotes parent nodes of x_j and $ch(j)$ denotes children nodes of x_j . When computing the conditional distribution of x_j , we only need to take into account its parents, its children and its children's parents (co-

parents). This set of variables is often referred as *Markov blanket* of x_j .

1. Initialize: Assign each x_i to one of its possible values $x_i^{(0)}$ ($1 \leq i \leq n$)
2. For $l=0$ to $(L-1)$
 - Sample $x_1^{(l+1)} \sim p(x_1 | x_2^{(l)}, x_3^{(l)} \dots x_n^{(l)}, \mathbf{e})$
 - Sample $x_2^{(l+1)} \sim p(x_2 | x_1^{(l+1)}, x_3^{(l)} \dots x_n^{(l)}, \mathbf{e})$
 - ...
 - Sample $x_j^{(l+1)} \sim p(x_j | x_1^{(l+1)} \dots x_{j-1}^{(l+1)}, x_{j+1}^{(l)} \dots x_n^{(l)}, \mathbf{e})$
 - ...
 - Sample $x_n^{(l+1)} \sim p(x_n | x_1^{(l+1)}, x_2^{(l+1)} \dots x_{n-1}^{(l+1)}, \mathbf{e})$

Figure 2: Gibbs sampling algorithm

When $l \rightarrow \infty$, the distribution of $(x_1^{(l)}, x_2^{(l)} \dots x_n^{(l)})$ is theoretically proved to converge to $p(\mathbf{x}|\mathbf{e})$ ([3]). So, if we select a large enough number p , the distribution of $(x_1^{(p)}, x_2^{(p)} \dots x_n^{(p)})$ is close to $p(\mathbf{x}|\mathbf{e})$ ($\forall i \geq k$). Then, from the sequence of L samples, we select q samples at positions $p, p+r, p+2r \dots p+q*r$ ($r > 1, p+q*r \leq L$) to approximately compute $p(\mathbf{x}'|\mathbf{e})$ or its properties (e.g. expected value). An alternative way is running the sampling algorithm q times with $L = p$ to get q Gibbs sample sequences, and selecting every p th sample (the last one) from each of the q sequences.

3.2 Inference in relevance-based topic model

Our relevance-based topic model is a special case of Bayesian networks. As we indicated in the previous subsection, the key step in Gibbs sampling is estimating the conditional distribution of each unknown variable given its Markov blanket.

The Markov blanket of z_i ($MB(z_i)$), the latent topic variable of token $w_i=k$ which appears in document $d_i=i$, includes θ_i, Φ and w_i .

$$p(z_i=j | \theta_i, \Phi, w_i=k) \propto p(z_i=j | \theta_i) p(w_i=k | z_i=j, \Phi) \propto \theta_{i,j} \Phi_{j,k}$$

Note that, if $j \notin \mathbf{x}_i$, then $\theta_{i,j} = 0$ and $p(z_i=j | \theta_i, \Phi, w_i=k) = 0$.

The Markov blanket of $\theta_i = (\theta_{i,1}, \theta_{i,2} \dots \theta_{i,k})$, the prior probabilities of topics in the i th document, includes α, \mathbf{x}_i and \mathbf{z} .

$$p(\theta_i | \alpha, \mathbf{x}_i, \mathbf{z}) \propto p(\theta_i | \alpha, \mathbf{x}_i) p(\mathbf{z} | \theta_i)$$

Non-zero elements in θ_i : $(\theta_{i,x_i(1)}, \theta_{i,x_i(2)}, \theta_{i,x_i(3)})$

follows Dir(α) distribution. Therefore:

$$p(\theta_i | \alpha, \mathbf{x}_i) \propto \theta_{i,x_i(1)}^{(\alpha-1)} \theta_{i,x_i(2)}^{(\alpha-1)} \theta_{i,x_i(3)}^{(\alpha-1)}$$

Assume $n_{i,j}$ ($1 \leq j \leq K$) is the number of times that the topic j is assigned to some token in the i th document in the current sample, then:

$$p(\mathbf{z} | \boldsymbol{\theta}_i) \propto \theta_{i,x_i(1)}^{n_{x_i(1)}} \theta_{i,x_i(2)}^{n_{x_i(2)}} \theta_{i,x_i(3)}^{n_{x_i(3)}}$$

$$p(\boldsymbol{\theta}_i | \alpha, \mathbf{x}_i, \mathbf{z})$$

$$\propto \theta_{i,x_i(1)}^{n_{x_i(1)} + \alpha - 1} \theta_{i,x_i(2)}^{n_{x_i(2)} + \alpha - 1} \theta_{i,x_i(3)}^{n_{x_i(3)} + \alpha - 1}$$

We see that the posterior distribution of $\boldsymbol{\theta}_i$ is also a Dirichlet distribution. Like previous work on topic models ([4][5][13][15][19]), for the sake of efficiency, instead of sampling $\boldsymbol{\theta}_i$ from its posterior, we deterministically assign it to the most probable values (*mode* of the posterior distribution):

$$\theta_{i,x_i(j)} = \frac{n_{i,x_i(j)} + \alpha - 2}{\sum_{j'=1}^3 (n_{i,x_i(j')} + \alpha - 2)}$$

$$= \frac{n_{i,x_i(j)} + \alpha'}{\sum_{j'=1}^3 (n_{i,x_i(j')} + \alpha')} \quad (\alpha' = \alpha - 2, 1 \leq j \leq 3)$$

And $\theta_{i,j} = 0$, if $j \notin \mathbf{x}_i$.

Similarly, the Markov blanket of $\boldsymbol{\Phi}_j = (\Phi_{j,1}, \Phi_{j,2}, \dots, \Phi_{j,W})$, the distribution over words of topic j , includes: β , \mathbf{w} and \mathbf{z} .

$$p(\boldsymbol{\Phi}_j | \mathbf{w}, \mathbf{z}, \beta) \propto p(\boldsymbol{\Phi}_j | \beta) p(\mathbf{w} | \boldsymbol{\Phi}_j, \mathbf{z})$$

$$\propto \Phi_{j,1}^{(\beta-1)} \Phi_{j,2}^{(\beta-1)} \dots \Phi_{j,W}^{(\beta-1)} p(\mathbf{w} | \boldsymbol{\Phi}_j, \mathbf{z})$$

Assume $m_{j,k}$ ($1 \leq k \leq W$) is the number of times that the k th word is assigned to topic j , then:

$$p(\mathbf{w} | \boldsymbol{\Phi}_j, \mathbf{z}) \propto \Phi_{j,1}^{m_{j,1}} \Phi_{j,2}^{m_{j,2}} \dots \Phi_{j,W}^{m_{j,W}}$$

$$p(\boldsymbol{\Phi}_j | \mathbf{w}, \mathbf{z}, \beta)$$

$$\propto \Phi_{j,1}^{m_{j,1} + \beta - 1} \Phi_{j,2}^{m_{j,2} + \beta - 1} \dots \Phi_{j,W}^{m_{j,W} + \beta - 1}$$

We also deterministically assign $\boldsymbol{\Phi}_j$ to the most probable values:

$$\Phi_{j,k} = \frac{m_{j,k} + \beta - 2}{\sum_{k'=1}^W (m_{j,k'} + \beta - 2)} = \frac{m_{j,k} + \beta'}{\sum_{k'=1}^W (m_{j,k'} + \beta')}$$

$$(\beta' = \beta - 2, 1 \leq k \leq W)$$

The sampling algorithm for relevance-based topic model is formally presented in Fig. 3.

1. Initialize variables: $\mathbf{z}^{(0)}$, $\boldsymbol{\theta}^{(0)}$ and $\boldsymbol{\Phi}^{(0)}$
2. For $l = 0$ to $(L-1)$:
 - 2.1 For $t = 1$ to N :
 - 2.1.1 Sample $z_t^{(l+1)}$ from:

$$p(z_t=j | \text{MB}(z_t)) = \theta_{i,j}^{(l)} * \Phi_{j,k}^{(l)} \quad (w_t=k, d_t=i)$$
 - 2.2 For $i = 1$ to D :
 - 2.2.1 Estimate $\boldsymbol{\theta}_i^{(l+1)}$:

$$\theta_{i,x_i(j)}^{(l+1)} = \frac{n_{i,x_i(j)}^{(l+1)} + \alpha'}{\sum_{j'=1}^3 (n_{i,x_i(j')}^{(l+1)} + \alpha')}$$

$$\theta_{i,j}^{(l+1)} = 0, \text{ if } j \notin \mathbf{x}_i$$
 - 2.3 For $j = 1$ to K :
 - 2.3.1 Estimate $\boldsymbol{\Phi}_j^{(l+1)}$:

$$\Phi_{j,k}^{(l+1)} = \frac{m_{j,k}^{(l+1)} + \beta'}{\sum_{k'=1}^W (m_{j,k'}^{(l+1)} + \beta')}$$

Figure 3: Gibbs sampling for relevance-based topic model

Although the convergence distribution of Gibbs samples is theoretically independent to initialization phase (Step 1, Fig.3), a poor choice of starting values can significantly increase the convergence time (burn-in period) ([3]). In this study, we initialize the topic-word distributions based on their intuitive meaning. Background topic is initialized by word frequencies in corpus (Step 1.1 Fig. 4). A topic t in T_θ is initialized by the difference between word frequencies in its training set C_t and the whole corpus (Step 1.2). The topic representing non-relevant parts in a document d focuses on words frequently appearing on d , but not in the training set or whole corpus (Step 1.3). Purpose of factors f' , f'' and denominators in (1.2) and (1.3) is to normalize $\Phi_{t,w}$ and $\Phi_{t \circ (d),w}$ into interval $[0,1]$. Three non-zero elements of topic mixtures $\boldsymbol{\theta}_i$ are initialized uniformly. We do not need to initialize z_j because given \mathbf{w} , $\boldsymbol{\theta}$ and $\boldsymbol{\Phi}$, z_j is conditionally independent to other variables.

$$\begin{aligned}
& 1.1 \quad \Phi_{b,w}^{(0)} = \text{freq}(w|C) \\
& 1.2 \quad \forall t \in T_0: \Phi_{t,w}^{(0)} = \\
& \quad \frac{\text{freq}(w|C_t) - \text{freq}(w|C) + f^*}{\sum_{w \in V} [\text{freq}(w|C_t) - \text{freq}(w|C) + f^*]} \\
& \quad \text{where } f^* = \text{Max}_{w \in V} \{ \text{freq}(w|C) - \text{freq}(w|C_t) \} \\
& 1.3 \quad \text{For every document } d: \Phi_{t,d,w}^{(0)} = \\
& \quad \frac{\text{freq}(w|d) - \max \{ \phi_{b,w}^{(0)}, \phi_{t,w}^{(0)} \} + f''}{\sum_{w \in V} [\text{freq}(w|d) - \max \{ \phi_{b,w}^{(0)}, \phi_{t,w}^{(0)} \} + f'']} \\
& \quad \text{where } d \text{ is a document in training set for topic } t \text{ and} \\
& \quad f'' = \text{Max}_{w \in V} \{ \max \{ \Phi_{b,w}^{(0)}, \Phi_{t,w}^{(0)} \} - \text{freq}(w|d) \}
\end{aligned}$$

Figure 4: Initialization

In conventional LDA, the resulting topics are not stable over different runs of the same sampling algorithm. Topic j in one Gibbs sequence (i.e. Markov chain) is not constrained to be the same as topic j in another sequence ([15]). In the relevance-based topic model, however, the meaning of each topic is predefined. So, topics in our proposed model are stable over runs. Consequently, we could estimate Φ^* by averaging over Gibbs samples from multiple sequences. This approach not only provides a more accurate result ([17]), but also makes it possible to run the sampling process in parallel.

From the samples, we count $m_{j,k}^*$ the number of times that the k th word is assigned to topic j , for every $1 \leq j \leq K$ and $1 \leq k \leq W$. In each topic j , a set V_j^* of top W^* words ranked by $m_{j,k}^*$ is selected as the feature set. The multinomial distribution of topic j is normalized and smoothed as follows:

$$\phi_{j,k}^* = \begin{cases} \frac{m_{j,k}^* + \lambda}{\sum_{k' \in V_j^*} m_{j,k'}^* + (W^* + 1)\lambda}, & k \in V_j^* \\ \frac{\lambda}{\sum_{k' \in V_j^*} m_{j,k'}^* + (W^* + 1)\lambda}, & k \notin V_j^* \end{cases}$$

where W^* and λ are parameters

3.3 Reducing Complexity

In the algorithm described in Fig. 3, the conditional distribution of each $z_i^{(l+1)}$ is only dependent on $\theta_{ij}^{(l)}$ and $\Phi_{j,k}^{(l)}$ ($w_i=k, d_i=i$) and independent on any other z_k ($k \neq i$) (Step 2.1.1). So, we can re-arrange the sampling order in Step 2.1 in any order without affecting the final results.

Sampling step (Step 2.1.1) could be implemented by Roulette wheel technique which includes the four substeps: computing $f(z)$, normalizing $f(z)$ (note that $f(z)$ computed by the formula in Step 2.1.1 is not normalized), randomly selecting a number between $[0,1)$ and picking up a topic corresponding to the selected number. Among the four substeps, the consuming time for the first and second is significantly larger than time for the third and the four. So, the time for the first and the second is almost the same for the whole sampling step.

In the preprocessing phase, we remove stop words from a standard list ([21]) and use Porter Stemmer algorithm ([12]) to convert each token in the corpus to its stem. Therefore, in a document the topics of the tokens having the same stem are sampled from the same distribution. So, the time taken for computing $f(z)$ and normalizing the distribution (the first and the second substeps) could be reduced if we group the tokens having the same stem into continuous segments.

The computational complexity could be reduced by a factor of $r = M/N$, where M is the average number of distinct stems, and N is the average number of distinct tokens in a document. Of course, M is always less than N . Additionally, by re-ordering tokens, storage space could also be reduced. Instead of storing n continuous tokens having the same stem, we just store one token and the number n . This idea is also applicable to other probabilistic topic models in [4][5][13][15][19].

4 An empirical illustration

In this section, we illustrate some important characteristics of the proposed relevance-based topic model by applying it to a very small and pseudo dataset. The small size of the dataset makes it easy and intuitive to analyze the returned results and illustrate model characteristics. More comprehensive experiments are presented in Section 5.

The dataset contains the training sets for two topics (classes) *machine learning* (ML) and *data mining* (DM) (Table 1). Each document comprises of three parts: general terms (e.g. “artificial” or “intelligence”), topical terms (e.g. “learning” respect to ML or “mining” respect to DM), and non-relevant terms (e.g. “NLP” in doc 1). Although the dataset is pseudo, it intuitively makes sense. One can imagine that doc 1 is a paper about applications of machine learning in natural language processing, or doc 5 is a data mining paper in biomedical domain.

Table 1 - A pseudo dataset

doc#	class	Document content
1	ML	Artificial intelligence machine learning machine learning training NLP NLP NLP
2	ML	Artificial intelligence machine learning machine learning training speech speech speech speech recognition recognition recognition waves waves waves
3	DM	Intelligence data mining data mining classification clustering
4	DM	Artificial intelligence data mining data mining time series
5	DM	Artificial intelligence data mining data mining classification clustering biomed biomed

We run the inference algorithm presented in Section 3.2 with $\alpha' = 0.005$ and $\beta' = 0.1$. Table 2 shows top words ranked by probability $p(\text{word}|\text{topic})$ of some topics.

First we note that “intelligence” and “artificial” occur frequently across the two training sets and so they are identified as domain based stop words and appear in the background distribution with high weights. As regards topic representative words, in topic ML “machine”, “learning” and “training” appear frequently in every training document (and not in other documents see Table 1). So as expected they are identified by our model as representative of the topic and have dominant roles in the ML distribution (Table 2). Similarly “mining”, “data” and “cluster” are frequent in the DM training set and not in other documents; thus they have high weights in the DM distribution. Notice also that these topical terms have very low weights in the background distribution (Table 2).

Table 2 - topic-word distributions

ML		DM		Background	
learning	.32	mining	.31	intellig	.34
machine	.31	data	.30	artificial	.30
train	.16	cluster	.12	machine	.06
intellig	.06	classifi	.11	learn	.05
artificial	.05	intellig	.08	data	.05
speech	.03	artificial	.04	mining	.05
recogni	.03	biomed	.02	train	.03
wave	.02	time	.01	classify	.03
nlp	.02	series	.01	cluster	.03

to(d ₁)		to(d ₂)		to(d ₅)	
NLP	.82	speech	.38	biomed	.75
machine	.07	wave	.28	classifi	.10
learn	.06	recogni	.28	cluster	.09
train	.02	machine	.02	artificial	.03
intellig	.01	learn	.02	mining	.01
artificial	.01	train	.01	data	.01
speech	5e-7	intellig	.004	intellig	.01
recogni	5e-7	artificial	.003	recogni	4e-7
wave	5e-7	nlp	1e-7	wave	4e-7

More interesting observations may be made by looking at the lower set of rows in Table 2. We see for example, that “biomed” is a highly weighted term for to(d₅). This is because of two reasons. One is that it as it appears in very few documents of the collection (1 out of 5) it is not in the background set. Second as it appears in very few documents of the training set (1 out of 3 for DM) it is not viewed as topical either. However, it does appear with high frequency in doc 5, therefore the model learns that there is some other topic (non relevant to DM, in the relative sense) that likely generated this word in this document. Hence it has a high weight in the to(d₅) distribution. Notice here that “biomed” has the same token frequency in the DM training set as “cluster” which was regarded as topical for DM. However their occurrences across the DM training set differ thus accounting for the difference in how the model treats the two words. Consequently also $p(\text{“cluster”}|\text{DM})$ is larger than $p(\text{“biomed”}|\text{DM})$ (0.12 versus 0.02).

Thus this small example illustrates how the model is able to correctly distinguish between the different roles of the words across the topics, background and other document specific non-relevant topics.

5 Classification performance

5.1 Methodology and datasets

Classification: After the training phase, each topic j is represented by W^* word features and a smoothed multinomial distribution Φ_j^* over words. To classify a new document $d = \{w_1, w_2 \dots w_{Nd}\}$, we apply Bayesian formula to compute the posterior probability:

$$p(\text{topic} = j | d) \propto p(\text{topic} = j) \prod_{k=1}^{Nd} \phi_{j,w_k}^*$$

$$\propto \prod_{k=1}^{Nd} \phi_{j,w_k}^*$$

In the equation, we assume that classes are uniformly distributed. Finally, the class to which document d is assigned is a topic j in T_0 that maximizes the posterior probability:

$$C(d) = \underset{j \in T_0}{\text{ArgMax}} \{p(\text{topic} = j | d)\}$$

Datasets and protocol: We use two datasets. The first one is a ML Cora collection of about 4000 machine learning papers from Cora corpus ([10]). Each of the papers is categorized into seven topics (Table 3). The second one is News5 dataset, a subset of 20 Newsgroup dataset, containing the five confusable *comp.** classes ([11]). Experimental results in ([14][18]) indicate that this is the most challenging subset in 20 Newsgroup dataset. In preprocessing phase, we remove very short documents out of the datasets. We also remove rare words appearing in less than five documents and stop words from a standard list ([21]) and then use Porter ([12]) for stemming.

Table 3 - Classes in ML Cora and News5

ML Cora	News5
Probabilistic methods	Comp.graphics
Genetic algorithms	Comp.os.ms-windows.misc
Neural networks	Comp.sys.ibm.pc.hardware
Rule learning	Comp.sys.mac.hardware
Reinforce learning	Comp.window.x
Case-based	
(Learning) Theory	

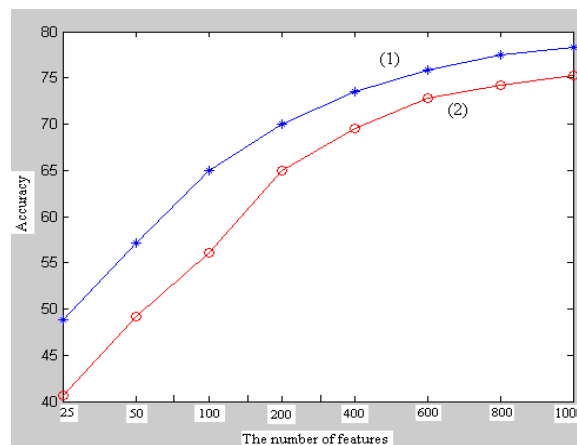
We use conventional naïve Bayes (NB) method ([9]) as the baseline. For each topic (class) t , top W^* words ranked by $p_{NB}(\text{word} | t)$ are selected as the feature set. In Section 5.2, we compare classification accuracies of NB and our proposed method, which is based on relevance-based topic model, on the two datasets. In Section 5.3, we introduce noise into the training sets, and show how the noise influences performances of each method. We use 3-fold cross validation in all experiments.

5.2 Classification accuracy

Fig. 5 shows classification accuracy of each method on ML Cora. We could see that the proposed method is consistently better than naïve Bayes. For instance, when we use 100 features, accuracies of naïve Bayes and our method are 56.1% and 64.9% respectively (14% improvement). When we use 800 features, accuracies of naïve Bayes and the proposed are 74.16% and 77.6% respectively (4.5% improvement). Note that, the best result reported in a

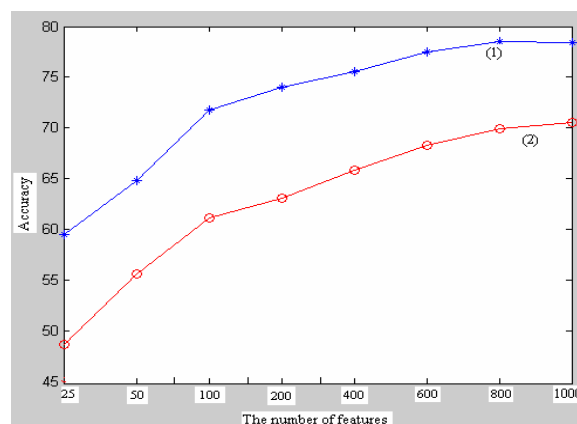
recent work ([8]) on the same dataset is only 72.5% of accuracy.

In Fig. 6, we see that when compared to naïve Bayes, the proposed method improves accuracy by 10-20%. Note that the results of our baseline are generally consistent with results published in [11] which use a semi-supervised method.



(1): Proposed method, (2): naïve Bayes

Figure 5 - Accuracy on ML Cora dataset



(1): Proposed method, (2): naïve Bayes

Figure 6 - Accuracy on News5 dataset

In [14], Sadler uses an unsupervised method based on pLSI that automatically discovers $K=5$ topics and classifies each document into one of the five topics. In order to compute accuracy, the five generated topics are heuristically matched to five topics in News5 dataset. The reported classification accuracy on News5 is only 54.7%. The method in [14] is similar to ours in the sense that it allows multiple topics per document, but it does not take any relevance information between documents and given topics into account. So, the generated topics are not

perfectly matched to the given ones. That is perhaps the cause of the significant difference in accuracy.

5.3 Robustness to noises

In order to demonstrate the robustness of our method to the non-relevant parts in training documents, we artificially introduce some noise in the training sets as follows. Each document in the training sets is made noisy by concatenating r documents from a non-relevant set. This set is a collection of documents that are not relevant to any of the topics we need to classify into. In the case of ML Cora, we use the remaining papers in the whole Cora corpus as the non-relevant set. In the case of News5, training documents for the 15 remaining classes are used as non-relevant set.

We train naïve Bayes and the proposed method on the noisy training sets and test on unseen (non noisy) documents. We compute accuracy reductions compared to the accuracies achieved using the “clean” training sets (Section 5.2) for each method. The reduction indicates how noise influences performance of the classification methods.

Table 4 - Accuracy Reduction on ML Cora dataset

# of features	25	100	500
Naïve Bayes	40.7	56	71.7
Proposed method	48.9	64.9	74.5

No noise

# of features	25	100	500
Naïve Bayes	30.5 (25%)	47 (16%)	64.53 (10%)
Proposed method	48.4 (1%)	61.7 (5%)	71.5 (4%)

Noise rate = 100% ($r = 1$)

# of features	25	100	500
Naïve Bayes	22.38 (45%)	36.96 (34%)	57.36 (20%)
Proposed method	42.54 (13%)	58.4 (10%)	68.5 (8%)

Noise rate = 200% ($r = 2$)

# of features	25	100	500
Naïve Bayes	19 (53%)	29.1 (48%)	53.1 (26%)
Proposed method	37.14 (24%)	50.6 (22%)	67 (10%)

Noise rate = 400% ($r = 4$)

Table 4 shows accuracy reductions of the two methods on ML Cora dataset with different noise

levels. In each cell, the first number is the accuracy, the other is accuracy reduction. From the table, we could see that the proposed method is significantly more robust to the presence of non-relevant information in the positive training texts than naïve Bayes on the ML Cora dataset.

Table 5 shows the comparison on News5 dataset. In term of accuracy, our method is better than naïve Bayes in all cases. In term of accuracy reduction, our method is better in 5 of 9 cases, and worse in 3 of 9 cases. The two methods are equivalent in the other 1 cases. The improvement in the sense of robustness of our method over naïve Bayes on News5 dataset is much less than on ML Cora dataset. Our explanation is that the topics on ML Cora are more well-defined and compact than topics on News5. The topics in News5 are such that there appears to be more correlation between the relevant parts and noisy parts.

Table 5 - Accuracy reduction on News5 dataset

# of features	25	100	500
Naïve Bayes	48	61	66
Proposed method	59.5	72	76

No noise

# of features	25	100	500
Naïve Bayes	38.4 (20%)	50 (18%)	58 (12%)
Proposed method	53.6 (10%)	63.4 (12%)	65.4 (14%)

Noise rate = 100% ($r = 1$)

# of features	25	100	500
Naïve Bayes	36 (25%)	45.7 (25%)	55.4 (16%)
Proposed method	51.2 (14%)	61.2 (15%)	63.8 (16%)

Noise rate = 200% ($r = 2$)

# of features	25	100	500
Naïve Bayes	30.7 (36%)	43.9 (28%)	52 (21%)
Proposed method	43.4 (27%)	50.4 (30%)	58.5 (23%)

Noise rate = 400% ($r = 4$)

6 Related work

Various supervised methods such as naïve Bayes text classification (McCallum et al. [9]) or Support Vector Machines (Joachims [7]), and semi-supervised methods ([11]) have been proposed for text classification. These methods, however, assume that

every part of a positive training document of a class is relevant to that class. That makes these methods susceptible to the quality of training set.

Probabilistic Latent Semantic Indexing (pLSI) (Hofmann [6]), which is also known as aspect model, is popularly applied in text modeling applications. In this approach, each document is a mixture of topics, where a topic is a multinomial distribution. Each word in a document is generated by a topic, and different words in the same document may be generated by different topics. Topics are automatically discovered from the corpus. Thus, the topics are synthetic and not always interpretable.

One limitation of pLSI is that it is not clear how the mixing proportions for topics in a document generated (Blei et al. [2]). To overcome the limitation, Blei et al. propose Latent Dirichlet Allocation (LDA), also referred as probabilistic topic models ([5][15]). In LDA, topic mixture of each document is a K-dimensional hidden variable randomly drawn from the same Dirichlet distribution, where K is the number of topics. Thus, LDA is a completely generative model. This model and its variants have been explored in many different applications such as document compression ([2][19]), finding scientific topics ([5]), topic trend mining ([19][16]), E-community discovery ([20]) and mixed-membership analysis ([4]). However, resulting topics in LDA might not match well with the prior classification knowledge of human beings. Therefore, the approach could not be applied directly for text classification, where topics (classes) are given upfront.

7 Conclusions

In this paper, we propose a learning approach based on relevance-based topic model for text classification. In the relevance-based topic model, a positive training document d of a class t is generated by a mixture of three topics: topic t corresponding to the class itself, a background topic, b , which captures words that are common in general and common in the particular domain, and $t_o(d)$ (other topics in the document) capturing non-relevant parts in the document. Only the parts generated by t (the really relevant parts) contribute to the resulting distribution over words for topic t . Therefore, the proposed approach is robust common words and noise in the training documents. And experimental results on original as well as “noisy” versions of ML Cora and News5 dataset demonstrate the improvement of our approach over naïve Bayes.

We also propose the idea of exploiting the assumption of exchangeability for the tokens in a document (“bag-of-words” assumption) to reduce the computational complexity of the learning algorithm.

This idea is not only applicable to our relevant based topic model, but also on conventional LDA model.

For future work, we intend to apply relevance-based topic model for text classification without any human-labeled training data. Instead, we will use documents returned from a global search engine (e.g. Google) or an intranet search engine, retrieved by topics themselves as training sets. The challenge of this approach is that there is a lot of noise in the returned sets. This is exactly the kind of problem that our model is designed to solve

References

- [1] Adrieu, C., Freitas, N., Doucet, A., Jordan, M., *An Introduction to Markov Chain Monte Carlo for Machine Learning*, Machine Learning, 50, (2003).
- [2] Blei, M., Ng, A., Jordan, M., *Latent Dirichlet Allocation*, Journal of Machine Learning Research, 3, (2003).
- [3] Casella, G., George, E., *Explaining the Gibbs Sampler*, The American Statistician, 46(3), (1992).
- [4] Erosheva, E., Fienberg, S., Lafferty, J., *Mixed-membership Models of Scientific Publication*, In Proceedings of National Academy of Science (PNAS), 2004.
- [5] Griffiths, T., Steyvers, M., *Finding Scientific Topics*, In Proceedings of National Academy of Science (PNAS), 2004.
- [6] Hofmann, T., *Probabilistic Latent Semantic Indexing*, In Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI), 1999.
- [7] Joachims, T., *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*, In Proceedings of 10th European Conference on Machine Learning (ECML), 1998.
- [8] Lu, Q., Lise, G., *Link-based Text Classification*, In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2003.
- [9] McCallum, A., Nigam K., *A Comparison of Event Models for Naïve Bayes Text Classification*, In Proceedings of AAAI-98 Workshop on Learning for Text Categorization, 1998.
- [10] McCallum, A., Nigam, K., Rennie, J., Seymore, K., *Automating the Construction of Internet Portal with Machine Learning*, Information Retrieval, 3, (2000).

- [11] Nigam, K., Ghani, R., *Analyzing Effectiveness and Applicability of Co-training*, In Proceedings of the 9th ACM Conference on Knowledge and Information Management (CKIM), 2000.
- [12] Rijsbergen, C., Robertson, S., Porter, M., *New Models in Probabilistic Information Retrieval*, British Library Research and Development Report, 5587, 1980.
- [13] Rosen-Zvi, M., Griffiths, T., Steyvers, M., Smyth, P., *The Author-Topic Model for Authors and Documents*, In Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI), 2004.
- [14] Sandler, M., *On the Use of Linear Programming for Unsupervised Text Classification*, In Proceedings of the 11th ACM SIG International Conference on Knowledge Discovery and Data Mining Conference (KDD), 2005.
- [15] Steyvers, M., Griffiths, T., *Probabilistic Topic Models*, In Landauer *et al.* (eds.) *Latent Semantic Analysis: A Road to Meaning*, Laurence Erlbaum, 2006.
- [16] Steyvers, M., Smyth, P., Griffiths, T., *Probabilistic Author-Topic Models for Information Discovery*, In Proceedings of the 10th ACM SIG International Conference on Knowledge Discovery and Data Mining Conference (KDD), 2004.
- [17] Wei, X., Croft, B., *LDA-based Document Models for Ad-hoc Retrieval*, In Proceedings of the 29th ACM SIG International Conference on Research and Development in Information Retrieval (SIGIR), 2006.
- [18] Zha, H., He, X., Dong, X., Simons, H., *Spectral Relaxation for K-Means Clustering*, In Proceedings of the 15th Annual Conference on Neural Information Processing Systems (NIPS), 2001.
- [19] Zhou, D., Ji, X., Zha, H., Giles, L., *Topic Evolution and Social Interactions: How Authors Effect Research*, In Proceedings of the 15th ACM Conference on Knowledge and Information Management (CKIM), 2006.
- [20] Zhou, D., Manavoglu, E. Li, J., Giles, L., Zha, H., *Probabilistic Models for Discovering E-Communities*, In Proceedings of the 15th ACM International World Wide Web Conference (WWW), 2006.
- [21] http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words.