The Journal of Engineering

Research Article

IET Journals
The Institution of Engineering and Technology

Uniform attribute-content model

eISSN 2051-3305
Received on 10th July 2018
Accepted on 11th December 2018
E-First on 12th June 2019
doi: 10.1049/joe.2018.5135
www.ietdl.org

Yingzhuo Xiang[1] ✉, *Jikun Yan*[1], *Ling You*[1], *Pu An*[1]

[1]*National Key Laboratory of Science and Technology on Blind Signal Processing, Chengdu, People's Republic of China*

✉ *E-mail: xiangyzh@foxmail.com*

Abstract: There have been growing needs for text processing, such as classifying, retrieving and clustering. The foundation of such a process is to extract features, which can best describe the text. Great progress has been made in text modelling. However, most of the text modelling methods are based only on the content, nor only on the attributes. Although there have been some combined models proposed in recent years, the lack of universality limits such models. In this study, the authors propose a uniform attribute-content model, which uses the attributes to influence the content feature extraction process. They design the attributes as a special filter to each feature extracted from the content. Thus the mixed features contain both content information and attribute information, which can describe the text more precise. They also propose a Monte Carlo method to solve this model. Experimental results on the Enron email dataset demonstrate the effectiveness of the authors' proposed models.

Nomenclature

Symbol

N_i	number of words in document i
k	number of topics
$\boldsymbol{D}$	corpus D
$\boldsymbol{w}$	word vector of a document
$\boldsymbol{\alpha}$	Dirichlet parameter
$\boldsymbol{\beta}_{1:K}$	work probably vector
$\boldsymbol{z}_{d,n}$	topic of nth word in document d
$\boldsymbol{y}$	attribute values
$\boldsymbol{\sigma}$	standard deviation of attributes
$\boldsymbol{b}$	normalisation parameters
$\boldsymbol{n}_{d,k}$	number of words assigned to topic k in document d
$\boldsymbol{n}_{w,k}$	number of word w assigned to topic k in the corpus

1 Introduction

The performance of text processing is largely based on feature extraction. There have been many methods proposed to extract features from the text, such as vector space model (VSM) [1, 2], latent sematic indexing (LSI) [3–5], latent Dirichlet allocation (LDA) [6], etc. VSM proposed by Gerard Salton and McGill in 1969 has been widely used in many occasions. This model uses a vector to represent a document; thus, the corpus of all documents is a matrix of these vectors. Each element of the matrix is a feature, which is computed by term frequency-inverse document frequency (TFIDF) [7–9]. Due to the high dimension of VSM, LSI uses singular value decomposition (SVD) [10] for dimensionality reduction. LDA extracts topics from the documents; each document is represented by several topics, which can make a further dimensionality reduction. All of the above methods and their modifications are based on the content of the text, which ignores the attributes of the text.

Intuitively, more information of the corpus mixed in features can lead to a better result. The text on the Internet has many attributes which are much valuable when we process it, such as the author of the text, the source, etc. For example, a piece of news contains the content, title, author, date of publication and source. Pon *et al.* [11] proposed the iScore model attempting to combine the content and attribute of the text. They abstract the text as several weak features and train a classifier based on these weak features. However, this model cannot be used for other datasets except news due to the inflexibility; especially when the text does not have many attributes.

In this paper, we develop a uniform content-attribute learning model, named uaLDA, combining the text content and text attributes to get better features. Our novel model is based on the LDA, integrating the attributes as parameters to affect the topic generation. As LDA abstract the text as a given number of topics with different weights. We use the attributes to influence the generation process of every topic. Thus the weight of the topics of the text is changed. This process is shown in Fig. 1. Two documents A, the red point, and B, the yellow point, are represented by two features, topic 1 and topic 2. This is derived from the LDA model. If the content of the two documents is similar, the distance of the two points is very short when showing in Fig. 1. If the attributes of the two documents are different, when we apply uaLDA to the same documents, point A moves to the position of A′ and point B moves to B′. The distance of the two documents becomes larger due to their different attributes. In contrast with iScore, our uaLDA does not need to extract a new feature from attributes.

The goal of our proposed method is to build a uniform content-attribute model to extract features from the text more effectively and representatively. The model can overcome the classification problem when the content of documents is similar except the attributes. The contributions of this paper are summarised as follows:

(i) By arguing that more information leads to a better classification result, we propose uaLDA to model the text with both content and attributes. The proposed method can handle text with various attributes, so we call it uaLDA where the letter 'u' represents *uniform* and the letter 'a' represents *attributes*.
(ii) The proposed method is suitable for text with attributes, especially different documents with similar contents. The advantage of the proposed method is that the feature extracted contains both contents and attributes information.
(iii) The LDA is a special case of the proposed uaLDA method when the attributes of text are not considered. The proposed uaLDA can be regarded as a generalised version of LDA. The LDA extracts topics from the content while the uaLDA extracts from both content and attributes, which contain much more useful information.

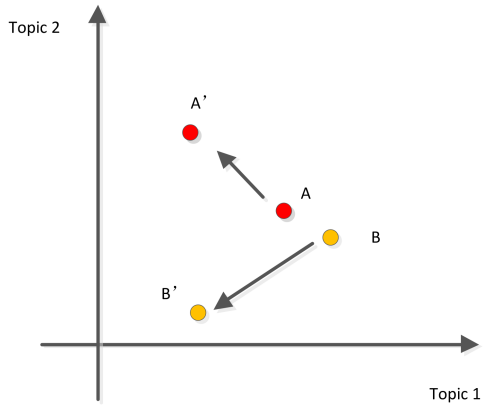

Fig. 1 *Illustration of the basic idea of uaLDA. The two points A and B are close as they have the similar content. We consider using the attributes to affect the topic extraction process, thus the point A moves to A′ and point B moves to B′, in order to expand the distance of the two documents*

The paper is organised as follows. Section 2 reviews related works of text processing. Section 3 presents the proposed uaLDA method. The experimental results are given in Section 4. Finally, Section 5 concludes this paper.

2 Related works

Many models have been proposed to model text so that text can be processed by computers. In this section, we briefly review some classic models related to the proposed method. We divide the models into two categories, a content-based method and a combined method.

2.1 Content-based method

The content-based method of text processing extracts features only based on the content of text. This has been a researching hotspot for decades. Some famous models such as VSM, LDA and their variants have been proved powerful in text processing. Here we give a short description of LDA, which is closely related to our proposed method. LDA, shorten as LDA, is a probabilistic topic model [12] that extracts the topics from the text as features. The graph model of LDA is shown in Fig. 2. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterised by a distribution over words.

Let D be a corpus of M documents with k topics. Each document is represented as $\boldsymbol{w}$, a vector of words. The generative process for each document in corpus is described below:

Choose $N\sim$ Poisson(ξ)
Choose $\theta\sim$ Dirichlet(α)
For each of the N words w_n:

(a) Choose a topic $z_n\sim$ Multinomial(θ)
(b) Choose a word w_n from $p(w_n \mid z_n; \beta)$, a multinomial probability conditioned on the topic z_n

Equation (1) is the probability equation derived from such a generation process. The rest work is to estimate parameters as follows: (see (1)) As the model is complicated and hard to solve directly, Gibbs sampling [13] and variational inference [6] are usually used to estimate the parameters [14].

Due to the perfect formulation and the performance of LDA, many variances of this model have been proposed in recent years. Roberts in [15] build a linear model of LDA cooperating the social data. Xia *et al.* in [16] specialise the LDA model that can be applied to bug triage. He *et al.* in [17] induce continuous

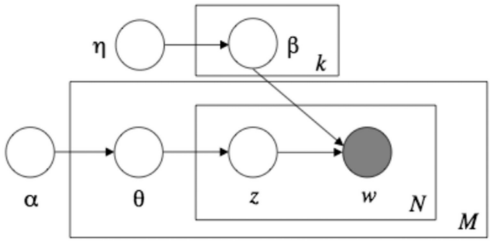

Fig. 2 *Graph model of LDA: there are three levels to the LDA representation*

distributed representations for latent topics rather than the *multinomial distribution*, in order to accelerate the calculation speed with a minor decrease in performance. In [18] Lim *et al.* propose to use LDA to model the hashtags and the texts in convenience of the Twitter data. Although the basic LDA model has been proposed decades ago, it is still one of the best text modelling methods.

2.2 Combined method

The combination of content and attributes is a challenge in modelling text. The model iScore unifies content and attributes as a set of weak features. For an article d, a set of feature extractors generate a set of feature scores $F(d) = \{f_1(d), f_2(d),\ldots, f_n(d)\}$. The features they extract are *topic relevancy*, *uniqueness*, *source reputation*, *writing style*, *freshness*, *subjectivity* and *polarity*. The *topic relevancy*, *uniqueness*, *subjectivity* and *polarity* are content related. The rest features are calculated based on the attributes. Each feature score is normalised to (0,1). Thus each article is represented as a vector of six dimensions.

This model largely relies on the attributes of articles. As introduced in [11], they extract six weak features, half of which are attributes related. If the attributes of an article are missing or incorrect, it seriously affects the feature computation, which may lead to bad classification results in the succeeding process. However, the iScore model is an excellent model to combine content and attributes, especially when modelling text similar to news, which contain lots of attributes.

Azzopardi *et al.* in [19] adopt the iScore model into an auto recommendation system of Digital Library of Serbian Ph.D. Dissertations. They compared this content-attribute method with the Collaborative method in the application of auto recommendation scope, showing the superiority of their model. Nair and Binesh in [20] model text with strongly associated terms mined through the association rule mining technique. Their experiments show their method can effectively classify 'novel' documents. However, these models are not universal; their performance is largely relaid on the data set. Also, the combination of the content and attributes of these models are not very beautiful in the formula.

3 Methodology

In this section, we introduce the proposed uaLDA and use Gibbs sampling to deal with the parameter estimation problem of this model. We give an explanation of why uaLDA works better than LDA and how it extracts features from content and attributes.

3.1 uaLDA

The content-based models such as LDA introduced in Section 2 and its variants work well when the text content is not similar. However, if two text contents are similar, but their attributes are different, the content-based models cannot figure out the two documents. Although the iScore model attempts to add attributes information into feature extraction, it performs poorly when

$$p(D|\alpha,\beta)=\prod_{d=1}^{M}\int p(\theta_d|\alpha)\left(\prod_{n=1}^{N_d}\sum_{z_{dn}}p(z_{dn}|\theta_d)p(w_{dn}|z_{dn},\beta)\right)\mathrm{d}\theta_d \quad (1)$$

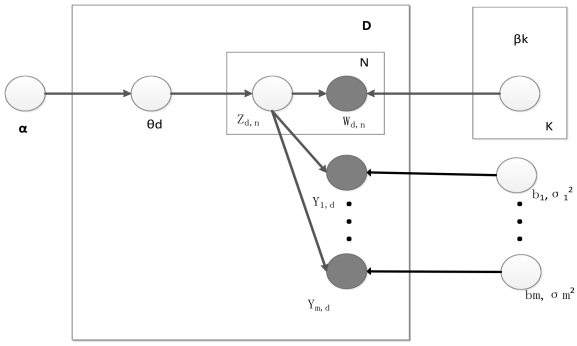

Fig. 3 *Graph model of uaLDA. Note that the black circle is the parameters that we can get from the corpus. The direction of the arrow is the sequence of the generation process. In uaLDA, we consider the topics to generate the attributes. Thus when we estimate the topic parameters $Z_{d,n}$, attributes information will affect the genitive process*

attributes are missing or incorrect. Moreover, the robustness of iScore is poor due to the short vector dimension. We propose uaLDA aiming to overcome such shortcomings of the above models.

The uaLDA is a probabilistic model of three levels, which is similar to LDA in Section 2. The difference is that we add attributes to this model as parameters to affect the topic extraction. The basic idea of uaLDA is that we add to LDA an attribute variable associated with each document. As mentioned, this attribute variable might be the data of the text, the source or the author of the document. We jointly model the documents and the attributes, to find latent topics, in other words, the features, which will best match the attributes and the content of each document. In uaLDA, we apply for a real number from 0–1 as the attribute variable, which each different attribute is regarded the same weight to latent topics.

To make the illustration of uaLDA clear, we list the symbols and their meaning in the Nomenclature section. We draw the graph model of uaLDA in Fig. 3. Under the sLDA model, each document and response arises from the following generative process:

(i) Choose $N\sim$ Poisson(ξ)
(ii) Choose $\theta\sim$ Dirichlet(α)
(iii) For each of the N words w_n:
(a) Choose a topic $z_n\sim$ Multinomial(θ)
(b) Choose a word w_n from $p(w_n \mid z_n; \beta)$, a multinomial probability conditioned on the topic z_n
(iv) Draw attributes variables $y|z_{1:N}, b, \sigma^2 \sim N(b^T\bar{z}, \sigma^2)$. Here $\bar{z} = (1/N)\sum_{n=1}^{N} z_n$.

The attributes values y_1, y_2,…,y_m are derived from each document. As each attribute has no relevance to the other one, y_1, y_2,…,y_m are all independent. Thus we normalise each attribute value to (0,1), which means the weight of each attribute is the same to latent topics.

3.2 Solving the model

We derive the total probability of latent topics as

$$\begin{aligned} p(\boldsymbol{z}|\alpha,\eta,\boldsymbol{w},\boldsymbol{y},\boldsymbol{b},\sigma) &\propto p(\boldsymbol{w}|\boldsymbol{z},\eta)p(\boldsymbol{z}|\eta)p(\boldsymbol{y}|\boldsymbol{z},\boldsymbol{b},\sigma) \\ &= \int\prod_d p(w_d|z_d,\beta)\mathrm{d}p(\beta|\eta)\int p(z_d|\theta_d)\mathrm{d}p(\theta_d|\sigma) \\ &\quad \cdot \prod_d p(y_d|z_d,\boldsymbol{b},\sigma) \end{aligned} \tag{2}$$

This illustration of (2) is shown in Fig. 4. As the model is complicated and hard to calculate the latent topic of each document, we use the Gibbs sampling method to estimate. To solve (2), we first calculate the prior probability distribution to get

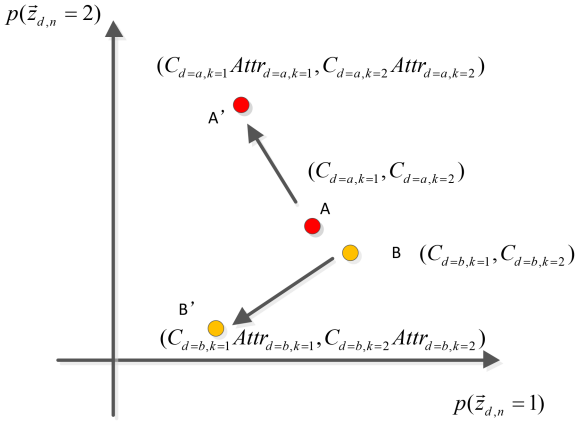

Fig. 4 *Illustration of how to attribute of document effect the feature extraction as (13) indicates. Here we extract two features from documents a and b, which means* $\boldsymbol{k}$ *= (1, 2). If we only consider the content, documents a and b are shown at the positions of A and B in the coordinate. When we add the attribute information into the features in the method of uaLDA, the position of a and b moves to A' and B', which the distance is larger*

$$\begin{aligned} \int p(z_d|\theta_d)\mathrm{d}p(\theta_d|\alpha) &= \int\prod_i \theta_{d,z_i}\frac{1}{B(\alpha)}\prod_k \theta_{d,k}^{\alpha_k}\mathrm{d}\theta_d \\ &= \frac{1}{B(\alpha)}\int\prod_k \theta_{d,k}^{n_{d,k}+\alpha_k}\mathrm{d}\theta_d \\ &= \frac{B(\boldsymbol{n}_{d,.}+\alpha)}{B(\alpha)} \end{aligned} \tag{3}$$

Given a number of topics, we get the likelihood function of words as

$$\begin{aligned} &\int\prod_d p(w_d|z_d,\beta)\mathrm{d}p(\beta|\eta) \\ &= \int\left(\prod_d\prod_i \beta_{w_{d,i},z_{d,i}}\right)\int\left(\prod_k\frac{1}{B(\eta)}\prod_w \beta_{w,k}^{\eta_w}\right)\mathrm{d}\beta \\ &= \prod_k\frac{1}{B(\eta)}\int\prod_w \beta_{w,k}^{\eta_w+n_{w,k}}\mathrm{d}\beta_k \\ &= \prod_k\frac{B(\eta+n_{.,k})}{B(\eta)} \end{aligned} \tag{4}$$

According to the generative process, we derive the probability of attributes y as

$$p(y_d|z_d,\boldsymbol{b},\sigma) \propto exp\sum_i\left(-\left(y_{d,i}-\boldsymbol{b}_i^T\overline{z_d}-\sigma_i\right)^2\right) \tag{5}$$

We substitute (3)–(5) into (2) and get

$$\begin{aligned} p(\boldsymbol{z}|\alpha,\eta,\boldsymbol{w},\boldsymbol{y},\boldsymbol{b},\sigma) \propto &\prod_d\frac{B(\boldsymbol{n}_{d,.}+\alpha)}{B(\alpha)}\cdot \\ &\prod_k\frac{B(\eta+n_{.,k})}{B(\eta)}\prod_d\exp\sum_i\left(-\left(y_{d,i}-\boldsymbol{b}_i^T\bar{z}_d-\sigma_i\right)^2\right) \end{aligned} \tag{6}$$

From (6) we can derive (7), the posterior probability equation after one assignment of Gibbs sampling

$$
\begin{aligned}
&p(z_{d,n}=k\,|\,\alpha,\eta,\boldsymbol{w},\boldsymbol{y},\boldsymbol{b},\sigma,\boldsymbol{z}_{\neg(d,n)})\\
&\quad\propto \prod_{d'}\frac{B(\boldsymbol{n}_{d',\cdot}+\alpha)}{B(\alpha)}\prod_{k'}\frac{B(\eta+\boldsymbol{n}_{\cdot,k'})}{B(\eta)}\\
&\quad\bullet\exp\sum_i\left(2\boldsymbol{b}_i^T\boldsymbol{z}_d(y_d-\sigma)-(\boldsymbol{b}_i^T\boldsymbol{z}_d)^2\right)\\
&\quad\propto \prod_{k'}[\Gamma(n_{d,k'}+\alpha_{k'})\frac{\Gamma(\eta_{w_{d,n}}+n_{w_{d,n},k'})}{\Gamma(\sum_w n_{w,k'}+\eta_w)}]\\
&\quad\bullet\exp\left(\sum_i 2\frac{b_{k,i}}{N_d}\left(y_{d,i}-\sigma_i-\boldsymbol{b}_i^T\bar{\bar{z}}_d^{\neg n}\right)-\left(\frac{b_{k,i}}{N_d}\right)^2\right)
\end{aligned}
\tag{7}
$$

In (7), $N_d=\sum_{k'}n_{d,k}$ which is the word number in document d. Notice that the Gamma function has the property shown in (8). We use (8) to simplify (7), and get

$$
\begin{aligned}
\frac{\Gamma(x+b)}{\Gamma(x)}&=\begin{cases}x & \text{if}\quad b=1\\ 1 & \text{if}\quad b=0\end{cases}\\
&=x^b,\quad b\in\{0,1\}
\end{aligned}
\tag{8}
$$

(see (9))

Here $1(k=k')$ means when $k=k'$, the value is 1, otherwise 0; $n_{d,k'}^{\neg d,n}=\sum_{i\neq n}1(z_{d,i}=k')$ means that the number of words assigned to topic k' in document d minus the words assigned to the current topic. As $n_{d,k'}^{\neg d,n}$ does not rely on the current assignment of $z_{d,n}$, it is a constant to the posterior probability. Thus (9) can be simplified as

$$
\Gamma(n_{d,k'}+\alpha_{k'})\propto(n_{d,k'}^{\neg d,n}+\alpha_{k'})^{1(k=k')}
\tag{10}
$$

Similarly, we can simplify the other Gamma function in (7) and get

$$
\Gamma\left(\eta_{w_{d,n}}+n_{w_{d,n},k'}\right)\propto\left(n_{w_{d,n},k'}^{\neg d,n}+\eta_{w_{d,n}}\right)^{1(k=k')}
\tag{11}
$$

$$
\Gamma\left(\sum_w n_{w,k'}+\eta_w\right)\propto\left(\sum_w\eta_w+n_{w,k'}^{\neg d,n}\right)^{1(k=k')}
\tag{12}
$$

Here $n_{w,k'}^{\neg d,n}=\sum_{d'}\sum_i 1(z_{k',i}=k'\wedge w_{d',i}=w\wedge(d,n)\neq(d',i))$. Substitute (10)–(12) into (7), we get

$$
\begin{aligned}
&p(z_{d,n}=k\,|\,\alpha,\eta,\boldsymbol{w},\boldsymbol{y},\boldsymbol{b},\sigma,\boldsymbol{z}_{\neg(d,n)})\\
&\quad\propto(n_{d,k}^{\neg d,n}+\alpha_k)\frac{n_{w_{d,n},k}^{\neg d,n}+\eta_{w_{d,n}}}{N_k^{\neg d,n}+W\eta}\\
&\quad\bullet\exp\sum_i\left(2\frac{b_{k,i}}{N_d}\left(y_{d,i}-\sigma_i-\boldsymbol{b}_i^T\bar{\bar{z}}_d^{\neg n}\right)-\left(\frac{b_{k,i}}{N_d}\right)^2\right)
\end{aligned}
\tag{13}
$$

That $N_k^{\neg d,n}$ in (13) means the number of words assigned to topic k except for the assignment of $z_{d,n}$. Here we derive the Gibbs sampling equation. Repeat (13) for each k of several times, the distribution of latent topics tends to be stable. Thus we get the final state of $p(z_{d,n})$, the topic distribution of document d, which is the feature extracted from the document d.

3.3 Why uaLDA works better

Here we depose (13) into two parts. Thus (13) is the production of

$$
C_{d,k}=\left(n_{d,k}^{\neg d,n}+\alpha_k\right)\frac{n_{w_{d,n},k}^{\neg d,n}+\eta_{w_{d,n}}}{N_k^{\neg d,n}+W\eta}
\tag{14}
$$

$$
\mathrm{Attr}_{d,k}=\exp\sum_i\left(2\frac{b_{k,i}}{N_d}\left(y_{d,i}-\sigma_i-\boldsymbol{b}_i^T\bar{\bar{z}}_d^{\neg n}\right)-\left(\frac{b_{k,i}}{N_d}\right)^2\right)
\tag{15}
$$

$$
p(z_{\mathrm{d},n}=k\,|\,z_{\neg(\mathrm{d},n)})\propto C_{\mathrm{d},k}\bullet\mathrm{Attr}_{\mathrm{d},k}
\tag{16}
$$

Look into the sampling (13), the first part of the production is related to the content, the other part is related to the attributes. During the process of deriving $p(z_{\mathrm{d},n})$, every dimension of $p(z_{\mathrm{d},n})$ is the product of content value $C_{\mathrm{d},k}$ and attributes value $\mathrm{Attr}_{\mathrm{d},k}$. As $\mathrm{Attr}_{\mathrm{d},k}$ is a function of k, the value of $\mathrm{Attr}_{\mathrm{d},k}$ changes with the topics. Thus the $\mathrm{Attr}_{\mathrm{d},k}$ can alter the $C_{\mathrm{d},k}$ for each k.

The process of one assignment of two documents is shown in Fig. 4. The uaLDA model utilises the value of the attribute to influence the content feature vector. The distance of the two documents is large when the attributes of the two are different, even though their content is similar. In another point of view, as the production relationship of content and attributes values shown in (13), the role of attributes value is as a filter to the content features. The content feature is projected to a special vector space, which can make the feature distance larger. Also, the sampling process of (13) can be treated as a transformation, which transfers the content feature space to the mix feature space. Thus the information of attributes is added to the final features of documents.

4 Experiment and results

In Section 3, we introduce the uaLDA model and explain why it works better. We analyse how the content features are influenced by the attributes. In this section, we do some experiment on the data set and make a compared to other feature extraction methods.

4.1 Experiment introduction

We use the Enron email data set to test the performance of uaLDA model. The emails in this data set have many attributes, such as the sender, the receiver, and the date. Also, most emails have rich content. Due to the attachment of the previous content, some emails' content is similar; this is a challenge to classify such emails. We manually select some emails from the catalog of *internal company policy* and sent by a group of given authors. Furthermore, we randomly add some error emails into the selected emails as noise. The selected set of emails mixed with error emails is the target data set. The goal is that whether the model can best describe the target data set from the whole data set after the training step.

In the experiment, we divide the target data set and the whole data set into six parts and numbered from 1 to 9. Each part contains several target emails. We use No.1 and No.2 parts as a training data set, No.3 as the test part. The rest 4–9 is treated the same as 1–3. Then we have three independent experiments.

To test the performance of our proposed method, we select two famous and classic models, the LDA model and iScore. Although the two models have been proposed for years, the two models are still state-of-art as the excellent performance and the representativeness in text modelling [21]. We test the LDA model, iScore model, XML-CNN model [22] and uaLDA model with the same classifier in this data set. Thus a better model of feature extraction can make the classify results better, as they share the same classifier.

4.2 Experiment with SVM classifier

We first test the three models with SVM [23–25] classifier in three experiments. The results are shown in Figs. 5 and 6.

Fig. 5 shows that the three models have a considerable high recall on the data set, especially the XML-CNN model. Fig. 6

$$
\begin{aligned}
\Gamma(n_{d,k'}+\alpha_{k'})&=\Gamma(n_{d,k'}-1(k=k')+\alpha_{k'}+1(k=k'))\\
&=\Gamma(n_{d,k'}-1(k=k')+\alpha_{k'})\cdot(n_{d,k'}-1(k=k')+\alpha_{k'})^{1(k=k')}\\
&=\Gamma(n_{d,k'}^{\neg d,n}+\alpha_{k'})\cdot(n_{d,k'}^{\neg d,n}+\alpha_{k'})^{1(k=k')}
\end{aligned}
\tag{9}
$$

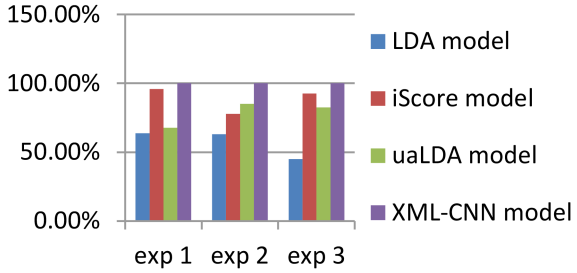

Fig. 5 *Recall of the three experiments with SVM. The recall is also called sensitivity, which means the fraction of relevant instances that have been retrieved over total relevant instance. Notice that the entire three models have a good performance on the recall*

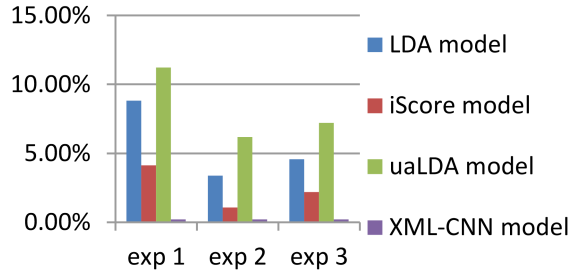

Fig. 6 *Precision of the three experiments with SVM. The precision is also called positive predictive values, which means the fraction of relevant instances among the retrieved instances. Obviously, uaLDA model performs the best in the three experiments*

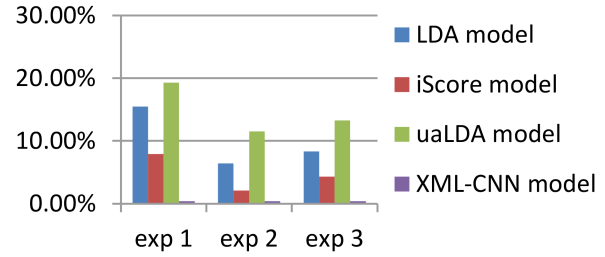

Fig. 7 *F1-measure of the three experiments with SVM. The calculation of f1-measure is* $f1 = 2{*}r{*}p/r + p$. *The higher of f1 means the better the performance of the algorithm*

shows the precision of the three models. Obviously, uaLDA outperforms the other two models. Although XML-CNN model has a higher recall, the precision is terrible low. This means that XML-CNN model selects all emails but takes none for the target set. The LDA model performs medium in the four. It has a recall of about 60% and a precision of 7%, which is 4% better than iScore. Our proposed uaLDA model performs the best. It has about 80% of recall and 12% precision for the best.

We also draw the F1-measure of the three algorithms in the three experiments in Fig. 7. The uaLDA model performs the best among the three during three independent experiments obviously. The classic LDA model, which only uses the content of the documents performs moderately, while the XML-CNN model performs the worst. The three experiments are all independent of each other; this means that the results are not occasional. Our proposed content-attributes model uaLDA performs better than the LDA model, which only considers the content; we can conclude that our proposed model extracts better features from the documents with attributes.

4.3 Experiment with EM classifier

The EM [26, 27] classifier is a famous unsupervised machine learning method, which is different from SVM. We have tested uaLDA with SVM and the performance shows that the features extracted from document are better than the LDA and iScore. Here we test the three models with EM classifier with the same data set. The recall and precision are shown in Figs. 8 and 9.

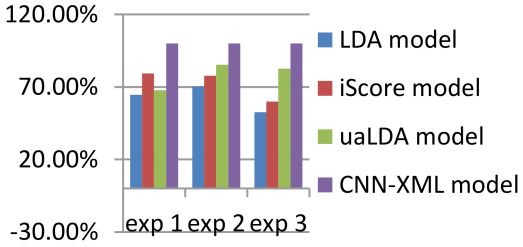

Fig. 8 *Recall of three experiments with EM*

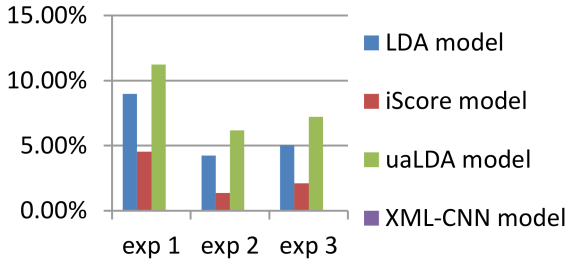

Fig. 9 *Precision of the three experiments with EM*

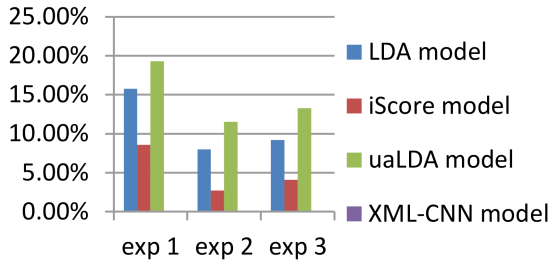

Fig. 10 *F1-measure of the three experiments with EM*

The results with EM classifier are similar to that with SVM. The recall of the three is considerable high and precision is lower. Our proposed uaLDA performs best in the precision while their recall is similar. Especially in experiment 3, the uaLDA is about 20 percent higher than iScore in recall and the score in precision is more than three times higher than iScore. The XML-CNN model classifies all emails as one class.

We also draw the F1-measure of the three algorithms with EM in Fig. 10. The F1-measure performance is similar to that shown in Fig. 7. The uaLDA model performs about 30% better than LDA model and about 200% than iScore.

4.4 Conclusion

In this section, we test the performance of uaLDA and make a comparison with LDA, XML-CNN and iScore. The quality of features extracted by uaLDA is examined by two classifiers, SVM and EM. Experiments show that uaLDA outperforms the other three. As the LDA model only extracts features from the content, we can conclude that our content-attribute model uaLDA efficiently utilises the attributes information and combines it into the features. Furthermore, our proposed uaLDA model performs on average two times better than iScore, which is another way to use attributes and content as features of the document.

5 Summary

In this paper, we have proposed uaLDA model to extract features from text with attributes. We use the attributes value to influence the content feature extraction process, which can combine the attributes information into the content features. Thus the features we extracted are more representable and can be efficiently used in classification. We built this model and give a resolution with Gibbs sampling. We gave an explanation of why uaLDA works well and how to combine attribute information into the features with the sampling equation. Experimental results on Enron email data set show that our proposed uaLDA model outperforms the original

LDA model which only extracts features from content. Furthermore, uaLDA outperforms iScore, which is a famous classical way to combine content and attributes.

6 References

[1] Salton, A., Wong, G., Yang, C.S.: 'A vector space model for automatic indexing', *Commun. ACM*, 1975, **18**, pp. 613–620

[2] Salton, G.: '*The SMART information retrieval system*' (Prentice-Hall, Englewood Cliffs, NJ, 1971)

[3] Hofmann, T.: 'Probabilistic latent semantic indexing'. Proc. of the 22nd Annual International ACM SIGIR Conf. on Research and Development in Information Retrieval (ACM), Berkeley, USA, 1999, pp. 50–57

[4] Landauer, T.K.: '*Latent semantic analysis*' (John Wiley & Sons, Ltd, 2006)

[5] Dumais, S.T.: 'Latent semantic analysis', *Annu. Rev. Inf. Sci. Technol.*, 2004, **38**, (1), pp. 188–230

[6] Blei, D.M., Ng, A.Y., Jordan, M.I.: 'Latent Dirichlet allocation', *J. Mach. Learn. Res.*, 2003, **3**, pp. 993–1022

[7] Yiming, Y.: 'An evaluation of statistical approach to text categorization', Technical Report CMU-CS-97–127, Computer Science Department, Carnegie Mellon Univ., 1997

[8] Aizawa, A.: 'An information-theoretic perspective of tf–idf measures', *Inf. Process. Manage.*, 2003, **39**, (1), pp. 45–65

[9] Martineau, J., Finin, T.: 'Delta TFIDF: an improved feature space for sentiment analysis'. 3rd Int. AAAI Conf. on Web and Social Media (ICWSM), San Jose, USA, 2009, p. 106

[10] Golub, G.H., Reinsch, C.: 'Singular value decomposition and least squares solutions', *Numer. Math.*, 1970, **14**, (5), pp. 403–420

[11] Pon, R.K., Cárdenas, A.F., Buttler, D.J*., et al.*: 'Iscore: measuring the interestingness of articles in a limited user environment'. IEEE Symp. on Computational Intelligence and Data Mining (CIDM), 2007, Paris, France, 2007, pp. 354–361

[12] Blei, D.M.: 'Probabilistic topic models', *Commun. ACM*, 2012, **55**, (4), pp. 77–84

[13] Griffiths, T.L., Steyvers, M.: 'Finding scientific topics', *Proc. Natl. Acad. Sci.*, 2004, **101**, (Suppl. 1), pp. 5228–5235

[14] Kumar, A.: 'A spectral algorithm for latent Dirichlet allocation', *Adv. Neural Inf. Process. Syst.*, 2012, pp. 917–925

[15] Roberts, M.E., Stewart, B.M., Airoldi, E.M.: 'A model of text for experimentation in the social sciences', *J. Am. Stat. Assoc.*, 2016, **111**, (515), pp. 988–1003

[16] Xia, X., Lo, D., Ding, Y*., et al.*: 'Improving automated bug triaging with specialized topic model', *IEEE Trans. Softw. Eng.*, 2017, **43**, (3), pp. 272–297

[17] He, J., Hu, Z., Berg-Kirkpatrick, T*., et al.*: 'Efficient correlated topic modeling with topic embedding', Proc. 23rd ACM SIGKDD Int. Conf. on Knowl. Discov. Data Min., Halifax, Canada2017, pp. 225–233

[18] Lim, K.W., Chen, C., Buntine, W.: 'Twitter-network topic model: a full Bayesian treatment for social network and text modeling', arXiv preprint arXiv:1609.06791, 2016

[19] Azzopardi, J, Ivanovic, D., Kapitsaki, G.: 'Comparison of collaborative and content-based automatic recommendation approaches in a digital library of serbian', PhD Dissertations, Semanitic Keyword-based Search on Structured Data Sources, Springer, Cham, 2016

[20] Nair, B.: 'A classifier to predict document novelty using association rule mining'

[21] Blei, D. M.: 'Improving and evaluating topic models and other models of text comment', 2016, pp. 1408–1410

[22] Liu, J., Chang, W.C., Wu, Y*., et al.*: 'Deep learning for extreme multi-label text classification'. Proc. 40th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, Tokyo, Japan 2017

[23] Cortes, C., Vapnik, V.: 'Support vector machine', *Mach. Learn.*, 1995, **20**, (3), pp. 273–297

[24] Suykens Johan, A.K., Vandewalle, J.: 'Least squares support vector machine classifiers', *Neural Process. Lett.*, 1999, **9**, (3), pp. 293–300

[25] Tong, S., Koller, D.: 'Support vector machine active learning with applications to text classification', *J. Mach. Learn. Res.*, 2001, **2**, pp. 45–66

[26] Bailey, T.L., Elkan, C.: '*Fitting a mixture model by expectation maximization to discover motifs in bipolymers*', (University of California at San Diego, Technical Report, 1994), pp. 28–36

[27] Moon, T.K.: 'The expectation-maximization algorithm', *IEEE Signal Process. Mag.*, 1996, **13**, (6), pp. 47–60