IDIAP RESEARCH REPORT

Theme Topic Mixture Model: A Graphical Model for Document Representation

Mikaela Keller [1] Samy Bengio [2]

IDIAP–RR 04-05

April 1, 2005

submitted for publication

[1] IDIAP, CP 592, 1920 Martigny, Switzerland, mkeller@idiap.ch
[2] IDIAP, CP 592, 1920 Martigny, Switzerland, bengio@idiap.ch

IDIAP Research Report 04-05

Theme Topic Mixture Model: A Graphical Model for Document Representation

Mikaela Keller Samy Bengio

April 1, 2005

submitted for publication

Abstract. Documents are usually represented in the bag-of-word space. However, this representation does not take into account the possible relations between words. We propose here a graphical model for representing documents: the Theme Topic Mixture Model (TTMM). This model assumes two types of relations among textual data. Topics link words to each other and Themes gather documents with particular distribution over the topics. This paper defines the TTMM, compares it to the related Latent Dirichlet Allocation (LDA) model [**?**] and reports some interesting empirical results.

Contents

1 Introduction

In order to be automatically processed, textual data must be represented formally. The most basic and widely used indexing method, for Text Categorization and other supervised related problems, is the *bag-of-words* document representation [?].

Several other document representations have been proposed in the literature, in particular, methods based on Graphical Models, such as Latent Dirichlet Allocation (LDA) [?] and Probabilistic Latent Semantic Analysis (PLSA) [?]. They estimate the density of the documents and try to overcome some problems inherent to the bag-of-words representation.

One weakness of bag-of-words is that it does not take into account the synonymic and polysemic properties of human languages. That is, it will respectively make a high distinction between the words *ocean* and *sea*, but will merge the different meanings of the word *surfing* (the Internet or in the sea).

A second problem with this simple representation is that the dimension of the representation space is equal to the size of the dictionary (order of magnitude 20 000 words). That means a lot of parameters to estimate in any system taking bag-of-words documents as inputs, which leads easily to a curse of dimensionality problem.

Here we present another Graphical Model, the Theme Topic Mixture Model (TTMM), which, like PLSA and LDA, tries to overcome these problems. This method leads to a representation which is constructed to highlight a small number of "concepts" or "topics" present in the documents, instead of a huge number of words. Furthermore, an advantage that density estimation methods have over indexing methods is the possibility to take profit of unlabeled data in order to improve the performance on supervised tasks.

In Section 2, we quickly explain the general document representation problem. In Section 3 we present the TTMM and in Section 4, the related LDA probabilistic model. Section 5 compares these two models on several theoretical aspects and Section 6 reports an experiment comparing different document representations. Finally, Section 7 concludes the paper.

But first we would like to emphasize a particular point: in this paper you will find words such as *concept*, *theme* or *topic*. They are used here by commodity in order to express the intuition of semantic links between textual data components, but they in fact simply refer to high level statistical correlations.

2 Document Representation

Most Corpus Information Access tasks make the assumption that the precise order of the words in documents can be neglected and that the word frequencies are sufficient information. Implications of these assumptions are reflected in the preprocessing step as well as the document representation itself.

As explained in [?], documents are often represented by a vector $d = (q_1, ..., q_{|\mathcal{V}|})$ of weights q_j, assigned to every word w_j in a vocabulary $\mathcal{V}$. This representation is called the *bag-of-words* representation or the Vector Space Model. The weight q_j is in general a function of the frequency of the j^{th} word of $\mathcal{V}$ in the document d. The vocabulary $\mathcal{V}$ is extracted from a training subset of the targeted corpus. Since the frequencies of words are the key point of this representation, selected *neutral* words, called *stop-words* (such as *a*, *the*, *about*, *as*, etc), which have usually high frequency but low discriminant properties, are in general removed from $\mathcal{V}$. Another possible step in the preprocessing of $\mathcal{V}$ is the so-called *stemming*, in which words in the corpus are replaced by their stem. For example *connecting, connected, connection, connections,* would be replaced by their common stem, *connect.* This step - not always performed - reduces the vocabulary size and attempts to reflect the fact that words with the same stem have similar meanings.

However, except for stemming, there is no information about possible links between words included in this representation. Nevertheless, other approaches to represent documents can be applied, taking into account this kind of information. Among these approaches, we find probabilistic approaches in which the density of documents in the Vector space is estimated according to a model. In the following, a document density estimation model is proposed in which high level statistical correlations between words in a corpus are assumed.

3 Theme Topic Mixture Model

The proposed model has a lot in common with LDA (see section 4), since it is inspired by it. In the Theme Topic Mixture Model, the documents are assumed to be sampled from a mixture over latent themes, each of which defines a particular mixture over latent topics as a distribution over words. As graphically displayed in Fig. 1, in this model the observed variable is the document d, seen as a set of words w_l, and the unobserved variables are the themes $h \in \{1, \ldots, J\}$ and the topics $t \in \{1, \ldots, K\}$, with J and K being hyper-parameters that must be chosen. The parameters π, τ and β represent respectively the mixing proportions of themes, the mixing proportions of topics given the themes and the probability of each word given each topic, that have to be estimated.

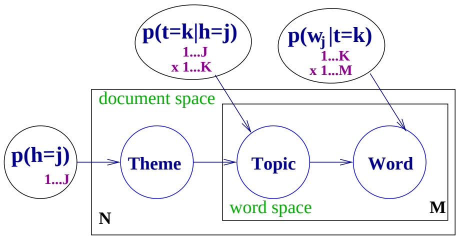

Figure 1: TTMM graphical representation. The boxes represent replicates. The outer box represents the repeated choice of themes, while the inner box represents the repeated choice of topics within a theme. π, τ, and β are the parameters of the model.

The underlying generative process for each document is the following:

1. Choose $|d| \sim Poisson(\xi)$: the document size.
2. Choose a theme $h = j$ from $P(h)$, a multinomial distribution with parameter $\pi = (\pi_1, ..., \pi_J)$: the mixing proportions.
3. For each of the $|d|$ words in d:
   (a) Choose a topic $t = k$ in $\{1, \ldots, K\}$ from $P(t|h = j)$, a multinomial distribution conditioned on the theme $h = j$.
   (b) Choose a word w_l from $P(w|t = k)$, a multinomial distribution conditioned on the topic $t = k$.

The randomness of the document size $|d|$, modeled for example with a Poisson distribution with parameter ξ, is necessary for the generative process. However, given that $|d|$ is independent of all the other data generating variables (h and t), it is not of real interest for the model.[1] Hence, it will be ignored.

According to the generative process, each word w is seen as a mixture of topics t, with different mixing proportions depending on the document's theme h:

$$P(w_l|h = j) = \sum_{k=1}^{K} \tau_{jk}\beta_{kl}, \tag{1}$$

where $\tau_{jk} = P(t = k|h = j)$ and $\beta_{kl} = P(w_l|t = k)$.

[1]In fact the log-likelihood will have this form: $\mathcal{L} = A(|d|) + B(h, t)$ and thus maximizing it will lead to two distinct problems.

The probability of a document d given that it was generated by the theme $h = j$, is then

$$\begin{aligned} P(d|h=j) &= \prod_{w_l \in d} \left[P(w_l|h=j)\right]^{n(w_l,d)} \\ &= \prod_{w_l \in d} \left[\sum_{k=1}^{K} \tau_{jk}\beta_{kl}\right]^{n(w_l,d)}, \end{aligned} \tag{2}$$

where $n(w_l, d)$ is the frequency of the term w_l in d.

Finally, each document d is seen as a mixture of themes h:

$$\begin{aligned} P(d) &= \sum_{j=1}^{J} \pi_j P(d|h=j) \\ &= \sum_{j=1}^{J} \pi_j \prod_{w_l \in d} \left[\sum_{k=1}^{K} \tau_{jk}\beta_{kl}\right]^{n(w_l,d)}, \end{aligned} \tag{3}$$

where $\pi_j = P(h = j)$.

Let D be a given corpus of N documents. The log-likelihood of the corpus D given the model then becomes:

$$\mathcal{L}(D;\pi,\tau,\beta) = \sum_{i=1}^{N} \ln \left[\sum_{j=1}^{J} \pi_j \prod_{w_l \in d_i} \left(\sum_{k=1}^{K} \tau_{jk}\beta_{kl}\right)^{n(w_l,d_i)}\right]. \tag{4}$$

Depending on the actual implementation of the various multinomial distributions (they could be represented as tables but also as Multilayer Perceptrons (MLPs) for instance), it can be maximized, either by Expectation-Maximization (EM) [?] or Stochastic Gradient Ascent [?] optimization techniques.

3.1 Expectation-Maximization Optimization

In order to perform an EM optimization of a TTMM one has first to get rid of the sum inside the logarithm in the log-likelihood equation (4). This could be done easily if we were given $\{h_{ij}\}$ the indicator variables specifying which theme j the document d_i was generated from, and $\{t_{jlk}\}$ the indicator variables specifying which topic k the word w_l was generated from given that we were in the theme j context. Indeed, the complete log-likelihood could be written as:

$$\begin{aligned} \mathcal{L}_{comp}(D;\pi,\tau,\beta) = \sum_{i=1}^{N}\sum_{j=1}^{J} h_{ij} \Big(\ln(\pi_j) \\ + \sum_{w_l \in d_i}\sum_{k=1}^{K} t_{jlk}[n(w_l,d_i)] \ln(\tau_{jk}\beta_{kl})\Big). \end{aligned} \tag{5}$$

Notice that the expected values of $\{h_{ij}\}$ and $\{t_{jlk}\}$ are respectively $P(h = j|d_i)$ and $P(t = k|w, h = j)$. Hence, the EM algorithm goes as follows.

In the **E-step** the complete log-likelihood is estimated, by estimating the posteriors of h_{ij} and t_{jkl} as follows:

$$\begin{aligned} P_{ij} &= E[h_{ij}] = P(h=j|d_i) \\ &= \frac{\pi_j P(d_i|h=j)}{\sum_{q=1}^{J} \pi_q P(d_i|h=q)} \\ &= \frac{\pi_j \prod_{w_l \in d_i} \left[\sum_{k=1}^{K} \tau_{jk}\beta_{kl}\right]^{n(w_l,d_i)}}{\sum_{q=1}^{J} \pi_q \prod_{w_l \in d_i} \left[\sum_{k=1}^{K} \tau_{qk}\beta_{kl}\right]^{n(w_l,d_i)}} \end{aligned} \tag{6}$$

$$Q_{jkl} = E[t_{jlk}] = P(t = k|w_l, h = j) = \frac{\tau_{jk}\beta_{kl}}{\sum_{p=1}^{K} \tau_{jp}\beta_{kp}}. \quad (7)$$

In the **M-step** the expected log-likelihood $E[\mathcal{L}_{comp}]$, is maximized under the normalization constraints, using the posteriors estimated in the previous step. The maximum is obtained for the following parameter values:

$$\pi_j = P(h = j) = \frac{\sum_{i=1}^{N} P_{ij}}{\sum_{q=1}^{J}\sum_{i=1}^{N} P_{iq}} = \frac{\sum_{i=1}^{N} P_{ij}}{N}, \quad (8)$$

given that $\sum_{q=1}^{J} P_{iq} = \sum_{q=1}^{J} P(h = q|d_i) = 1$,

$$\begin{aligned}\tau_{jk} &= P(t = k|h = j) \\ &= \frac{\sum_{i=1}^{N} P_{ij} \sum_{w_l \in d_i} Q_{jkl} n(w_l, d_i)}{\sum_{p=1}^{K}\sum_{i=1}^{N} P_{ij} \sum_{w_l \in d_i} Q_{jpl} n(w_l, d_i)} \\ &= \frac{\sum_{i=1}^{N} P_{ij} \sum_{w_l \in d_i} Q_{jkl} n(w_l, d_i)}{\sum_{i=1}^{N} P_{ij} |d_i|}, \quad (9)\end{aligned}$$

$$(10)$$

given that $\sum_{p=1}^{K} P(t = p|w_l, h = j) = 1$, and

$$\begin{aligned}\beta_{kl} &= P(w_l|t = k) \\ &= \frac{\sum_{i=1}^{N}\sum_{j=1}^{J} P_{ij} Q_{jkl} n(w_l, d_i)}{\sum_{m=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{J} P_{ij} Q_{jkm} n(w_m, d_i)}. \quad (11)\end{aligned}$$

$$(12)$$

where M is the size of the dictionary.

3.2 Stochastic Gradient Ascent Optimization

If tables τ and β are represented as MLPs, a Gradient Ascent optimization algorithm can be used in order to learn the corresponding parameters. This can be used to represent tables in a more distributed manner, as well as a way to control the capacity of these tables. We propose a Stochastic Gradient Ascent algorithm optimizing the log-likelihood criterion (4) under the normalization constraints:

$$\mathcal{H} = \mathcal{L}(D; \pi, \tau, \beta) + \rho\left(1 - \sum_j \pi_j\right) \quad (13)$$

$$+\sum_j \lambda_j \left(1 - \sum_k \tau_{jk}\right) + \sum_k \eta_k \left(1 - \sum_l \beta_{kl}\right). \quad (14)$$

where ρ, λ_j and η_k are Lagrange multipliers.

For each document d_i, the gradient of $\mathcal{H}$ with respect to the log-parameters will be:

$$\frac{\partial \mathcal{H}}{\partial[\ln \pi_j]} = P_{ij} - \sum_{q=1}^{J} P_{iq} \quad (15)$$

$$\begin{aligned}\frac{\partial \mathcal{H}}{\partial[\ln \tau_{jk}]} &= P_{ij}\Bigg[\sum_{w_l \in d_i} Q_{jkl} n(w_l, d_i) \\ &\quad - \sum_{p=1}^{K} \sum_{w_l \in d_i} Q_{jpl} n(w_l, d_i)\Bigg] \end{aligned} \tag{16}$$

$$\begin{aligned}\frac{\partial \mathcal{H}}{\partial[\ln \beta_{kl}]} &= \sum_{j=1}^{J} P_{ij} Q_{jkl} n(w_l, d_i) \\ &\quad - \sum_{m=1}^{M} \sum_{j=1}^{J} P_{ij} Q_{jkm} n(w_m, d_i). \end{aligned} \tag{17}$$

The parameters ζ_m of the model can thus be updated after each document d_i, as follows:

$$\zeta_m = \zeta_m + \epsilon \frac{\partial \mathcal{H}(d_i; \zeta_m)}{\partial \zeta_m} \tag{18}$$

where ϵ represents a small learning rate.

4 A Related Model: LDA

The Latent Dirichlet Allocation model (LDA) [**?**] is very similar to TTMM. The main difference is that instead of considering the number of themes to be finite, in LDA it is considered as infinite. This infinity of choices for the proportions of the mixture over the latent topics is obtained by a Dirichlet distribution. Thus, under LDA model, the probability of a document d can be written as:

$$P(d|\alpha, \beta) = \int P(\theta|\alpha) \prod_{w_l \in d} \left[\sum_{k=1}^{K} \beta_{kl} P(t = k|\theta)\right]^{n(w_l, d)} d\theta, \tag{19}$$

where $n(w_l, d)$ is the frequency of the word w_l in d, $\beta_{kl} = P(w_l|t = k)$, θ is the K-dimensional Dirichlet random variable (with $\sum_{k=1}^{K} \theta_k = 1$) and $P(\theta|\alpha)$ is the Dirichlet probability density of θ.

This difference makes the computation of equation (19) intractable by exact inference. Hence, in order to learn the parameters of the model a variational approximation is proposed [**?**]. Indeed, the log-probability of each document is approximated by a lower bound depending on the variational distribution $q(\theta, t|\gamma, \phi)$, which is an approximation for fixed α, β and d of the posterior distribution $p(\theta, t|d, \alpha, \beta)$. The document log-probability can be decomposed as follows:

$$\begin{aligned}&\ln[P(d|\alpha, \beta)] = L_d((\gamma, \phi); (\alpha, \beta)) \\ &+ D_{KL}(q(\theta, t|\gamma, \phi) \| p(\theta, t|d, \alpha, \beta))\end{aligned} \tag{20}$$

where γ and ϕ are the variational parameters, $L_d((,); (\alpha, \beta))$ is $\ln[P(d|\alpha, \beta)]$'s lower bound $\forall \alpha, \beta$ and $D_{KL}(\|)$ is the Kullback-Leibler divergence. For the log-likelihood maximization, two aims must be reached:

1. The lower bound has to be the closest possible to the log-probability, which is obtained for γ_d^* and ϕ_d^* maximizing $L_d((\gamma, \phi); (\alpha, \beta))$.

2. The log-likelihood has to be maximum with respect to the original parameters, which is obtained by α^*,β^* maximizing $\sum_d L_d((\gamma_d^*, \phi_d^*); (\alpha, \beta))$.

This leads to the variational EM proposed in [**?**], where in the **E-step** an iterative algorithm is run to find γ_d^* and ϕ_d^* and in the **M-step** the optimal α^*,β^* are computed.

5 TTMM vs LDA

In this section we compare TTMM to LDA on several characteristics, namely their ability in being applied to a Dimensionality Reduction task, a Clustering task and a Supervised task, as well as Time and Space Complexity.

Dimensionality Reduction application : These two density estimation methods can be used, for instance, as a Dimensionality Reduction method for the *bag-of-words* representation. The idea is that instead of considering words as basic units of document representation we could consider a topic basis, with the hope that a few topics would capture more information than the huge amount of words.

In the case of LDA, it was proposed in [**?**] to use the variational parameter $\gamma_d^* \in \mathbf{R}^K$ as representing document d. Since γ_d^* is a distribution that approximates the Dirichlet parameters $P(\theta_d|\alpha)$, it provides a representation in the topic space.

In the case of TTMM, we could choose for instance the posterior of topic components given the document: $P(t=k|d) = \frac{P(t=k,d)}{P(d)}$, where

$$P(t=k,d) = \sum_{j=1}^{J} \pi_j P(t=k,d|h=j) \tag{21}$$

$$= \prod_{w_l \in d} [P(w_l|t=k)]^{n(w_l,d)} \sum_{j=1}^{J} \pi_j \tau_{jk}. \tag{22}$$

Similarly, we could represent documents using a theme basis, or even a combination of both.

Clustering application : Contrary to LDA, TTMM density estimation can also be seen as a soft clusterization of documents in few themes. This can be a useful corpus representation, for example, in order to speed up an Information Retrieval task [**?**].

Supervised task application : We could also use TTMM directly in a supervised task such as Text Categorization. Indeed, we can identify themes with categories, and let for instance the probability of theme j be $\pi_j = freq(category\, j)$. In this case, the modelisation is very similar to the one proposed by H. Li and K. Yamanishi [**?**]. We can also imagine applying LDA directly to a Text Categorization task by learning as many LDA models as categories. But the parameters of the TTMM solution would probably be better estimated than those of LDA since a same parameter could help solving several different classification problems, and thus would have more data to estimate it.

Time Complexity : Let N be the number of documents in a corpus, M the size of the dictionary associated to the corpus, $|d|$ the number of words in the document d, K the number of topics and J the number of themes. As displayed in Table 1, each EM iteration for maximizing the TTMM likelihood has a complexity in time of $\mathcal{O}(NKJ[|d|+M])$, while each variational EM iteration seems to have one of $\mathcal{O}(NK|d|[|d|+M])$. Both TTMM and LDA are well-defined generative models, thus we are able to infer the probability of any new document d. In the case of TTMM we can infer the exact $P(d)$ with a complexity in time of $\mathcal{O}(JK|d|)$. For LDA we can only infer the optimal lower bound of the probability, and this operation has a complexity in time of $\mathcal{O}(K|d|^2)$. Comparing the complexities of TTMM and LDA, we notice that the number of themes J in TTMM is replaced by the size $|d|$ of a document in LDA's formula. Thus we can imagine that for a corpus containing long documents TTMM will have a better time complexity than LDA.

Space Complexity : As shown in Table 2, from a memory complexity point of view, LDA is more parsimonious than TTMM. Indeed, LDA with parameters (α, β) has a space complexity of $\mathcal{O}(KM)$, while TTMM with (π, τ, β) has one of $\mathcal{O}(K[M+J])$.

	LDA	TTMM
EM-step	$\mathcal{O}(NK\lvert d\rvert[\lvert d\rvert+M])$	$\mathcal{O}(NKJ[\lvert d\rvert+M])$
Inference	$\mathcal{O}(K\lvert d\rvert^2)$	$\mathcal{O}(KJ\lvert d\rvert)$

Table 1: Time Complexities for LDA and TTMM

	LDA	TTMM
Parameters	$\mathcal{O}(KM)$	$\mathcal{O}(K[M+J])$

Table 2: Space Complexities for LDA and TTMM

6 Experiments

In this section, an experiment comparing LDA, TTMM, and the bag-of-word representation is reported. In [?], LDA's features and bag-of-words document representations were compared on a Text Categorization task using support vector machines (SVMs) as classifiers. Using the same data (a subset of Reuters-21578), splits and experimental protocol, the experiment is repeated here with TTMM.

For this experiment, the authors of [?] have selected 8529 Reuters-21578's documents (almost all the training data of ModApte split), they stopped, but did not stemmed the data, and from the resulting vocabulary they discarded the less frequent words to finally obtain a vocabulary of 15810 words. An LDA model with 50 topics was trained on all the documents, without reference to their class labels. For each proportion $p \in \{0.01, 0.05, 0.1, 0.2\}$ of the training data, they trained a support vector machine (SVM) on the LDA's features document representation as explained in section 5 (in the Dimensionality Reduction paragraph), for a binary classification problem[2].

For several numbers of themes TTMMs were trained on all the documents without references to their class labels. The models were optimized with the EM algorithm, using tables to represent the various conditional distributions.

For several proportions p of the training data, an SVM with a Gaussian kernel was trained on TTMM's topic-based features document representation for the same binary classification problem as described in the experimental section of [?]. The standard deviation of the kernel was tuned using a 5-fold cross-validation procedure on the training data. The results on the remaining $1-p$ proportion of the data were compared to the ones of SVMs trained on the bag-of-words representation, and SVMs trained on LDA's features, as reported in [?]. Fig. 2 summarizes these results for category GRAIN.

Note however that these results are optimistic, and not comparable with other Text Categorization published results, since the vocabulary was extracted from both training and test sets. Thus the problem of having unseen words in the test set is not addressed. Nevertheless, to make a comparison between TTMM and LDA, we followed the same experimental protocol as described in [?].

For the reported numbers of themes J (500, 1000) and topics K (50), the features obtained with TTMM give in general as good results as the LDA features and even a better one for proportion $p = 0.05$. Furthermore, we can see in this experiment that TTMM does capture important information from the data, since even with 99.6% less features than the bag-of-words representation (50 vs 15810), the results are better for small values of p.

[2]Reuters-21578 documents are labeled with one or several categories among 115 possibles. A one-against-the-others approach is here considered for the GRAIN category.

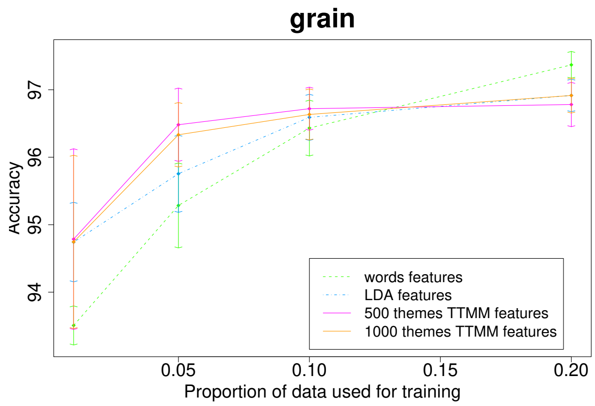

Figure 2: Classification results on GRAIN vs. NOT GRAIN binary classification problem for several proportions of training data, and several features. TTMM features where computed for 50 topics and several numbers of themes, using EM optimization.

7 Conclusion

In this paper, we presented a new document density estimation model, the Theme Topic Mixture Model (TTMM), and we compared it to LDA, a very similar model. TTMM appears to reach reasonable performance, close to LDA. Advantages of TTMM over LDA were discussed in the paper. For instance, contrary to LDA, TTMM can be inferred exactly; moreover, viewing TTMM as a discretized version of LDA, we could use it to solve some applications that are not accessible to LDA. Using TTMM or LDA for document representation instead of the bag-of-words representation has proved to give a good dimensionality reduction of the input space without performance loss, at least when training on small corpora. We plan to do further experiments on exploring TTMM advantages over LDA, and over bag-of-words representation. For instance, using TTMM directly to solve a text categorization task may be a promising issue.

Acknowledgments

This research has been carried out in the framework of the Swiss NCCR project (IM)2 and in the framework of the PASCAL European Network of Excellence, funded by the Swiss OFES.

References