# Theme Topic Mixture Model: A Graphical Model for Document Representation

**Mikaela Keller** MKELLER@IDIAP.CH
**Samy Bengio** BENGIO@IDIAP.CH
*IDIAP, CP 592, rue du Simplon 4, 1927 Martigny, Switzerland*

## 1. Introduction

In order to be automatically processed, textual data must be represented formally. The basic and widely used indexing method, for Text Categorization and other supervised related problems, is the *bag-of-words* document representation [4].

Several other document representations have been proposed in the literature, in particular, some methods based on Graphical Models, such as Latent Dirichlet Allocation (LDA) [2] and Probabilistic Latent Semantic Analysis (PLSA) [3]. 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 every machine having 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.

## 2. Theme Topic Mixture Model

The proposed model has a lot in common with LDA (see section 3), 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 $\pi, \tau$ and $\beta$ 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.

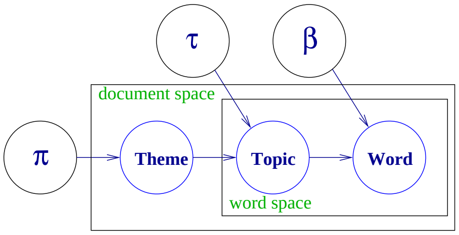

Figure 1: TTMM graphical model

The 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 $\{1, \dots, K\}$ 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 $\xi$, 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 mixture 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 $\beta_{kl} = P(w_l|t = k)$.

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 $d$.

[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.

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

The log-likelihood of the model on the corpus then becomes:

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

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 Descent optimization techniques.

### 2.1 Expectation-Maximization Optimization

In order to perform an EM optimization 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 variable specifying which theme $j$ the document $d_i$ was generated from, and $\{t_{jlk}\}$ the indicator variable specifying which topic $k$ the word $w_l$ was generated from given that we are in the theme $j$ context. Indeed, the complete log-likelihood could be written as:

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

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 as follows:

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

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

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:

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

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

$$\begin{aligned} \tau_{jk} = P(t=k|h=j) &= \frac{\sum_{i=1}^{N} P_{ij} \sum_{w_l \in d_i} Q_{jkl} n(w_l, d_i)}{\sum_{p=1}^{K}\sum_{i=1}^{N} P_{ij} \sum_{w_l \in d_i} Q_{jpl} n(w_l, d_i)} \\ &= \frac{\sum_{i=1}^{N} P_{ij} \sum_{w_l \in d_i} Q_{jkl} n(w_l, d_i)}{\sum_{i=1}^{N} P_{ij} \#(d_i)}, \qquad (9) \\ & \qquad (10) \end{aligned}$$

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

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

where $M$ is the size of the dictionary.

### 2.2 Stochastic Gradient Descent Optimization

If the tables $\tau$ and $\beta$ are represented as MLPs, a Gradient Descent optimization algorithm can be used in order to learn the corresponding parameters. We propose a Stochastic Gradient Descent algorithm optimizing the log-likelihood criterion (4) under the normalization constraints:

$$\mathcal{H} = \mathcal{L}(\pi, \tau, \beta) + \rho\left(1 - \sum_j \pi_j\right) + \sum_j \lambda_j \left(1 - \sum_k \tau_{jk}\right) + \sum_k \eta_k \left(1 - \sum_l \beta_{kl}\right). \tag{12}$$

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

$$\frac{\partial \mathcal{H}}{\partial[\ln \pi_j]} = P_{ij} - \sum_{q=1}^{J} P_{iq} \tag{13}$$

$$\frac{\partial \mathcal{H}}{\partial[\ln \tau_{jk}]} = P_{ij}\left[\sum_{w_l \in d_i} Q_{jkl} n(w_l, d_i) - \sum_{p=1}^{K}\sum_{w_l \in d_i} Q_{jpl} n(w_l, d_i)\right] \tag{14}$$

$$\frac{\partial \mathcal{H}}{\partial[\ln \beta_{kl}]} = \sum_{j=1}^{J} P_{ij} Q_{jkl} n(w_l, d_i) - \sum_{m=1}^{M}\sum_{j=1}^{J} P_{ij} Q_{jkm} n(w_m, d_i). \tag{15}$$

## 3. A Related Model: LDA

The Latent Dirichlet Allocation model (LDA) [2] 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, the probability of a document can be written as:

$$P(d|\alpha,\beta) = \int P(\theta|\alpha) \prod_{w_l \in d} \left[ \sum_{k=1}^{K} \beta_{kl} P(t=k|\theta) \right]^{n(w_l,d)} d\theta, \tag{16}$$

where $n(w_l, d)$ is the frequency of the word $w_l$ in $d$, $\beta_{kl} = P(w_l|t = k)$, $\theta$ 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 $\theta$.

This difference makes the computation of equation (16) intractable by exact inference. Hence, in order to learn the parameters of the model a variational approximation is proposed [2]. 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 $\alpha$, $\beta$ and $d$ of the posterior distribution $p(\theta, t|d, \alpha, \beta)$. The document log-probability can be decomposed as follows:

$$\ln\left[P(d|\alpha,\beta)\right] = L_d((\gamma,\phi);(\alpha,\beta)) + D_{KL}(q(\theta,t|\gamma,\phi)\|p(\theta,t|d,\alpha,\beta)) \tag{17}$$

where $\gamma$ and $\phi$ 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 $\gamma_d^*$ and $\phi_d^*$ 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 $\alpha^*$,$\beta^*$ maximizing $\sum_d L_d((\gamma_d^*, \phi_d^*);(\alpha,\beta))$.

This leads to the variational EM proposed in [2], where in the **E-step** an iterative algorithm is run to find $\gamma_d^*$ and $\phi_d^*$ and in the **M-step** the optimal $\alpha^*$,$\beta^*$ are computed.

## 4. TTMM vs LDA

In this section we compare TTMM to LDA on several characteristics:

**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 has been proposed in [2] to use the variational parameter $\gamma_d^* \in \mathbb{R}^K$ as representing document $d$. Since $\gamma_d^*$ is a distribution that approximates the Dirichlet parameters $P(\theta_p|\alpha)$, it provides a representation in the topic space.

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

$$P(t = k, d) = \sum_{j=1}^{J} \pi_j P(t = k, d|h = j) = \prod_{w_l \in d} [P(w_l|t = k)]^{n(w_l,d)} \sum_{j=1}^{J} \pi_j \tau_{jk}. \qquad (18)$$

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 [1].

**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 $\pi_j = freq(category\ j)$. 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 should be better estimated than those of LDA since a same parameter could help solving two different classification problems, and thus will 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. Each EM iteration for maximizing the TTMM likelihood has a complexity in time of $\mathcal{O}(NK[J|d| + JM])$, while each variational EM iteration seems to have one of $\mathcal{O}(NK[|d|^2 + JM])$. 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 complexity than LDA.

**Space Complexity** : From a memory complexity point of view LDA is more parcimonious than TTMM. Indeed, LDA with parameters $(\alpha, \beta)$ has a space complexity of $\mathcal{O}(KM)$, while TTMM with $(\pi, \tau, \beta)$ has one of $\mathcal{O}(K[M + J])$.

## 5. Experiments

In this section, an experiment comparing LDA and TTMM is reported. In [2], LDA's features and bag-of-words document representations are compared in a Text Categorization task using support vector machines (SVMs). Using the same data (a subset of Reuters-21578), splits and experimental protocol, the experiment is repeated here with TTMM.

For several numbers of themes a TTMM was trained on all the documents without references to their class labels. For several proportions $p$ of the training data, an SVM was trained on TTMM's features document representation for a category binary classification problem[2]. 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 LDA's features reported in [2]. The results for category GRAIN are illustrated in Fig. 2.

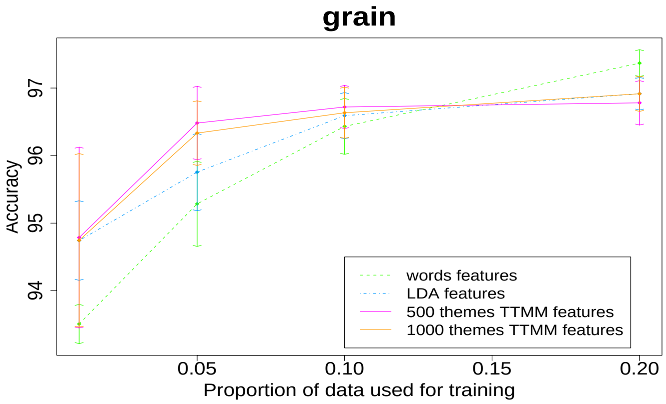

Figure 2: Classification results on GRAIN vs. NOT GRAIN binary classification problem for several proportions of training data, and several features. TTMM features where computed for 50 topics and several numbers of themes, using EM optimization.

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 $p$.

## 6. 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 results, close to LDA performances. Advantages of TTMM over LDA were discussed in the paper. For instance, contrary to LDA, TTMM can be infered 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 little proportion of training data. We plan to do futher experiments on 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.

[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.

## 7. Acknowledgments

This research has been carried out in the framework of the Swiss NCCR project (IM)2 and in the framework of PASCAL.

## References

[1] Mayank Bawa, Roberto J. Bayardo Jr., and Rakesh Agrawal. Sets: Search Enhanced by Topic-Segmentation. In *Proceedings of the 26nd Annual ACM Conference on Research and Development in Information Retrieval*, Berkeley, California, August 2003.

[2] David Blei, Andrew Ng, and Michael Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, January 2003.

[3] Thomas Hofmann. Unsupervised learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42:177–196, 2001.

[4] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.