Theme Topic Mixture Model for Document Representation

Mikaela Keller MKELLER@IDIAP.CH
Samy Bengio BENGIO@IDIAP.CH
IDIAP Research Institute, CP 592, rue du Simplon 4, 1920 Martigny, Switzerland

Abstract

In Automatic Text Processing tasks, documents are usually represented in the bag-of-words space. However, this representation does not take into account the possible relations between words. We propose here a review of a family of document density estimation models for representing documents. Inside this family we derive another possible model: 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. An experiment reports the performance of the different models in this family over a common task.

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 [10].

Starting from this simple representation, several other document representations have been proposed in the literature trying to overcome some problems inherent to the bag-of-words representation. Some, like, the Latent Semantic Indexing (LSI) [5], are based on an algebraic linear transformation of the term by document matrix. Others, based on Graphical Models, such as Latent Dirichlet Allocation (LDA) [2] and Probabilistic Latent Semantic Analysis (PLSA) [7] estimate the density of the documents given a particular model.

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 are required to estimate in any system taking bag-of-words documents as inputs, which leads easily to the curse of dimensionality.

In this paper, we present another Graphical Model, the Theme Topic Mixture Model (TTMM). This model, like LSI, the linear algebraic method, or like PLSA and LDA from the same family of models, tries to overcome the bag-of-words representation problems. Indeed, with all these methods we can achieve 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, gathering together the words in “concepts” is meant to disambiguate the cases of synonymic or polysemic use of language.

The paper is organized as follows. In Section 2, we quickly explain the general document representation problem. In Section 3 we first present PLSA, LDA and TTM models and then compare them on several theoretical aspects. Finally, Section 4 reports an experiment comparing different document representations.

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 for convenience 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 of the data as well as the document representation itself.

As explained in [10], documents are often represented by a vector $d = (q_1, ..., q_M)$ of weights q_j, assigned to every word w_j in a vocabulary $\mathcal{V}$ of size M. 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 usually have 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 the semantic links between words included in this representation. Nevertheless, there are other approaches to represent documents, which take into account this kind of information. Among these approaches, we find Latent Semantic Indexing (LSI) [5] which is a linear transformation of the data, and also probabilistic approaches in which the density of documents in the Vector space is estimated according to a model. In the LSI approach a Singular Value Decomposition of the matrix A, whose columns are documents in the bag-of-words representation, is performed. Documents are then represented by their projections on the K first eigenvectors of A. Each eigenvector is a linear combination of the words' space basis' vectors. These combinations explain the name of "latent semantic" representation, since they tend to gather together words according to their co-occurrence rate. However since our aim is to represent the documents by a few components, highlighting some "concepts" present in the document rather than its words, shouldn't we learn that directly?

In the following, a family of document density estimation models is presented in which high level statistical correlations between words in a corpus are assumed through the use of a hidden variable that we will call **Topic**.

3. Document Density Estimation

The graphical models mentioned before, *ie* the Probabilistic Latent Semantic Analysis (PLSA) from Hofmann [7], the Latent Dirichlet Allocation (LDA) from Blei *et al* [2] (also referred to as multinomial Principal Component Analysis (mPCA), proposed by Buntine *et al* [3]), and the Theme Topic Mixture Model (TTMM) which will be defined in the following, have in common their main idea. This idea is to assume that the choice of words in the generation of a document is independent of the document given a hidden variable which is called **Topic** or sometimes Aspect.

Fig. 1 is a graphical representation of the generic structure of these models. In the Figure, the boxes represent replicates. The outer box represents the repeated trials of a random variable X in the document space (N is the number of documents), while the inner box represents the repeated choice of topics within the word space (M is the vocabulary size) for a given value of the variable X. As shown in this Figure the variable X in the document space and the **Word** variable in the word space have no direct dependencies. In addition, in these three models, multiple topics dependencies over words are not taken into account and the probability of a word given the variable X is seen as a mixture of multinomials over the topics.

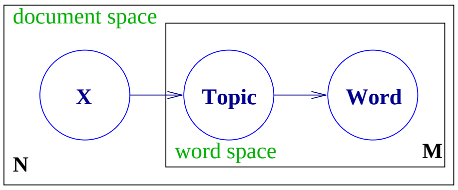

Figure 1: A graphical representation of the generic structure of a document density model.

The main difference between these models lies in the way variable X is defined and from which distribution its values are drawn. In the following a description of each of these models is given as well as a comparison between them.

3.1 Probabilistic Latent Semantic Analysis

The main distinctive feature of PLSA is that it seeks a generative model for **word/document co-occurrences**, rather than a model for documents themselves. From that, it follows that the variable in the document space called δ (see Fig. 2), is a variable that picks one document among the others in the database, since we need to model each word occurrence in each document. To say it differently, δ takes a value among the document indexes $\{1, \ldots, N\}$, the probability $P(\delta)$ being proportional to the length of the δ^{th} document. The assumption is that each word w_j in a given document d_δ is generated from a latent **Topic** t taking values among $\{1, \ldots, K\}$, K being a chosen hyperparameter. The data generation process can be described as follows:

1. Select a document index δ with probability $P(\delta)$
2. Pick a latent topic $t = k$ with probability $P(t = k|d_\delta)$

3. Generate a word w_j with probability $P(w_j|t=k)$

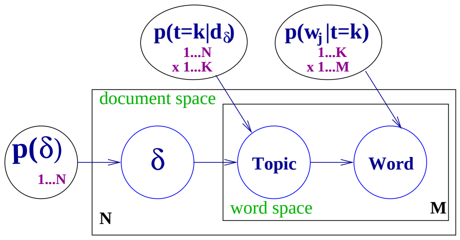

Figure 2: A graphical representation of PLSA

This generative process is summarized by the joint distribution of a word w_j, a latent topic $t=k$, and a document d_δ :

$$P(w_j, t=k, d_\delta) = P(\delta)P(t=k|d_\delta)P(w_j|t=k),$$

and the joint distribution of the observed data is thus:

$$P(d_\delta, w_j) = P(\delta)\sum_{k=1}^{K} P(t=k|d_\delta)P(w_j|t=k). \tag{1}$$

So each word in a document is seen as a sample from a mixture model where mixture components are multinomial $P(w_j|t=k)$ and the mixing proportions are $P(t=k|d_\delta)$.

The log-likelihood of the model,

$$\mathcal{L} = \sum_{\delta=1}^{N}\sum_{j=1}^{M} n_j^\delta \log P(d_\delta, w_j)$$

with n_j^δ the frequency of the word w_j in d_δ, is maximized by Expectation-Maximization (EM) [6], as follows:

The **E-step** consists of computing the posterior probabilities for the latent variable, based on the current estimates of the parameters, that is:

$$P(t=k|w_j, d_\delta) = \frac{P(w_j|t=k)P(t=k|d_\delta)}{\sum_l P(w_j|t=l)P(t=l|d_\delta)}.$$

The **M-step** consists of the maximization of the expected joint log-likelihood of the observed and latent variables given the estimations of the previous step. This is achieved in PLSA with the following parameters re-estimation:

$$P(w_j|t=k) = \frac{\sum_{\delta=1}^{N} n_j^\delta P(t=k|w_j, d_\delta)}{\sum_{m=1}^{M}\sum_{\delta=1}^{N} n_m^\delta P(t=k|w_m, d_\delta)}$$

$$P(t=k|d_\delta) = \frac{\sum_{j=1}^{M} n_j^\delta P(t=k|w_j, d_\delta)}{\sum_{j=1}^{M} n_j^\delta}.$$

PLSA can be used to replace the original document representation by a representation in a low-dimensional "latent" space, in order to perform a Text Categorization or a Document Retrieval task. In [7], the components of the document in the low-dimensional space are chosen to be $P(t = k|d), \forall k$, and for each unseen document or query they are computed by maximizing the log-likelihood with $P(w_j|t = k)$ fixed. This representation scheme is referred to as PLSI, for Probabilistic Latent Semantic Indexing. It has been pointed out in [2] that PLSA is not a well-defined generative model of documents, since there is no direct way to assign probability to an unseen document. However, some experiments in [7] report a comparison between LSI and PLSI, on several corpora. They point out a better performance of PLSI in all cases. In particular PLSI performs well even in the cases where LSI fails completely.

The other weakness of PLSA can be described as follows. The parameters of a K-topics PLSA model are the K multinomials of size M and the K mixing proportions for each of the N documents. Hence, the number of parameters equals $KM + KN$ and therefore grows linearly with the number of documents. This suggests that the model is prone to overfitting. In practice, to try to overcome this problem, a tempered EM (TEM) is performed instead of the EM. During the TEM iterations the parameters are smoothed in order to achieve an acceptable predictive performance on a validation set. However, according to [2], overfitting can occur even with the TEM version and it is likely with large corpora.

3.2 Latent Dirichlet Allocation

In LDA the documents are assumed to be sampled from a random mixture over latent topics, where each topic is characterized by a distribution over words. In this model the observed variable is the document d, seen as a set of words w_j, $j \in \{1, \ldots, M\}$, each of these words being dependent of the unobserved variable t, the **Topic**, with possible values in $\{1, \ldots, K\}$, and K being an hyperparameter that must be chosen. In the document space the LDA model has another unobserved variable, $\theta = (\theta_1, \ldots, \theta_K)$, $\theta_k > 0$, $\sum_{k=1}^{K} \theta_k = 1$ (see Fig. 3), responsible for the mixing proportions of the topics in each document. The generative process for each document d is the following:

1. Choose $n(d) \sim Poisson(\xi)$: the document size
2. Choose $\theta \sim Dirichlet(\alpha)$: the random mixing proportions
3. For each of the $n(d)$ words of d:
   (a) Choose a topic $t = k$ from $P(t|\theta)$, a multinomial probability with parameter θ
   (b) Choose a word w_j from $P(w|t = k)$, a multinomial probability conditioned on the topic $t = k$

where $\alpha = (\alpha_1, \ldots, \alpha_K)$, $\alpha_k > 0$, is a parameter to estimate. The randomness of the document size $n(d)$, modeled for example by a Poisson distribution with parameter ξ, is necessary for the generative process. However, given that $n(d)$ is independent of all the other data generating variables (θ and t), it is not of real interest for the modelisation.[1] Hence, it will be ignored.

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

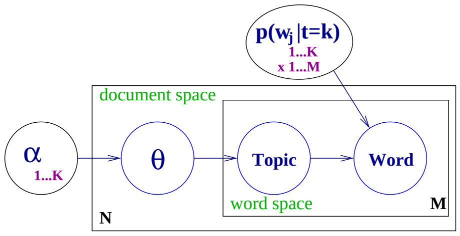

Figure 3: A graphical representation of LDA/mPCA

Given this generative model, we can write the joint distribution of a word and a topic as:

$$P(w_j, t = k|\theta, \alpha) = P(w_j|t = k)P(t = k|\theta).$$

Summing over k, we obtain the marginal distribution of a word,

$$P(w_j|\theta, \alpha) = \sum_{k=1}^{K} P(w_j|t = k)P(t = k|\theta).$$

Hence, the joint distribution of document d (*i.e.* a set of $|d|$ words) and a topic mixture θ is given by:

$$P(\theta, d|\alpha) = P(\theta|\alpha) \prod_{j=1}^{M} \left[\sum_{k=1}^{K} P(w_j|t = k)P(t = k|\theta) \right]^{n_j^d}$$

where n_j^d is the frequency of the word w_j in d and $P(\theta|\alpha)$ the Dirichlet probability density of θ. Finally, integrating over θ, we obtain the marginal distribution of a document,

$$P(d|\alpha) = \int P(\theta|\alpha) \prod_{j=1}^{M} \left[\sum_{k=1}^{K} P(w_j|t = k)P(t = k|\theta) \right]^{n_j^d} d\theta. \tag{2}$$

LDA, contrary to PLSA, is a true generative model of documents since both observed and unseen documents can be generated by the process described above. The parameters of a K-topics LDA model are $\alpha \in \mathbf{R}^{+K}$ the Dirichlet parameter and the M parameters of each of the K multinomial estimating $P(w|t = k), \forall k$. That is $K + KM$ parameters for LDA, which is less than for PLSA and in addition independent of the number of documents.

However, in order to estimate these parameters one has to compute the posterior distribution $P(\theta, t|d)$, which is intractable in general, according to [2]. Therefore, instead of doing an exact inference for LDA, the authors of the paper propose an approximate inference algorithm based on a variational method [8]. Indeed, their algorithm maximizes a lower bound on the log-likelihood based on a variational distribution that approximates the posterior distribution $P(\theta, t|d)$. This maximization is done by a so-called variational EM algorithm, which consists in the iteration of the following two steps:

1. **(E-step)** Variational approximation of the posterior distribution. This is performed by an iterative algorithm, which requires approximatively $[n(d)]^2K$ operations for each document, according to [2].

2. **(M-step)** Maximize the resulting lower bound on the log-likelihood with respect to the parameters of the model.

In order to represent the documents in a space with a lower dimension than the bag-of-words space (for instance to perform a supervised task), the authors of the paper have chosen K variational parameters from the posterior distribution approximation. This gives a representation of documents in terms of topics instead of a representation in terms of words.

For many reasons mentioned above LDA is an interesting model of document density. However, the approximate inference algorithm is not easy to implement. Therefore, as a first step, another model, tractable by exact inference, is proposed in the following section.

3.3 Theme Topic Mixture Model

In TTMM, the variable in the document space is called **Theme**. Each theme is characterized by a particular value for the mixing proportions over the topics. TTMM is very similar to LDA, but instead of using a continuous space for the choice of the mixing proportions of the topics, the choice is constrained to a discrete finite set. 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 of the model that have to be estimated are the tables representing the mixing proportions of themes, the mixing proportions of topics given the themes and the probability of each word given each topic.

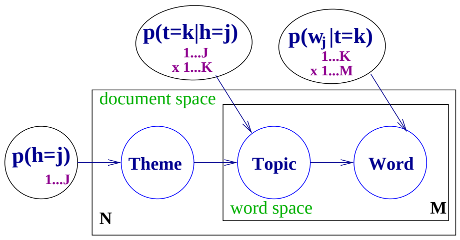

Figure 4: A graphical representation of TTMM

The underlying generative process for each document is the following:

1. Choose $n(d) \sim Poisson(\xi)$: the document size.
2. Choose a theme $h = j$ from $P(h)$, a multinomial distribution representing the mixing proportions.
3. For each of the $n(d)$ 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 $n(d)$, as for LDA, is necessary for the generative process, and for the same reason will be ignored (*cf* footnote 1).

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:

$$P(w_l|h=j) = \sum_{k=1}^{K} P(w_l|t=k)P(t=k|h=j).$$

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

$$P(d|h=j) = \prod_{l=1}^{M} [P(w_l|h=j)]^{n_l^d} = \prod_{l=1}^{M} \left[\sum_{k=1}^{K} P(w_l|t=k)P(t=k|h=j)\right]^{n_l^d},$$

where n_l^d is the frequency of the term w_l in d, with $\sum_l n_l^d = n(d)$. Finally, each document d is seen as a mixture of themes h:

$$P(d) = \sum_{j=1}^{J} P(h=j)P(d|h=j) = \sum_{j=1}^{J} P(h=j) \prod_{l=1}^{M} \left[\sum_{k=1}^{K} P(w_l|t=k)P(t=k|h=j)\right]^{n_l^d}. \tag{3}$$

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

$$\mathcal{L}(D) = \sum_{i=1}^{N} \log \left[\sum_{j=1}^{J} P(h=j) \prod_{l=1}^{M} \left(\sum_{k=1}^{K} P(w_l|t=k)P(t=k|h=j)\right)^{n_l^{d_i}}\right].$$

The maximization of this log-likelihood can be done by EM as for PLSA or by Gradient Ascent Optimization [9].

In the **E-step** the posterior probabilities of the latent variables are estimated, as follows:

$$\begin{aligned} P_{ij} &= P(h=j|d_i) = \frac{P(h=j)P(d_i|h=j)}{\sum_{q=1}^{J} P(h=q)P(d_i|h=q)} \\ &= \frac{P(h=j)\prod_{l=1}^{M}\left[\sum_{k=1}^{K} P(t=k|h=j)P(w_l|t=k)\right]^{n_l^{d_i}}}{\sum_{q=1}^{J} P(h=q)\prod_{l=1}^{M}\left[\sum_{k=1}^{K} P(t=k|h=q)P(w_l|t=k)\right]^{n_l^{d_i}}} \end{aligned}$$

$$Q_{jkl} = P(t=k|w_l, h=j) = \frac{P(t=k|h=j)P(w_l|t=k)}{\sum_{p=1}^{K} P(t=p|h=j)P(w_l|t=p)}.$$

The **M-step** as explained in Section 3.1, leads to a re-estimation of the parameters of the model:

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

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

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

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

$$P(w_l|t=k) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{J} n_l^{d_i} P_{ij} Q_{jkl}}{\sum_{m=1}^{M} \sum_{i=1}^{N} \sum_{j=1}^{J} n_m^{d_i} P_{ij} Q_{jkm}}.$$

As for PLSA and LDA, this density estimation method can then be used as a Dimensionality Reduction method on the *bag-of-words* representation. The idea is that instead of considering words as basic units of document representation we will consider a topic basis, with the hope that a few topics will capture more information than the huge amount of words. We can choose as topic component its posterior given the document, $P(t = k|d) = \frac{P(t=k,d)}{P(d)}$, where,

$$P(t=k,d) = \prod_{w_l \in d} [P(w_l|t=k)]^{n_l^d} \sum_{j=1}^{J} P(h=j) P(t=k|h=j).$$

3.4 Comparison

In the following we discuss and compare the three models described in the previous subsections. First note that these models are not completely new from a statistical point of view. Similar modelisations can be found in others domains such as for example the Tied Mixture used in Speech Processing [1].

Let us take a closer look at their main equations, (1), (2), and (3), since they summarize quite well the similarities and dissimilarities between the underlying models.

$$\textbf{PLSA:}\ P(d_\delta, w_l) = P(\delta) \sum_{k=1}^{K} P(t=k|d_\delta) P(w_l|t=k). \tag{1}$$

$$\textbf{LDA:}\ P(d) = \int P(\theta) \prod_{l=1}^{M} \left[\sum_{k=1}^{K} P(w_l|t=k) P(t=k|\theta) \right]^{n_l^d} d\theta. \tag{2}$$

$$\textbf{TTMM:}\ P(d) = \sum_{j=1}^{J} P(h=j) \prod_{l=1}^{M} \left[\sum_{k=1}^{K} P(t=k|h=j) P(w_l|t=k) \right]^{n_l^d}. \tag{3}$$

As it has been mentioned before, and can be noticed in the three equations, there is a common core which is the mixture of word multinomial over the topics. However, TTMM is something like an hybrid between PLSA and LDA. Even if PLSA is a word/document occurrence model and not a document model, we can say that PLSA would be a TTMM where the themes would have been identified with the documents. Furthermore, we can almost say that TTMM is a discretized LDA, or LDA a continuous TTMM. Indeed, where we have a discrete mixture of themes in TTMM, we have a continuous mixture of θ, both themes and θs defining the mixture proportions over the topics. Note however that by discretizing LDA in order to obtain TTMM, we loose something else appart from the continuity. Using a Dirichlet distribution for choosing the parameters of the multinomial over the topics constrains these parameters in a way that does not exist in TTMM. Another way of seeing that is that where LDA has K degrees of freedom, in the parameter $\alpha \in \mathbf{R}^{+K}$, TTMM has at least $J \times K$ degrees of freedom, in J multinomials $P(t \mid h = j)$ in the $(K-1)$-simplex.

TTMM needs $J(1+K)+KM$ parameters while LDA only needs $K+KM$. This is due to the fact that the continuous distribution with one parameter, which generates the mixing proportions θ in LDA, is replaced in TTMM by two discrete distributions, the multinomials representing $P(h)$ and $P(t|h=j)\forall j$. The number of parameters is possibly less than with PLSA (number of parameters: $KM+KN$), since we hope that documents can be clustered together by themes, and so $J < N$.

On the other hand, LDA optimization is intractable, so it has to be approximated. The variational EM algorithm has a complexity in time of $\mathcal{O}(NK\overline{n(d)}[\overline{n(d)}+M])$ at each step, where $\overline{n(d)}$ is the mean of the documents' lengths. Futhermore, because of its structure, PLSA tends to overfit, but training can be smoothed in order to reach and acceptable solution using Tempered EM. Each EM step for PLSA has a time complexity of $\mathcal{O}(NK\overline{n(d)}[\overline{n(d)}+M])$ like LDA. TTMM optimization can be reached with an exact inference but with a higher time complexity ($\mathcal{O}(NKJ[\overline{n(d)}+M])$) than the two others, if the mean of documents' lengths is smaller than the number of themes J.

4. Experiment

In this section, an experiment comparing LDA, PLSA, TTMM, and the bag-of-words representation is reported. In [2], LDA's features and bag-of-words document representations were compared on a Text Categorization task using support vector machines (SVMs) [4] as classifiers. Using the same data (a subset of Reuters-21578), splits and experimental protocol, the experiment is repeated here with TTMM and PLSA features.

The general procedure of the experiment is as follows:

1. A document density estimation model (LDA, PLSA or TTMM) is trained on a set of documents D.

2. The set D is split into a training set Tr_p containing a proportion p of the data, and a test set Te_p containing the remaining data.

3. An SVM is trained on Tr_p using for document representation the features extracted from the document density estimation model.

4. This SVM is tested on Te_p.

5. The steps 2., 3. and 4. are repeated for several splits and several values of p.

6. The experiment goes through steps 1. to 5. for each of the document density estimation models. The results obtained on the Te_p sets are compared to results obtained using SVMs trained on the bag-of-words representation of documents.

Note however that the results of this experiments are optimistic, and not comparable with other Text Categorization published results, since the vocabulary was extracted, and the models trained from the concatenation of Tr_p and Te_p sets. Thus the problem of having unseen words in the test set is not addressed. Nevertheless, in order to make a comparison between TTMM and LDA, we followed the same experimental protocol as described in [2].

We give here some details about the training of the models:

- Data: the authors of [2] have selected 8529 Reuters-21578's documents (almost all the training data of ModApte split) for the set called D below.
- Bag-of-words: They stopped but did not stem the data, and from the resulting vocabulary they discarded the less frequent words to finally obtain a vocabulary of 15810 words.
- LDA: A model with 50 topics was trained on all the documents, without reference to their class labels.
- PLSA: A model with 50 topics was trained by simple EM optimization rather than by TEM. We make this choice for the sake of simplicity (fewer hyperparameters to tune). However note that in this experiment letting PLSA overfit should favor it instead of harming its performance. Indeed, in this experiment, there is no true test set since the features are learned over the whole set D. The value 50 for the number of topics has been chosen to match LDA's choice.
- TTMM: Models with 50 topics and several values for the number of themes have been trained by EM using early stopping to control the capacity. With 500 and 1000 themes we obtained the highest likelihood among the number of themes values we tried. The value 50 for the number of topics has been chosen to match LDA's choice.
- SVMs: For PLSA and TTMM features, linear and Gaussian kernels were tried. The choice of the Gaussian kernel standard deviation was made using K-Fold cross-validation ($K = 5$) on each of the splits[2]. The Gaussian kernels give the best results, which are the ones that we report on the graphics shown in Fig. 5 and 6.

As can be seen in Fig. 5 and 6 the results obtained with the features extracted from the document density estimation models are comparable.

We can see in this experiment that the document density estimation models do 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.

However, when the proportion of data used for training the SVMs is bigger, the results do not show a clear advantage of the models' features over the bag-of-words representation.

2. We have not been able to sort out, what kind of kernel was used in the SVM trained on LDA features in [2], and have supposed thus in the analysis of the results that it may be sub-optimal. The same happens with the choice of the number of topics.

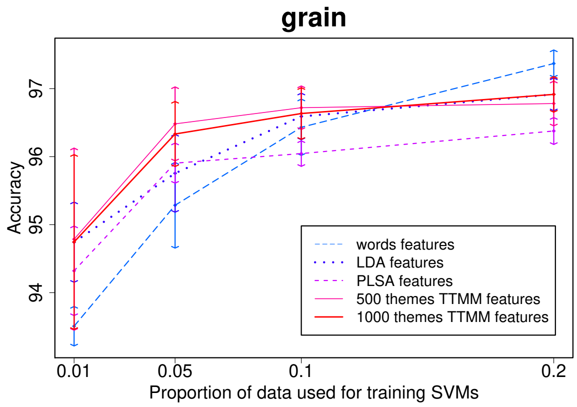

Figure 5: Classification results on GRAIN vs. NOT GRAIN binary classification problem for several proportions of training data, and several features.

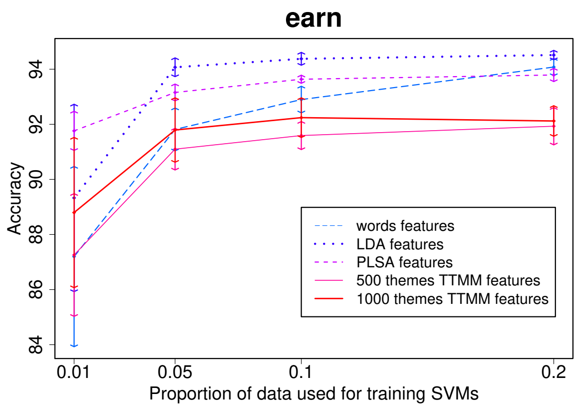

Figure 6: Classification results on EARN vs. NOT EARN binary classification problem for several proportions of training data, and several features.

This may be because there is not enough capacity, and the number of features is too small. This hyperparameter (the number of latent topics), should probably be tuned according to a criterion depending on the number of labeled documents that we have for training the SVMs, and not only on the maximum likelihood of the parameters over D. Another explanation could be that all these density models rely on constraints which may be too strong when there is enough data.

Looking in more detail the results for each of the two categories GRAIN (Fig. 5) and EARN (Fig. 6), we notice that the three models do not perform equally. Indeed, TTMM give overall better results for GRAIN than for EARN, while PLSA has the converse behaviour and LDA an overall good performance in both cases. We can explain this by the fact that classifying document as belonging to category EARN or category GRAIN, are different kinds of tasks for which the three models are adapted to greater or lesser degree.

Indeed, GRAIN is a category that is only represented in 5% of the data[3] while 35% of the documents are labeled EARN.

5. Conclusion

In this paper, we have presented an overview of several density estimation models for document representation. We have furthermore proposed yet another model in this family, namely the Theme Topic Mixture Model (TTMM), which lies in between LDA and PLSA, sharing some advantages of both of them. A theoretical comparison between the models was then presented, highlighting advantages and problems of each method. This was followed by an empirical analysis, which shows that no one model is always better than the others, and that the ultimate choice may depends on the actual data configuration. Interestingly, all of the proposed density estimation models fail with respect to the simple bag-of-words representation when the size of the dataset becomes sufficiently large. This probably means that the constraints that have been purposely integrated into all these models (the choice of words in a document is independent of the document itself given a hidden topic variable) may be useful when the data is scarce but too strong when it is abundant, in which case constraints should be relaxed somehow.

Acknowledgments

We would like to thank David Blei for giving us his experiment's data and Florent Monay for helping with the PLSA experiments. 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

[1] J. R. Bellegarda and D. Nahamoo. Tied mixture continuous parameter models for large vocabulary isolated speech recognition. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, pages 13–16, 1989.

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

[3] W. Buntine. Variational Extensions to EM and Multinomial PCA. In T. Elomaa, H. Mannila, and H. Toivonen, editors, *Machine Learning: ECML 2002: 13th European Conference on Machine Learning, Helsinki, Finland, August 19-23, 2002. Proceedings*, pages 23 – 34. Springer-Verlag Heidelberg, 2002.

[4] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

3. 5% of the document belongs to the category GRAIN means that the trivial rejector has already an accuracy of 95%...

[5] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc. B.*, 39:1–38, 1977.

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

[8] M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[9] M. Keller and S. Bengio. Theme Topic Mixture Model: A Graphical Model for Document Representation. IDIAP-RR 05, IDIAP, 2004.

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