Optimization for Machine Learning

Editors:

Irina Rish rish@us.ibm.com
IBM T.J. Watson Research Center
1101 Kitchawan Rd., Yorktown Heights, NY

Guillermo Cecchi gcecchi@us.ibm.com
IBM T.J. Watson Research Center
1101 Kitchawan Rd., Yorktown Heights, NY

Aurelie Lozano aclozano@us.ibm.com
IBM T.J. Watson Research Center
1101 Kitchawan Rd., Yorktown Heights, NY

This is a draft containing only `sra_chapter.tex` and an abbreviated front matter. Please check that the formatting and small changes have been performed correctly. Please verify the affiliation. Please use this version for sending us future modifications.

The MIT Press
Cambridge, Massachusetts
London, England

Contents

1 Sparsity in Topic Models

Jagannadan Varadarajan vjagann@idiap.ch
Idiap Research Institute,
École Polytechnique Fédérale de Lausanne,
Switzerland

Rémi Emonet remonet@idiap.ch
Idiap Research Institute
Martigny, Switzerland

Jean-Marc Odobez odobez@idiap.ch
Idiap Research Institute
École Polytechnique Fédéral de Lausanne,
Switzerland

Recently, topic models have become an effective tool in mining dominant patterns in the data in an unsupervised fashion. Eventually, these models have found relevance in numerous areas such as text analysis, recommendation systems and computer vision. Topic models use co-occurrence analysis to discover latent structures called *topics*, which are dominant co-occurring sets of words in the data. In practice, one often wants to impose a constraint on this learning, wherein each topic has only a subset of the vocabulary or each document is represented using only a few dominant topics. Such a sparsity constraint have shown to improve the learning performance even under adverse conditions such as noise. The objective of this article is to provide a brief overview of various methods employed within the framework of Topic models to achieve sparsity. After this review, a demonstration is provided by applying an information theoretic sparsity approach applied to Probabilistic Latent Sequential Motifs (PLSM), a topic model approach developed to discover temporal motifs from videos and time-series in general.

1.1 Introduction

There is an overwhelming amount of data being accumulated these days through various sources such as web pages, news articles, blogs, videos, and various other sensor logs. The sheer enormity of the data available makes it very difficult to find relevant information quickly. Therefore it has become important to develop efficient data mining and analysis tools that could help an end-user browse through this vast amount of data. Recently, topic models have emerged as a powerful data mining tool by allowing us to obtain a concise representation of the data set by capturing dominant patterns from simple un-ordered feature counts. While topic models were first proposed to solve text mining, document clustering and trend analysis, they have also been successfully employed in other domains like computer vision to address problems such as scene classification, object class recognition and activity analysis.

To quickly review the ideas of topic models, let us consider Probabilistic Latent Semantic Analysis (PLSA) by Hofmann (2001), which is one the earliest topic models proposed, and perhaps one of the simplest and easiest to understand and implement.

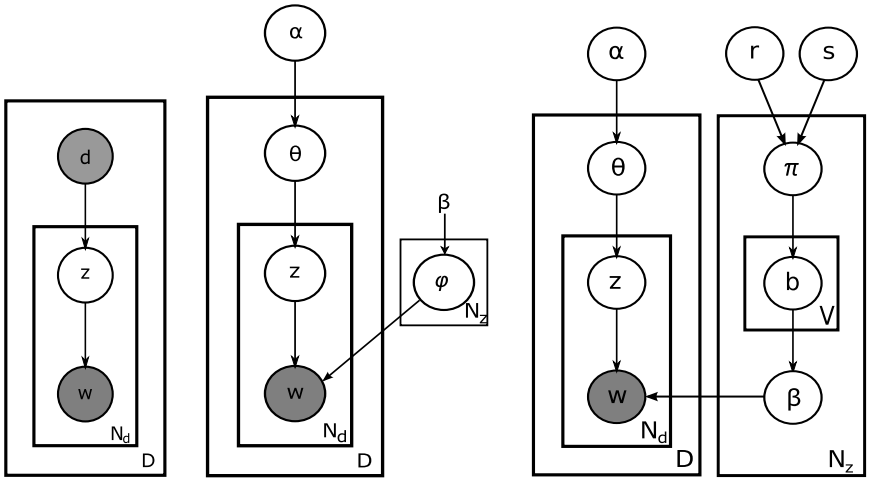
1.1.1 PLSA and Sparsity issue

PLSA and LDA are generative models, meaning, they are based on probabilistic sampling rules that describe how words in a document are generated. To get an intuition of the generative process of PLSA, let us consider that a columnist for *Wall Street Journal* decides to write an article on the "Global Economic Crisis". He would first plan his article based on some sub-topics that could possibly be *Economy*, *Stocks* and *Banking* for example. Then, he might decide the importance to be given for each of the sub-topics possibly reflected by the number of words or paragraphs dedicated to each subtopic. For instance, he might decide to write about each of the above topics in about $\{5, 5, 7\}$ paragraphs of the same size respectively. Then, for each topic, he would choose the most appropriate words to convey his ideas on the subject. Let us consider for a moment that a computer, ignorant of language grammar and word order is assigned a job to generate a number of such articles using an algorithm. Then if each word is indicated by the variable w, each topic by z and document by d, perhaps it might have the method given in algorithm (1.1) in its RAM for drawing a "bag of N_d words" for each document d,

Distributions: The importance given to each topic is given by a categorical

Algorithm 1.1 The PLSA generative model

for $d = 1$ to D; **do**
  for for $j = 1$ to N_d; **do**
    draw a topic $z \sim P(z|d)$
    draw a word $w \sim P(w|z)$
  end for
end for



(a) PLSA (b) LDA (c) Sparse LDA

Figure 1.1: Differences between the PLSA, LDA topic models, and the Sparse LDA model.

distribution $p(z|d)$. In the example taken, this would simply be the proportion of the three topics in the article given by $\{5/17, 5/17, 7/17\}$. Similarly, the number of times each word occurring in a topic gives the categorical distribution $p(w|z)$. This would mean that words like *fiscal, deficit, banks* and *GDP* will have high probability under the topic "Economy" and, words like *profit, booking, NASDAQ, LSE* and *banks*[1] may occur more frequently under the "Stocks" topic.

Graphical Model: The procedure described in algorithm (1.1) is called a generative process and its pictorial version in Figure 1.1(a) is called the PLSA graphical model. In this notation, the nodes represent the random variables in the circles. Shaded circles indicate observed variables and transparent circles represent latent variables.

1. Note that the term *banks* can occur in more than one topic. For instance, banks can also occur in documents that talk about rivers and water bodies, which is an example if polysemy.

In the case of Figure 1.1(a), w and d are observed and z is called a latent variable which needs to be estimated. The directed edges indicate conditional dependencies. Here, we have w depending on z and the presence of z introduces a conditional independence: a word w and document d are conditionally independent given the topic z, indicated as $w \perp\!\!\!\perp d|z$. Intuitively, this means that words depend only on the topic and not on the document for which it is generated. The plates indicate repetition of the sampling process, where the variable in the bottom right of the plate indicates the number of samples. In Figure 1.1(a), the plate surrounding w and z indicates that z is sampled N_d times, each time followed by a w sample. In other words, for each document d, there are N_d (z, w) pairs.

Our objective in creating a graphical model as in Figure (1.1) is to simplify the joint distribution into simpler factors as in equation (1.1) and infer them. More specifically, we would like to learn the topics: $P(w|z)$, and their weights in a document: $P(z|d)$, from a corpus of documents and observations represented as a *word-count* matrix $P(w, d)$. The conditional independence assumption in the model is used to split the joint distribution of the model into smaller factors, i.e., the joint distribution of all the variable triplets (w, z, d) can be written as

$$P(w, z, d) = P(d)P(w|z)P(z|d) \tag{1.1}$$

Furthermore, the probability of an observation pair (w, d) can be obtained by marginalizing out the topic variable in the joint distribution:

$$P(w, d) = \sum_{z=1}^{N_z} P(w, z, d) = P(d) \sum_{z=1}^{N_z} P(z|d)P(w|z). \tag{1.2}$$

A closer look at equation(1.2) reveals that the model decomposes the conditional probabilities of words in a document $p(w|d)$ as a convex combination of the topic specific word distributions $p(w|z)$, where the weights are given by the topic distribution $p(z|d)$ in a document.

Sparsity issue – While the distributions learned from PLSA give us a concise representation of the corpus, they are often loosely constrained, resulting in non-sparse process representations which are often not desirable in practice. For instance, in PLSA, one would like each document d to be represented by only a small number of topics z with high weights $p(z|d)$, or each topic z to be represented by a small number of words with high $p(w|z)$ weights. This would provide a more compact representation of the data and in many cases improve efficiency in storage and computation. But nothing in the modeling encourages such a learning mechanism. Recently, there have been some attempts in including such an objective in learning

the distributions that have also shown improvement in performance. In the following sections we will review some of these proposed methods.

1.1.2 Matrix factorization methods

Historically matrix factorization methods like Singular Vector Decomposition (SVD) were used to identify concepts hidden in the data. Given a document-term matrix D, SVD factorizes D as $D = U\Sigma V'$, where U and V are matrices with orthonormal columns and Σ contains the singular values. By taking the top K singular values and setting the rest to zero ($\tilde{\Sigma}$) we get our concept space from the rows of $U\tilde{\Sigma}$. Though SVD is one of the simplest matrix factorization methods, it suffers from several problems. For instance there is no clear interprestation of the magnitude of the vectors that define the concept space in SVD. Furthermore, there is a possibility of obtaining negative values while reconstructing D with the top K singular values. This has motivated several other alternatives such as the Non-negative Matrix Factorization (NMF) and Probabilistic Topic Models (PTM).

Non-negative matrix factorization (NMF) by Lee and Seung (1999) is an improvement over SVD, where the matrix D is decomposed into non-negative factors W and H, i.e., $D = WH$. This is preferred over SVD because we often want to decompose the count matrix into additive components of non-negative factors. This was successfully used in many applications including text mining and face-recognition by Lee and Seung (1999). In NMF the matrix W represents the set of basis vectors and H represents the coefficients of linear decomposition. But many a times, depending upon the application domain a sparse set of coefficients or the basis vectors is desired. Hoyer (2005) proposed a method wherein sparse basis vectors W and the coefficients H can be obtained for a desired degree of sparsity. To this end, a measure to describe the degree of sparsity of any vector as

$$\text{sparseness}(X) = \frac{\sqrt{n} - |X|_1/|X|_2}{\sqrt{n} - 1} \tag{1.3}$$

was proposed, where n is the dimension of the vector X. The measure takes values in the interval $[0, 1]$, where a sparseness value of 0 indicates that all the coefficients have equal non zero value, and a sparseness of 1 indicates a single non-zero component. From the above equation we see that one can obtain the desired degree of sparsity of a vector by manipulating its L_1 and L_2 norms. To achieve this at each iteration of the estimation, first W and H are estimated by proceeding along the negative gradient that minimizes the error $||D - WH||^2$, then, based on whether the constraints apply to W or H or both, each column or row respectively of the matrices are projected

to have unchanged L_2 norm and desired L_1 norm.

The Sparse-NMF method was applied to face image datasets and natural image datasets. It was shown that by imposing sparsity constraint on the basis vectors obtained more local features which otherwise was not possible in situations where faces are not well aligned. Similarly, by seeking sparse coefficients H on a natural image dataset, Sparse-NMF learned oriented features resembling edges and lines. The method thus enables us to control sparsity explicitly with a parameter that can be easily interpreted.

At this point, it is relevant to mention that it has also been shown by Gaussier and Goutte (2005) that PLSA is equivalent to NMF with the Kullback-Leibler (KL) divergence. However, the probabilistic framework in which PLSA works gives us several advantages. It gives a clear interpretation of the matrix decomposition in terms of conditional distributions. Additionally, the graphical model framework enables us to create principled hierarchical extensions, which can be solved by well established inference tools like Expectation-Maximization, Mean-field approximation and Gibbs sampling.

1.1.3 Sparsity in LDA and HDP

PLSA is not a fully generative model. While the method gives the topic weights for all the training documents indexed by d, it does not explain how topic weights $p(z|d)$ can be drawn for an unseen document. Also, a Bayesian treatment to this requires that all parameters of the model are drawn from a prior distribution. In Latent Dirichlet Allocation Figure 1.1(b), this is solved by having the topic weights θ_d as a random variable drawn from a Dirichlet distribution $\mathrm{Dir}(\alpha)$ with hyper-parameter α, and the topic parameters φ_z drawn from a Dirichlet distribution $\mathrm{Dir}(\beta)$ with hyper-parameter β. But in practice, due to lack of any prior knowledge on the topic presence in documents or word participation in topics, a symmetric Dirichlet prior which has the same scalar value for all the components of the vector is used for α and β. Let us consider the case of the β prior first, such a non-informative prior has two main consequences: a) large values of the scalars of β, provides more smoothing over the terms of the vocabulary, and b) when β value goes to zero, the role of the smoothing prior reduces resulting in empirical estimates of φ_z) (topics that place their weights only on few terms or less smooth distribution over words). In order to circumvent this effect of priors on smoothing and sparsity, Wang and Blei (2009) proposed a model that decouples the request for sparsity and the smoothing effect of Dirichlet prior. Although the model was presented as a sparse version of Hierarchical

Dirichlet Process (HDP)[2], it can be simplified and understood even in the context of LDA.

Sparse LDA – The graphical model of Sparse LDA is provided in Figure 1.1(c) and the generative process is as follows: For each topic $z = 1, 2, \ldots, N_z$, a term selection proportion π_z is first drawn from Beta(r, s). Then for each term $v, 1 \leq v \leq N_w$ (where N_w is the number of words in the vocabulary), a selector $\{0, 1\}$ is drawn from a Bernoulli(π_z). Furthermore, drawing the topic proportions is akin to the LDA model as discussed above, i.e., the topic weights θ_d for each document is drawn from Dir(α). For each term w_{di}, the topic assignment z_{di} is drawn from a categorical distribution Categ(θ_d) and each word w_{di} is drawn from another categorical distribution Categ$(\beta_{z_{di}})$.

We can observe that by using the selector variables for each term in the topic, the topics are defined only over a sub-simplex and the smoothing prior is applied to only the selected terms. From a sparsity perspective what the model achieves by having explicit selector variables is the effect of introducing the L_0 norm on the vocabulary for each topic. We see that an elegant generative process is used to solve an otherwise very complex problem in the combinatorial sense.[3] While this method allows us to tune the expected level of sparsity from each topic by adjusting the Beta parameters, this does not improve the sparsity of the topic decomposition if desired. This was exactly addressed using a different generative process by Williamson et al. (2009).

Focused topic model – In this mode proposed by Williamson et al. (2009), the goal is to explain each document using only a small set of topics. Overlooking the nitty-gritties of HDP in focused topic model, we see that it relies on the Indian Buffet Process (IBP) by Griffiths and Ghahramani (2005) to generate a sparse binary matrix which serves as a prior (switching variable) to indicate if a topic is present in a document or not. Thus a sparse prior results in using only a few topics to explain the document hence the name "Focused". The two main steps of this generative model that differentiates this from the HDP model concerns the generation of topic specific weights for each document i.e.,: first, a binary matrix $B \sim \text{IBP}(\alpha)$ is created. The entries of the binary matrix are given by b_{mk} taking values 0 or 1. Second, for each topic k, a global topic proportion is sampled according to,

2. HDP uses non-parametric methods like Dirichlet process to obtain topics. Since the number of topics is unlimited it is often called infinite LDA.

3. In a naive method, a desired L_0 norm sparsity can be achieved by generating $\binom{V}{N}$ subsets of words and checking all the combinations for each topic.

$\phi_k \sim \text{Gamma}(\gamma, 1)$. Finally, $n_k^{(m)}$, the number of words for k^{th} topic in the m^{th} document is drawn according to, $n_k^{(m)} \sim Poisson - Gamma(b_{mk}\phi_k, 1)$. The entries of the matrix serve as switching/decision variables for the topic selection, thus a sparse binary matrix would result in a sparse topic decomposition of the document. The performance of the model is evaluated using two measures called topic presence frequency (fraction of documents in the corpus with an incidence of the topic) and topic proportion (fraction of words in the corpus assigned to the topic). A sparse decomposition should have the least correlation between these two measures.

We can conclude from studying the above models that the Sparse LDA and Focused topic model, are different generative models for achieving sparsity on two different distributions, they eventually rely on priors to generate binary variables to decide whether to select a word for a topic in the former case or a topic for a document in the latter case.

1.1.4 Information theoretic sparsity methods

A different view of the sparsity problem in the context of probabilistic topic modeling is to seek more peaky distributions. In such cases, as we are searching the space of distributions, a natural choice would be to guide the learning process towards attaining more peaky distributions characterized by a smaller entropy instead of a norm based regularization constraint, although not in a probabilistic context. Traditionally, information theoretic measures like entropy and Kullback-Leibler (KL) divergence have been used as a regularization constraint in several inverse or under-constrained problems (Besnerais et al. (1999)). Recently, KL divergence has also been used successfully as a means to achieve sparsity. In Bradley and Bagnell (2009), a sparse coefficient vector with respect to a fixed bases are learned by optimizing the generalized KL divergence (Bregman divergence) with uniform distribution. They show that this achieves a higher degree of sparsity in a classification task when compared to the well known L_1 or L_2 optimization.

In another application of topic model for video scene analysis by Varadarajan et al. (2010a), the goal is to learn distributions over time that indicate the start of a certain activity in the scene. Using the regular EM optimization procedure is loosely constrained, and therefore we obtain a sub-optimal solution that gives a smooth distribution for the activity start times. To solve this, a regularization constraint in the EM optimization procedure is added to select a peaky distribution by maximizing the KL divergence between the uniform distribution and the learned distribution. This results in a simple procedure that can be applied to any distribution for which such a sparsity constraint is desirable.

In the following section we describe how the KL divergence based sparsity constraint was applied to Probabilistic Latent Sequential Motifs model, provide its motivations, modeling details and further go on to explain how sparsity constraint is imposed in this model.

1.2 Probabilistic Latent Sequential Motif Model

In this section, we introduce the PLSM model, its motivation, the generative model along with details of the learning procedure. Then the inference procedure and how it is improved by using a KL divergence based sparsity constraint is presented. The model and its properties are then validated on synthetic experiments and illustrated on real surveillance videos.

1.2.1 Motivation

Let us consider for example a temporally ordered set of observations from which one would like to extract sequential patterns called motifs (e.g. a text document or a speech signal). Our observation here at any point in time would be a single word in the case of text, or a single phoneme in the case of speech. But if we consider a video signal, it would contain multiple observations at any point in time. These observations could be due to multiple local activities occurring simultaneously. For example, consider a video signal obtained by recording a busy traffic scene. In such scenes many activities occur simultaneously due to more than one object present in the scene. These activities occur without any particular synchrony or order resulting in the superposition of multiple overlapping observations, making any analysis a complex problem. From these observations, we are interested in identifying the dominant activity patterns in the scene and their time of occurrence. This is similar to the case of topic models applied to text, where topics that model dominant co-occurrences are obtained. But the added difficulty here is due to observations caused by multiple activities simultaneously and the lack of a-priori knowledge of how many activities occur in the scene.

In Varadarajan et al. (2010b), we introduced the Probabilistic Latent Sequential Motif (PLSM) topic model to discover dominant sequential activity patterns from sensor data logs represented by word$\times$time count documents. Its main features are: i) the estimated patterns are not merely defined as static word distributions but also incorporate the temporal order in which words occur; ii) automatic estimation of activity starting times, and iii) the ability to deal with multiple temporally overlapping activities in the scene.

This model is detailed in the following sections.

1.2.2 Model overview and generative process

Figure 1.2(a) illustrates how the documents are generated. Let D be the number of documents in the corpus, indexed by d, each spanning T_d discrete time steps. Let $V = \{w_i\}_{i=1}^{N_w}$ be the vocabulary of words that can occur at any given instant $t_a \in \{1, \cdots, T_d\}$. A document is then described by its count matrix $\mathrm{n}(w, t_a, d)$ indicating the number of times a word w occurs at the absolute time t_a. These documents are generated from a set of N_z temporal patterns or motifs $\{z_i\}_{i=1}^{N_z}$ represented by the distributions $P(w, t_r|z)$. The motifs have a maximal duration of T_z time steps, where t_r denotes the relative time at which a word occurs within a motif. Each motif can start at any time instant t_s, $t_s \in \{1, \cdots, T_{ds}\}$ within the document. Qualitatively, documents triplets (w, t_a, d) are generated by sampling words from the motifs and placing them in the document relative to a sampled starting time according to (cf Figure 1.2(a)). The PLSM graphical model is given in Figure 1.2(b) and the procedure to generate the triplets (w, t_a, d) is as follows:

Algorithm 1.2 The PLSM generative model

draw a document $d \sim P(d)$
for each word w in the document d **do**
  draw a latent motif $z \sim P(z|d)$
  draw the starting time $t_s \sim P(t_s|z, d)$ *% where $P(t_s|z, d)$ denotes the probability that the motif z starts at time t_s within the document d.*
  draw the relative time $t_r \sim P(t_r|z)$ *% where $P(t_r|z)$ denotes the probability of observing any word w at time t_r.*
  draw a word $w \sim P(w|t_r, z)$ *% where $P(w|t_r, z)$ denotes the probability that the word w within the motif z occurs at time t_r.*
  set $t_a = t_s + t_r$ *% this assumes that $P(t_a|t_s, t_r) = \delta(t_a - (t_s + t_r))$, that is, the probability density function $P(t_a|t_s, t_r)$ is a Dirac function.*
end for

The main assumption with the above model is that the occurrence of a word only depends on the motif, not on the time instant when a motif occurs. Given the deterministic relation between the three time variables $(t_a = t_s + t_r)$, the joint distribution of all variables can be written as:

$$P(w, t_a, d, z, t_s) = P(d)P(z|d)P(t_s|z, d)P(w|z)P(t_a - t_s|w, z) \tag{1.4}$$

1.2.3 Model inference with sparsity

Our goal is to discover the motifs and their starting times given the data $\mathcal{D}$ defined by the count matrices $\mathrm{n}(w, t_a, d)$. The model parameters

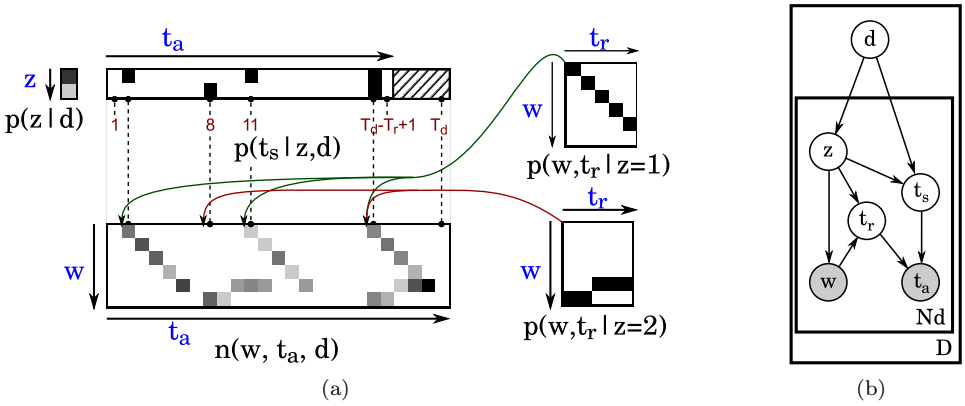


Figure 1.2: a) document $\mathrm{n}(w, t_a, d)$ generation. Words $(w, t_a = t_s + t_r)$ are obtained by first sampling the motifs and their starting times from the $P(z|d)$ and $P(t_s|z,d)$ distributions, and then sampling the word and its temporal occurrence within the motif from $P(w, t_r|z)$. b) graphical model.

$\Theta = \{P(z|d), P(t_s|z,d), P(t_r|z), P(w|t_r,z)\}$ can be estimated by maximizing the log-likelihood of the observed data $\mathcal{D}$, which is obtained through marginalization over the hidden variables $Y = \{t_s, z\}$:

$$\mathcal{L}(\mathcal{D}|\Theta) = \sum_{d=1}^{D} \sum_{w=1}^{N_w} \sum_{t_a=1}^{T_d} n(w, t_a, d) \log \sum_{z=1}^{N_z} \sum_{t_s=1}^{T_{ds}} P(w, t_a, d, z, t_s) \tag{1.5}$$

Such an optimization can be performed using an Expectation-Maximization (EM) approach, maximizing the expectation of the complete log-likelihood. However, as motivated in the introduction, the estimated distributions may exhibit a non-sparse structure that is not desirable in practice. In our model this is the case of $P(t_s|z,d)$: one would expect this distribution to be peaky, exhibiting high values for only a limited number of time instants t_s. To encourage this, we propose to guide the learning process towards sparser distributions characterized by smaller entropy, and achieve this indirectly by adding to the data likelihood a regularization constraint to maximize the KL divergence $D_{KL}(U||P(t_s|z,d))$ between the uniform distribution U (maximum entropy) and the distribution of interest. This gives a constrained log-likelihood function given by:

$$\mathcal{L}_c(\mathcal{D}|\Theta) = \mathcal{L}(\mathcal{D}|\Theta) + \sum_{t_s,z,d} \lambda_{z,d} \cdot \frac{1}{T_{ds}} \cdot \log\left(\frac{1/T_{ds}}{P(t_s|z,d)}\right) \tag{1.6}$$

After development and removing the constant term, our constrained objective function is now given by:

$$\mathcal{L}_c(\mathcal{D}|\Theta) = \mathcal{L}(\mathcal{D}|\Theta) - \sum_{t_s,z,d} \frac{\lambda_{z,d}}{T_{ds}} \cdot \log(P(t_s|z,d)) \tag{1.7}$$

The EM algorithm can be easily applied to the modified objective function. In the E-step, the posterior distribution of hidden variables is calculated as (the joint probability is given by equation(1.4)):

$$P(z,t_s|w,t_a,d) = \frac{P(w,t_a d,z,t_s)}{P(w,t_a,d)} \text{ with } P(w,t_a,d) = \sum_{z=1}^{N_z}\sum_{t_s=1}^{T_{ds}} P(w,t_a,d,z,t_s) \tag{1.8}$$

In the M-step, the model parameters (the probability tables) are updated according to:

$$P(z|d) \propto \sum_{t_s=1}^{T_{ds}}\sum_{t_r=0}^{T_z-1}\sum_{w=1}^{N_w} \mathrm{n}(w,t_s+t_r,d)P(z,t_s|w,t_s+t_r,d) \tag{1.9}$$

$$P(t_s|z,d) \propto \max\left(\varepsilon, \left(\sum_{w=1}^{N_w}\sum_{t_r=0}^{T_z-1} \mathrm{n}(w,t_s+t_r,d)P(z,t_s|w,t_s+t_r,d)\right) - \frac{\lambda_{z,d}}{T_{ds}}\right) \tag{1.10}$$

$$p_w(w|z) \propto \sum_{d=1}^{D}\sum_{t_s=1}^{T_{ds}}\sum_{t_r=0}^{T_z-1} \mathrm{n}(w,t_s+t_r,d)P(z,t_s|w,t_s+t_r,d) \tag{1.11}$$

$$p_{t_r}(t_r|w,z) \propto \sum_{d=1}^{D}\sum_{t_s=1}^{T_{ds}} \mathrm{n}(w,t_s+t_r,d)P(z,t_s|w,t_s+t_r,d) \tag{1.12}$$

Qualitatively, in the E-step, the responsibilities of the motif occurrences in explaining the word pairs (w,t_a) are computed (high responsibilities are obtained for informative words, i.e. words appearing in only one motif and at a specific time), whereas the M-steps aggregates these responsibilities to infer the motif patterns and occurrences. Importantly, thanks to the E-steps, the multiple occurrences of an activity in documents are implicitly aligned in order to learn its pattern.

Sparsity analysis – A closer look at equation(1.7), reveals that while maximizing the KL divergence between the uniform distribution and $P(t_s|z,d)$ amounts to maximizing the factor $H = -\sum(1/T_{ds})log(P(t_s|z,d))$ which is nothing but the cross entropy between uniform distribution and $P(t_s|z,d)$.

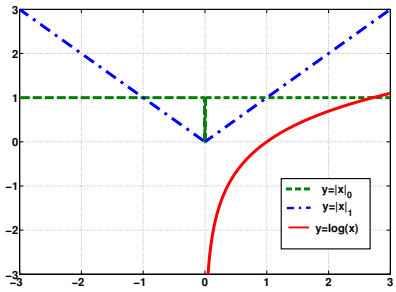
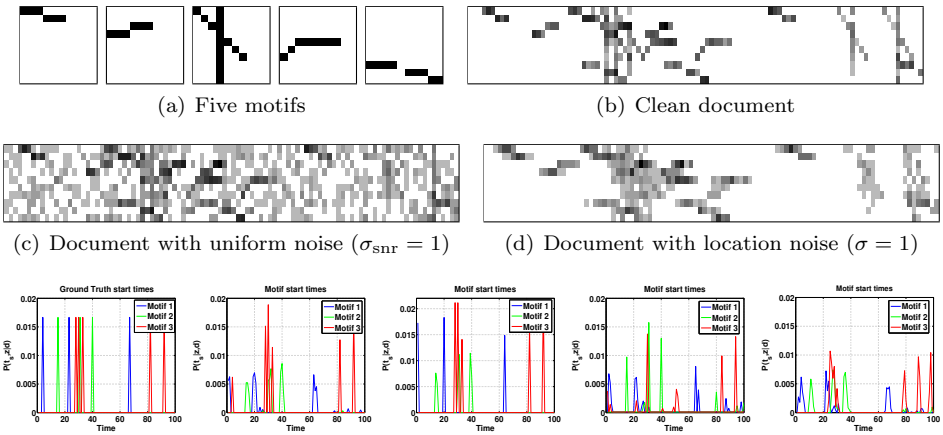


Figure 1.3: Comparison of L_1 norm sparsity and KL divergence based sparsity: L_1 norm curve has a slower rate of decay than $\log(x)$ in the range $[0, 1]$

Ideally, this factor reaches its maximum when $P(t_s|z, d)$ takes value 0 for all t_s. But due to the constraint that sum of probability values over t_s should sum to one, we obtain a sparse vector with only few non-zero values. This is again revealed in the equation(1.10), where we see that the effect of the introduced constraint is to the probability of terms to 0 which are lower than $\frac{\lambda_{z,d}}{T_{ds}}$, thus increasing the sparsity as desired.[4]

It might be worth to do a comparison of the usual L_1 norm based penalty which is widely used in the sparsity community with the KL divergence based penalty for achieving sparsity, specifically when the vector values lie in $[0, 1]$. Figure 1.3 gives the plot of three functions: i) $y = |x|_0$ that is used in L_0 norm based sparsity optimization, ii) $y = |x|_1$ that is used in L_1 norm based sparsity optimization, and iii) $y = \log(x)$ that is used in the KL divergence based sparsity constraint. In the L_1 norm optimization, where at each step the L_1 of the vector is minimized, the vector takes steps along the gradient of the L_1 curve which is constant throughout its range. While using KL divergence of the vector with the uniform distribution, the minimization proceeds along the gradient of the log function. The log function has a gradient similar to the L_1 norm at values near 1, but it becomes much higher at values near 0. This phenomenon of the log function ensures a faster rate of decay for small values of the vector and hence results in a sparse solution much faster than the L_1 norm.

4. In practice, during optimization one needs to set to a small value ε instead of 0 so that the constraint remains defined.



(a) Five motifs (b) Clean document

(c) Document with uniform noise ($\sigma_{\text{snr}} = 1$) (d) Document with location noise ($\sigma = 1$)

(e) Starting times (ground truth) (f) For (b) (g) For (b), with sparsity (h) For (c), with sparsity (i) For (d), with sparsity

Figure 1.4: Synthetic experiments. (a) the five motifs, (b) a segment of a generated document, (c,d) the same segment perturbed with (c) uniform noise added by sampling (w, t_a) uniformly ($\sigma_{\text{snr}} = 1$) and (d) Location noise added to each word time occurrence ($\sigma = 1$). (e)–(i) the true motif occurrences $P(t_s|z, d)$ (only 3 of them are shown for clarity). (e) ground truth of document segment shown in (b). (f-i) the recovered motif occurrences $P(t_s|z, d)$; (f) the clean document (cf b) with no sparsity $\lambda = 0$ (g) the clean document with sparsity $\lambda = 0.5$; (h) the noisy document (c) with sparsity $\lambda = 0.5$ (i) the noisy document (d) with sparsity $\lambda = 0.5$.

1.3 Experiments on synthetic data

We first demonstrate the PLSM model's performance and the effect of sparsity constraint using synthetic data. Using a vocabulary of 10 words, we created five motifs with duration ranging between 6 and 10 time steps (see Figure 1.4(a)). Then, we created 10 documents of 2000 time steps assuming equi-probable motifs and 60 random occurrences per motif. In the rest of the article, average results from the 10 documents and corresponding error-bars are reported. One hundred time steps of one document are shown in Figure 1.4(b), where the intensities represents the word count (larger counts are darker), and Figure 1.4(e) shows the corresponding starting times of three out of the five motifs. We can observe that there is a large amount of overlap between the motif occurrences. Finally, in equation(1.10) we defined $\lambda_{z,d} = \lambda \frac{\mathrm{n}_d}{N_z}$, where n_d denotes the total number of words in the document, and use λ to denote the sparsity level. As a result, note that when $\lambda = 1$, the correction term $\frac{\lambda_{z,d}}{T_{ds}}$ is, on average, of the same order of magnitude than the first part of the right hand side in equation(1.10).

Results on clean data – Figure 1.5(b) and Figure 1.5(a) illustrate the recovered topics with and without the sparsity constraint respectively. We

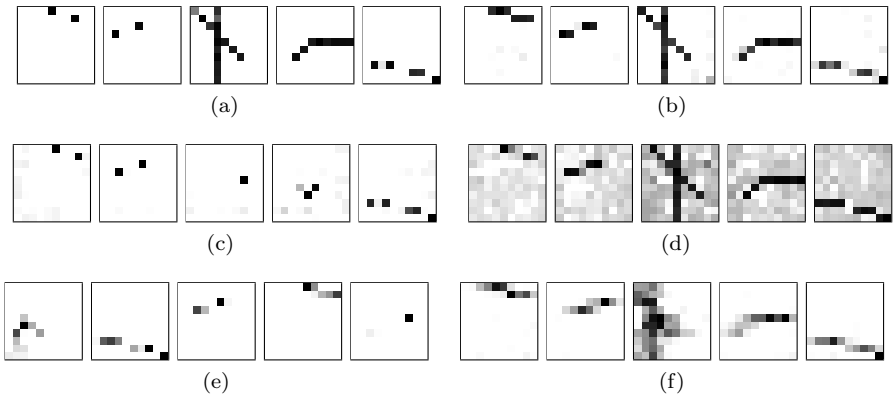


Figure 1.5: Recovered motifs without (a,c,e) and with (b,d,f) sparsity constraints $\lambda = 0.5$ (a,b) from clean data; (c,d) from documents perturbed with random noise words, $\sigma_{snr} = 1$, cf Figure 1.4(c); (e,f) from documents perturbed with Gaussian noise on location $\sigma = 1$, cf Figure 1.4(d).

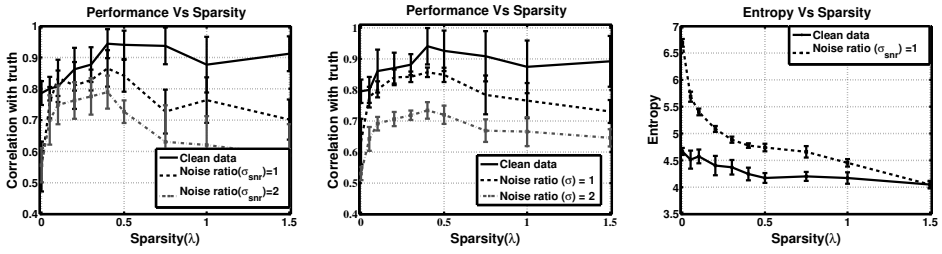


Figure 1.6: Average motif correlation between the estimated and the ground truth motifs for different sparsity weight λ and for different levels of (a) the uniform noise, (b) the Gaussian noise on a word time occurrence t_a. (c) Average entropy of $P(ts|z,d)$ in function of the sparsity λ.

can observe that two of the obtained motifs are not well recovered without the sparsity constraint. This can be explained as follows. Consider the first of the five motifs. Samples of this motif motif starting at a given instant t_s in the document can be equivalently obtained by sampling words from the learned motif Figure 1.5(a) and sampling the starting time from three consecutive t_s values with lower probabilities instead of one t_s value. This can be visualized in Figure 1.4(f), where the peaks in the blue curve $P(t_s|z = 1, d)$ are three times wider and lower than in the ground truth. When using the sparsity constraint, the motifs are well recovered, and the starting time occurrences better estimated (see Figure 1.5(b) and Figure 1.4(g)).

Robustness to Noise and sparsity effect – Two types of noise were used to test the method's robustness. In the first case, words were added to the clean documents by randomly sampling the time instant t_a and the word w

from a uniform distribution, as illustrated in Figure 1.4(c). The amount of noise is quantified by the ratio $\sigma_{snr} = N_w^{noise}/N_w^{true}$ where, N_w^{noise} denotes the number of noise words added and N_w^{true} the number of words in the clean document. The learning performance is evaluated by measuring, the average normalized cross correlation between the learned motifs $\hat{P}(t_r, w|z)$ and the true motifs $P(t_r, w|z)$ (see Figure 1.6).

Noise can also be due to variability in the temporal execution of the activity. This "location noise" was simulated by adding random shifts (sampled from Gaussian noise with $\sigma \in [0, 2]$) to the time occurrence t_a of each word, resulting in blurry documents (see Figure 1.4(d)). Figure 1.5(c-f) illustrates the recovered motifs. Without sparsity constraint, the motif patterns are not well recovered (even the vertical motif). With the sparsity constraint, motifs are well recovered, but reflect the effects of the generated noise, i.e. uniform noise in the first case, temporal blurring in the second case. Figure 1.6 shows that the model is able to handle quite a large amount of noise in both cases, and that the sparsity approach provide significantly better results. Finally, we validate that, as desired, there is an inverse relation between the sparsity constraint and the entropy of $P(t_s|z, d)$ which is clearly seen in Figure 1.6(c).

1.4 Scene activity patterns

1.4.1 Activity words

We also applied our PLSM model to discover temporal activity patterns from real life scenes. This work flow is summarized in Figure 1.7. To apply the PLSM model on videos, we need to define the words w forming its vocabulary. Instead of using low-level visual features directly, we perform a dimensionality reduction step on the low level features as done in Varadarajan et al. (2010b) by applying PLSA on low level features $w^{ll} = (p, v)$, where p is a quantized image location (obtained by dividing the image into 10×10 grids) and v is a quantized direction of the optical flow feature (we used the 4 cardinal directions as our bins). The low-level documents for applying PLSA are created from these feature counts accumulated over overlapping clips of 1 second duration. As a result, we obtain temporally and spatially localized activity (TSLA) patterns z^{ll} from the low-level features and use the occurrences of these as our words to discover sequential activity motifs in PLSM model. Thus, N_A dominant TSLA patterns obtained from PLSA define our words for PLSM i.e., $N_w = N_A$, and the word count for each time instant d_{t_a} is given by $n(w, d_{t_a}) \propto P(z^{ll}|d_{t_a})$. The word counts defining the PLSM documents d are then built from the amount of presence of these TSLA patterns.

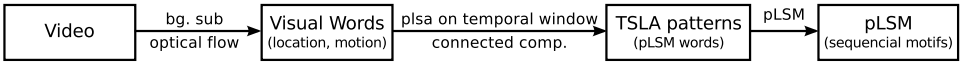


Figure 1.7: Flowchart for discovering sequential activity motifs in videos.

1.4.2 Data

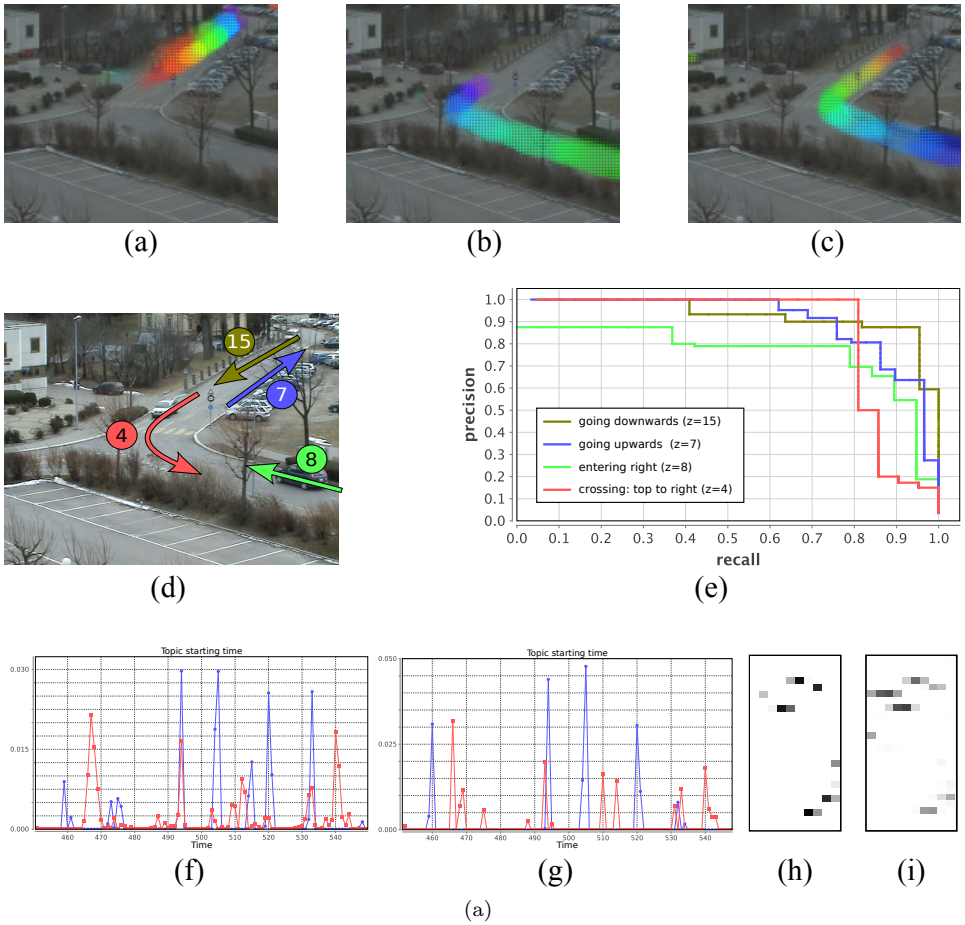
Experiments were carried out on two complex scenes. The Far-field video (Varadarajan et al. (2010b)) contains 108 minutes of a three-road junction captured from a distance, where typical activities are moving vehicles. As the scene is not controlled by a traffic signal, activities have large temporal variations. The Traffic Junction video is 45 minutes long and captures a portion of a busy traffic-light-controlled road junction. Activities include people walking on the pavement or waiting before crossing the road, and vehicles moving in and out of the scene.

Given the scene complexity and the expected number of typical activities, we arbitrarily set the number N_z of sequential motifs to 15 and the motif duration T_z to 10 time steps (10 seconds). Some top ranking sequential motifs from the Far-field dataset are shown in Figure 1.8(a,b,c). They exactly correspond to the dominant patterns in the scene namely, vehicle moving along the main road in both directions in the Far-field data. In the interest of space and better illustration we have provided sample clips and comprehensive results at `http://www.idiap.ch/paper/1930/sup.html`. In the Traffic Junction scene, despite the low amount of data, we could recover motifs that correspond to vehicular movements, pedestrian activities, and complex interactions between vehicles and pedestrians.

1.4.3 Event detection and Sparsity effect

We also did a quantitative evaluation of how well PLSM can be used to detect particular events. We created an event detector by considering the most probable occurrences $P(t_s, z|d)$ of a topic z in a test document d. By setting and varying a threshold on $P(t_s, z|d)$, we can control the trade-off between precision and completeness. For this event detection task, we labelled a 10 minute video clip from the Far-field scene, distinct from the training set, and considered 4 events depicted in Figure 1.8(d). To each event type, we manually associated a motif, built an event detector and varied the decision threshold to obtain precision/recall curves. Figure 1.8(e) shows the obtained results.

The sparsity constraint employed on $P(t_s, z|d)$ distribution resulted in clear peaks for the motif start times (see Figure 1.8(g)) as opposed to smoother distributions obtained without the sparsity constraint in Figure 1.8(f). This was useful in removing some of the false alarms and im-



(a)

Figure 1.8: (a–c) Sample motifs from Far-field data representing dominant vehicular activities in the scene. (d–e) Event detection experiments. d) Four motifs from PLSM representing four events in the scene. e) Interpolated Precision/Recall curves for the detection of the four types of events evaluated on a 10 minute test video. (f–i) Effect of sparsity constraint on $P(t_s|z,d)$. (f,h) without sparsity, (g,i) with sparsity constraint.

proving the quantitative results in the event detection task. However, looking at the motifs qualitatively revealed that a sparse $p(t_s, z|d)$ (and hence more peaky) distribution results in smoother motifs: the uncertainty in start times is transfered to the time axis of the motifs as could be already seen on synthetic data (cf. Figure 1.5(f) and Figure 1.5(b)) or in the real case (see Figure 1.8(h) vs Figure 1.8(i))

1.5 Conclusion

In this article we reviewed some of the methods used to impose sparsity constraint within the framework of topic models. We provided a more detailed look at PLSM, a topic-based method for temporal activity mining that extracts temporal patterns from documents where multiple activities occur simultaneously. We provided a simple yet effective approach to encourage sparsity in the model, and more specifically on the motif start time distributions of the PLSM model. Experiments carried out both on synthetic data under variety of noise and real life data have shown that the sparsity constraint improves the quality of recovered activity patterns and increases the model's robustness to noise. The formulation of the sparsity regularization constraint as an entropy minimization makes it straightforward to introduce in the EM optimization. This can be similarly introduced in most topic models like PLSA and LDA.

1.6 Acknowledgements

This work was supported by the European Union under the integrated Project VANAHEIM (Video/Audio Networked surveillance system enhAncement through Human-cEntered adaptIve Monitoring), 248907, as well as the Swiss National Science Foundation under the project HAI (Human Activity Interactivity), FNS-198. The authors gratefully thank the EU and Swiss NSF for their financial support, and all project partners for a fruitful collaboration. More information about EU-VANAHEIM and SNSF-HAI is available from the project web sites www.vanaheim-project.eu and www.snf.ch.

References

G. Besnerais, J. Bercher, and G. Demoment. A new look at entropy for solving linear inverse problems. *IEEE Transactions on Information Theory*, 45(5):1565–1578, 1999.

D.M. Bradley and J. A. Bagnell. Differentiable sparse coding. In *Advances in Neural Information Processing Systems*, volume 21, page 113120, 2009.

E. Gaussier and C. Goutte. Relation between plsa and nmf and implications. In *28th International ACM SIGIR conference on Research and development in information retrieval*, pages 601–602, 2005.

Thomas L. Griffiths and Zoubin Ghahramani. Infinite latent feature models and the indian buffet. In *Advances in Neural Information Processing Systems*, 2005.

T. Hofmann. Unsupervised learning by probability latent semantic analysis. *Machine Learning*, 42:177–196, 2001.

P. O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5(2):1457–1470, 2005.

D. Lee and S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, (401):788–791, 1999.

J. Varadarajan, R. Emonet, and J.-M. Odobez. A sparsity constraint for topic models - application to temporal activity mining. In *NIPS Workshop on Practical Applications of Sparse Modeling: Open Issues and New Directions*, 2010a.

J. Varadarajan, Remi Emonet, and Jean-Marc Odobez. Probabilistic latent sequential motifs: Discovering temporal activity patterns in video scenes. In *British Machine Vision Conference*, pages 117.1–117.11, Aberystwyth, 2010b.

C. Wang and D.M. Blei. Decoupling sparsity and smoothness in the discrete hierarchical Dirichlet process. In *Neural Information Processing Systems*, pages 1982–1989, 2009.

Sinead Williamson, Chong Wang, Katherine Heller, and David Blei. Focused topic models. In *NIPS workshop on Applications for Topic Models: Text and Beyond*, Whistler, Canada., 2009.