TEMPORAL MOTIF MINING WITH SEQUENTIAL MINING TECHNIQUES

Wang Yuxin
Laboratoire Hubert Curien
University Jean Monnet, France
yuxin.wang@etu.univ-st-etienne.fr

Baptiste Jeudy
Laboratoire Hubert Curien
University Jean Monnet, France
baptiste.jeudy@univ-st-etienne.fr

Rémi Emonet
Laboratoire Hubert Curien
University Jean Monnet, France
remi.emonet@univ-st-etienne.fr

June 24, 2019

ABSTRACT

This report presents a new approach for faster convergence speed in the temporal motif inferring, investigating the combined use of the probabilistic model and the sequence mining technique to make good for deficiency. Specifically, we mainly focus on Probabilistic Latent Sequential Motifs (PLSM) - an extended version of Latent Dirichlet Allocation (LDA), and Interesting Sequence Miner (ISM) - a sequential pattern mining algorithm. Our work attempts to elucidate (1) How is the performance of PLSM model; (2) How the initialization affect the inferred patterns of PLSM model; (3) How to combine ISM with PLSM; (4) What is the impact the combined approach have on the inference algorithm of PLSM model. Using synthetic documents, We first test the performance of PLSM for various aspects. Furthermore, we demonstrate that our approach is able to efficiently speed up the rate of convergence of PLSM model. Through the competition between the single PLSM and the combined approach, we show that efficiency is the direct result of using ISM to find a good initialization for PLSM.

***K*eywords** Probabilistic model · PLSM · Sequential mining techniques · ISM

1 Introduction

Discovering the most important topics from temporal data is an active area. Topic models like Latent Dirichlet Allocation (LDA) and Probabilistic Latent Semantic Analysis (PLSA) have shown a vast potential in successful mining inherent patterns or topics from temporal databases, where each document can be seen as the mixture of the topics. [10] demonstrates that PLSA outperforms traditional approaches in scene classification tasks and is able to extract features, to mine meaningful visual patterns based on the quantized local invariant descriptors. The probability distribution of spatial-temporal visual features and the human-action classes relevant topics in the video sequences can be learned using LDA model.
However, most topic models are not able to deal with the nature of sequential activities by only relying on the exploration of unordered word co-occurrence using the sliding time windows. A Markov clustering topic model is introduced in [9] to discover behaviors and detect the static events on space video data. But it does not consider the independent multiple activities that happen concurrently. In order to solve the correlation of activities over a period of time, [3] manually partition the video to make sure the whole story of the clip happens within the same traffic cycles. Nevertheless, the manual partition could be tedious. A model that could discovery repetitive sequence topics embedded in the documents within a temporal window is introduced in [11]. The model is solved by calling Expectation-Maximization(EM) algorithm, whereas the nature of EM algorithm limits the model to be sensitive to its initialization.

Fortunately, the probabilistic graphical model is not a unique approach to find the inherent topics in the sequential data. The traditional sequential pattern mining techniques tend to find the most frequent or "interesting" sub-sequences contained in a database. But due to the nature of the frequent sequences, the patterns found by these approaches are usually highly redundant and lengthy, which are hard for human beings to explain and make use of. The introduction to the theory of *minimum description length (MDL)* [8] effectively solves the redundancy issue in pattern mining. The exploration based on *MDL* shows the better performance in founded patterns' quality than frequency-based approaches, but MDL-based algorithms depend on the encoding scheme to compress sequence data to a great extent. *Interesting Sequence Miner* in [4] is a sequential mining algorithm viewed from the probabilistic angle. By taking advantages of *structural expectation maximization*[5], the most relevant sequential patterns are inferred effectively in ISM. Moreover, as ISM is based on probabilistic model, it easily incorporates into other probabilistic models.
In this report, we propose a combined approach for inferring the latent variables that integrates the sequential mining technique and the probabilistic model. The contribution of our report is as follows: 1) The complete analysis of PLSM model; 2) An approach for improving the inference speed based on the combined use of ISM and PLSM.
The rest of this report is organized as follows: Section 2 presents the detailed solution and analysis of PLSM model. Section 3 describes the combination method of ISM and PLSM. The advantage of using ISM is also demonstrated through the comparison between single PLSM and the combined method. Section 4 concludes the report and provides future perspectives.

2 Probabilistic Latent Sequential Motif Model

In this section, we first give an overview of the model and then show how to call EM algorithm to infer patterns under PLSM. At the end, we demonstrate the strength and analyse various aspects of PLSM using synthetic documents.

2.1 Model overview

To simplify the problem, we assume there are N_d documents in the database. All span N_{T_a} discrete time steps.
The corpus consist of N_w words, each word w could appear at any absolute time $ta(0 \leq ta \leq N_{T_a})$. The corpus shares N_z latent motifs $p(w, tr|z)$ with a fixed duration of N_{T_r} time steps. Each document has an unique starting time table $p(z, ts|d)$ that spans N_{T_s} time steps. The document is described by the triplets $n(w, ta, d)$ indicating the specific word w appears at the absolute time ta in document d for $n(w, ta, d)$ times. To be clear, $X = (w, ta, d)$ are called observed data in PLSM model, where w, ta, d can be observed as the name implies. We let Z to denote all latent variables, then $(X, Z) = (w, ta, d, z, ts)$ are called complete data in PLSM model. We ignore the relative time in motifs since tr can be decidable as $tr = ta - ts$ when latent variable ts is assigned.

2.2 Generative Process

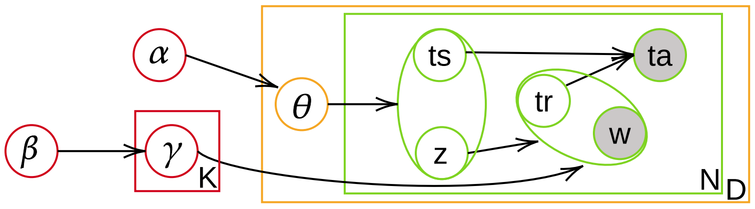


Figure 1: Probabilistic Graphical Model

Figure 1 illustrates the graphical model of PLSM, three layers of PLSM are marked by three colors:

1. corpus-level(red): α ,β and γ are corpus-level parameters, which means that we only need to sample them once during the whole corpus generation process; α and β are prior Dirichlet distributions for θ and γ respectively. γ represents latent motifs.
2. document-level(orange):θ are document-level parameters. Each document needs to sample θ once. It represents the starting time table.
3. word-level(green): ts, z, tr, w, ta are word-level variables. For each word in each document, they should be sampled once. The eclipses around ts, z and w, tr means they are inferred together.

Figure 2 depicts the generation process of one specific document. In our story, for each document d, the generative process is defined as:

- Draw motifs from its prior distribution $\vec{\gamma} \sim Dir(\vec{\beta})$, where γ is a matrix with $N_z \times N_w \times N_{Tr}$ dimensions. This sampling is done once for the whole corpus;
- Draw the starting time from its prior distribution $\vec{\theta} \sim Dir(\vec{\alpha})$, where θ is a $N_z \times N_{Ts}$ matrix;
- Draw the starting time ts and latent topic z, where $p(z, ts|d) \sim$ Multinomial($\vec{\theta}$);
- Draw the word w and relative time tr, where $p(w, tr|z) \sim$ Multinomial($\vec{\gamma}$);
- The absolute time ta of the word is decidable given ts and tr;

Consider the truth that:

$$\begin{aligned} p(z, ts|d) &= p(z|d)p(ts|z, d) \\ p(w, tr|z) &= p(w|z)p(tr|w, d) \end{aligned} \tag{1}$$

the draw of $p(z|d)$ and $p(ts|z, d)$ can be combined into one draw, which is $p(z, ts|d)$. It also applies to $p(w|z)$ and $p(tr|w, z)$, which is $p(w, tr|z)$. Here, $p(z, ts|d)$ is treated as the time table depicted by figure 2, while $p(w, tr|z)$ are considered as motifs.

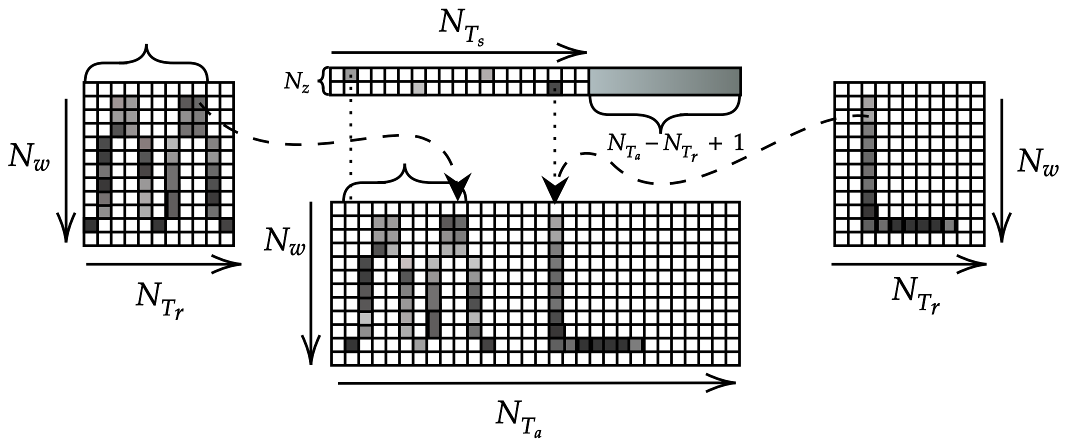


Figure 2: Generative Process

2.3 Model Analysis

Now, we have defined the whole model, all we want is to get the posterior distribution of latent variables. The key to solve this model is to infer two posterior distributions given the documents: $p(w, tr|z)$ and $p(z, ts|d)$. According to the Bayes Rules, they can be written as:

$$\begin{aligned} p(z, ts|w, tr, ta, d) &= \frac{p(z, ts, w, tr, ta, d)}{\sum\limits_{z,ts} p(z, ts, w, ta, tr, d)} \\ p(w, tr|z, ts, ta, d) &= \frac{p(z, ts, w, tr, ta, d)}{\sum\limits_{w,tr} p(z, ts, w, ta, tr, d)} \end{aligned} \tag{2}$$

We can easily factorize the numerator based on the graphical model:

$$p(w, t_a, d, z, t_s) = p(d)p(z|d)p(t_s|z, d)\, p(w, t_a - t_s|z) \tag{3}$$

The heart of the problem is about the denominator. We want to infer the latent motifs and their start time behind each word w in the document d. The denominator of the posterior distribution $p(z, ts|d)$ can be represented as:

$$\begin{aligned} p(w, ta, d) &= \sum_{l}^{N_{T_s}} \sum_{k}^{N_z} p(w, ta, d, tr, z = k, ts = l) \\ &= \sum_{l}^{N_{T_s}} \sum_{k}^{N_z} p(w, ta, d, tr) p(z = k, ts = l) \end{aligned} \tag{4}$$

Therefore, the denominator is the probability of the overall words in the corpus.

$$p(\overrightarrow{w}, \overrightarrow{ta}, \overrightarrow{d}) = \prod_{i=1}^{N_w} \sum_{l}^{N_{T_s}} \sum_{k}^{N_z} p(w_i, ta_i, d_i, tr_i) p(z_i = k, ts_i = l) \tag{5}$$

The denominator gets stuck in the crux of $(N_z + N_{T_s})^{N_w}$. The discrete state space is too large to enumerate [7].

2.4 Model inference

It is intractable to directly compute the posterior distributions. In this section, we will discuss how to call EM algorithm to solve posterior distributions. First, we introduce the concepts of Jensen inequality and KL divergence. Then the formal derivations are given.

Generally, given the observed data, the natural idea is to find model parameters that could maximize the log probability to generate the data in hand. Recalling the generative model, the log-likelihood of the observed data $X = (w, ta, d)$ can be written as:

$$\begin{aligned} \mathcal{L}(W, Ta, D|\theta) &= \prod_{w=1}^{N_w} \prod_{ta=1}^{N_{Ta}} \prod_{d=1}^{N_d} logp(w, ta, d)^{n(w,ta,d)} \\ &= n(w, ta, d) \prod_{w=1}^{N_w} \prod_{ta=1}^{N_{Ta}} \prod_{d=1}^{N_d} logp(w, ta, d) \end{aligned} \tag{6}$$

By decomposing the joint distribution based on the graphical model, the log-likelihood function can be rewritten as:

$$\mathcal{L}(W, Ta, D|\theta) = \sum_{w=1}^{N_w} \sum_{ta=1}^{N_{Ta}} \sum_{d=1}^{N_d} n(w, ta, d) \underbrace{log \sum_{z}^{N_z} \sum_{ts}^{N_{T_s}} p(d)p(z|d)p(ts|z, d)p(w, tr|z)}_{hard\ to\ solve} \tag{7}$$

The natural idea to maximize the log-likelihood function is to take the derivative and set it to be zero. As shown in (7), it seems to be impossible to follow that simple rule as the summation terms inside the log functions that hinder us to step forward. We need to think of a way to simply it.
Here is the part where Jensen inequality comes.The general form of Jensen inequality for a concave function f can be represented as:

$$f(\mathbb{E}(x) \geq \mathbb{E}(f(x)) \tag{8}$$

By making use of Jensen inequality and concavity of the log function, we could apply scaling to the log-likelihood (7) by introducing the artificial distribution $Q(\vec{\gamma})$.

$$\begin{aligned}\mathcal{L}(W,Ta,D|\theta) &= \sum_{w=1}^{N_w}\sum_{ta=1}^{N_{T_a}}\sum_{d=1}^{N_d} n(w,ta,d)\underbrace{log\sum_{z=1}^{N_z}\sum_{ts=1}^{N_{T_s}} Q(\vec{\gamma})\frac{p(w,z,ta,ts,tr,d)}{Q(\vec{\gamma})}}_{log(\mathbb{E}(x))}\\ &\geq \sum_{w=1}^{N_w}\sum_{ta=1}^{N_{T_a}}\sum_{d=1}^{N_d} n(w,ta,d)\underbrace{\sum_{z=1}^{N_z}\sum_{ts=1}^{N_{T_s}} Q(\vec{\gamma})log\,\frac{p(w,z,ta,ts,tr,d)}{Q(\vec{\gamma})}}_{\mathbb{E}[log(x)]}\\ &= \sum_{w=1}^{N_w}\sum_{ta=1}^{N_{T_a}}\sum_{d=1}^{N_d} n(w,ta,d)\sum_{z=1}^{N_z}\sum_{ts=1}^{N_{T_s}} p(z,ts|w,ta,tr,d)log\,\frac{p(w,z,ta,ts,tr,d)}{p(z,ts|w,ta,tr,d)}\end{aligned} \tag{9}$$

After using Jensen inequality, we take the summation terms out of the log function. Please note that we have replaced the artificial distribution $Q(\vec{\gamma})$ with the posterior distribution of latent variables $p(z,ts|w,ta,tr,d)$. Please check 2.4.1 for more details.
Now we have the lower bound of the log-likelihood, which is called ELBO (Evidence Lower Bound). We let $B(\theta)$ denote it. Since $B(\theta)$ is the lower bound of $\mathcal{L}(\theta)$, any θ that can increase $B(\theta)$ can make $\mathcal{L}(\theta)$ grow as well. So, the goal has became to find a θ^{i+1} to make $B(\theta^i)$ maximize. Hence, we could find a good solution for log-likelihood function $\mathcal{L}$.

$$\theta^{i+1} = \arg\max_{\theta} B(\theta^i) \tag{10}$$

We could eliminate the constant term for the maximization of θ. Based on (9) and (10), we could rewrite the function as (11). Here we just split the log function into two parts by using its nature.

$$B(\theta) = \sum_{w=1}^{N_w}\sum_{ta=1}^{N_{Ta}}\sum_{d=1}^{N_d} n(w,ta,d)\sum_{z=1}^{N_z}\sum_{ts=1}^{N_{T_s}} p(z,ts|w,ta,tr,d)log\,p(w,z,ta,ts,tr,d) \tag{11}$$

2.4.1 Why conditional probabilities

We introduced the artificial distributions $Q(\vec{\gamma})$ to help take the summation out of the log function. But why let $Q(\vec{\gamma})$ equal to conditional probabilities of latent variables? It is the nature of Jensen inequality. If we want to achieve the equal-sign in inequality, one constraint must be satisfied:

$$\frac{p(w,z,ta,ts,tr,d)}{Q(\vec{\gamma})} = c \tag{12}$$

where c is a constant. Recall that $Q(\vec{\gamma})$ is the introduced probability density function, so it satisfies:

$$\int_{\gamma} Q(\vec{\gamma})d\gamma = 1 \tag{13}$$

Based on (12) and (13), we could get:

$$\begin{aligned}Q(\vec{\gamma}) &= \frac{p(w,z,ta,ts,tr,d)}{\sum_{z=1}^{N_z}\sum_{ts=1}^{N_{T_s}} p(w,z,ta,ts,tr,d)}\\ &= \frac{p(w,z,ta,ts,tr,d)}{p(w,ta,tr,d)}\\ &= p(z,ts|w,ta,tr,d)\end{aligned} \tag{14}$$

2.4.2 Sparsity Constraint

In order to make inferred patterns as clear as possible, we expect the starting time table to exhibit much higher value at limited positions than others. To encourage this, a regularization constraint is added in the log-likelihood function aiming at maximizing the Kullback-Leibler (KL) divergence between the uniform distribution and $p(z, ts|d)$. The idea behind is simple. The flatter the posterior distribution is, the less peaky it is.

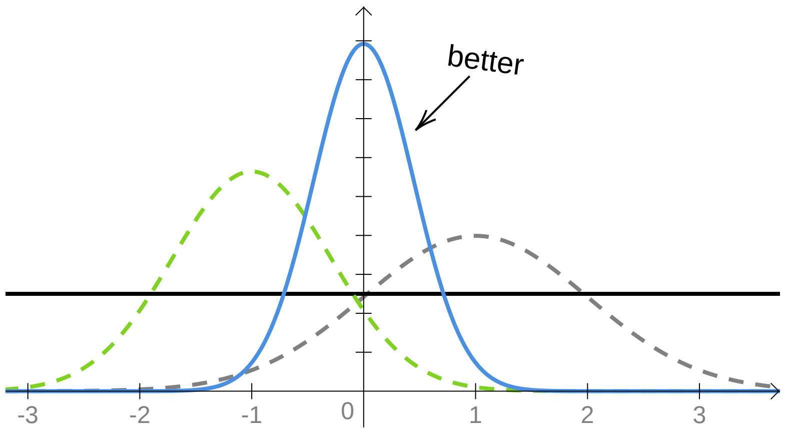


Figure 3: KL Divergence

The "closeness" of two distributions measured by KL divergence can be written as:

$$\begin{aligned} \mathbb{KL}(\mathrm{U}|p(z,ts|d)) &= \sum_{t_s,z} \frac{1}{T_s} \cdot log(\frac{\frac{1}{T_s}}{(p(z,t_s|d)}) \\ &= \sum_{t_s,z} \frac{1}{T_s} \cdot [\underbrace{log(\frac{1}{T_s})}_{constant} - log(p(z,t_s|d))] \\ &= -\sum_{t_s,z} \frac{1}{T_s} log(p(z,t_s|d)) \end{aligned} \tag{15}$$

Again, because we are going to maximize the function, the constant term is thrown away. The expression of θ^{i+1} can be written as:

$$\begin{aligned} \theta^{i+1} &= \arg\max_{\theta}(B(\theta^i) - \sum_{t_s,z} \frac{1}{T_s} logp(z,t_s|d)) \\ &= \arg\max_{\theta} Q(\theta^i) \end{aligned} \tag{16}$$

2.4.3 E step

Consider the truth that the only thing we can handle is $B(\theta)$, on the basis of (14), we first let the lower bound $B(\theta) = \mathcal{L}(\theta)$. This is the task for E step. Figure 4 provides the illustration.

$$p(z,ts|w,ta,d) = \frac{p(z,ts,w,ta,d)}{\sum_{z=1}^{N_z}\sum_{ts=1}^{N_{T_s}} p(z,ts,w,ta,d)} = \frac{p(z,ts,w,ta,d)}{p(w,ta,d)} \tag{17}$$

Next, EM algorithm will execute M step to find the next point θ^{i+1} to make B(θ^{i+1}) reach its maximization. Hence the value of $\mathcal{L}$ can also increase. After M step, we will repeat E -step and M-step until it converges. As illustrated at Figure 4, EM algorithm can not guarantee to find the global maximum.

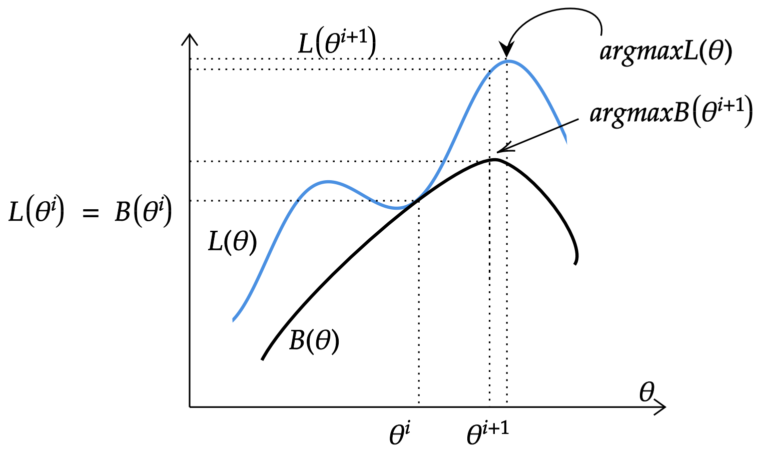


Figure 4: EM algorithm

2.4.4 M step

At the moment, everything is ready. We are about to solve the maximization problem with constraints.

$$\begin{aligned} \theta^{(t+1)} &= \arg\max_{\theta} \sum_{z,ts} P(z,ts|\theta^{(t)}) log P(w,ta,d,z,ts|\theta) \\ \text{s.t.} &\sum_{w=1}^{N_w} \sum_{tr=1}^{N_{T_r}} p(w,tr|z) = 1 \\ &\sum_{z=1}^{N_z} \sum_{ts=1}^{N_{T_s}} p(z,ts|d) = 1 \end{aligned} \tag{18}$$

The natural idea is to introduce the Lagrange multiplier. We could build the Lagrange function as follows:

$$\mathscr{L} = Q(\theta) + \sum_{z=1}^{N_z} \lambda_z (1 - \sum_{w=1}^{N_w} \sum_{tr=1}^{N_{T_r}} p(w,tr|z)) + \sum_{d=1}^{N_d} \mu_d (1 - \sum_{z=1}^{N_z} \sum_{ts=1}^{N_{T_s}} p(z,ts|d)) \tag{19}$$

where $Q(\theta)$ is:

$$\begin{aligned} Q(\theta) = &\sum_{w=1}^{N_d} \sum_{ta=1}^{N_{T_a}} \sum_{d=1}^{N_d} n(w,ta,d) \sum_{z=1}^{N_z} \sum_{ts=1}^{N_{T_s}} p(z,ts|w,ta,d) log\, p(w,ta,d,z,ts) \\ &- \sum_{t_s,z,d} \frac{\lambda_{z,d}}{T_s} \cdot \log(p(z,t_s|d)) \end{aligned} \tag{20}$$

The ultimate goal is to derive two posterior distributions $p(w, tr|z)$ and $p(z, ts|d)$.
Solve p(w,tr|z). First, we need to solve the Lagrange function regarding $p(w, tr|z)$. Take the derivative of $\mathscr{L}$ with respect to $p(w, tr|z)$ and set the derivative to be zero, we could get:

$$
\begin{aligned}
\frac{\partial \mathscr{L}}{\partial p(w,tr|z)} &\Rightarrow \frac{\sum_{d=1}^{N_d}\sum_{ts=1}^{N_{ts}} n(w,ts+tr,d)p(z,ts|w,ts+tr,d)}{p(w,tr|z)} - \sum_{z=1}^{N_z}\lambda_z = 0 \\
&\Rightarrow \sum_{d=1}^{N_d}\sum_{ts=1}^{N_{T_s}} n(w,ts+tr,d)p(z,ts|w,ts+tr,d) = p(w,tr|z)\sum_{z=1}^{N_z}\lambda_z \\
&\Rightarrow \sum_{w=1}^{N_w}\sum_{tr=1}^{N_{T_r}}\sum_{d=1}^{N_d}\sum_{ts=1}^{N_{T_s}} n(w,ts+tr,d)p(z,ts|w,ts+tr,d) = \underbrace{\sum_{w=1}^{N_w}\sum_{tr=1}^{N_{T_r}} p(w,tr|z)}_{1}\sum_{z=1}^{N_z}\lambda_z \\
&\Rightarrow \sum_{z=1}^{N_z}\lambda_z = \sum_{w=1}^{N_w}\sum_{tr=1}^{N_{T_r}}\sum_{d=1}^{N_d}\sum_{ts=1}^{N_{T_s}} n(w,ts+tr,d)p(z,ts|w,ts+tr,d)
\end{aligned}
\tag{21}
$$

After we have λ_z in hand, it could be taken back to the formula to get $p(w, tr|z)$:

$$
\begin{aligned}
&\sum_{d=1}^{N_d}\sum_{ts=1}^{N_{ts}} n(w,ts+tr,d)p(z,ts|w,ts+tr,d) = p(w,tr|z)\sum_{z=1}^{N_z}\lambda_z \\
&\Rightarrow p(w,tr|z) = \frac{\sum_{d=1}^{N_d}\sum_{ts=1}^{N_{T_s}} n(w,ts+tr,d)p(z,ts|w,ts+tr,d)}{\sum_{z=1}^{N_z}\lambda_z} \\
&\Rightarrow p(w,tr|z) = \frac{\sum_{d=1}^{N_d}\sum_{ts=1}^{N_{T_s}} n(w,ts+tr,d)p(z,ts|w,ts+tr,d)}{\sum_{w=1}^{N_w}\sum_{tr=1}^{N_{T_r}}\sum_{d=1}^{N_d}\sum_{ts=1}^{N_{T_s}} n(w,ts+tr,d)p(z,ts|w,ts+tr,d)}
\end{aligned}
\tag{22}
$$

Solve p(z,ts|d). The solution provided by this report has some differences. In [11], two latent parameters $p(z|d)$ and $p(ts|z, d)$ are solved separately due to the truth $p(z, ts|d) = p(z|d)p(ts|z, d)$, which means that the starting time table $p(z, ts|d)$ is split into two probability tables. Here we combine them. Figure 5 shows the differences between both.

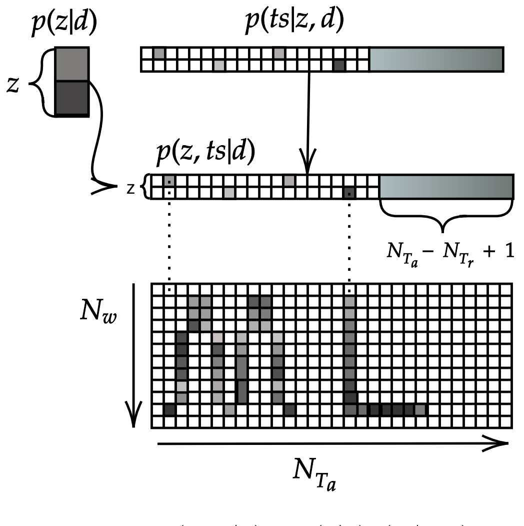


Figure 5: $p(z, ts|d) = p(z|d)p(ts|z, d)$

Take the derivative of $\mathscr{L}$ with respect to $p(z,ts|d)$ and set it to be zero, we could get:

$$\begin{aligned}
\frac{\partial \mathscr{L}}{\partial p(z,ts|d)} &\Rightarrow \frac{\sum_{tr=0}^{N_{T_r}-1}\sum_{w=1}^{N_w} n(w,ts+tr,d)p(z,ts|w,ts+tr,d)}{p(z,ts|d)} - \frac{\lambda_{z,d}}{T_s p(z,ts|d)} - \sum_{d=1}^{N_d}\mu_d = 0 \\
&\Rightarrow \frac{1}{p(z,ts|d)}(\sum_{tr=0}^{N_{T_r}-1}\sum_{w=1}^{N_w} n(w,ts+tr,d)p(z,ts|w,ts+tr,d) - \frac{\lambda_{z,d}}{T_s}) = \sum_{d=1}^{N_d}\mu_d \\
&\Rightarrow \sum_{tr=0}^{N_{T_r}-1}\sum_{w=1}^{N_w} n(w,ts+tr,d)p(z,ts|w,ts+tr,d) - \frac{\lambda_{z,d}}{T_s} = p(z,ts|d)\sum_{d=1}^{N_d}\mu_d \\
&\Rightarrow \sum_{z=1}^{N_z}\sum_{ts=1}^{N_{T_s}}\sum_{tr=0}^{N_{T_r}-1}\sum_{w=1}^{N_w} n(w,ts+tr,d)p(z,ts|w,ts+tr,d) - \frac{\lambda_{z,d}}{T_s} = \sum_{d=1}^{N_d}\mu_d \sum_{z=1}^{N_z}\sum_{ts=1}^{N_{T_s}} p(z,ts|d) \\
&\Rightarrow \sum_{d=1}^{N_d}\mu_d = \sum_{z=1}^{N_z}\sum_{ts=1}^{N_{T_s}}\sum_{tr=0}^{N_{T_r}-1}\sum_{w=1}^{N_w} n(w,ts+tr,d)p(z,ts|w,ts+tr,d) - \frac{\lambda_{z,d}}{T_s}
\end{aligned} \tag{23}$$

When solving the Lagrange function $\mathscr{L}$, we introduce the constraint $\sum_{z,ts} p(z,ts|d) = 1$. For each document, the starting time table is a probability table that sums up to 1. After getting the expression of μ_d, we take it back to the derivative expression and solve $p(z,ts|d)$.

$$\begin{aligned}
&\Rightarrow p(z,ts|d) = \frac{\sum_{tr=0}^{N_{T_r}-1}\sum_{w=1}^{N_w} n(w,ts+tr,d)p(z,ts|w,ts+tr,d) - \frac{\lambda_{z,d}}{T_s}}{\sum_{d=1}^{N_d}\mu_d} \\
&\Rightarrow p(z,ts|d) = \frac{\sum_{tr=0}^{N_{T_r}-1}\sum_{w=1}^{N_w} n(w,ts+tr,d)p(z,ts|w,ts+tr,d) - \frac{\lambda_{z,d}}{T_s}}{\sum_{z=1}^{N_z}\sum_{ts=1}^{N_{T_s}}\sum_{tr=0}^{N_{T_r}-1}\sum_{w=1}^{N_w} n(w,ts+tr,d)p(z,ts|w,ts+tr,d) - \frac{\lambda_{z,d}}{T_s}}
\end{aligned} \tag{24}$$

Because $p(w,tr|z)$ and $p(z,ts|d)$ are both probability tables, we know the denominators are nothing but the normalization parameters. We could say that they are proportional to the numerator. After finishing the inferring, the final normalization will be done at a time.

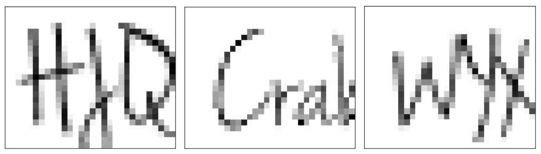
$$\begin{aligned}
p(w,tr|z) &\propto \sum_{d=1}^{N_d}\sum_{ts=1}^{N_{T_s}} n(w,ts+tr,d)p(z,ts|w,ts+tr,d) \\
p(z,ts|d) &\propto max(\epsilon, \sum_{tr=0}^{N_{T_r}-1}\sum_{w=1}^{N_w} n(w,ts+tr,d)p(z,ts|w,ts+tr,d) - \frac{\lambda_{z,d}}{T_s})
\end{aligned} \tag{25}$$

The reason why we add a max function when solving $p(z,ts|d)$ is that we apply a normalization term to guide it to be sparse. Max could ensure it to be a positive number. Otherwise, it will be negative sometimes.

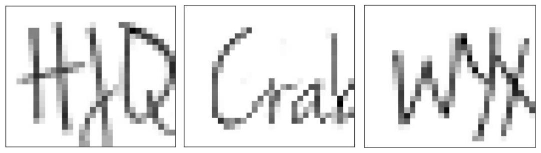
2.5 Experiments on synthetic data

In this section, we investigate the performance and various aspects of PLSM model using synthetic data. In experiments, PLSM is tested on documents with the overlapping in varying degrees.

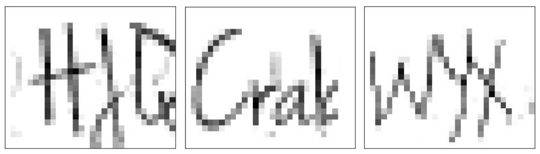
Data Generation. For the sake of explainable patterns, we created three topics ('HJQ','Crab','WYX') filled in $N_w \times N_{T_r}$ matrices $p(w,tr|z)$ as latent motifs($N_z = 3$) to generate two documents($N_d = 2$) of 180 time steps($N_{T_a} = 180$). We assume that each motif has 25 words($N_w = 25$) and spans 40 time steps($N_{T_r} = 40$). The intensities of the motif mean the level of probabilities (the higher, the darker), while the intensities of the document mean the count of the words(the higher, the darker). Figure 6 shows the inferred motifs(a,c,e) from the corresponding documents(b,d,f) along with the increasing level of overlapping. As can be seen, the motifs are well recovered. Even though in the most overlapped document (f), the "Q" in the inferred topic "HJQ" looses partial patterns, but are still easy to identify.


(a) easy motif

(b) easy-level document


(c) mid motif

(d) mid-level document


(e) hard motif

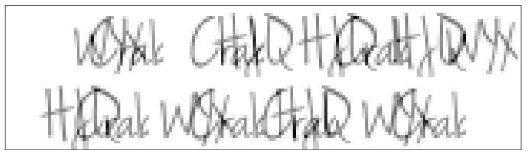
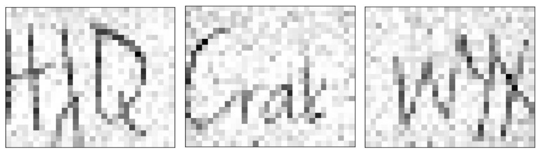
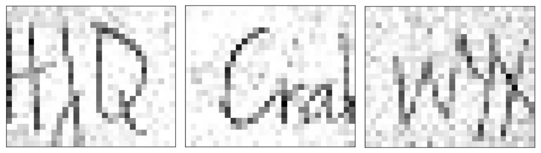

(f) hard-level document

Figure 6: Experiments on clean Documents

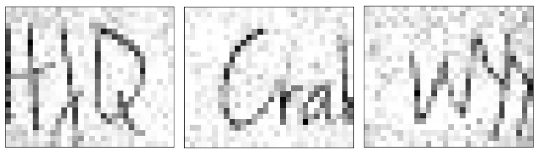
Noisy Document. In order to test the model's robustness to noise, Gaussian noise that follows $N(0, 60)$ is added to documents. In reality, the noise can be viewed as the variability of random events. Figure 7 shows the inferred motifs from noisy documents. We could see that, with the same iteration steps, even though the motifs are not as clear as the inferred in the clean documents, they are still recognizable for human beings.


(a) noisy easy motif

(b) noisy easy-level document


(c) noisy mid motif

(d) noisy mid-level document


(e) noisy hard motif

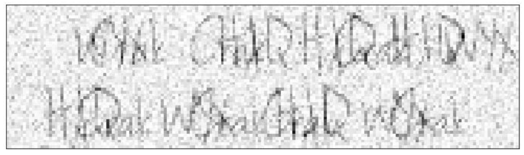
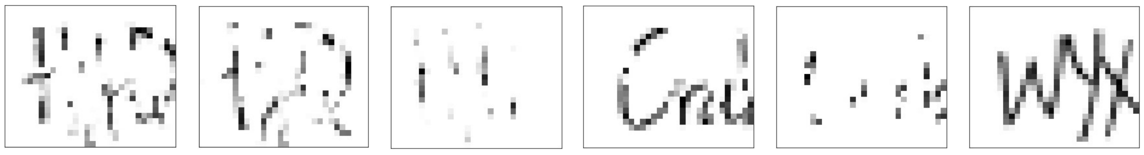

(f) noisy hard-level document

Figure 7: Experiments on Noisy Documents

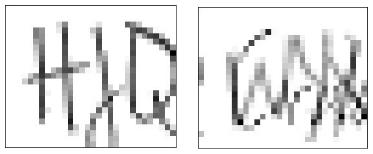
Number of latent topics. In PLSM model, N_z is an hyper-parameter. In this experiment, we will tune the number of N_z to make it higher or lower than the real number of N_z.The easy-level document will be used to illustrate how the change of N_z will affect the performance of the model and the inferred motifs.

As illustrated in Figure 8 and Figure 9, the different number of latent topics will force the real motifs to be split into pieces where each of them captures partial patterns or to be merged together.



(a) HJQ_1 (b) HJQ_2 (c) HJQ_3 (d) $Crab_1$ (e) $Crab_2$ (f) wyx

Figure 8: nz = 6



(a) HJQ (b) $Crab - wyx$

Figure 9: nz = 2

Prior initialization. As we assumed the prior distribution on the parameters, 5 different prior distributions were tested on $p(w, tr|z)$, where one of them is non-informative distribution. Since in reality, we can not obtain an updated prior knowledge that can be applied to all documents, the non-informative Dirichlet prior distribution seems to be the common situation. Our assumption can be validated in Figure 10. Various prior distributions show the disparity on the rate of convergence. But non-informative prior shows the best performance, which is the most reasonable choice under the case that we have no knowledge on $p(w, tr|z)$.

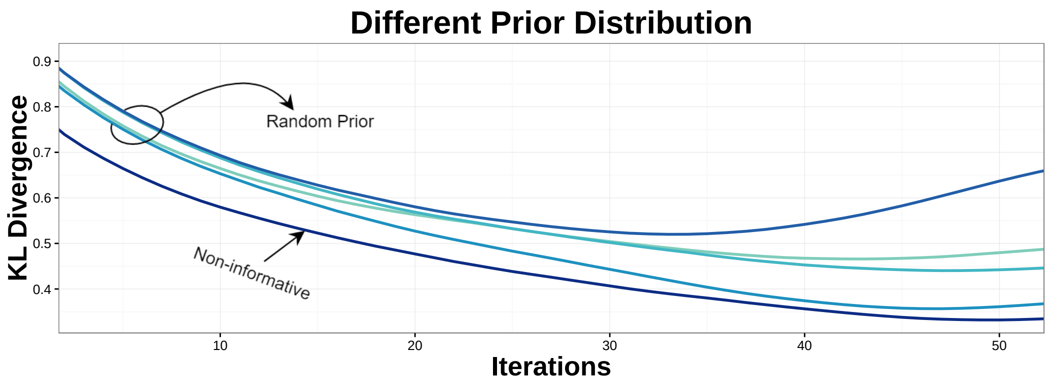


Figure 10: Different Prior Distribution

3 Sequential Mining Techniques and PLSM

As shown by the experiments in 2.5, EM algorithm is sensitive to the prior initialization. It is clear that we can not have any prior knowledge on the starting time table since the starting time could be totally random. But we have the observed data. Consider the essence of motifs, they are real patterns that describe the whole documents. So, is it possible to discover some important patterns using the sequential mining techniques before we conduct the EM algorithm? If possible, therefore, we could use those sequences as the prior knowledge of the motif and point at a good direction for EM algorithm. Hence, maybe we could get better results. In the following, we introduce *Interesting Sequence Miner* [4].

3.1 Interesting Sequence Mining

In the story of ISM, the time property will not be considered. The universe consist of various items $U = \{Item_1, Item_2, Item_3, \ldots\}$. Each sequence S is a list of items. If each item i in $S_a = \{i_5, i_6, i_7, \ldots\}$ also belongs to $S_b = \{i_4, i_5, i_6, i_7, i_8, \ldots\}$, we could say that sequence S_a is the sub-sequence of S_b. For example,$S_a = \{(2, 3)\}$, while $S_b = \{(1), (2, 3)\}$. In this case, $S_a \subset S_b$.

3.1.1 Generative Model

The sequence database $X_1, X_2, X_3, \ldots$ are generated from sub-sequences of Interesting set I, which is the set we want to infer. Consider the truth that the sub-sequence $S \in I$ may be included multiple times in the generated sequence X_i, we could sample the multiplicity z_s for $\forall S_i \in I$, where the indicator variable $z_s \sim Categorical(\vec{\pi})$. After sampling, these sequence from I are used to randomly interleave to construct the database X. Figure 11 shows what is the interleaving. Assuming that there are two sequences $S_1 = \{(1, 2)\}$ and $S_2 = \{(3, 4)\}$. Interleaving of S_1 and S_2 generates the set of sequence $P = \{(1, 2, 3, 4), (1, 3, 4, 2), (3, 1, 2, 4), (3, 1, 4, 2), (1, 3, 2, 4), (3, 4, 1, 2)\}$

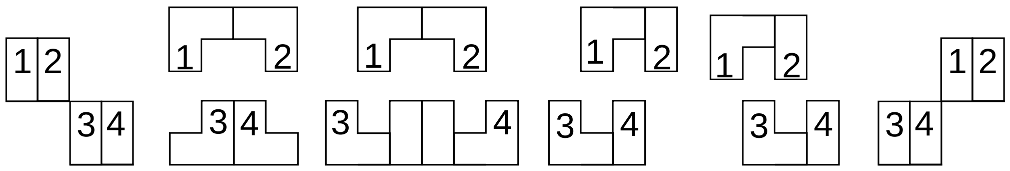


Figure 11: Interleaving

For the sake of clarity, we set the subscript of the variables to be indices, while superscript means different values of one variable. Take $X = \{X_1, X_2, \ldots\}$ for example, X_i means which items we choose, while X_i^j means the j_{th} value of X_i. The formal generation process for each sequence $X_i \in X$ works as follows:

1. $\forall S_i \in I$, sample the multiplicity $z_{S_i} \sim Categorical(\vec{\pi_{S_i}})$, where $\pi_{S_i} \in \Pi$ is a vector of multiplicity probability for on sequence S_i. $z_{S_i} \in \mathbb{Z}$ represents the multiplicity for S_i.
2. If the multiplicity $z_{S_i} \geq 1$, the corresponding S_i is included in the sequence set $\mathbb{S}$. z_{S_i} is an indicator variable.
3. Generate set $\mathbb{P}$ by interleaving all $\forall S \in \mathbb{S}$.
4. Sample X from $\mathbb{P}$ uniformly.

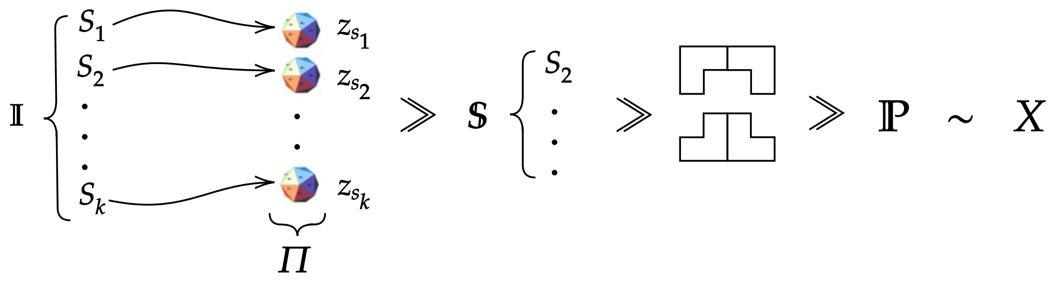


Figure 12: Generative Model of ISM

3.1.2 Inferring and Learning

Given Π and $\mathcal{Z}$, we could then write the posterior probability of generating X:

$$P(X, \mathbb{Z}|\Pi) = \begin{cases} \frac{1}{|P|} \prod_{i=1}^{|\mathbb{I}|} \prod_{m=0}^{|\pi_i|-1} \pi_{S_i^m}^{[z_{S_i}=m]}, & if X \in \mathbb{P} \\ 0, & otherwise \end{cases} \tag{26}$$

where $\prod_{i=1}^{|\mathbb{I}|}$ means to iterate all sequences $S_i \in \mathbb{I}$. $|\mathbb{I}|$ is the number of sequences in the set $\mathbb{I}$. The second product ranging from 0 to $|\pi_i| - 1$ represents the iteration on the vector of multiplicity sampling. Intuitively, it can be treated as the number of faces a dice has. $\pi_{S_i^m}$ is the probability of sampling m from S_i's dice. At last, $[z_{S_i} = m]$ evaluates to 1 only when $z_{S_i} = m$.
It is intractable to directly calculate the length of $\mathbb{P}$ due to the fact that $S_i \in \mathbb{I}$ can appear multiple times. However, recalling the meaning of interleaving, it turns out that $|\mathbb{P}|$ is bounded by the permutations of all items $i_i^k \subset S_i$, where $S_i \in \mathbb{S}$. The set $\mathbb{S}$ has two subsets $S_1 = \{(1, 2)\}$ and $S_2 = \{(3, 4)\}$, for example, the interleaving set $\mathbb{P}$ is bounded by (2+2)!, which is 24. The formal upper bound of $|\mathbb{P}|$ can be written as:

$$|\mathbb{P}| \leq (\sum_{i=1}^{|\mathbb{S}|} |S_i|)! \tag{27}$$

Taking back the upper bound of $|\mathbb{P}|$ into (26) leads to a lower bound of the posterior distribution $P(X, \mathbb{Z}|\Pi)$. It is a familiar scenario. We can not maximize the posterior probability directly. But maximizing its lower bound can also increase the value of $P(X, \mathbb{Z}|\Pi)$ as discussed in section 2.3.3. Maximizing the log-likelihood of the posterior distribution $P(X, \mathbb{Z}|\Pi)$ over $\mathbb{Z}$ [4] can be written as:

$$\mathbb{Z} = \max_{\mathbb{Z}} \sum_{i=1}^{|\mathbb{I}|} \sum_{m=0}^{|\pi_i|-1} [z_{S_i^m} = m] log \pi_{S_i^m} - \sum_{j=1}^{\sum_{i=1}^{|\mathbb{S}|} |S_i|} log j \\ s.t. X \in \mathbb{P} \tag{28}$$

(28) is an NP-hard problem. Hence, we introduce the definition of Submodular Multiset Function.

Definition 1 *For a function f, if there are sets A, B, V where A $\subseteq$ B $\subseteq$ V, and there is e $\in$ V - B, then we have:*

$$f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B) \tag{29}$$

Viewed from the angle of its practical meaning, we could treat a submodular function as a formal way to describe the "diminishing marginal effect". Consider the set as the goods, along with the increasing number of goods you have, the satisfaction of gaining additional things is decreasing. Let τ be the supported sequence of the interesting sequence set $\mathbb{I}$ and define $g(\mathcal{C}) = |\cup_{S \in \mathcal{C}} S|$. As a result, (28) can be transformed to a submodular function f [4]. Formally,

$$f(\mathcal{C}) = \max_{\mathcal{C}} \sum_{i=1}^{|\mathcal{C}|} \sum_{m=0}^{|\pi_i|-1} [\#c(S_i^m) = m] log \pi_{S_i^m} - \sum_{j=1}^{\sum_{i=1}^{|\mathcal{C}|} |S_i|} log j \tag{30}$$

Let τ to be the supported multiset of $\mathbb{I}$ and $g(\mathcal{C}) = |\sup_{S \in \mathcal{C}^{\mathbb{S}}}|$. The NP-hard problem (28) can be transformed as: Find the set $\mathcal{C}$ to maximize the function $f(\mathcal{C})$ given the constraints $\mathcal{C} \in \tau$ [4]. The greedy algorithm thus can be applied to solve the restated problem: Add the items who has the maximal gain, which satisfies the constraint meanwhile.

Algorithm 1 Greedy Algorithm to maximize f

```
Input: Sequence Database X, Interesting supported set $\tau$, Initialize multiset $\mathcal{C} \leftarrow \emptyset$;
  while g($\mathcal{C}$) $\neq |X|$ do
      Choose $S \in \tau$ maximizing $\frac{f(\mathcal{C} \cup \{S\}) - f(\mathcal{C})}{|S|}$;
      $\mathcal{C} \leftarrow C \cup \{S\}$
  end while
  return $\mathcal{C}$
```

At the moment, we have three tasks to do in ISM.

1. Given the interesting set $\mathbb{I}$ and the initialization of the model parameter Π , optimize the latent variables [2];
2. Given Π and $\mathbb{I}$ and current optimal value in (28) by using Greedy algorithm, infer new sequence S' added to the interesting set $\mathbb{I}$ [5].
3. Given Π and $\mathbb{Z}$, rank the retrieved sequence $\mathbb{I}$

Regarding the first problem, ISM takes the hard-EM algorithm[2] to make the parameter estimation. The pseudo code of the algorithm is as follows [4]:

Algorithm 2 HARD-EM

```
Input: Interesting set $\mathbb{I}$;
  The initialization of $\Pi$;
  k $\leftarrow$ 0;
  while $\left\|\Pi^{k-1} - \Pi^{k}\right\|_F > \epsilon$ do
      k $\leftarrow$ k + 1;
      E-STEP: $\forall X^i$ solve (26) by using the greedy algorithm to get $z_S^i \forall S \in \tau_i$;
      M-STEP: $\pi_{S_m^k} \leftarrow \frac{1}{N}\sum_{i=1}^{N}[z_S^i = m] \forall S \in \tau, \forall m$;
  end while
  Remove from $\mathbb{I}$ sequences S with $\pi_{S_0}$ = 1
  return $\mathbb{I}, \Pi^k$
```

In order to infer the new sequence, the candidate sequence S' should be generated first. In ISM, the candidate generation is slightly different, it follows *Highest support first* rule. The candidate generation process can be defined as [4]:

Algorithm 3 CANDIDATE-GEN

```
Input: Interesting set $\mathbb{I}$, cached support $\delta$, queue length q
  if $\nexists$ priority queue $\mathcal{Q}$ for $\mathbb{I}$ then
      Initialize $\delta$-ordered priority queue $\mathcal{Q}$;
      Sort $\mathbb{I}$ by decreasing sequence support suing $\delta$
      for all ordered pairs $S_1, S_2 \in \mathbb{I}$, highest ranked first do
          Generate candidate $S' = S_1 S_2$
          Cache support of $S'$ in $\delta$ and add $S'$ to $\mathcal{Q}$
          if $\mathcal{Q}$ = q then
              break
          end if
      end for
  end if
  Pull the highest-ranked candidate $S'$ from $\mathcal{Q}$
  return S'
```

As for inferring and adding new sequences to the interesting set $\mathbb{I}$, the STRUCTURAL-EM algorithm from [5] is integrated into ISM. The rule for adding a new sequence is simple: only when the inferred sequence S' can improve the averaged value $\bar{q}$ (28) coming from the whole database, S' is added. The pseudo code of STRUCTURAL-EM for one iteration can be described as [4]:

Algorithm 4 Structural-EM (one iteration)

```
Input: Interesting set $\mathbb{I}$; $\Pi$; Optimal value $q^i$ for all sequences $S_i \in X$; Averaged value $\bar{q} \leftarrow \frac{1}{N}\sum_{i=1}^{|\mathbb{I}|} q^i$
  while $\bar{q}' > \bar{q}\{$ until one good candidate found $\}$ do
      Generate candidate S' using Candidate-Gen;
      $\mathbb{I} \leftarrow \mathbb{I} \sup\{S'\}, \pi_{S'} \leftarrow (0, 1, ..., 1)^T)$;
      E-STEP: $\forall X^i, solve(26) toget z_S^i \forall S \in \tau_i$;
      M-STEP: $\pi'_{S_m} \leftarrow \frac{1}{N}\sum_{i=1}^{|\mathbb{I}|[z_S^i=m]} \forall S \in \mathbb{I}, \forall m, \forall X^i$, solve (26) using $\pi'_S, z_S^i \forall S \in \tau_i$ to get the optimal $q^i$;
      Set new profit $\bar{q'} \leftarrow \frac{1}{N}\sum_{i=1}^{|\mathbb{I}|} q^i$;
      $\mathbb{I} \leftarrow \mathbb{I} \backslash \{S'\}$
  end while
  $\mathbb{I} \leftarrow \mathbb{I} \cup \{S'\}$
  return $\mathbb{I}, \Pi'$
```

The first two problems have been solved, when we need to rank the sequences in the interesting set $\mathbb{I}$, how could we measure the corresponding "importance"? Instead of ranking by its support, an alternative way called *interestingness* is used in ISM. The definition is as follows:

Definition 2 *The interestingness for $\forall S_i \in \mathbb{I}$ is defined as:*

$$int(S_i) = \frac{\sum_{m=1}^{|z_{S_i}|}[z_{s_i}^m \geq 1]}{Supp(S_i)} \tag{31}$$

interestingness ranges from 0 to 1, where 1 means the most interesting.

The complete Interesting Sequence Miner algorithm is formed as[4]:

Algorithm 5 ISM

```
Input: Input $\forall X_i \in$ database X;
  Initialize I with singletons, $\Pi$ with their supports;
  while not converged do
      Add sequence to $\mathbb{I}$,$\Pi$ using STRUCTURAL-EM;
      Optimize parameters for $\mathbb{I}$,$\Pi$ using HARD-EM;
  end while
  return $\mathbb{I}, \Pi$
```

3.2 Combination of PLSM and ISM

We have introduced the PLSM and ISM. The ultimate goal in this section is to discuss the way to combine two methods in order to improve the inference algorithm in PLSM. But first, we need to face the truth that ISM does not take the time property into consideration. So there are two problems in front of us:

1. how could we feed data of PLSM model into ISM?
2. After getting important sequences from ISM, how could we reconstruct the sequences to be motifs' prior distribution?

Recall the generative model of PLSM, the latent variables $p(w, tr|z) \sim Multinomial(\vec{\gamma})$ and $p(z, ts|d) \sim Multinomial(\vec{\theta})$. Hence, data-likelihood is a product of multinomial distributions. Besides, both posterior distributions have a Dirichlet prior distribution. Based on the truth of the conjugation of Dirichlet-Multinomial, we could have the following proved relations [1]:

$$Dir(\vec{p}|\vec{\theta}) + Multinomial(\vec{m}) = Dir(\vec{p}|\vec{\theta} + \vec{m}) \tag{32}$$

Therefore, we could discover important patterns by using ISM to be the prior distribution of the motif. After the initialization, it still follows the Dirichlet distribution. We could set it as the prior distribution for PLSM and start the inference algorithm.

3.2.1 Generate sequences for ISM

In the sequence generation, the number of times that one word appears will not be considered, which means that we only care if the word w appears at the specific instant time point or not.
Figure 13 illustrates the way how sequences are generated from $n(nw, ta, d)$. First, a window with the same size of latent motifs($N_w \times N_{T_r}$) is used to slide through the whole database (documents by documents) with a sliding step a. The sliding step a is a hyper-parameter in the generation. It controls the 'density' of the generated sequence. The smaller the step is, the more sequences it will generate. On the contrary, the larger sliding step may loss some patterns inside the document. Furthermore, as can be seen in figure 13, each square in the sliding window has been assigned a number in order to capture the position of the pattern. The intuition is that "important patterns" will repeat several times along with the sliding. After getting sequences, we could feed them into ISM and get the "Interested Sequences", which can be further used to initialize the motif.

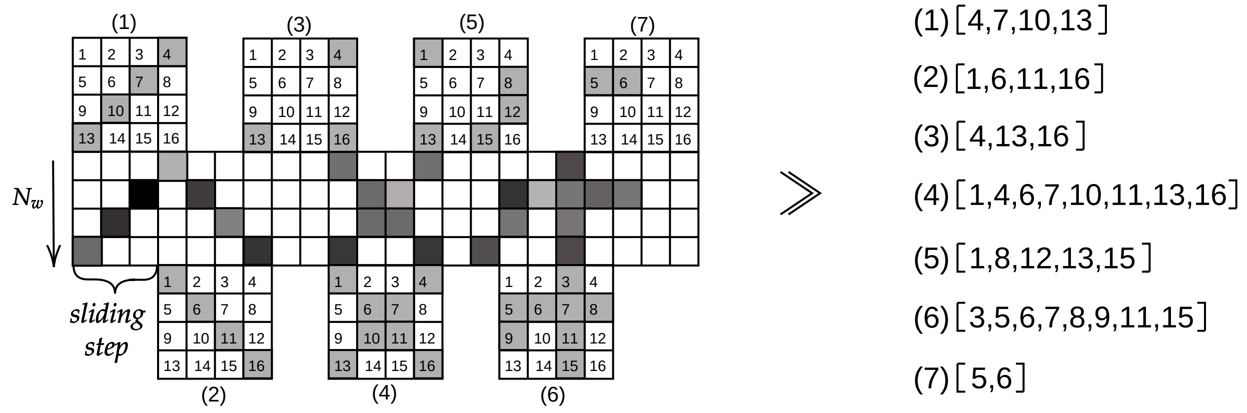


Figure 13: Sequence Generation

3.2.2 Reconstruction from the sequences to the motif

Before making use of the sequences returned by ISM, there are two problems in the way. Due to the nature of ISM, it will select the most 'interesting' sequences ordered by its *interestingness* as mentioned at (31). In most cases, these interesting sequences are singletons, which is also the common case within Data Mining field. Therefore, since We want some sequences that could capture partial important patterns of the document, we will first weed out the singletons from returned sequences. Then, we will order the rest of sequences by their *interestingness*.
Secondly, even though we have eliminated all singletons. The number of left sequences is still much larger than the number of latent motifs. But we should use sequences that belong to the same latent motif to do the initialization. Under this circumstance, in this report, we only choose top k-ranked sequences to conduct the initialization, where k is the same number with the number of the hypothetical motifs N_z.
In addition to the problem above, ISM also return some redundant sequences. For example $S_1 = [8, 15]$ and $S_2 = [3, 5, 8, 15]$, where $S_1 \subset S_2$. In this case, we will select the higher-ranked sequence and eliminate its sub-sequence in the following.
Do not confuse the element in each sequence with the initialized weight of N_z latent motifs. The elements in the sequences represent the position. After selecting the sequences, we first create N_z initialized motifs with one as their weights, then slide those motifs with assigned the position through the whole database. By making use of the conjugation of Dirichlet-Multinomial, we could simply add weights in the document to the corresponding position in the motif. And we do this for N_z motifs. So far, we finish the initialization of the motifs. The reconstruction can be defined as follows:

1. Eliminate the singletons. According to the order of interestingness, eliminate the redundant sequences;
2. Order the rest sequences by its interestingness;
3. Assign sequences into N_z motifs;
4. Make motifs slide through the whole database to initialize weights;

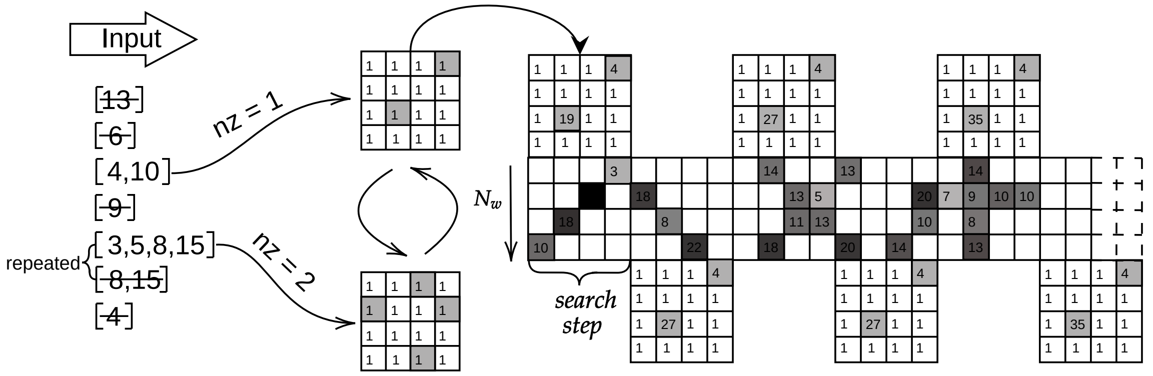


Figure 14: Search Pattern

As illustrated in Figure 14, we assume there are two latent motifs in the document. The input is ISM sequences ordered by interestingness. First, the singletons are weeded put. Then, repeated sequences are eliminated based on interestingness. During the pattern search, the sliding step(search step in the figure) could be different from the sliding step in Figure 13. The smaller the step, the higher the weight.

3.2.3 Combination

The complete combination method can be concluded as follows:

1. Feed data into ISM following the sequence generation rule;
2. Clean sequences from ISM based on rules at 3.2.2;
3. Select sequences according to N_z;
4. Initialize the weights by sliding through the documents.

Figure 15 illustrates the complete combination process of using ISM to initialize the latent motifs. After getting the initialized motifs, we could feed motifs into the PLSM model as discussed in section 2.

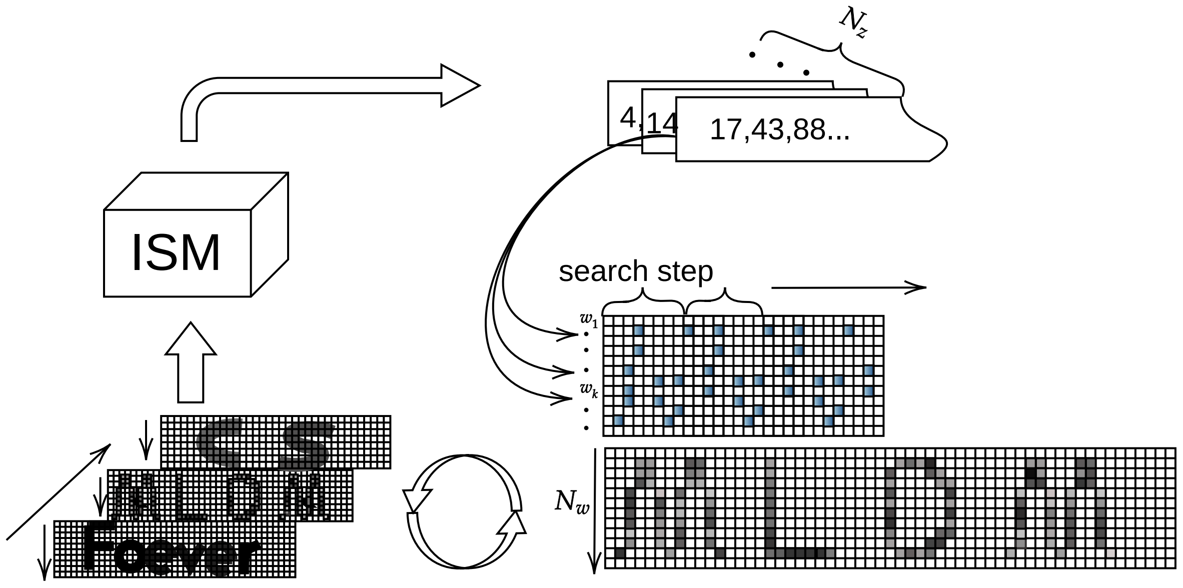


Figure 15: ISM and PLSM

3.3 Model Selection

Now, we are able to use ISM to initialize the prior distribution of latent motifs. However, there are still too many sequence candidates in hand after the elimination at 3.2.2. Furthermore, different sequences may belong to one real motif. If we assign two sequences that originally belong to the same real motif to two hypothetical motifs, it is predictable that such the initialization will bring some distractions to the inference algorithms. So, can we judge if two sequences belong to one latent motif or not? Unfortunately, we do not know what are real motifs not even to judge. Also, the most usual scenario in graphical models is to use non-informative Dirichlet distribution to be the prior distribution as we have no knowledge about latent variables normally. As shown in [6], under the non-informative prior distribution, LDA model shows the equivalent performance with PLSA model. So, in order to validate the strength of our method, we need to present its superiority compared with Non-informative prior distribution.

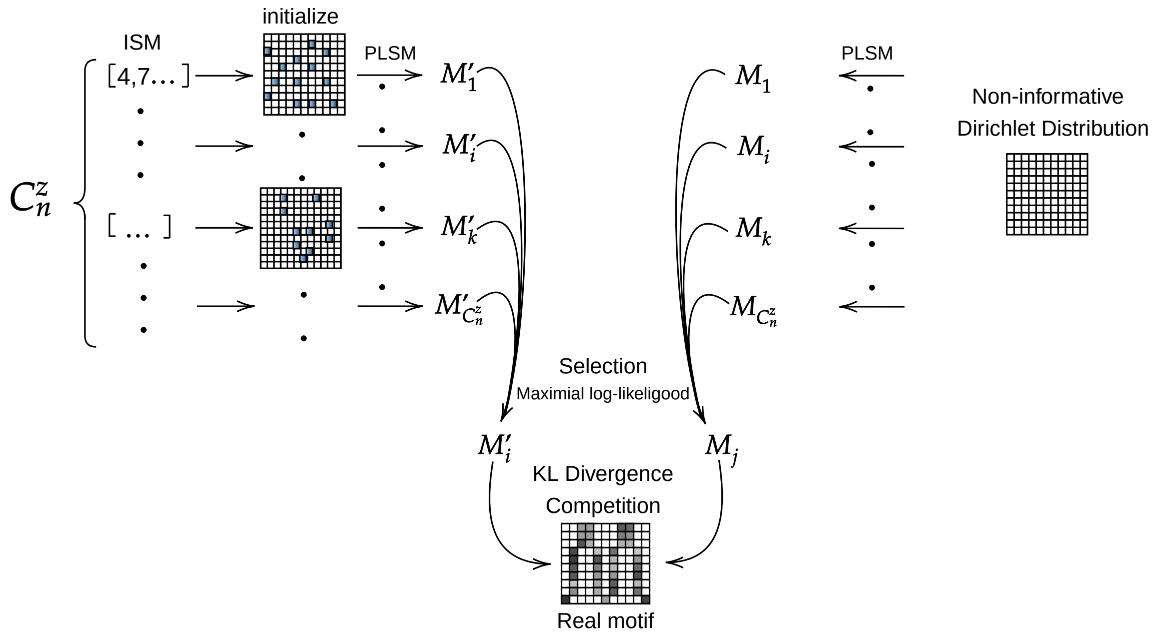


Figure 16: Model selection

Since we have larger number sequence candidates, We first try all the combinations of sequences. We choose z sequences(hypothetical number of motifs) from n sequence candidates, which is C_n^z. Recalling the meaning of the latent motifs, it is the posterior distribution in the log-likelihood function. So the inferred motifs should be able to increase the value of the log-likelihood function. Specifically, as illustrated in Figure 16, each item in the full combination set will be used to initialize the motif that will be further fed into PLSM to get the inferred motifs. Meanwhile, once the inferred motifs are obtained, we are able to calculate the corresponding value of the log-likelihood function. In order to evaluate the performance of the methods using ISM, the best combination of sequences is selected from which the full combination set based on the value of the log-likelihood function(M'_i in the figure). Correspondingly, since we choose the best from C_n^z possibilities, to be fair, we test setting non-informative distribution as the prior distribution and draw different multinomial initialization of motifs for C_n^z times as well. The best-inferred motif is chosen (M_j in the figure) following the same rule. The real motifs are introduced at the end of the competition to calculate the KL Divergence between the two best-inferred motifs from two methods.

Time Complexity. In our experiments, the size of the database is $2 \times 25 \times 180$, where 2 represents documents N_d, 25 means number of words N_w, 180 is the absolute times steps N_{T_a}. The average running time of PLSM (up to 100 iterations) is 10.8 seconds, which is calculated by 100 experiments. When it faces C_n^z, the time price is too high to pay. Check section 4 for details.

Improvement. Good prior distribution should generate good posterior distributions $p(w, tr|z)$ with high probabilities that could give higher value on the log-likelihood function. Instead of running PLSM for C_n^z times, we can calculate the log-likelihood value from the first sampling of the prior distribution initialized by ISM. We pick up the one with the highest value on the log-likelihood function. We do the same for the non-informative prior distribution. By doing this, we skip the PLSM and select the best one, which is a trick to save time.

3.4 Experiment results

We conduct experiments to validate the strength of the combined method. Using synthetic data, we show that the prior distribution initialized by ISM promote the convergence rate of the inference algorithm in PLSM. Specifically, the non-informative prior distribution and the prior motifs initialized by ISM are tested on each database. The closeness between the inferred motifs and the real motifs is compared after each iteration step, where the closeness is measured by KL divergence.

Documents. The documents used in the experiments are the same as the documents in section 2.5. The documents are split into three levels: easy(clean), middle(overlap), hard(most overlap).
KL Divergence. In experiments, we mainly focus on the quality of inferred motifs. But in PLSM model, $p(z, ts|d)$ is also the latent variable. It controls the starting time of the latent motif, which can be considered as the deviations of the motifs. Figure 17 shows the deviations in different inferred motifs (2) and (3). In such a circumstance, the KL divergence between the real motifs and the inferred motif is larger. But those are still good motifs from our perspective. Therefore, the inferred motif could have some deviations compared to the real motif used to generate documents. In this case, we limit the font(for the sake of recognizable pattern, we use font to simulate the pattern in the document.) to full fill the motif matrix. By doing so, we could bound the algorithm to infer the latent motif with small shift compared to the original one.

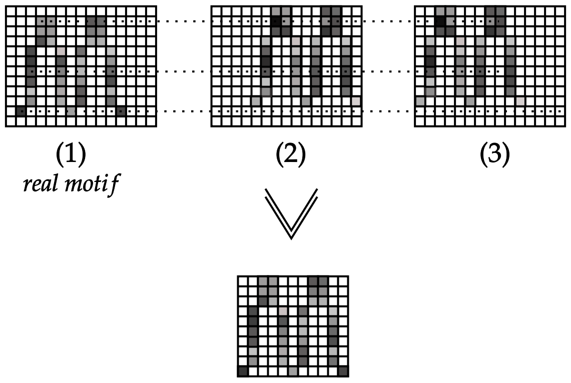
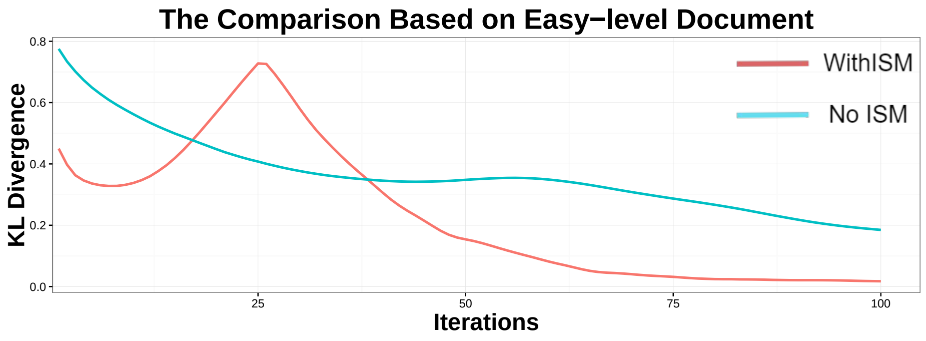


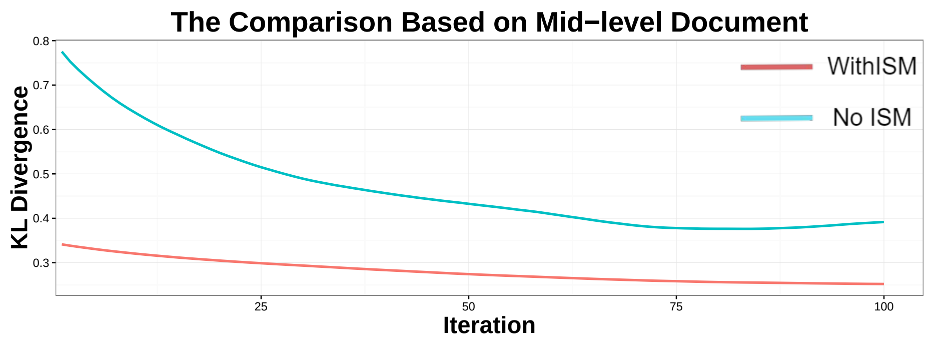
Figure 17: Bounded motif

Comparison Rules. In experiments, the number of iterations will be fixed to 100 for both methods. The temporal motifs during the inference are recorded after each iteration. The final evaluation depends on the "closeness" between the inferred motifs and the real motifs.

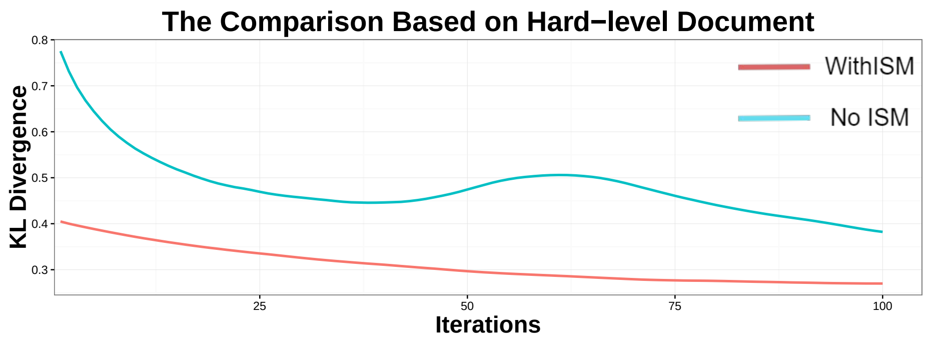
Figure 18 shows the experimental results taken on documents. For the sake of clarity, the calculated KL Divergence has been normalized to 1. As can be seen in Figure 18, the motifs initialized by ISM show the higher "closeness" with the real motifs at the starting point. Meanwhile, it shows a faster rate of convergence. on the easy-level documents, even though the KL divergence increases in red line due to the deviations, after around 40 iterations, it shows better performance than the method without ISM. On the mid-level and the hard-level documents, the ISM-initialized method clearly shows its power, it is always lower than the non-informative one.



(a) Easy-level



(b) mid-level



(c) hard-level

Figure 18: Competition of using ISM and not Using ISM

4 Conclusion

In this report, we proposed a combined approach to improve the inference algorithm in the probabilistic model using sequential mining techniques. Using synthetic data, various aspects of PLSM and the combined approach of PLSM and ISM were analyzed separately. We demonstrated the superiority of the combined approach on three datasets with different overlapping levels, showing that the initialized prior by ISM outperforms the usual non-informative prior. However, we discovered some underlying thorny problems in our approach. The alternative directions for the combined use of ISM and PLSM were provided along with the following conclusions:

1. The growth rate of the size of the full combination set;
2. The limitation length of the single sequence ISM can handle;

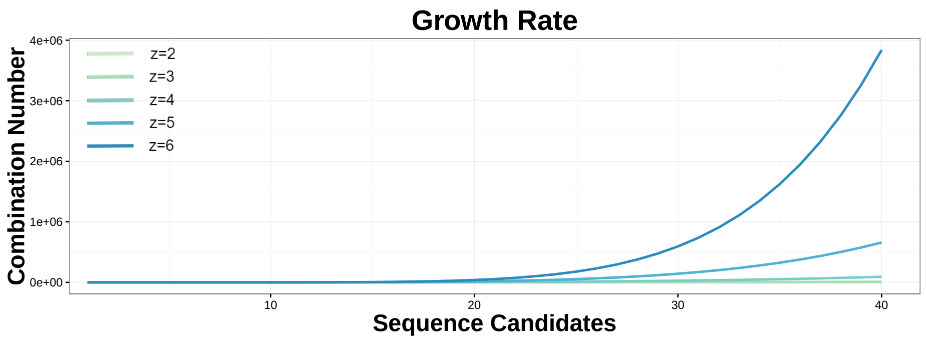


Figure 19: Growth Rate

Growth Rate of the combination set. As shown at 3.3, in order to choose the best combination of sequences to initialize the motifs, we need to try all possible combinations, which is C_n^z. This number grows exponentially along with the increasing number of sequence candidates and the guess of latent motifs as shown in Figure 19. We could see that the size of the combination set has reached 4×10^6 when sequence candidates are merely 40 and the hypothetical number of motifs is 6. However, even for the small documents we tested in the experiments $(2, 25, 180)$, we have around 60 candidate sequences after the elimination. For complex documents, it is also common that there are more than 6 latent motifs. Therefore, the full combination set size will become prohibitively large to deal with.
Possible Solution. One way to handle this is to apply the genetic algorithm on the sequence selection. In this case, we could set the log-likelihood function as its fitness function as we discussed in 3.3. Therefor, we may find a good combination in a acceptable amount of time.

Limitation sequence length of ISM. Because of the way ISM works: it uses greedy algorithm to decide if the sequence candidates could make good compressing patterns, there is an known shortcoming for ISM. It can not handle a single long sequence (roughly around 10^4 items). Recalling the way how we combine two models: the sliding window with the same size of latent motif $N_w \times N_{T_r}$ is used. The single sequence length depends on the size of motifs. Along with the increase of N_w and N_{T_r}, ISM is not able to solve it in a reasonable amount of time. But this case is also the common the case when we face larger documents (either N_w or N_{T_r} is large).
Possible solution. One way to deal with this problem is to split sequences into pieces. The danger with this is also clear: we may loss some patterns, especially for the longer one.

Future Perspective. In the experiments, we have demonstrated the strength of the combined approach. But if we do more iterations, the disparity between the non-informative prior and our approach will decrease. Viewed the inference algorithm from the angle of Optimization, we were trying to optimize the lower bound of the log-likelihood. In the experiments, this report called the simple and the most natural one - Gradient Descent. So, instead of combining PLSM with other methods, focusing on the optimization methods may bring us different experience.

References

[1] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[2] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[3] Tanveer A Faruquie, Prem Kumar Kalra, and Subhashis Banerjee. Time based activity inference using latent dirichlet allocation. In *BMVC*, pages 1–10, 2009.

[4] Jaroslav Fowkes and Charles Sutton. A subsequence interleaving model for sequential pattern mining. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 835–844. ACM, 2016.

[5] Nir Friedman. The bayesian structural em algorithm. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 129–138. Morgan Kaufmann Publishers Inc., 1998.

[6] Mark Girolami and Ata Kabán. On an equivalence between plsi and lda. In *SIGIR*, volume 3, pages 433–434, 2003.

[7] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.

[8] Peter D Grünwald. *The minimum description length principle*. MIT press, 2007.

[9] Timothy Hospedales, Shaogang Gong, and Tao Xiang. A markov clustering topic model for mining behaviour in video. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1165–1172. IEEE, 2009.

[10] Pedro Quelhas, Florent Monay, Jean-Marc Odobez, Daniel Gatica-Perez, and Tinne Tuytelaars. A thousand words in a scene. *IEEE transactions on pattern analysis and machine intelligence*, 29(9):1575–1589, 2007.

[11] Jagannadan Varadarajan, Rémi Emonet, and Jean-Marc Odobez. A sequential topic model for mining recurrent activities from long term video logs. *International journal of computer vision*, 103(1):100–126, 2013.