Journal of the Royal Statistical Society
SERIES B Statistical Methodology

*J. R. Statist. Soc.* B (2014)
**76**, *Part* 2, *pp.* 319–347

# Association pattern discovery via theme dictionary models

Ke Deng,

*Harvard University, Cambridge, USA, and Tsinghua University, Beijing, People's Republic of China*

Zhi Geng

*Peking University, Beijing, People's Republic of China*

and Jun S. Liu

*Harvard University, Cambridge, USA*

[Received January 2011. Final revision April 2013]

**Summary.** Discovering patterns from a set of text or, more generally, categorical data is an important problem in many disciplines such as biomedical research, linguistics, artificial intelligence and sociology. We consider here the well-known 'market basket' problem that is often discussed in the data mining community, and is also quite ubiquitous in biomedical research. The data under consideration are a set of 'baskets', where each basket contains a list of 'items'. Our goal is to discover 'themes', which are defined as subsets of items that tend to co-occur in a basket. We describe a generative model, i.e. the theme dictionary model, for such data structures and describe two likelihood-based methods to infer themes that are hidden in a collection of baskets. We also propose a novel sequential Monte Carlo method to overcome computational challenges. Using both simulation studies and real applications, we demonstrate that the new approach proposed is significantly more powerful than existing methods, such as association rule mining and topic modelling, in detecting weak and subtle interactions in the data.

*Keywords*: Association rule mining; Co-occurrence pattern recognition; Market basket analysis; Rejection control sampling; Sequential Monte Carlo methods; Topic modelling

## 1. Introduction

In many research areas ranging from data mining to bioinformatics, a key task is to identify associations between various 'items'. To be concrete, we let $\mathbf{X} = (X_1, \ldots, X_p)$ denote a vector of binary variables, where $X_j = 1$ or $X_j = 0$ indicates the presence or absence of item $j$. Given a set of observations on $\mathbf{X}$, we are interested in discovering 'patterns' among the items, defined as subsets of the items that tend to co-occur more frequently than expected by chance. These patterns can be more generally interpreted as interactions between the binary variables. A well-known example of the problem is the *market basket analysis* (MBA) that was proposed by Piatetsky-Shapiro (1991). Table 1 shows a list of transaction records in a supermarket, a typical data set in MBA, where each row records a 'basket' containing several items. The data can be presented as a binary matrix with rows for different transactions or baskets and columns for

*Address for correspondence*: Jun S. Liu, Department of Statistics, Harvard University, 1 Oxford Street, Cambridge, MA 02138, USA.
E-mail: jliu@stat.harvard.edu

 1369–7412/14/76319

**Table 1.** Typical subset of the data in MBA

| *Customer* | *Basket* |
|---|---|
| $C_1$ | Chips, salsa, cookies, crackers, Coke, beer |
| $C_2$ | Lettuce, spinach, oranges, celery, apples, grapes |
| $C_3$ | Chips, salsa, frozen pizza, frozen cake |
| $C_4$ | Lettuce, spinach, milk, butter |

the presence or absence of different items. The goal of MBA is to find whether some items tend to be sold together, which may reveal consumers' behaviours and help the managers to improve their advertising and promotion strategies.

An intuitive first analysis of the MBA data appears trivial: for any pair of items (such as Coke and frozen pizza), one only needs to count their respective times of occurrences, both individually and as pairs. Then, one can compute the time-honoured $\chi^2$-statistic to test whether they have co-occurred more frequently than expected by chance. In the literature, computer scientists have developed a highly efficient algorithm called association rule mining (ARM) (Piatetsky-Shapiro, 1991; Agrawal *et al.*, 1993; Agrawal and Srikant, 1994), which not only finds co-occurring pairs on the basis of $\chi^2$-statistics (though ARM does not use any formal statistical test) but also searches recursively for multi-item associations. In ARM, all *frequent item sets* are enumerated and *association rules* are generated from these frequent item sets. However, this strategy may encounter difficulties when we try to analyse item sets with weak pairwise but significant multi-item associations. Although much effort (Zaki, 2000; Han *et al.*, 2004; Webb, 2007) has been made to improve the sensitivity and specificity of ARM, this method still tends to produce many redundant or false association patterns.

Some off-the-shelf statistical tools such as hierarchical clustering and $k$-means clustering can also be applied to the MBA-type data. But it is generally difficult to obtain satisfactory results by using these approaches when the data under consideration have the following characteristics:

(a) the potential patterns can heavily overlap;
(b) the potential patterns involve many items;
(c) some of the multi-item patterns are marginally weak, which manifests in very low pairwise correlations.

In these cases, most of the aforementioned methods fail because they lack the ability to handle more than two items at a time, usually resulting in very high false positive and false negative rates.

To consider multiple items simultaneously, we introduce a probabilistic generative model named the *theme dictionary model* (TDM), which is inspired by the dictionary model of Bussemaker *et al.* (2000), and propose a few novel methods for discovering co-occurrence patterns and conducting parameter estimations. In TDM, we treat each item as a basic unit and potential patterns as *themes*. Each transaction, which is generally termed a *collection*, is constructed by mixing a small number of themes selected from the dictionary. A probabilistic model can be prescribed to govern the theme selection process. What we can observe, however, is only the aggregation of all items in each collection instead of the actual themes that make up the collection. Under this framework, the pattern identification problem is converted to a model selection problem with missing data, which can be solved by either Bayesian model selection or a stepwise

strategy that employs the expectation–maximization (EM) algorithm (Dempster *et al.*, 1977) to estimate the parameters and a testing procedure to identify new themes.

The TDM is closely related to topic models, such as latent Dirichlet allocation (Blei *et al.*, 2003), the dynamic topic model (Blei and Lafferty, 2006) and the correlated topic model (Blei and Lafferty, 2007). The themes in the TDM and the topics in topic models share many similarities, both containing a group of items, aiming to discover item associations, and being basic building blocks to generate the observed data under a probabilistic missing data framework. The key difference between the TDM and topic models is that they focus on different types of relationships. In a TDM, each theme usually contains only a few items, but the number of themes can be very large. When a theme is chosen, all items in it are chosen, i.e. the items in one theme must act together. In topic models, however, the number of topics is usually small, but each topic contains a large number of items. When a topic is chosen, its associated items appear independently with given probabilities so the items in a topic do not need to occur altogether. As a consequence, the TDM focuses on tight association patterns of items, whereas topic models focus on loose correlations on a global scale.

Another important difference is that the TDM and topic models favour different types of data. Because of the computational bottleneck, a TDM is efficient only when the number of items in the baskets is relatively small, e.g. a few dozens. Large baskets containing too many items may significantly slow down the learning process of TDMs. In contrast, topic models work well only for large baskets. To recognize topics, a large number of repeated samples from each relevant topic of a basket are needed, which is feasible only when the baskets contain a large number of items. If most of the baskets are small, topic models usually fail to detect any useful pattern from the data. Thus, generally speaking, we recommend topic models if the baskets or documents concerned contain a large number of items or words, and the main interest is to do basket or document clustering or classification based on a small number of features. In contrast, we recommend TDMs for association discovery in a group of small baskets or documents, where we emphasize detecting tight co-occurrence patterns of items or words.

The remainder of the paper is organized as follows. In Section 2, we define the TDM formally and prove the identifiability of the model. In Section 3, we describe both a Bayesian model selection procedure and a stepwise method for discovering the unknown theme dictionary, with the latter method being useful for large data sets. In Section 4, we discuss how to solve the computation problems in TDM estimation via sequential Monte Carlo (SMC) approaches. A novel SMC approach, called sequential rejection control sampling (SRCS), is proposed, and its performance is evaluated and compared with existing methods. In Section 5 we present a simulation study to illustrate the general performance of our method and to compare with other methods. In Section 6, we apply the TDM to several real data sets. At the end, we discuss potential extensions of the proposed method in Section 7.

The data that are analysed in the paper can be obtained from

`http://wileyonlinelibrary.com/journal/rss-datasets`

## 2. Theme dictionary model

Let $\mathcal{L} = \{L_1, \ldots, L_p\}$ denote the set of all basic units called *items*. A *theme* $\alpha$ is defined as a multiset on $\mathcal{L}$, which can be represented by a vector $\alpha = (n_1^\alpha, \ldots, n_p^\alpha)$, where $n_j^\alpha$ is the number of occurrences of item $L_j$ in theme $\alpha$. An alternative representation of a theme is to list all its items directly. For example, suppose that $\mathcal{L} = \{A, B, C, D\}$. Theme $\alpha_{AB} = (1, 1, 0, 0)$ can also be represented as $\{A(1), B(1)\}$, or more conveniently $AB$; and theme $\alpha_{AA} = (2, 0, 0, 0)$ can be

represented by $\{A(2)\}$ or $AA$. In this paper, we shall use these different theme representations interchangeably without further notice. If two themes $\alpha$ and $\beta$ satisfy $n_j^\alpha \leqslant n_j^\beta$ for all $1 \leqslant j \leqslant p$, we say that $\alpha$ is *covered* by $\beta$, which is denoted by $\alpha \subseteq \beta$; furthermore, if $n_j^\alpha < n_j^\beta$ for some $j$, we say that $\alpha$ is *strictly covered* by $\beta$, which is denoted by $\alpha \subset \beta$.

A *theme dictionary* $\mathcal{D} = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$ is a set of themes. A *collection* $S$ is the sum of all items contained in a number of themes selected from $\mathcal{D}$. The *binary TDM*, which is denoted as $\mathcal{M}_B$, postulates the following probability model for producing a collection $S$:

$$P(S|\mathcal{D}, \boldsymbol{\theta}) = \left( \prod_{\alpha \in S} \theta_\alpha \right) \prod_{\alpha \notin S} (1 - \theta_\alpha), \tag{1}$$

where $\boldsymbol{\theta} = \{\theta_\alpha\}_{\alpha \in \mathcal{D}}$, and $0 \leqslant \theta_\alpha \leqslant 1$ for all $\alpha \in \mathcal{D}$. In words, model $\mathcal{M}_B$ assumes that a collection is generated by selecting each theme $\alpha$ independently from the dictionary $\mathcal{D}$ with probability $\theta_\alpha$. A collection can also be represented by a vector of binary indicators $\{I_\alpha\}_{\alpha \in \mathcal{D}}$, where $I_\alpha = 1$ if theme $\alpha$ is present in the collection and $I_\alpha = 0$ otherwise. To illustrate, Table 2 shows the generating process of the collection $S = \{A, B, CD\}$ from dictionary $\mathcal{D} = \{A, B, C, D, AB, CD\}$ under $\mathcal{M}_B$. The binary vector of $S$ under $\mathcal{M}_B$ is $(1, 1, 0, 0, 0, 1)$; thus the probability of $S$ is

$$P(S|\mathcal{D}, \boldsymbol{\theta}) = \theta_A \theta_B (1 - \theta_C)(1 - \theta_D)(1 - \theta_{AB})\theta_{CD}.$$

For collection $S$, we assume that we can only observe its *scrambled version*, which can be expressed as the mapping of the scramble function

$$\mathcal{T}(S) = \sum_{\alpha \in S} \alpha, \tag{2}$$

where the summation of two themes $\alpha$ and $\beta$ is defined as

$$\alpha + \beta = (n_1^\alpha + n_1^\beta, \ldots, n_p^\alpha + n_p^\beta).$$

We call $O = \mathcal{T}(S)$ the *observation* of a collection $S$. Considering that $O$ is also a multiset on $\mathcal{L}$, we can present it by a vector as well, i.e. $O = (n_1^O, \ldots, n_p^O)$. If a theme $\alpha$ and an observation $O$ satisfy $n_j^\alpha \leqslant n_j^O$ for all $1 \leqslant j \leqslant p$, we say that $\alpha$ is *covered* by $O$, which is denoted by $\alpha \subseteq O$.

In many applications, such as the MBA problem, we do not observe the theme partition of a collection and it is our main goal to infer the themes and likely partitions of an observation. For example, our observation of $S$ in Table 2 is $O = \mathcal{T}(S) = \{A, B, C, D\}$, which can in fact be derived from any of the following four possible collections:

$$S_1 = \{A, B, C, D\},$$

$$S_2 = \{A, B, CD\},$$

**Table 2.** Collection generating process of $S = \{A, B, CD\}$ in $\mathcal{M}_B$

| *Theme* $\alpha$ | $\theta_\alpha$ | *A collection* |
|---|---|---|
| $A$ | $\theta_A$ | 1 |
| $B$ | $\theta_B$ | 1 |
| $C$ | $\theta_C$ | 0 |
| $D$ | $\theta_D$ | 0 |
| $AB$ | $\theta_{AB}$ | 0 |
| $CD$ | $\theta_{CD}$ | 1 |

$$S_3 = \{C, D, AB\},$$

$$S_4 = \{AB, CD\}.$$

Thus, the probability of observing $O = \{A, B, C, D\}$ is

$$P(O|\mathcal{D}, \boldsymbol{\theta}) = \sum_{k=1}^{4} P(S_k|\mathcal{D}, \boldsymbol{\theta}).$$

We denote the TMD with the collection generating process $\mathcal{M}_B$ and the scramble function $\mathcal{T}$ as TDM($\mathcal{M}_B, \mathcal{T}$). Given a group of observations $\mathcal{O} = \{O_1, \ldots, O_n\}$ generated from the model, our goal is to discover the underlying dictionary $(\mathcal{D}, \boldsymbol{\theta})$. Theorem 1 guarantees the identifiability of TDM($\mathcal{M}_B, \mathcal{T}$). The proof can be found in Appendix A.

*Theorem 1.* Let $\mathcal{O}_\mathcal{D}$ be the set of all possible observations generated by TDM($\mathcal{M}_B, \mathcal{T}$) based on the dictionary $(\mathcal{D}, \boldsymbol{\theta})$, and $\mathcal{P}_{\boldsymbol{\theta}}$ be the corresponding probability distribution on $\mathcal{O}_\mathcal{D}$. If two dictionaries $(\mathcal{D}_1, \boldsymbol{\theta}_1)$ and $(\mathcal{D}_2, \boldsymbol{\theta}_2)$ lead to the same distribution on observations, i.e. $\mathcal{O}_{\mathcal{D}_1} = \mathcal{O}_{\mathcal{D}_2}$ and $\mathcal{P}_{\boldsymbol{\theta}_1} = \mathcal{P}_{\boldsymbol{\theta}_2}$, we have $\mathcal{D}_1 = \mathcal{D}_2$ and $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2$.

## 3. Statistical inference of the theme dictionary

### 3.1. Full Bayesian approach for theme discovery

Since the number of possible themes (which is of the order of $2^p$) is much larger than the number of observations in practice, we constrain our interests only to *proper themes*, which satisfy the following conditions:

(a) the number of items in the theme is bounded above,

$$L(\alpha) \triangleq \sum_{j=1}^{p} n_j^\alpha \leqslant \tau_L;$$

(b) the support of the theme is bounded below,

$$\phi(\alpha) \triangleq \frac{1}{n} \sum_{i=1}^{n} I(\alpha \subseteq O_i) \geqslant \tau_F.$$

In practice, the two thresholds $\tau_L$ and $\tau_F$ can be specified on the basis of both one's prior knowledge and computational concerns. We note that a too large $\tau_L$ or a too small $\tau_F$ may greatly increase the search space and significantly slow down the computation.

Let $\mathcal{D}_\mathrm{c} = \{\alpha_1, \alpha_2, \ldots, \alpha_N\}$ be the dictionary of all proper themes, called the *complete dictionary*, and let $\boldsymbol{\theta} = \{\theta_\alpha\}_{\alpha \in \mathcal{D}_\mathrm{c}}$ be the corresponding parameters ($\theta_\alpha \in [0, 1]$ for $\forall \alpha \in \mathcal{D}_\mathrm{c}$). In practice, $\mathcal{D}_\mathrm{c}$ can be efficiently generated by ARM. Since usually only a small portion of the themes in $\mathcal{D}_\mathrm{c}$ are needed to explain the set of observations $\mathcal{O}$ under the model TDM($\mathcal{M}_B, \mathcal{T}$), our goal is to discover this small set of themes, the actual dictionary.

Since the complete dictionary has a majority of the $\theta_\alpha = 0$, we give $\boldsymbol{\theta}$ a mixture prior distribution:

$$\pi(\boldsymbol{\theta}) = \prod_{\alpha \in \mathcal{D}_\mathrm{c}} \{(1 - q)\, \delta_0(\theta_\alpha) + q\, h(\theta_\alpha)\}, \tag{3}$$

where $\delta_0(x)$ is a point mass at 0, $h(x)$ is a given probability density on $(0, 1]$ and $q \in (0, 1)$ is the expected fraction of non-zero $\theta$s *a priori*. (In most cases, a natural choice for $h$ is the uniform distribution, i.e. $h(x) \equiv 1$.) The problem of theme discovery is equivalent to finding which $\theta_\alpha$ is non-zero. The posterior distribution of $\boldsymbol{\theta}$ is

$$f(\boldsymbol{\theta}|\mathcal{O}) \propto \prod_{i=1}^{n} P(O_i|\mathcal{D}_c, \boldsymbol{\theta})\,\pi(\boldsymbol{\theta}) = \prod_{i=1}^{n} \sum_{S \in \mathcal{S}_i} P(S|\mathcal{D}_c, \boldsymbol{\theta})\,\pi(\boldsymbol{\theta}), \tag{4}$$

where $\mathcal{S}_i$ denotes the set of collections that are compatible with observation $O_i$. The size of $\mathcal{S}_i$ is dependent on the dictionary size and can be astronomically large when the dictionary is complex and the number of items in $O_i$ is large.

It is possible to use Gibbs sampling (Geman and Geman, 1984) to draw samples from the above posterior distribution, although it can be inefficient when $N$ is large. Because

$$f(\theta_\alpha|\mathcal{O}, \boldsymbol{\theta}_{[-\alpha]}) \propto \prod_{i=1}^{n} \sum_{S \in \mathcal{S}_i} P(S|\mathcal{D}_c, \boldsymbol{\theta})\,\{(1-q)\,\delta_0(\theta_\alpha) + q\,h(\theta_\alpha)\}, \tag{5}$$

we have the following mixture density for the conditional distribution

$$f(\theta_\alpha|\mathcal{O}, \boldsymbol{\theta}_{[-\alpha]}) = (1-p_\alpha)\,\delta_0(\theta_\alpha) + p_\alpha\, g(\theta_\alpha|\mathcal{O}, \boldsymbol{\theta}_{[-\alpha]}), \tag{6}$$

where constant $p_\alpha$ and density $g(x|\mathcal{O}, \boldsymbol{\theta}_{[-\alpha]})$ are of the form

$$p_\alpha = 1 - \frac{(1-q)\prod_{i=1}^{n} r_{\alpha,i}}{(1-q)\prod_{i=1}^{n} r_{\alpha,i} + q\int_0^1 \prod_{i=1}^{n} \{r_{\alpha,i} + (1-2r_{\alpha,i})x\}\,h(x)\,\mathrm{d}x}, \tag{7}$$

$$g(x|\mathcal{O}, \boldsymbol{\theta}_{[-\alpha]}) \propto \prod_{i=1}^{n} \{r_{\alpha,i} + (1-2r_{\alpha,i})x\}\,h(x), \tag{8}$$

with $r_{\alpha,i} = A_{\alpha,i}/(A_{\alpha,i} + B_{\alpha,i})$ and

$$\begin{aligned} A_{\alpha,i} &= \frac{1}{1-\theta_\alpha} \sum_{S \in \mathcal{S}_i} P(S|O_i, \mathcal{D}, \boldsymbol{\theta})\, I(\alpha \notin S), \\ B_{\alpha,i} &= \frac{1}{\theta_\alpha} \sum_{S \in \mathcal{S}_i} P(S|O_i, \mathcal{D}, \boldsymbol{\theta})\, I(\alpha \in S). \end{aligned} \tag{9}$$

The $p_\alpha$ defined in equation (7), which is called the *activity rate* of $\alpha$, is a natural measurement of the importance of a theme: a higher $p_\alpha$ means that $\alpha$ has a higher posterior probability to be included in the dictionary and thus is more important. Given $q$, $\{r_{\alpha,i}\}_{i=1}^n$ and $h(x)$, the integration in equation (7) can be approximated numerically. Monte Carlo techniques such as rejection sampling can be used to draw samples from distribution (8). However, considering that the calculation of $\{r_{\alpha,i}\}_{i=1}^n$ involves computing expression (9), which can be very expensive even to approximate, this full Bayesian method is realistic only when the size of $\mathcal{D}_c$ is small. In the next subsection, we propose an approximation strategy for large data sets.

### 3.2. Top-down stepwise method for inferring the theme dictionary

Although it is desirable to employ the full Bayesian approach as described previously, in many real applications the required computation is too costly to be practical. Each systematic scan step of the Markov chain Monte Carlo procedure can be very time consuming when $\mathcal{D}_c$ contains a large number of themes, and the Markov chain Monte Carlo algorithm may need many iterations to converge. To cope with the difficulty, we propose the following top-down stepwise procedure to discover the dictionary.

*Step 1*: start with the complete dictionary $\mathcal{D}_c$.
*Step 2*: for the current dictionary $\mathcal{D}$, find the maximum likelihood estimator of the theme usage probabilities $\hat{\boldsymbol{\theta}} = \{\hat{\theta}_\alpha\}_{\alpha \in \mathcal{D}}$ from the observations $\mathcal{O}$ by using the EM algorithm.
*Step 3*: calculate the significance score $\psi(\alpha)$ for each theme $\alpha \in \mathcal{D}$, which is the logarithm of the likelihood ratio statistic between the current model $(\mathcal{D}, \hat{\boldsymbol{\theta}})$ and the alternative model $(\mathcal{D}, \hat{\boldsymbol{\theta}}_{[\alpha=0]})$:

$$\psi(\alpha) = \sum_{i=1}^{n} [\log\{P(O_i|\mathcal{D}, \hat{\boldsymbol{\theta}})\} - \log\{P(O_i|\mathcal{D}, \hat{\boldsymbol{\theta}}_{[\alpha=0]})], \tag{10}$$

where $\hat{\boldsymbol{\theta}}_{[\alpha=0]} = \{\hat{\theta}_\beta I(\beta \neq \alpha)\}_{\beta \in \mathcal{D}}$. If $\psi(\alpha) \leqslant \tau_S$, we call $\alpha$ an insignificant theme.
*Step 4*: prune the theme dictionary $\mathcal{D}$ by removing the insignificant themes from it.
*Step 5*: iterate steps 2–4 until no themes can be removed from $\mathcal{D}$. Rank the themes in $\mathcal{D}$ by the significance score decreasingly at the end.

In practice, the threshold $\tau_S$ can be determined empirically or based on model selection principles. For example, $\tau_S = \frac{1}{2}\log(n)$ on the basis of the Bayesian information criterion. In this paper, however, we set $\tau_S = \frac{3}{2}\log(n)$, as a large range of simulations suggest that this higher penalty leads to a better overall performance. It is also feasible to do a 'bottom-up' strategy, but it is computationally more demanding.

The EM algorithm for finding the maximum likelihood estimator $\hat{\boldsymbol{\theta}}$ for dictionary $\mathcal{D}$ proceeds as follows. Let $\mathcal{S}_i$ denote the set of all possible partitions of observation $O_i$ under dictionary $\mathcal{D}$. The conditional probability for a partition $S \in \mathcal{S}_i$ given observation $O_i$ and the current estimate $\boldsymbol{\theta}^{(r)}$ is

$$P(S|O_i, \mathcal{D}, \boldsymbol{\theta}^{(r)}) = \frac{P(S|\mathcal{D}, \boldsymbol{\theta}^{(r)})}{P(O_i|\mathcal{D}, \boldsymbol{\theta}^{(r)})} = \frac{P(S|\mathcal{D}, \boldsymbol{\theta}^{(r)})}{\sum_{S' \in \mathcal{S}_i} P(S'|\mathcal{D}, \boldsymbol{\theta}^{(r)})}. \tag{11}$$

Then, the $Q$-function of the EM algorithm, which is defined as the expectation of the complete-data log-likelihood given the observations $\mathcal{O} = \{O_1, \ldots, O_n\}$ and the current estimate $\boldsymbol{\theta}^{(r)}$, is

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)}) = E[l(\boldsymbol{\theta})|\mathcal{O}, \mathcal{D}, \boldsymbol{\theta}^{(r)}] = \sum_{i=1}^{n} \sum_{S \in \mathcal{S}_i} P(S|O_i, \mathcal{D}, \boldsymbol{\theta}^{(r)}) \log\{P(S|\mathcal{D}, \boldsymbol{\theta})\}.$$

At the M-step, by maximizing $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)})$ we obtain the updated estimate

$$\boldsymbol{\theta}_\alpha^{(r+1)} = \mathbf{M}_\alpha(\boldsymbol{\theta}^{(r)}) = \frac{1}{n} \sum_{i=1}^{n} f(\alpha|O_i, \mathcal{D}, \boldsymbol{\theta}^{(r)}) \qquad \forall \alpha \in \mathcal{D}, \tag{12}$$

where

$$f(\alpha|O_i, \mathcal{D}, \boldsymbol{\theta}^{(r)}) \triangleq \sum_{S \in \mathcal{S}_i} I(\alpha \in S) P(S|O_i, \mathcal{D}, \boldsymbol{\theta}^{(r)}) = E[I(\alpha \in S)|O_i, \mathcal{D}, \boldsymbol{\theta}^{(r)}]$$

represents the contribution of observation $O_i$ to theme $\alpha$. The summation in the denominator of equation (11) can be expensive to compute for a large sized $O_i$ with many items. In Section 4, we describe an efficient Monte Carlo approximate method.

The above algorithm can be further accelerated by standard EM acceleration techniques based on Newton or quasi-Newton methods (see Jamshidian and Jennrich (1993, 1997) for a comprehensive review), which make usage of the exact or approximated Jacobian matrix $\mathbf{J}$ of the parameter updating function (12). Note that the Jacobian matrix $\mathbf{J}$ can be organized into the form

$$\mathbf{J}_{\alpha\beta} \triangleq \frac{\partial \mathbf{M}_\alpha(\boldsymbol{\theta})}{\partial \theta_\beta} = \frac{\sum_{i=1}^{n} \{f(\alpha,\beta|O_i,\mathcal{D},\boldsymbol{\theta}) - f(\alpha|O_i,\mathcal{D},\boldsymbol{\theta})\, f(\beta|O_i,\mathcal{D},\boldsymbol{\theta})\}}{n\theta_\beta(1-\theta_\beta)}, \tag{13}$$

where

$$f(\alpha,\beta|O_i,\mathcal{D},\boldsymbol{\theta}) \triangleq \sum_{S\in\mathcal{S}_i} I(\alpha,\beta\in S)\, P(S|O_i,\mathcal{D},\boldsymbol{\theta}) = E[I(\alpha,\beta\in S)|O_i,\mathcal{D},\boldsymbol{\theta}] \qquad \forall \alpha,\beta\in\mathcal{D}.$$

### 3.3. *Inferring the construction of each 'basket'*

Given the dictionary $\mathcal{D}$ with the probability vector $\boldsymbol{\theta}$, the conditional probability $P(S|O_j,\mathcal{D},\boldsymbol{\theta})$ gives us information on likely ways to parse an observed basket, revealing how it was constructed. The partition with the highest posterior probability, or the smallest partition set with certain coverage (i.e. the optimal confidence interval), can be obtained. This kind of theme level information allows us to understand the observations better and to make decisions accordingly.

The following toy example shows the difference between our understanding of the observations at the item level *versus* that at the theme level. Suppose that we have two pairs of observations: pair A,

$$O_{a1} = ABCDEF,$$
$$O_{a2} = BCDEF,$$

and pair B,

$$O_{b1} = DEFG,$$
$$O_{b2} = ABDEF,$$

which were generated from the dictionary $\mathcal{D}$

| theme$_\alpha$ | $ABC$ | $CDE$ | $DEF$ | $BF$ | $A$ | $B$ | $C$ | $D$ | $E$ | $F$ | $G$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\theta_\alpha$ | 0.005 | 0.005 | 0.005 | 0.01 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 | 0.015 |

The best partitions of the four observations are respectively

$$S_{a1} = \{ABC, DEF\} \qquad P(S_{a1}|O_{a1},\mathcal{D},\boldsymbol{\theta}) \approx 0.969,$$
$$S_{a2} = \{BF, CDE\} \qquad P(S_{a2}|O_{a2},\mathcal{D},\boldsymbol{\theta}) \approx 0.956,$$
$$S_{b1} = \{DEF, G\} \qquad P(S_{b1}|O_{b1},\mathcal{D},\boldsymbol{\theta}) \approx 0.993,$$
$$S_{b2} = \{A, B, DEF\} \qquad P(S_{b2}|O_{b2},\mathcal{D},\boldsymbol{\theta}) \approx 0.993.$$

The two observations in pair $A$ are very similar at the item level but are quite different at the theme level. In contrast, the two observations in pair $B$ look quite different at the item level but their sharing of a common theme $DEF$ makes them highly related. In Section 6.1, we display a few most likely parses of several sentences in a Chinese novel, demonstrating that these parses are indeed grammatically sensible.

## 4. Approximation with a sequential rejection control sampler

### 4.1. *Sequential Monte Carlo sampling in the inference of the theme dictionary model*

A brute force calculation of either the quantity (9) in the Bayesian method or values of $f(\alpha|O_i,\mathcal{D},\boldsymbol{\theta})$ and $f(\alpha,\beta|O_i,\mathcal{D},\boldsymbol{\theta})$ in the stepwise method needs to enumerate exhaustively all possible par-

titions of observation $O_i$ under a given dictionary $(\mathcal{D}, \boldsymbol{\theta})$. It is computationally infeasible when $O_i$ contains many items and $\mathcal{D}$ is moderately large. However, considering that both $f(\alpha|O_i, \mathcal{D}, \boldsymbol{\theta})$ and $f(\alpha, \beta|O_i, \mathcal{D}, \boldsymbol{\theta})$ are conditional expectations, and that $r_{\alpha,i}$ can be reorganized as

$$r_{\alpha,i} = \frac{1}{1 + B_{\alpha,i}/A_{\alpha,i}}, \qquad \frac{B_{\alpha,i}}{A_{\alpha,i}} = \frac{1-\theta_\alpha}{\theta_\alpha} \frac{E[I(\alpha \in S)|O_i, \mathcal{D}, \boldsymbol{\theta}]}{1 - E[I(\alpha \in S)|O_i, \mathcal{D}, \boldsymbol{\theta}]},$$

we can approximate these terms via Monte Carlo methods. To achieve this, we need to draw partition $S$ from a distribution that should be reasonably close to the conditional distribution $P(S|O_i, \mathcal{D}, \boldsymbol{\theta})$. SMC sampling appears suitable for this task.

In a standard SMC framework (Liu and Chen, 1998), of which the particle filter is a special case, we need a set of 'growing' random vectors, $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_T$, so that $\mathbf{x}_{t+1} = (\mathbf{x}_t, x_{t+1})$, and a sequence of auxiliary distributions $\pi_t(\mathbf{x}_t)$ with the properties that $\mathbf{x}_T \equiv \mathbf{x}$, where $\mathbf{x}$ is the final random vector of interest, and $\pi_T(\mathbf{x}) = \pi(\mathbf{x})$, the target distribution. One should choose $\pi_t(\mathbf{x}_t)$ as close to the marginal distribution $\pi(\mathbf{x}_t)$ as possible. Then, given $\{w_t^{(j)}, \mathbf{x}_t^{(j)}\}_{j=1}^m$, a set of weighted 'particles' (i.e. Monte Carlo samples) at step $t$, a main goal of SMC procedures is to evolve to a set of weighted particles at time $t+1$, $\{w_{t+1}^{(j)}, \mathbf{x}_{t+1}^{(j)}\}_{j=1}^m$. For example, we can let $\mathbf{x}_{t+1}^{(j)} = (\mathbf{x}_t^{(j)}, x_{t+1}^{(j)})$ with $x_{t+1}^{(j)}$ drawn from a trial distribution $q(x_{t+1}|\mathbf{x}_t^{(j)})$, and update the weight as $w_{t+1}^{(j)} = w_t^{(j)} \pi_{t+1}(x_{t+1}^{(j)}|\mathbf{x}_t^{(j)})/q(x_{t+1}|\mathbf{x}_t^{(j)})$ for $j = 1, \ldots, m$.

We can reformulate the SMC procedure under the framework of a filtration of $\sigma$-fields: $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \ldots \subset \mathcal{F}_T$, of which the standard 'growing' random-vector setting is a special case. An intuitive analogue of the sequence of increasing $\sigma$-fields is a sequence of 'pictures' on the same object with increasingly higher resolutions. Suppose that we have a sequence of auxiliary probability measures (distributions) defined on the corresponding $\sigma$-fields: $\pi_0(\mathbf{x}), \pi_1(\mathbf{x}), \ldots, \pi_T(\mathbf{x})$. Although we use a common $\mathbf{x}$ to denote the random variable that is involved in different $\sigma$-fields, the variable takes values at different 'resolutions', and our goal is finally to generate samples from the highest resolution distribution $\pi_T(\mathbf{x})$. The particle filter (or general SMC method) can be applied to this framework, but there are two distinctive features that need special attention:

(a) one often cannot reach all possible configurations at level $\mathcal{F}_{t+1}$ from a particle at level $\mathcal{F}_t$ and
(b) a configuration at level $\mathcal{F}_{t+1}$ can be generated from different particles at level $\mathcal{F}_t$.

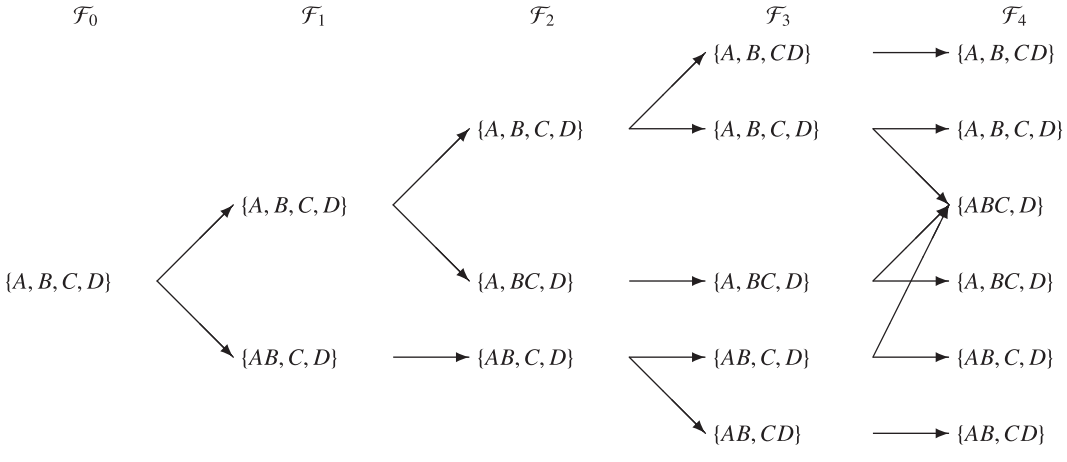
Thus, the weight updating rule for traditional SMC sampling needs to be modified to make the sampler proper and efficient.

To apply the generalized SMC method to our problem, we consider a sequence of dictionaries that can lead to the current dictionary $\mathcal{D}$. If a theme in dictionary $\mathcal{D}$ cannot be further decomposed into a combination of other themes in $\mathcal{D}$, we call it a *basic theme* of $\mathcal{D}$. Let $\mathcal{D}_0$ be the set of all basic themes of $\mathcal{D}$. In many cases, it is a natural choice to let $\mathcal{D}_0 = \mathcal{L}$, i.e. the collection of single-item sets. We can always construct a sequence of bridging dictionaries $\mathcal{D}_0 \subset \mathcal{D}_1 \subset \ldots \subset \mathcal{D}_T = \mathcal{D}$, such that $\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{\alpha_{t+1}\}$, and $T = \#\mathcal{D} - \#\mathcal{D}_0$, i.e. we can fill the gap between $\mathcal{D}_0$ and $\mathcal{D}$ by adding one theme at each step.

Without loss of generality, we consider one observation $O$. The filtration $\mathcal{F}_t$ can then be defined as all the partitions of the observation $O$ that are allowable by the current dictionary $\mathcal{D}_t$. Let $\boldsymbol{\theta}_t = \{\theta_\alpha\}_{\alpha \in \mathcal{D}_t}$ be the constrained version of $\boldsymbol{\theta}$ on $\mathcal{D}_t$; then, for any partition $S_t$ measurable in $\mathcal{F}_t$, we let

$$\pi_t(S_t) = P(S_t|O, \mathcal{D}_t, \boldsymbol{\theta}_t),$$

which defines a probability measure on $\mathcal{F}_t$. The target distribution $\pi$ can be formally written as $\pi(S) = P(S|O, \mathcal{D}_T, \boldsymbol{\theta}_T) = \pi_T(S)$. The expansion from $\mathcal{F}_t$ to $\mathcal{F}_{t+1}$ can be constructed via the




$$\mathcal{L} = \{A, B, C, D\},\ \mathcal{D} = \mathcal{L} \cup \{AB, BC, CD, ABC\},\ O = \{A, B, C, D\}$$

$$\mathcal{D}_0 = \mathcal{L},\ \mathcal{D}_1 = \mathcal{D}_0 \cup \{AB\},\ \mathcal{D}_2 = \mathcal{D}_1 \cup \{BC\},\ \mathcal{D}_3 = \mathcal{D}_2 \cup \{CD\},\ \mathcal{D}_4 = \mathcal{D}_3 \cup \{ABC\}$$

**Fig. 1.** Illustration of the evolution from $\mathcal{F}_0$ to $\mathcal{F}_4$: we call this process a particle division process

set (or particle in general) division process shown in Fig. 1. For example, because a new theme $\{ABC\}$ is added to $\mathcal{D}_3$ to form $\mathcal{D}_4$, a new partition $\{ABC, D\}$, which is not allowable in $\mathcal{F}_3$, is allowed in $\mathcal{F}_4$. An arrow is drawn from partition $S_t \in \mathcal{F}_t$ to partition $S_{t+1} \in \mathcal{F}_{t+1}$, indicating a parent–child relationship, if and only if either $S_{t+1} = S_t$ or $S_{t+1}$ includes the new theme $\alpha_{t+1}$ in the new dictionary $\mathcal{D}_{t+1}$ and $\alpha_{t+1}$ happens to be the summation of a few themes in partition $S_t$. This particle division can be easily proven to generate all possible partitions in $\mathcal{F}_{t+1}$ starting from all partitions in $\mathcal{F}_t$.

### 4.2. Sequential Monte Carlo sampler

The sequential Monte Carlo sampler (SMCS) that was proposed by Del Moral *et al.* (2006) fits our goal perfectly. Given the evolutionary structure from $\mathcal{F}_{t-1}$ to $\mathcal{F}_t$, many *Markov transition kernels* $K_t(x_{t-1}, x_t)$ can be employed for moving from $\mathcal{F}_{t-1}$ to $\mathcal{F}_t$. A natural choice in this case is

$$K_t(x_{t-1}, x_t) = \frac{\pi_t(x_t)}{\sum_{x'_t \in \mathcal{B}(x_{t-1})} \pi_t(x'_t)} I\{x_t \in \mathcal{B}(x_{t-1})\}, \qquad \forall (x_{t-1}, x_t) \in \mathcal{F}_{t-1} \times \mathcal{F}_t.$$

For a sequence of Markov transition kernels $\{K_t\}_{1 \leqslant t \leqslant T}$, we can introduce a sequence of *backward Markov kernels* $\{L_t\}_{0 \leqslant t \leqslant T-1}$, from which a sequence of auxiliary distributions $\{\tilde{\pi}_t(x_{0:t})\}$ can be constructed, where

$$\tilde{\pi}_t(x_{0:t}) = \pi_t(x_t) \prod_{k=0}^{t-1} L_k(x_{k+1}, x_k).$$

Since the dimension of $\tilde{\pi}_t$ increases over time, the 'standard' SMC framework, which was proposed in Liu and Chen (1998), can be used to draw weighted samples of $\tilde{\pi}_T$. Considering that $\tilde{\pi}_T(x_{0:T})$ admits $\pi_T(x_T)$ as a marginal distribution, weighted samples of $\pi_T$ can be obtained by marginalizing the weighted samples of $\tilde{\pi}_T$. The algorithm of Del Moral *et al.* (2006) is as follows.

(a) Assume that a group of $m$ particles $\{x_{1:t-1}^{(j)}, \omega_{1:t-1}^{(j)}\}_{1\leqslant j\leqslant m}$ have been obtained at step $t-1$.
(b) If the effective sample size is smaller than a threshold, resample the particles and set $\omega_{1:t-1}^{(j)} = 1/m$.
(c) For $j = 1, \ldots, m$, draw $x_t^{(j)} \sim K_t(x_{t-1}^{(j)}, \cdot)$, and assign to the new generated particle $x_{1:t}^{(j)} = (x_{1:t-1}^{(j)}, x_t^{(j)})$ the weight

$$\omega_{1:t}^{(j)} = \omega_{1:t-1}^{(j)}\, \tilde{\omega}_t(x_{t-1}^{(j)}, x_t^{(j)}), \qquad \tilde{\omega}_t(x_{t-1}, x_t) = \frac{\pi_t(x_t)\, L_{t-1}(x_t, x_{t-1})}{\pi_{t-1}(x_{t-1})\, K_t(x_{t-1}, x_t)}.$$

(d) After obtaining $m$ particles $\{x_{1:T}^{(j)}, \omega_{1:T}^{(j)}\}_{1\leqslant j\leqslant m}$ with respect to $\tilde{\pi}_T(x_{0:T})$, $\{x_T^{(j)}, \omega_{1:T}^{(j)}\}_{1\leqslant j\leqslant m}$ form a group of properly weighted samples of $\pi_T$.

The choice of backward Markov kernels $\{L_t\}_{0\leqslant t\leqslant T-1}$ has a great effect on the efficiency of the algorithm, and the *optimal backward Markov kernels* $\{L_t^{\text{opt}}\}_{0\leqslant t\leqslant T-1}$, which minimizes the variance of the unnormalized importance weight $\omega_n(x_{1:n})$, is given by

$$L_{t-1}^{\text{opt}}(x_t, x_{t-1}) = \frac{\eta_{t-1}(x_{t-1})\, K_t(x_{t-1}, x_t)}{\eta_t(x_t)},$$

where $\eta_0 = \pi_0$ in this case, and

$$\eta_t(x_t) = \eta_0\, K_{1:t}(x_t) \triangleq \int \eta_0(x_0) \prod_{k=1}^{t} K_k(x_{k-1}, x_k)\, \mathrm{d}x_{1:t}.$$

Considering that the computation that is involved in the optimal backward Markov kernels is usually prohibitive in practice, a few suboptimal backward kernels were also recommended in Del Moral *et al.* (2006), e.g.

$$L_{t-1}(x_t, x_{t-1}) = \frac{\pi_{t-1}(x_{t-1})\, K_t(x_{t-1}, x_t)}{\pi_{t-1} K_t(x_t)}, \tag{14}$$

or

$$L_{t-1}(x_t, x_{t-1}) = \frac{\pi_t(x_{t-1})\, K_t(x_{t-1}, x_t)}{\pi_t(x_t)}. \tag{15}$$

However, they cannot be used here directly because the support of $\pi_t$ increases exponentially with $t$. We use the following approximated version of kernel (14) in this paper:

$$L_{t-1}(x_t, x_{t-1}) = \frac{\pi_{t-1}(x_{t-1})\, K_t(x_{t-1}, x_t)}{\hat{\pi}_{t-1} K_t(x_t)}, \tag{16}$$

where $\hat{\pi}_{t-1} K_t(x_t) = \Sigma_{j=1}^m \omega_{t-1}^{(j)} K_t(x_{t-1}^{(j)}, x_t)$, and $\{x_{t-1}^{(j)}, \omega_{t-1}^{(j)}\}_{1\leqslant j\leqslant m}$ are the $m$ weighted samples of $\pi_{t-1}$ obtained in the sampling process. This kernel leads to the incremental weight

$$\tilde{\omega}_t(x_{t-1}, x_t) = \frac{\pi_t(x_t)\, L_{t-1}(x_t, x_{t-1})}{\pi_{t-1}(x_{t-1})\, K_t(x_{t-1}, x_t)} = \frac{\pi_t(x_t)}{\hat{\pi}_{t-1} K_t(x_t)}.$$

More details on related SMC methods can be found in Liu (2001), Doucet *et al.* (2001) and Del Moral (2004).

### *4.3. Sequential rejection control sampler*

Since the system that is studied here is discrete, it is preferable not to do independent sampling directly, but to explore all possibilities of the next step (i.e. *expansion*) and then to perform rejection control to reduce the sample size (i.e. *shrinkage*), as suggested in Fearnhead and Clifford (2003). Because a particle at level $\mathcal{F}_{t+1}$ can be generated from different particles at level $\mathcal{F}_t$, however, the Fearnhead–Clifford algorithm must be modified to make the sampler proper.

*4.3.1. Rejection control sampling*

Before we proceed to the sequential sampling set-up, we first review the rejection control sampling (RCS) procedure that was proposed in Liu *et al.* (1998). Let $\pi$ be defined on a discrete domain $\mathcal{F}$, and let $c > 0$ be a constant chosen in advance. Then

(a) $\forall x \in \mathcal{F}$, we define its *surviving probability* as $p_x = \min\{1, \pi(x)/c\}$ and compute its *importance weight* as $\omega_x = \pi(x)/p_x = \max\{\pi(x), c\}$;
(b) for each $x \in \mathcal{F}$, draw a binary variable $Z_x \sim \text{Bernoulli}(p_x)$, and denote $Z = \{Z_x\}_{x\in\mathcal{F}}$;
(c) weighted samples $\{x, \omega_x Z_x\}_{x\in\mathcal{F}}$ are called *rejection control samples*, on the basis of which the following estimator of $\mu = E_\pi[h(X)]$ can be constructed,

$$\hat{\mu}_Z = \sum_{x\in\mathcal{F}} h(x)\omega_x Z_x \Big/ \sum_{x\in\mathcal{F}} \omega_x Z_x.$$

In practice, the constant $c$ can be determined to maintain a fixed *mean sample size* (MSS) given *a priori*. For example, if we want the MSS to be $n_c$, we can solve $c$ via linear programming so that

$$n_c \triangleq \sum_{x\in\mathcal{F}} p_x = \sum_{x\in\mathcal{F}} \min\{1, \pi(x)/c\}. \tag{17}$$

Fearnhead and Clifford (2003) provided a fast algorithm for this computation. We note the following simple facts:

(a) $n_c$ is a monotonously non-increasing function of $c$ and
(b) $cn_c \leqslant 1$.

Fearnhead and Clifford (2003) showed that the discrete distribution that is represented by the $n_c$ Monte Carlo samples resulting from the RCS procedure is the 'optimal' representation of the original distribution $\pi$ under the total variation distance. The following theorem shows that the weighted samples generated by RCS enjoy a better statistical efficiency in estimation than independent identically distributed samples. (The proof can be found in Appendix B.)

*Theorem 2.* The estimator $\hat{\mu}_t$ based on RCS is asymptotically unbiased, and statistically more efficient than the sample mean of $n_c$ independent identically distributed samples from $\pi$, i.e.

$$E_Z[\hat{\mu}_Z] = \mu + O(c),$$

$$\text{MSE}(\hat{\mu}_Z) \leqslant \text{var}_\pi\{h(X)\}/n_c.$$

The numerical experiment shown in Fig. 2 illustrates the relative efficiency of RSC samples and independent identically distributed samples under different mean sample sizes.

*4.3.2. Sequential rejection control sampling*

In our current SMC setting, we assume that at step $t-1$ we have already obtained a good particle approximation of $\pi_{t-1}$, which is denoted $\tilde{\pi}_{t-1}$. Then, we can proceed with the following recursive RC procedure, which is similar in spirit to that proposed in Fearnhead and Clifford (2003) for standard SMC sampling with a discrete state space.

(a) Run RCS for $\tilde{\pi}_{t-1}$ with MSS $n_c$ to generate a vector of surviving indicators $Z^{(t-1)} = \{Z_x^{(t-1)}\}_{x\in\mathcal{F}_{t-1}}$, where the probability for $Z_x^{(t-1)} = 1$ is $r_x^{(t-1)} = \min\{1, \tilde{\pi}_{t-1}(x)/c_{t-1}\}$, with $\sum_{x\in\mathcal{F}_{t-1}} r_x^{(t-1)} = n_c$.

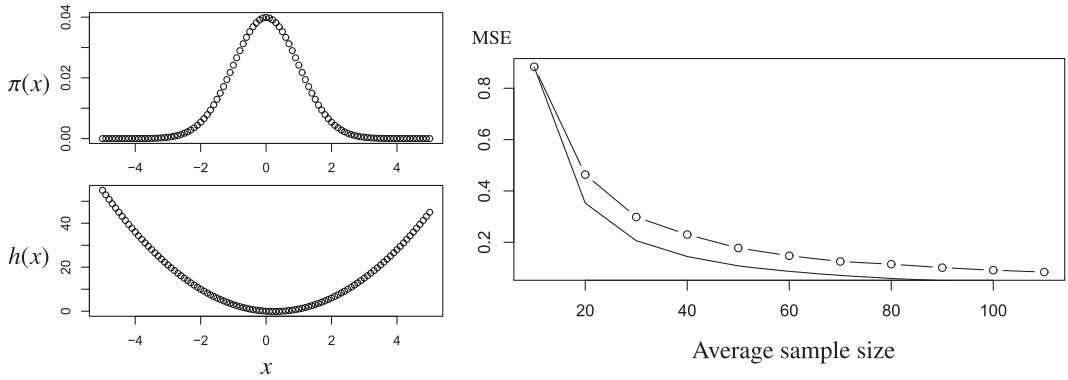



**Fig. 2.** Comparison of the statistical efficiency of RCS (———) with independent identically distributed sampling (○—○): $\mathcal{F} = \{-5, -4.9, \ldots, 4.9, 5\}$; $\pi(x) \propto \text{dnorm}(0, 1, x)$, $h(x) = 2x^2 - x$; the mean-square error MSE is calculated from 1000 independent runs

(b) For each $x \in \mathcal{F}_t$, we consider all its ancestors and define its *induced surviving indicator* as $\tilde{Z}_x^{(t)} = \vee_{x' \in \mathcal{P}_{t-1}(x)} Z_{x'}^{(t-1)}$. Thus, the probability for $\tilde{Z}_x^{(t)} = 1$ is

$$\tilde{p}_x^{(t)} = 1 - \prod_{x' \in \mathcal{P}_{t-1}(x)} (1 - r_{x'}^{(t-1)}), \tag{18}$$

and the surviving child's importance weight is $\tilde{\omega}_x^{(t)} = \pi_t(x)/\tilde{p}_x^{(t)}$. From $\tilde{\omega}^{(t)}$ and $\tilde{Z}^{(t)}$, we have the following particle approximation of $\pi_t$:

$$\tilde{\pi}_t(x) \triangleq \tilde{\omega}_x^{(t)} \tilde{Z}_x^{(t)} \Big/ \sum_{x'} \tilde{\omega}_{x'}^{(t)} \tilde{Z}_{x'}^{(t)}, \qquad \forall x \in \mathcal{F}_t.$$

This completes the recursion.

Considering that the support of $\tilde{\pi}_t$ is typically much larger than the prescribed MSS $n_c$, step (a) in the next recursion is necessary to control the Monte Carlo sample size. Compared with the procedure of Fearnhead and Clifford (2003), our SRCS gives additional consideration to those rejected particles. Theorem 3 below (which is a direct corollary of theorem 2) shows that SRCS enjoys a high statistical efficiency when the MSS $n_c$ is reasonably large.

*Theorem 3.* For a proper function $h(x)$ defined on $\mathcal{F}_t$, let

$$\hat{\mu}_{Z^{(t)}} = \sum_{x \in \mathcal{F}_t} h(x) \omega_x^{(t)} Z_x^{(t)} \Big/ \sum_{x \in \mathcal{F}_t} \omega_x^{(t)} Z_x^{(t)}.$$

If $\tilde{\pi}_{t-1} = \pi_{t-1}$, we have $E[\hat{\mu}_{Z^{(t)}}] = E_{\pi_t}[h(X)] + O(n_c^{-1})$, and

$$\lim_{n_c \to \infty} P\left[\text{MSE}(\hat{\mu}_{Z^{(t)}}) \leqslant \frac{\text{var}_{\pi_t}\{h(X)\}}{n_c}\right] = 1.$$

### *4.4. Evaluation of the sequential Monte Carlo methods*

To evaluate the performances of the SMC sampler of Del Moral *et al.* (2006) and the new SRCS method, we design the following numerical experiment. The filtration of $\sigma$-fields, $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \ldots \subset \mathcal{F}_{100}$, is shown in Fig. 3. A uniform distribution $\pi_t$ is defined in each $\sigma$-field $\mathcal{F}_t$, and the goal is to draw samples of the target uniform distribution $\pi = \pi_{100}$. Standard SIS without adjusting the

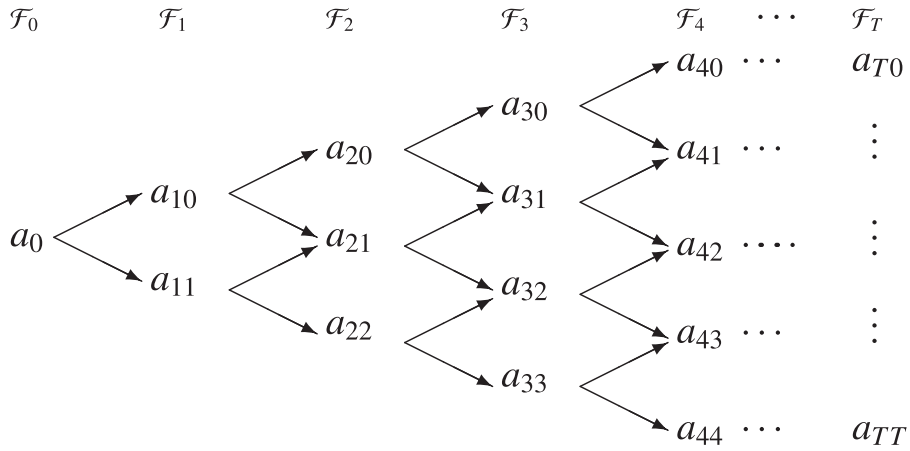



**Fig. 3.** $T$-layer lattice illustrating the evolution from $\mathcal{F}_0$ to $\mathcal{F}_1,\ldots,\mathcal{F}_T$: a uniform distribution $\pi_t(a_{tj}) = 1/(t+1)$ is assigned on $\mathcal{F}_t$, the $\sigma$-field on finite set $\{a_{t0}, a_{t1},\ldots,a_{tt}\}$; the goal is to draw samples from the target distribution $\pi = \pi_T$ via SMC sampling

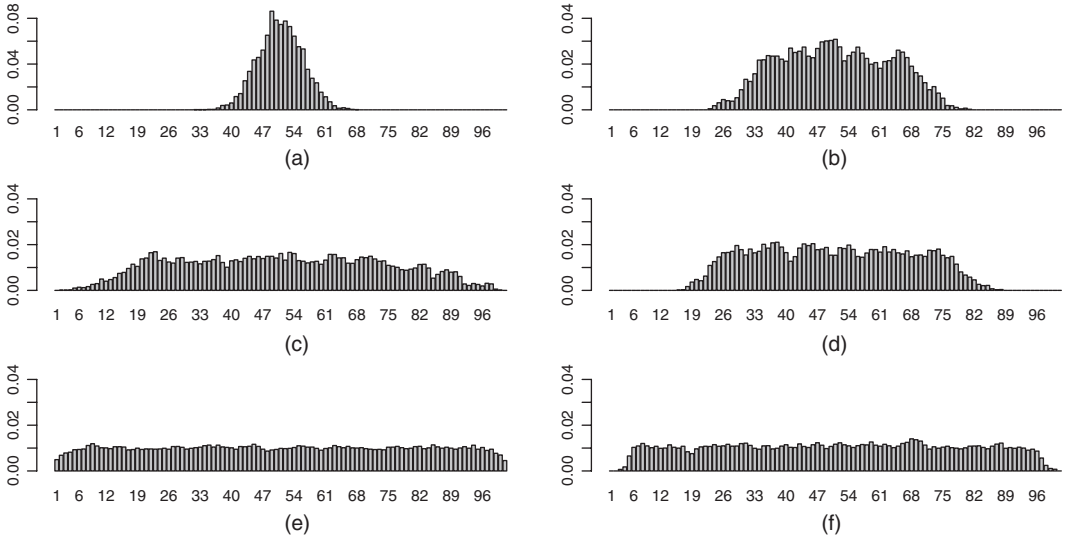



**Fig. 4.** Comparison of performance between various SMC methods (the histograms illustrate the estimations of target distribution $\pi_{100}$ based on the weighted samples from SMC methods; standard SIS leads to a serious bias, even when 5000 particles are used; SRCS and the SMCS can remove the bias significantly with much smaller Monte Carlo sample sizes, and SRCS is more efficient than SMCS): (a) standard SIS ($m = 5000$); (b) SMCS ($m = 50$); (c) SRCS ($m = 20$); (d) SMCS ($m = 100$); (e) SRCS ($m = 50$); (f) SMCS ($m = 500$)

multipath effect, the SMCS, and SRCS were applied to this example. For each method with the given sample size, we generated 100 groups of weighted samples in 100 independent runs, based on which the average weight for each element in the support of $\pi$ was calculated. The results are summarized in Fig. 4. It shows that a direct use of standard SIS resulted in a serious bias in estimation, and SRCS performed the best among the three. In Section 5, we further demonstrate the superior performance of SRCS for computations with the TDM.

## 5. Simulation study

In this study, the item set consists of 26 English letters, i.e. $\mathcal{L} = \{\text{A,B,C},\ldots,\text{Z}\}$. We create a theme dictionary $\mathcal{D}_{\text{true}}$ as shown in Fig. 5, which contains 50 themes. The 16 letters in italics were not included in the theme dictionary although they appear in observations as parts of other themes. Observations were generated from model TDM($\mathcal{M}_B, \mathcal{T}$) with probabilities indicated above each theme in Fig. 5. A typical set of the simulated data is illustrated in Fig. 6(a), and the size distribution of the observations that was generated by the model is displayed in Fig. 6(b).

### *5.1. Evaluation of sequential Monte Carlo approximation*

We applied both SRCS and the SMCS to approximate the sufficient statistics

$$f(\alpha|O,\mathcal{D}_{\text{true}}) = \sum_{S\in\mathcal{S}} I(\alpha\in S)\, P(S|O,\mathcal{D}_{\text{true}}) \qquad \forall\alpha\in\mathcal{D},$$

under the dictionary $\mathcal{D}_{\text{true}}$ for observation $O = \{\text{A,B},\ldots,\text{Z}\}$, which contains 26 items and has about 260 000 partitions under $\mathcal{D}_{\text{true}}$.

We conducted 100 independent replications of SRCS and the SMCS with Monte Carlo sample size $m = 200$. From each replication of each method, we obtained an approximation of the

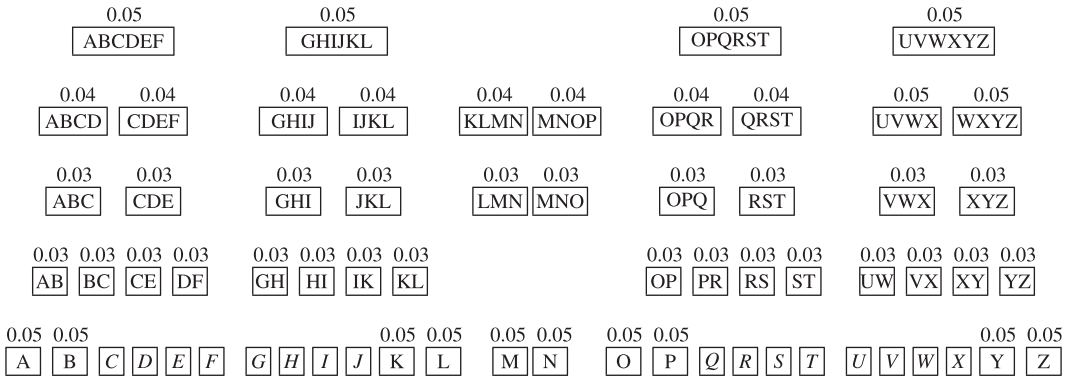



**Fig. 5.** True theme dictionary $\mathcal{D}_{\text{true}}$ underlying the simulation study: (theme type size, number of themes, probability) = (1,10,0.5), (2,16,0.03), (3,10,0.03), (4,10,0.04), (6,4,0.05)

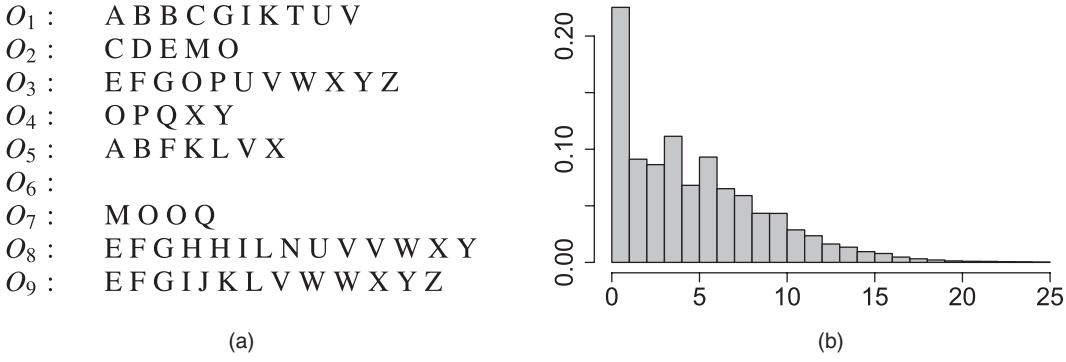



**Fig. 6.** Illustration of the data simulated from $\mathcal{D}_{\text{true}}$: (a) typical part of the simulated data; (b) length distribution of the simulated data

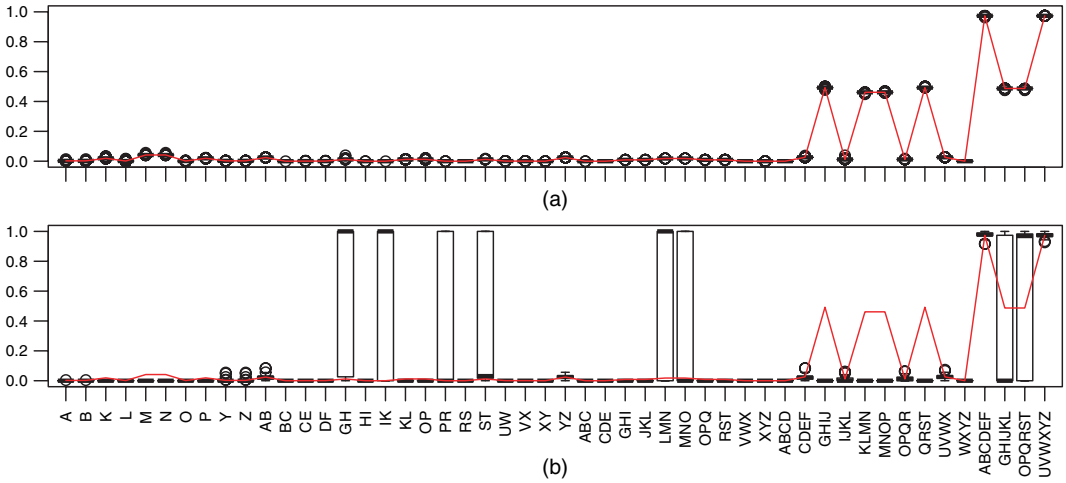



**Fig. 7.** Performances of (a) SRCS and (b) the SMCS with Monte Carlo sample size $m = 200$: each boxplot corresponds to the 100 replicated estimates for one theme in $\mathcal{D}_{\text{true}}$ (———, true values of the sufficient statistics)

sufficient statistics $f(\alpha|O, \mathcal{D}_{\text{true}})$ for each of the 50 themes in $\mathcal{D}_{\text{true}}$. These results are summarized in the boxplots in Fig. 7. The grey curves show the true values of these sufficient statistics, which were obtained by brute force enumeration. SRCS gave very accurate approximations and clearly outperformed the SMCS in this case. In the TDM analysis, our general strategy is to use Monte Carlo approximations by SRCS when the number of allowable partitions is beyond 1000, and to do brute force enumeration otherwise.

### 5.2. *Evaluation of the full Bayesian method*

We first generated a data set of 300 baskets according to the theme dictionary $\mathcal{D}_{\text{true}}$ that is depicted in Fig. 5. Applying ARM to this data set, we obtained a complete dictionary $\mathcal{D}_{\text{c}}$ with 806 candidate themes with thresholds $\tau_F = 0.03$ and $\tau_L = 8$.

From this data set, we found 806 theme candidates with thresholds $\tau_F = 0.03$ and $\tau_L = 8$. We constructed the complete dictionary with these theme candidates, i.e. $\mathcal{D}_{\text{c}} = (\alpha_1, \ldots, \alpha_N)$ where $N = 806$, and used the systematic scan Gibbs sampler to update the value of $\boldsymbol{\theta} = (p_{\alpha_1}, \ldots, p_{\alpha_N})$.

We used the non-informative prior $h \sim \text{unif}(0, 1]$, and the following initial value for parameter $\boldsymbol{\theta}$:

$$\theta_\alpha = \phi(\alpha)\, I\{L(\alpha) = 1\}, \qquad \alpha \in \mathcal{D}_{\text{c}}.$$

The hyperparameter $q$ in the prior distribution (3) is the prior expected fraction of 'active' themes (i.e. those with non-zero frequencies), and $Nq$ the expected number of active themes. We tried the following three values for $Nq$: 25, 50 and 100. In each case, the Markov chain converges after just a few iterations, with posterior number of active themes hovering between 50 and 60, which are very close to the true number of active themes. Fig. 8 displays Markov chain Monte Carlo trace plots for the number of active themes in 100 Gibbs steps under different hyperparameters. Using the posterior probability of 0.5 as the cut-off for declaring a theme active, the Bayesian method committed only one false negative error and zero false positive errors in all the prior settings that we tested. We conducted 10 independent replications of the Bayesian method under the same setting. At each time, we observed a similar pattern for an

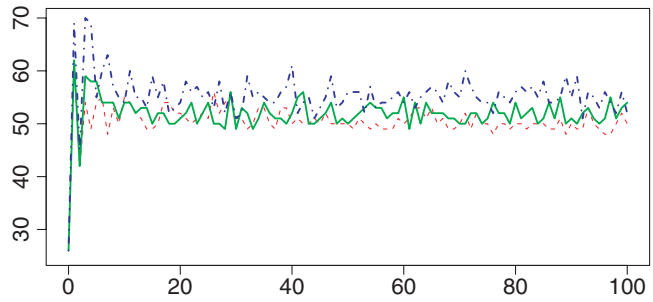



**Fig. 8.** Number of active themes in each of the 100 Gibbs steps: ·-·-·-·, $Nq = 100$; ———, $Nq = 50$; - - - - - -, $Nq = 25$

independently generated data set. On average, the Bayesian method took 22 min and achieved 99.3% sensitivity and 2% false positive rate.

### *5.3. Evaluation of the stepwise method*

We simulated 100 independent data sets of sample size 100, 200, 300 and 500 and applied the stepwise method to these simulated data sets. Thresholds $\tau_L = 8$ and $\tau_F = 0.03$ were the same as in the Bayesian approach. For a given sample size, the sensitivity and the false positive rate of the stepwise method were calculated from the 100 independent runs. The results are summarized in Table 3 and Fig. 9, from which we can see that the performance of the stepwise method improved substantially with increasing sample sizes. Compared with the full Bayesian

**Table 3.** Performance of the stepwise method for simulated data from $\mathcal{D}_{\text{true}}$

| *Sample size* | *Average sensitivity (%)* | *Average false positive rate (%)* |
|---|---|---|
| 100 | 68.8 | 14.6 |
| 200 | 93.3 | 1.0 |
| 300 | 98.4 | 0.6 |
| 500 | 99.8 | 0.4 |

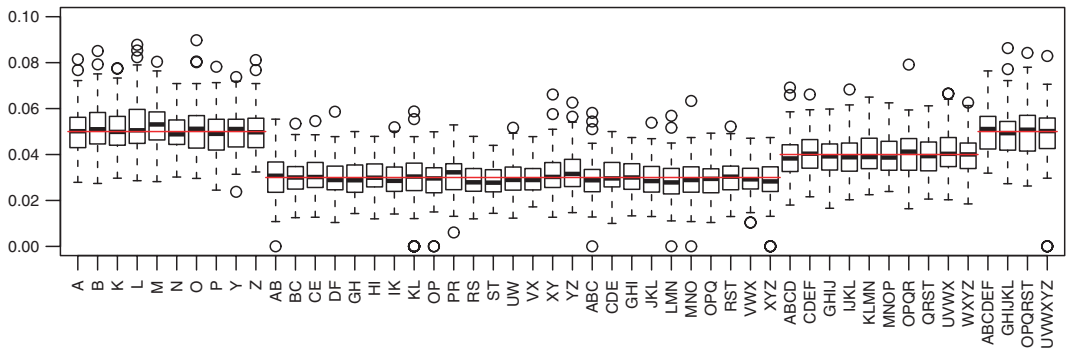



**Fig. 9.** Boxplots of the estimates from the 100 parallel runs when the sample size $n = 500$

method, the stepwise method appeared to have achieved a similar performance, but with a much shorter run time. For example, when the sample size was 300, the average running time for the stepwise method was 27 s, achieving 98.4% sensitivity and 0.6% false positive rate on average. The computational advantage of the stepwise method over the Bayesian method becomes even more significant as the size of the complete dictionary $\mathcal{D}_c$ becomes larger.

We also applied ARM to the simulated data to find frequent item sets under different minimal support thresholds. No matter which threshold was used, either the sensitivity or the false positive rate or both were much larger than for the TDM approach. The performance of ARM also was not improved when we increased the sample size.

## 6. Real data applications

### *6.1. Chinese text data mining*

The famous Chinese classic novel *Dream of the Red Chamber* (《红楼梦》) has had many millions of readers and inspired many Chinese literature researchers. It contains 4502 distinct Chinese characters and 108 296 sentences (i.e. observations). The average number of Chinese characters contained in each sentence is 6.72. Fig. 10 illustrates more details about the data. It is somewhat surprising that the number of distinct Chinese characters is so much smaller than the total number of distinctive English words that Shakespeare wrote in all his work (31534 distinct word types of 884647 total words published; see Efron and Thisted (1976)).

In this application, we treat each sentence in the novel as a basket and the Chinese characters in the sentence as items. Thus, we deliberately discard all the information regarding the ordering of the Chinese characters in each sentence, making it just an unordered basket of Chinese characters. We are interested in testing whether, by inferring co-occurrence of the Chinese characters, the TDM approach can recover some key names and phrases contained in these sentences as discovered 'themes'. We also hope that the TDM can even give us combinations of names of people (such as two lovers) and/or places that may reveal useful relationships.

We first applied ARM to the data set to generate the complete theme dictionary of all candidate themes. With thresholds $\tau_F = 0.0001$ and $\tau_L = 8$, ARM discovered 116 870 candidate themes, the majority of which cannot be transformed to any grammatically correct expressions, and often correspond to fragments of some frequently used Chinese phrases or sentences.

We then applied the stepwise method to fit the TDM for the data set starting from the complete theme dictionary obtained, which took about 4 h and came up with about 7315

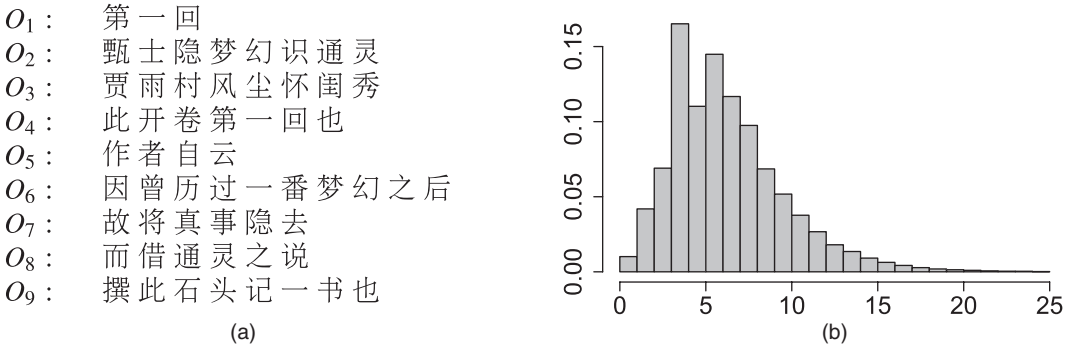



**Fig. 10.** Illustration of the text data from the Chinese novel *Dream of the Red Chamber*: (a) typical part of the Chinese text data; (b) length distribution of the Chinese text data

**Table 4.** Top 100 themes discovered by the TDM

| *Identifier* | *Theme* | *Meaning* | *Identifier* | *Theme* | *Meaning* | *Identifier* | *Theme* | *Meaning* | *Identifier* | *Theme* | *Meaning* | *Identifier* | *Theme* | *Meaning* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 宝玉 | Name | 21 | 方才 | Just now | 41 | 宝琴 | Name | 61 | 焙茗 | Name | 81 | 拌嘴 | Quarrel |
| 2 | 凤姐 | Name | 22 | 邢夫人 | Name | 42 | 雪雁 | Name | 62 | 阿弥陀佛 | Amitabha | 82 | 寂寞 | Lonely |
| 3 | 袭人 | Name | 23 | 里头 | Inside | 43 | 侄儿 | Nephew | 63 | 体面 | Dignity | 83 | 螃蟹 | Crab |
| 4 | 黛玉 | Name | 24 | 欢喜 | Happy | 44 | 司棋 | Name | 64 | 恍惚 | Trance | 84 | 嬷嬷 | Nanny |
| 5 | 王夫人 | Name | 25 | 媳妇 | Wife | 45 | 史湘云 | Name | 65 | 疑惑 | Puzzle | 85 | 翠墨 | Name |
| 6 | 宝钗 | Name | 26 | 东西 | Thing | 46 | 秦氏 | Name | 66 | 大观园 | Address | 86 | 标致 | Cute |
| 7 | 贾琏 | Name | 27 | 李纨 | Name | 47 | 衣服 | Cloth | 67 | 菩萨 | Bodhisattva | 87 | 唱戏 | Sing |
| 8 | 贾政 | Name | 28 | 贾蓉 | Name | 48 | 茗烟 | Name | 68 | 聪明 | Smart | 88 | 之母 | Someone’s mother |
| 9 | 姑娘 | Girl | 29 | 湘云 | Name | 49 | 林之孝 | Name | 69 | 辛苦 | Hard | 89 | 咬牙 | Grit one’s teeth |
| 10 | 什么 | What | 30 | 丫头 | Girl | 50 | 衣裳 | Cloth | 70 | 鲍二 | Name | 90 | 脂粉 | Cosmetics |
| 11 | 奶奶 | Grandma | 31 | 妹妹 | Younger sister | 51 | 规矩 | Rule | 71 | 嘱咐 | Tell | 91 | 陈设 | Furnishings |
| 12 | 出去 | Come out | 32 | 惜春 | Name | 52 | 孩子 | Kid | 72 | 二叔 | Name | 92 | 麒麟 | Kylin |
| 13 | 紫鹃 | Name | 33 | 芳官 | Name | 53 | 商议 | Discuss | 73 | 女孩儿 | Girl | 93 | 葫芦 | Gourd |
| 14 | 鸳鸯 | Name | 34 | 吩咐 | Enjoin | 54 | 妥当 | Properly | 74 | 仔细 | Be careful | 94 | 玫瑰 | Rose |
| 15 | 薛姨妈 | Name | 35 | 妙玉 | Name | 55 | 利害 | Interest | 75 | 伶俐 | Clever | 95 | 不迭 | Incessantly |
| 16 | 姐姐 | Elder sister | 36 | 金桂 | Name | 56 | 巧姐 | Name | 76 | 睁眼 | Eyes open | 96 | 十八 | 18 |
| 17 | 贾珍 | Name | 37 | 兄弟 | Brother | 57 | 琥珀 | Name | 77 | 蒋玉菡 | Name | 97 | 朦胧 | Cloudily |
| 18 | 薛蟠 | Name | 38 | 伏侍 | Serve | 58 | 师父 | Mentor | 78 | 讲究 | Be particular about | 98 | 燕窝 | Cubilose |
| 19 | 晴雯 | Name | 39 | 丫鬟 | Servant girl | 59 | 更比 | More than | 79 | 玻璃 | Glass | 99 | 惦记 | Miss |
| 20 | 香菱 | Name | 40 | 宝蟾 | Name | 60 | 周瑞 | Name | 80 | 调唆 | Instigate | 100 | 凤凰 | Phoenix |

**Table 5.** Subset of meaningful themes found by the TDM

| *Group I, combination of names* | *Group II, address* | *Group III, name* | *Group IV, Chinese phrase* |
|---|---|---|---|
| 王夫人, 邢夫人 | 晴雯, 麝月 | 青峰埂 | 宝二爷 | 冯紫英 | 宝玉 | 玫瑰 | 打嘴巴 | 阿弥陀佛 |
| 凤姐, 刘姥姥 | 贾母, 鸳鸯 | 潇湘馆 | 林玉黛 | 柳湘莲 | 晴雯 | 辉煌 | 打千儿 | 大惊小怪 |
| 邢王二夫人 | 贾珍, 贾琏 | 铁槛寺 | 薛宝钗 | 尤二姐 | 秦钟 | 玻璃 | 见世面 | 拿取笑儿 |
| 贾母, 薛姨妈 | 宝钗, 湘云 | 栊翠庵 | 王熙凤 | 琏二爷 | 鸳鸯 | 寂寞 | 吃果子 | 一宿无话 |
| 邢夫人, 尤氏 | 宝钗, 袭人 | 藕香榭 | 史湘云 | 甄士隐 | 紫鹃 | 凤凰 | 起诗社 | 满面泪痕 |
| 宝钗, 薛姨妈 | 贾母, 贾政 | 香稻村 | 贾雨村 | 甄宝玉 | 薛蝌 | 钥匙 | 茯苓霜 | 吓了一跳 |
| 凤姐, 王夫人 | 凤姐, 平儿 | 怡红院 | 刘姥姥 | 王一贴 | 探春 | 螃蟹 | 劳什子 | 说说话儿 |
| 宝钗, 王夫人 | 黛玉, 紫鹃 | 梨香院 | 林如海 | 林之孝 | 李纨 | 洗澡 | 老世翁 | 过来请安 |
| 宝玉, 林黛玉 | 秋纹, 麝月 | 水月庵 | 王夫人 | 王子腾 | 薛蟠 | 妯娌 | 小丫鬟 | 也不知道 |
| 贾政, 王夫人 | 宝钗, 黛玉 | 大观园 | 邢夫人 | 花自芳 | 宝蟾 | 芭蕉 | 两口子 | 不与相干 |
| 贾母, 王夫人 | 袭人, 麝月 | 蘅芜苑 | 老太太 | 邢岫烟 | 贾政 | 麒麟 | 儿媳妇 | 只得出来 |
| 贾珍, 贾蓉 | 紫鹃, 雪雁 | 沁芳亭 | 薛姨妈 | 尤三姐 | 代儒 | 菩萨 | 模样儿 | 放声大哭 |
| 贾兰, 贾环 | 宝玉, 宝钗 | 宁国府 | 薛大爷 | 李宫裁 | 宝钗 | 吩咐 | 银子钱 | 不好意思 |
| 贾赦, 贾政 | 李纨, 探春 | 荣国府 | 北静王 | 赖尚荣 | 迎春 | 规矩 | 不希罕 | 打发去了 |
| 贾珍, 尤氏 | 宝玉, 袭人 | 外书房 | 蒋玉菡 | 吴新登 | 茗烟 | 丫鬟 | 给请安 | 岂有之理 |

non-trivial themes (themes with more than one item). More than 90% of these themes have clear grammatical meanings, including about 400 names, 32 combinations of names and thousands of Chinese phrases. Table 4 lists the top 100 themes discovered by the TDM, which are composed of either people's names or well-known Chinese phrases. Another selected subset of meaningful themes that was discovered by the TDM are listed in Table 5. Since we estimated that the full Bayesian method will have to take more than 100 h to finish because of the large number of candidate themes, we did not apply it to this data set.

Since the 'truth' is difficult to define in this real data analysis, the false positive and false negative rates of the TDM and ARM are not easily determined and compared. To overcome this difficulty, we decided first to create a 'surrogate truth' by using a modified version of the *word dictionary model* (WDM) that was proposed by Bussemaker *et al.* (2000), which takes advantage of the ordering information of the Chinese characters in each sentence, to infer a list of common phrases (which are called 'words'). We then use this surrogate truth to evaluate the performance of the TDM and ARM.

Like the TDM, the WDM assumes the existence of a word dictionary, i.e. the phrase dictionary in the Chinese language, which also needs to be discovered from the analysis. But, unlike the TDM, the WDM assumes that each sentence is generated by an ordered concatenation of a sequence of words and phrases randomly selected from the dictionary. In Bussemaker *et al*. (2000), a stepwise method was suggested to learn the dictionary. Here we take a top-down approach:

(a) enumerate all existing strings in the text that satisfy the length and support constraints, $L(\alpha) \leqslant \tau_L = 10$ and $\phi(\alpha) \geqslant \tau_F = 0.0001$, and use them as word candidates;
(b) estimate the usage frequencies of the words, via the EM algorithm;
(c) rank the words on the basis of the importance score calculated in a way similar to that described in expression (10), and remove unimportant words.

A total of 7089 words were discovered by the WDM. By ignoring the order of Chinese characters in a word, we can convert a word into a theme. The 7089 words that were discovered by the WDM correspond to 6906 distinct themes, of which 4649 were also discovered by the TDM (the discovery rate is 67.3%). There are about 2700 themes that were reported by the TDM but not discovered by the WDM. Most of them are combinations of names or Chinese phrases with gaps and/or varying ordering, which cannot be captured by the WDM. To illustrate the

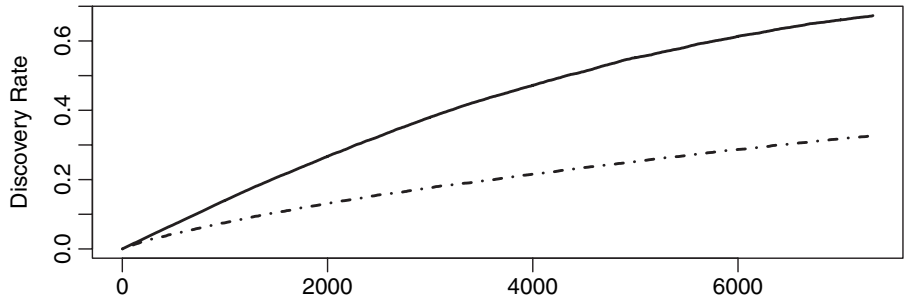



**Fig. 11.** Discovery rate of the top $K$ list from TDM (———) and ARM (·-·-·-)

**Table 6.** Three most probable parses for three sentences in the novel†

| | | | |
|---|---|---|---|
| $O_1$, 话 说 周 瑞 家 的 送 了 刘 姥 姥 去 后 | | | |
| 1 | 0.584322 | 0.584322 | 了 送(P)+说 话(M)+去 后(M)+ 姥 姥 刘(N)+ 的 家 周 瑞(N) |
| 2 | 0.252751 | 0.837072 | 了+后+说 话(M)+去 送+姥 姥 刘(N)+ 的 家 周 瑞(N) |
| 3 | 0.050561 | 0.887633 | 后+送+了 去+说 话(M)+姥 姥 刘(N)+ 的 家 周 瑞(N) |
| $O_2$, 夫 人 只 得 吩 咐 一 路 来 奔 荣 国 府 | | | |
| 1 | 0.761984 | 0.761984 | 一 路(P)+来 奔(M)+人 夫(N)+ 只 得(F)+咐 吩(M)+荣 府 国(A) |
| 2 | 0.149364 | 0.911348 | 一+路+来 奔(M)+人 夫(N)+只 得(F)+ 咐 吩(M)+ 荣 府 国(A) |
| 3 | 0.028287 | 0.939635 | 得+路+一 只+来 奔(M)+人 夫(N)+ 咐 吩(M)+ 荣 府 国(A) |
| $O_3$, 宝 玉 便 走 近 黛 玉 身 边 坐 下 | | | |
| 1 | 0.342864 | 0.342864 | 走+近+玉 宝(N)+下 坐(M)+身 边(P)+ 玉 便 黛(NM) |
| 2 | 0.325594 | 0.668458 | 便+走+近+下 坐(M)+身 边(P)+ 玉 玉 宝 黛(NN) |
| 3 | 0.206278 | 0.874736 | 便+走+近+玉 宝(N)+玉 黛(N)+ 下 坐(M)+身 边(P) |

†The second and third columns display the probabilities and cumulated probabilities of parses respectively. The English letters in parentheses highlight the type of corresponding themes: 'N' for names, 'M' for movements, 'A' for addresses, 'P' for phrases, 'NN' for name–name combinations and 'NM' for name–movement combinations.

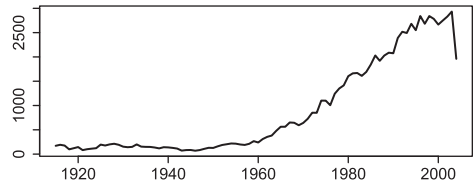
superiority of the TDM over ARM better, we calculated the discovery rate of the top $K$ list from the TDM as well as ARM for $1 \leqslant K \leqslant 7315$. (To obtain the top $K$ list from ARM, the theme candidates discovered by ARM were ranked by support decreasingly.) The results are displayed in Fig. 11, from which we can see that the discovery rate of the TDM is uniformly larger than ARM.

We also applied linear discriminant analysis (LDA), which is the fundamental method for topic modelling, to the data set. We downloaded the R-package for LDA from `http://www.cs.princeton.edu/blei/topicmodeling.html`, and ran the program under the default setting for 5000 iterations with different choices of topic number. In all cases, the topics reported do not have a clear meaning. The detailed results from ARM, the TDM, WDM and topic modelling can be found at `http://wileyonlinelibrary.com/journal/rss-datasets`, file 'Supplementary A'.

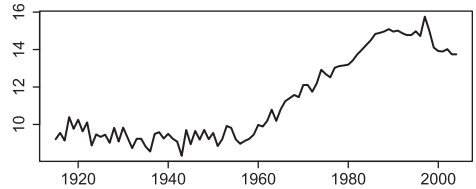
As discussed in Section 3.3, the inferred dictionary also helps us to parse the sentences into themes, which provide higher level information to the user. Table 6 shows the three most probable parses for each of the three sentences in the novel. These parses all correctly identify the basic sentence structures (main characters, verbs and places), even though no orderings of the words were provided to the algorithm.

$O_1$ : archaeology of Barbados
$O_2$ : a revision of the atomic weight of cadmium
$O_3$ : critique of the hypothesis of anomalous dispersion in certain solar phenomena
$O_4$ : discovery of the ninth satellite of Jupiter
$O_5$ : on the radial velocities of nebulae
$O_6$ : phoradendron
$O_7$ : preliminary note on nebular proper motions
$O_8$ : report on the autumn meeting
$O_9$ : spherical aberration in astronomical objectives due to changes of temperature
$O_{10}$ : the phi subgroup of a group of finite order

(a)




(b)




(c)

**Fig. 12.** Illustration of the *Proceedings of the National Academy of Sciences of the USA* paper title data set: (a) typical subset of the paper titles; (b) number of papers in each year; (c) average length of title in each year

### *6.2. Discover themes in journal paper title database*

We next applied the TDM to a data set containing titles of about 80 000 papers published in the *Proceedings of the National Academy of Sciences of the USA* from 1915 to 2005 (the data can be freely downloaded from `http://cs.nyu.edu/~roweis/data/pnas_all.tar`). Fig. 12 shows the number of papers published in the journal and the average title length in each year. The data for the last period are not complete, which is why the number of publications in 2004 shows a dip. A typical subset of these titles is illustrated in Fig. 12. To parse these title sentences 'intelligently', the first task is to recognize what the key scientific phrases are. We thus treat the English words in these title sentences as items, and each paper title as an observed basket. Our goal is to find association patterns between the English words used in these titles, which may reveal hot topics and possible research trends. Because research topics change over time, we are interested in knowing how different hot topics arose and faded over the years. We partitioned the title database into several groups according to time and inferred a theme dictionary for each data group. More specifically, we divided the database into nine groups according to nine time periods: 1915–1949; 1950–1969; 1970–1974; 1975–1979; 1980–1984; 1985–1989; 1990–1994; 1995–1999; 2000–2004. The number of papers in each group varies from 4000 to 14 000.

Note that the motivation of this attempt is very similar to that of Blei and Lafferty (2006), whose goal was to capture the changes of topics over time. However, differently from their dynamic topic model, which chains the parameters of different time spots in a state space model that evolves with Gaussian noise, we simply treat them as independent parameters and deal with the data sets from different time periods separately. It is feasible to adapt their dynamic framework to our theme model.

With the theme thresholds $\tau_L = 8$ and $\tau_F = 0.003$, we fitted the TDM for each of the nine data groups. The Bayesian method took 10–200 min for each of these data sets. In contrast, the stepwise method took only 4–400 s for the same data sets and gave similar results. Table 7 lists the top 10 two-item themes and top five multi-item themes discovered by the stepwise method for each period. Themes are sorted by the estimated parameter $\hat{\theta}_\alpha$ decreasingly. From Table 7, we observe how the research topics in the *Proceedings of the National Academy of Sciences*

**Table 7.** Hot terms in the *Proceedings of the National Academy of Sciences of the USA* from 1915 to 2004

| | *Results for the following years:* | | |
|---|---|---|---|
| | *1915–1949* | *1959–1969* | *1970–1974* |
| 1 | X-ray | *Escherichia coli* | *Escherichia coli* |
| 2 | Subgroup group | *e coli* | RNA messenger |
| 3 | *Drosophila melanogaster* | Protein synthesis | RNA polymerase |
| 4 | Ultra violet | Ribonucleic acid | RNA transfer |
| 5 | Differential equation | Amino acid | DNA polymerase |
| 6 | House mouse | Nucleic acid | Simian virus |
| 7 | Crossing over | *Drosophila melanogaster* | DNA replication |
| 8 | Effect upon | Deoxyribonucleic acid | Amino acid |
| 9 | Quantum theory | X-ray | Protein binding |
| 10 | Star magnitude | *Neurospora crassa* | DNA synthesis |
| 1 | Extragalactic galactic study | Tobacco mosaic virus | Cyclic adenosine monophosphate |
| 2 | Induced X-ray | Free cell system | DNA dependent RNA polymerase |
| 3 | National Academy Science | Amino acid sequence | Amino acid sequence |
| 4 | Linear differential equation | Nuclear magnetic resonance | RNA tumor viruses |
| 5 | X-ray effect | *Escherichia coli* mutant | Messenger RNA translation |
| | *1975–1979* | *1980–1984* | *1985–1989* |
| 1 | *Escherichia coli* | *Escherichia coli* | *Escherichia coli* |
| 2 | Virus simian | Gene expression | Gene expression |
| 3 | Adenylate cyclase | DNA sequence | Protein binding |
| 4 | Messenger RNA | t cell | t cell |
| 5 | Cyclic amp | Monoclonal antibodies | Cell line |
| 6 | Rat liver | Nucleotide sequence | DNA sequence |
| 7 | Cell surface | Simian virus | Monoclonal antibody |
| 8 | t cell | Rat liver | Binding site |
| 9 | Protein binding | Protein binding | Amino acid |
| 10 | Acetylcholine receptor | Cell line | Nucleotide sequence |
| 1 | Amino acid sequence | Epstein Barr virus | Human immunodeficiency virus |
| 2 | Rous sarcoma virus | t cell antigen | Protein kinase c |
| 3 | Nerve growth factor | *Escherichia coli* protein | Amino acid sequence |
| 4 | Chinese hamster cell | Low density lipoprotein | Platelet derived growth factor |
| 5 | Nuclear magnetic resonance study | Protein dependent kinase | Protein dependent kinase |
| | *1990–1994* | *1995–1999* | *2000–2004* |
| 1 | *Escherichia coli* | *Escherichia coli* | Gene expression |
| 2 | Gene expression | Gene expression | *Escherichia coli* |
| 3 | Protein binding | Protein binding | Protein binding |
| 4 | Transcription factor | t cell | Crystal structure |
| 5 | Amino acid | Transcription factor | Inaugural article |
| 6 | t cell | Transgenic mice | Nitric oxide |
| 7 | *Saccharomyces cerevisiae* | Growth factor | Stem cell |
| 8 | Transgenic mice | Not but | Transcription factor |
| 9 | Molecular cloning | Deficient mice | t cell |
| 10 | Binding site | *Saccharomyces cerevisiae* | Growth factor |
| 1 | Human immunodeficiency virus type | cd t cell | cd t cell |
| 2 | Human immunodeficiency virus | Mitogen activated protein kinase | Supramolecular chemistry self-assembly special feature |
| 3 | Protein kinase c | Nitric oxide synthase | nf kappa b |
| 4 | Tumor necrosis factor | Tumor necrosis factor | Asymmetric catalysis special feature part |
| 5 | Transforming growth factor szlig | Major histocompatibility complex class | Bioinorganic chemistry special feature |

*of the USA* were gradually dominated by biological sciences. In the first period (1915–1949), top themes included *drosophila melanogaster*, *crossing over* and *house mouse* (biology), *X-ray*, *quantum theory* and *ultra violet* (physics and chemistry), *group theory* and *differential equation* (mathematics), and *star magnitude* and *galactic extragalactic study* (astronomy). These reflect a balanced representation of science and mathematics. During 1950–1969, the dominating themes were mainly biological research topics and models, with only the exception of the theme *nuclear magnetic resonance*, which is also closely related to biomedical research. In biology, studies on proteins (its synthesis and composition) and nucleic acids (deoxyribonucleic acid (DNA) and ribonucleic acid (RNA)) appear to be balanced. Note that the first correct double-helix model of DNA was only proposed by James Watson and Francis Crick in 1953 on the basis of the single X-ray diffraction image data taken by Rosalind Franklin and Raymond Gosling. This important discovery establishes DNA as the key information unit and inheritance material of living beings.

Starting from the 1950s, bacteria *Escherichia coli* was solidly established as an important model organism for biological studies, and we can see that this theme persists in all the following periods. In contrast, the theme nuclear magnetic resonance quickly faded after the mid-1970s, indicating its maturity as a scientific topic in the 20-year period. The studies of *gene expression*, *protein (dependent) kinase*, *growth factor* and *t cell (antigens, receptors)* started to pick up the pace in the early 1980s and have lasted till nowadays. *Complementary DNA cloning* and other cloning techniques were hotly studied in late 1980s, paving the way for the development of complementary DNA microarray technologies in the 1990s. The studies of *protein binding* and *transcription factors* became very active since late 1980s and have lasted till now, partly because of the development of high throughput technologies for investigating gene expressions.

The study of *human immunodeficiency virus* clearly began in the late 1980s and remained on the hot topic list till 2000. Researches on *stem cells* started to show up in the last period, in the years from 2000 to 2004, indicating that it is a rising and exciting new direction in biological research. It is indeed true that, since the early 2000s, much effort and private funding have been directed to stem cell researches and much progress has been made.

To have a better understanding about the history of one particular research topic $\alpha$, we can plot its 'life curve' on the basis of its usage frequency at different periods of time. Fig. 13 shows

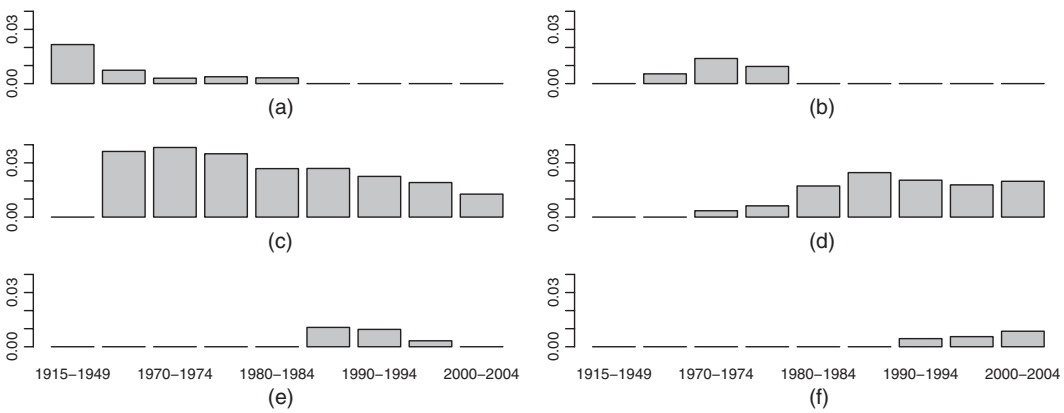



**Fig. 13.** 'Life curves' of a few research topics: (a) X-rays; (b) messenger RNA; (c) *Escherichia coli*; (d) gene expression; (e) human immunodeficiency virus; (f) stem cells

the life curves of a few hot research topics in different time periods. We also applied LDA to these data sets. For periods before 1980, LDA reported topics corresponding to mathematics, physics and chemistry as well as biology. For later periods, however, all topics reported are composed of biological terminologies. Compared with the results from the TDM, LDA's topics are made up of more, but loosely connected, terminologies. It also requires a prior specification of the topic number. The detailed results from the TDM and LDA can be found at `http://wileyonlinelibrary.com/journal/rss-datasets`, file 'Supplementary B'.

### *6.3. Analysis of the Netflix data*

The last example is based on the well-known 'Netflix prize' data set provided by Netflix Inc., which is an American company providing an on-line movie rental service. The Netflix prize was an open competition for the best prediction of unobserved customer ratings for films, based only on a training set of previous movie ratings from a small random subset of movie renters. We focus on the training data set in this study, which contains 100 480 507 ratings for 17 770 movies (and television series) provided by 480 189 customers. This can be viewed as a huge matrix with its $(i, j)$th entry taking values in $\{?, 1, 2, \ldots, 5\}$ representing the $i$th customer's rating for the $j$th movie, where '?' indicates that the rating of the movie is not available from the customer. Different from the original goal of the Netflix competition, we are interested in discovering sets of movies (modules in our TDM) from the data, which may provide a better framework for recommending movies. For example, each movie module may reveal some common features among the movies in the module so that a particular set of customers may wish to watch them all.

**Table 8.** Top themes found by the TDM before and after movies in one movie series are collapsed

| *Identifier* | *Top themes before collapse* | *Top themes after collapse* |
|---|---|---|
| 1 | *Kill Bill*: 1–2 | *Friends*; *The Best of Friends* |
| 2 | *American Pie*: 1–2 | *Snatch*; *Lock, Stock and Two Smoking Barrels* |
| 3 | *Men in Black*: 1–2 | *Indiana Jones and the Last Crusade*; *Raiders of the Lost Ark* |
| 4 | *Shrek*: 1–2 | *The Godfather*; *GoodFellas* |
| 5 | *Spider-Man*: 1–2 | *Bowling for Columbine*; *Fahrenheit 9/11* |
| 6 | *Happy Gilmore*; *Billy Madison* | *Sex and the City*; *Friends*; *The Best of Friends* |
| 7 | *Before Sunrise*; *Before Sunset* | *Lord of the Rings*; *The Matrix*; *Star Wars* |
| 8 | *Star Wars*: 4–6 | *The Royal Tenenbaums*; *Rushmore*; *The Big Lebowski* |
| 9 | *24*: 1–3 | *The Sixth Sense*; *The Shawshank Redemption*; *The Green Mile* |
| 10 | *Lord of the Rings*: 1–3 | *Amelie*; *Lost in Translation*; *Being John Malkovich* |
| 11 | *Harry Potter*: 1–3 | *The Green Mile*; *The Negotiator*; *A Few Good Men*; *A Time to Kill* |
| 12 | *Alias*: 1–3 | *The Fugitive*; *Air Force One*; *Clear and Present Danger*; *Patriot Games* |
| 13 | *The Matrix*: 1–3 | *Reservoir Dogs*; *The Godfather*; *Pulp Fiction*; *GoodFellas* |
| 14 | *Six Feet Under*: 1–3 | *Forrest Gump*; *The Green Mile*; *Saving Private Ryan*; *Gladiator* |
| 15 | *Lethal Weapon*: 1–4 | *Monsters, Inc.*; *Finding Nemo*; *Toy Story*; *Aladdin*; *The Lion King* |
| 16 | *CSI*: 1–4 | *The Rock*; *Gone in 60 Seconds*; *Entrapment*; *Swordfish*; *Con Air* |
| 17 | *The Best of Friends*: 1–4 | *The Bone Collector*; *High Crimes*; *Kiss the Girls*; *Along Came a Spider*; *Double Jeopardy* |
| 18 | *The Sopranos*: 1–5 | *The Green Mile*; *The Fugitive*; *Air Force One*; *Clear and Present Danger*; *Patriot Games*; *Ransom* |
| 19 | *Lord of the Rings*: 1–3 and 1–3 extended version | *Monsters, Inc.*; *Finding Nemo*; *Lord of the Rings*; *The Incredibles*; *Star Wars*; *Toy Story* |
| 20 | *Sex and the City*: 1–7 | *Maid in Manhattan*; *Pretty Woman*; *Sweet Home Alabama*; *Runaway Bride*; *How to Lose a Guy in 10 Days*; *Two Weeks Notice* |

We first filtered out non-popular movies that have been rated by fewer than 2% of the customers. Then, we dichotomize the observed ratings so that each movie will only be 'liked' or 'disliked or unrated' by each customer. More precisely, we constructed a sparse binary matrix with 480 189 rows or customers and 2042 columns or movies, where the $(i, j)$th entry equals 1 if customer $i$ rated movie $j$ as 4 or 5, and 0 if customer $i$ rated $j$ lower or never rated it.

Considering that it is computationally expensive to include all the 480 189 customers in our analysis, we randomly sampled 10 000 customers who have rated no more than 30 movies. With threshold $\tau_L = 8$ and $\tau_F = 0.001$, we fitted a TDM for this data set by the stepwise method, resulting in a theme dictionary with about 1400 non-trivial themes of movies. Most of the themes discovered have quite appealing meanings, such as those belonging to a movie series (e.g. the *Harry Potter* series and *Star Wars* series), movies of a similar type (e.g. action or love stories) and movies by the same actor, actress or director.

Since sets of movies belonging to a movie series are too obvious and thus less interesting, we collapsed the movies in a movie series into one single item with the following rule: if a customer likes any one movie from a movie series, we say that the customer likes the movie series. We refitted a TDM for the collapsed data set with the same setting and discovered more than 1400 non-trivial themes of movies and movie series. A subset of top themes discovered from the two data sets is listed in Table 8; the full theme lists can be found at `http://wileyonlinelibrary.com/journal/rss-datasets`, file 'Supplementary C'.

On the basis of the module results from our fitted TDM, we can formulate a new movie recommendation system for each user. Given customer $c$, for example, who has watched the set of movies $O_c$, we list all the modules that overlap with $O_c$, i.e. $\mathcal{D}_c = \{\alpha \in \mathcal{D} : \alpha \cap O_c \neq \emptyset\}$. Then, the union of all the movies in $\mathcal{D}_c$ excluding those that have already been watched form the recommendation list for the customer, i.e. $R_c = \{\cup_{\alpha \in \mathcal{D}_c} \alpha\} \setminus O_c$. The ranking of each movie $M$ in the recommendation list $R_c$ can be constructed either as the cumulative sum of frequencies (i.e. $\theta_\alpha$) of the modules in $\mathcal{D}_c$ that include $M$, or simply the number of modules in $\mathcal{D}_c$ that include $M$.

Another application of the TDM in discovering herbal functional groups of traditional Chinese medicine can be found in He *et al.* (2012).

## 7. Discussion

We propose a novel stochastic model, the TDM, to aid in the discovery of association patterns, named themes, among a large set of binary variables (indicating the presence or absence of an item) with sparse observations. A typical example of this kind of analysis is MBA, in which one wishes to infer item associations by analysing transactions of items purchased by the customers. Compared with other methods in the literatures such as ARM, the new approaches based on the TDM allow the association patterns to be composed of many variables, of which each has a very weak signal. It also allows the association pattern to overlap. Our simulation studies as well as real data applications show that the new methods are much more sensitive and specific than ARM, albeit at the cost of a much higher computational need. A practical and attractive strategy as demonstrated in our studies is first to use an ARM-type approach to obtain a set of candidate themes and then to use the TDM to thin down fragmental and redundant themes.

Another attractive feature of the TDM approach is its ease of incorporating the knowledge of field experts. If some patterns are known by field experts, we can simply include them in the dictionary *a priori*; however, if some patterns are known to be impossible or meaningless, we can put them in a blacklist for avoidance.

The new approach can be applied to a wide range of applications. It is particularly interesting as a text mining tool as it can provide a 'high level' understanding of sentences or paragraphs,

revealing relationships between different entities. By discovering themes and parsing each observation into the most likely combination of themes, it can provide a basis for understanding similarities and differences between observations, thus enabling us to do observation clustering and feature selection for statistical learning at the theme level. Since the existence of a theme automatically implies interactions between the items that are included in the theme, we can also use a TDM to help to identify interactive variables to build better predictive models.

Model $\mathcal{M}_B$ shown in equation (1) and studied in the previous sections is not the only choice for generating the sets of observations. The following *random-selection model*, denoted $\mathcal{M}_{\mathrm{R}}$, can be a good alternative:

$$P(S|\mathcal{D},\boldsymbol{\theta}) = \left(\prod_{\alpha\in S}\theta_\alpha\right)\theta_\tau, \qquad \theta_\alpha,\theta_\tau\in(0,1), \quad \theta_\tau+\sum_{\alpha\in\mathcal{D}}\theta_\alpha = 1. \tag{19}$$

In words, $\mathcal{M}_{\mathrm{R}}$ postulates that the items in a collection $S$ are generated by an imaginary monkey who draws the themes independently (with replacement) from an imaginary box containing all the themes until a special stopping symbol $\tau$ is drawn. The probability of obtaining theme $\alpha$ at each draw is $\theta_\alpha$. The theme discovery under model TDM($\mathcal{M}_{\mathrm{R}},\mathcal{T}$) is almost the same as that under TDM($\mathcal{M}_B,\mathcal{T}$). The identifiability of model TDM($\mathcal{M}_{\mathrm{R}},\mathcal{T}$) can be proved in a similar way to theorem 1. The theme discovery methods that were proposed in Section 3 for model TDM($\mathcal{M}_B,\mathcal{T}$) can be applied directly to model TDM($\mathcal{M}_{\mathrm{R}},\mathcal{T}$).

It is possible to explore more complex and 'intelligent' models based on the TDM framework. For example, it may be desirable to capture some aspects of the grammatical rules of the natural language for selecting the themes to build up a collection. We may assume that the themes fall into a few large clusters (e.g. clusters of *names*, *addresses* and *phases for movements*). These clusters may need to be learned separately on the basis of other information such as experts' inputs. Then, the generation of a 'collection' may prefer a certain order of the theme clusters (e.g. a person's name followed by a movement phase and then by a place name), which can be modelled as a hidden Markov chain. We may also be able to consider models such as context-free grammar or some hierarchical structure between the themes.

## Acknowledgements

This research was supported in part by National Science Foundation grants DMS-0706989, DMS-1007762 and DMS-1208771, NSFC grants 11171365 and 10931002, and Shenzhen Special Fund for Strategic Emerging Industry grant ZD201111080127A.

## Appendix A

### *A.1. Proof of theorem 1*

Let $O_\emptyset = \emptyset$ be the empty observation which contains no items. For $i=1$ or $i=2$, define

$$\lambda_i \triangleq \mathcal{P}_{\boldsymbol{\theta}_i}(O_\emptyset) = \prod_{\alpha\in\mathcal{D}_i}(1-\theta_{i,\alpha}).$$

For each $\alpha\in\mathcal{D}_i$, let $\xi_{i,\alpha}=\theta_{i,\alpha}/(1-\theta_{i,\alpha})$, and $O_\alpha=\mathcal{T}(\{\alpha\})=\alpha$ be the observation of single-theme collection $\{\alpha\}$. Let $\mathcal{D}_i^{(0)}=\{\alpha\in\mathcal{D}_i : \nexists\beta\in\mathcal{D}_i, \text{ subject to } \beta\subset\alpha\}$; we have

$$\mathcal{P}_{\boldsymbol{\theta}_i}(O_\alpha)=\xi_{i,\alpha}\lambda_i, \qquad \forall\alpha\in\mathcal{D}_i^{(0)}.$$

If two dictionaries $(\mathcal{D}_1,\boldsymbol{\theta}_1)$ and $(\mathcal{D}_2,\boldsymbol{\theta}_2)$ satisfy $\mathcal{O}_{\mathcal{D}_1}=\mathcal{O}_{\mathcal{D}_2}$ and $\mathcal{P}_{\boldsymbol{\theta}_1}=\mathcal{P}_{\boldsymbol{\theta}_2}$, we have

(a) $\lambda_1=\lambda_2\triangleq\lambda$, and $\mathcal{D}_1^{(0)}=\mathcal{D}_2^{(0)}\triangleq\mathcal{D}^{(0)}$.

(b) For any $\alpha \in \mathcal{D}^{(0)}$, $\xi_{1,\alpha}\lambda_1 = \mathcal{P}_{\theta_1}(O_\alpha) = \mathcal{P}_{\theta_2}(O_\alpha) = \xi_{2,\alpha}\lambda_2$; thus,

$$\xi_{1,\alpha} = \xi_{2,\alpha} \triangleq \xi_\alpha \quad \text{and} \quad \theta_{1,\alpha} = \theta_{2,\alpha} \triangleq \theta_\alpha, \qquad \forall \alpha \in \mathcal{D}^{(0)}.$$

(c) Define

$$\tilde{\mathcal{D}}_i^{(t)} = \bigcup_{k \leqslant t} \mathcal{D}_i^{(k)},$$

and

$$\mathcal{D}_i^{(t+1)} = \{\alpha \in \mathcal{D}_i : \nexists \beta \in \mathcal{D}_i - \tilde{\mathcal{D}}_i^{(t)}, \text{ subject to } \beta \subset \alpha\}.$$

If $\tilde{\mathcal{D}}_1^{(t)} = \tilde{\mathcal{D}}_2^{(t)} = \tilde{\mathcal{D}}^{(t)}$, and $\theta_{1,\alpha} = \theta_{2,\alpha} = \theta_\alpha$ for all $\alpha \in \tilde{\mathcal{D}}^{(t)}$, we have

(i) $\mathcal{D}_1^{(t+1)} = \mathcal{D}_2^{(t+1)} =^{\Delta} \mathcal{D}^{(t+1)}$
(ii) $\mathcal{P}_{\theta_1}(O_\alpha) = g(\{\xi_\beta : \beta \in \tilde{\mathcal{D}}^{(t)}\}, \xi_{1,\alpha})\lambda_1 = g(\{\xi_\beta : \beta \in \tilde{\mathcal{D}}^{(t)}\}, \xi_{2,\alpha})\lambda_2 = \mathcal{P}_{\theta_2}(O_\alpha)$ for $\forall \alpha \in \mathcal{D}^{(t+1)}$, where $g$ is a strictly monotone function of $\xi_{i,\alpha}$ and
(iii) since $\lambda_1 = \lambda_2$, we also have $\theta_{1,\alpha} = \theta_{2,\alpha} =^{\Delta} \theta_\alpha, \forall \alpha \in \mathcal{D}^{(t+1)}$.

Because $\tilde{\mathcal{D}}_i^{(t)} \uparrow \mathcal{D}_i$ when $t$ increases, these facts give a recursive way to prove $\mathcal{D}_1 = \mathcal{D}_2$ and $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2$ given $\mathcal{P}_{\theta_1} = \mathcal{P}_{\theta_2}$, i.e. TDM($\mathcal{M}_B, \mathcal{T}$) is identifiable.

### A.2. Proof of theorem 2

Let $H_Z = \Sigma_{x \in E} h(x)\omega_x Z_x$ and $W_Z = \Sigma_{x \in E} \omega_x Z_x$; we have

$$E_Z(H_Z) = \mu,$$

$$E_Z(W_Z) = 1,$$

$$\text{var}_Z(H_Z) = \sum_{x \in E} h^2(x)\pi^2(x)(1 - p_x)/p_x = \sum_{\pi(x) < c} h^2(x)\pi(x)\{c - \pi(x)\},$$

$$\text{var}_Z(W_Z) = \sum_{x \in E} \pi^2(x)(1 - p_x)/p_x = \sum_{\pi(x) < c} \pi(x)\{c - \pi(x)\},$$

$$\text{cov}_Z(H_Z, W_Z) = \sum_{x \in E} h(x)\pi^2(x)(1 - p_x)/p_x = \sum_{\pi(x) < c} h(x)\pi(x)\{c - \pi(x)\}.$$

By the delta method, we see that

$$\begin{aligned} E_Z(\hat{\mu}_Z) &\approx E_Z[H_Z\{1 - (W_Z - 1) + (W_Z - 1)^2 + \ldots\}] \\ &\approx \mu - \text{cov}_Z(H_Z, W_Z) + \mu\,\text{var}_Z(W_Z). \end{aligned}$$

Considering that

$$|\text{cov}_Z(H_Z, W_Z) - \mu\,\text{var}_Z(W_Z)| \leqslant \sum_{\pi(x) < c} |h(x) - \mu|\pi(x)\{c - \pi(x)\} \leqslant c\,E_\pi|h(X) - \mu| \to 0,$$

when $c \to 0$ (or, equivalently, $n_c \to \#(E)$), we have

$$E_Z[\hat{\mu}_Z] = \mu + O(c),$$

i.e. $\hat{\mu}_Z$ is asymptotically unbiased. The variance of $\hat{\mu}_Z$ can be explored by using the standard delta method for ratio statistics:

$$\text{var}_Z(\hat{\mu}_Z) = \text{var}_Z\left(\frac{H_Z}{W_Z}\right) \approx \mu^2\,\text{var}_Z(W_Z) + \text{var}_Z(H_Z) - 2\mu\,\text{cov}_Z(H_Z, W_Z).$$

Hence, the mean-squared error of $\hat{\mu}_Z$ is

$$\begin{aligned}\mathrm{MSE}(\hat{\mu}_Z) &= (E_Z[\hat{\mu}_Z] - \mu)^2 + \mathrm{var}_Z(\hat{\mu}_Z) \\ &= \mu^2 \mathrm{var}_Z(W_Z) + \mathrm{var}_Z(H_Z) - 2\mu \mathrm{cov}_Z(H_Z, W_Z) + O(c^2) \\ &= \sum_{\pi(x)<c} \{h(x) - \mu\}^2 \pi(x)\{c - \pi(x)\} + O(c^2).\end{aligned}$$

Considering that

$$\sum_{\pi(x)<c} \{h(x) - \mu\}^2 \pi(x)\{c - \pi(x)\} < \mathrm{var}_\pi\{h(X)\}c,$$

and $c \leqslant 1/n_c$, we have $\mathrm{MSE}(\hat{\mu}_Z) \leqslant \mathrm{var}_\pi\{h(X)\}/n_c$, i.e. the rejection control samples are statistically more efficient than independent, identically distributed samples with the same sample size.

## References


Agrawal, R., Imielinski, T. and Swami, A. (1993) Mining association rules between sets of items in large databases. In *Proc. SIGMOD Int. Conf. Management of Data*, pp. 207–216. New York: Association for Computing Machinery.

Agrawal, R. and Srikant, R. (1994) Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. Very Large Data Bases, Santiago*, pp. 487–499. San Francisco: Morgan Kaufmann.

Blei, D. and Lafferty, J. (2006) Dynamic topic models. In *Proc. 23rd Int. Conf. Machine Learning*, pp. 113–120. New York: Association for Computing Machinery.

Blei, D. and Lafferty J. (2007) A correlated topic model of science. *Ann. Appl. Statist.*, **1**, 17–35.

Blei, D., Ng, A. and Jordan, M. (2003) Latent Dirichlet allocation. *J. Mach. Learn. Res.*, **3**, 993–1022.

Bussemaker, H., Li, H. and Sigga, E. (2000) Building a dictionary for genomes: identification of presumptive regulatory sites by statistical analysis. *Proc. Natn. Acad. Sci. USA*, **97**, 10096–10100.

Del Moral, P. (2004) *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. New York: Springer.

Del Moral, P., Doucet, A. and Jasra, A. (2006) Sequential Monte Carlo samplers. *J. R. Statist. Soc.* B, **68**, 411–436.

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977) Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J. R. Statist. Soc.* B, **39**, 1–38.

Doucet, A., de Freitas, J. F. G. and Gordon, N. J. (eds) (2001) *Sequential Monte Carlo Methods in Practice*. New York: Springer.

Efron, B. and Thisted, R. (1976) Estimating the number of unknown species: how many words did Shakespeare know? *Biometrika*, **63**, 435–437.

Fearnhead, P. and Clifford, P. (2003) On-line inference for hidden Markov models via particle filters. *J. R. Statist. Soc.* B, **65**, 887–899.

Geman, S. and Geman, D. (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images.. *IEEE Trans. Pattn Anal. Mach. Intell.*, **6**, 721–741.

Han, J., Pei, J., Yin, Y. and Mao, R. (2004) Mining frequent patterns without candidate generation. *Data Mining Knowl. Discov.*, **8**, 53–87.

He, P., Deng, K., Liu, Z., Liu, D., Liu, J. and Geng, Z. (2012) Discovering herbal functional groups of traditional Chinese medicine. *Statist. Med.*, **31**, 636–642.

Jamshidian, M. and Jennrich, R. I. (1993) Conjugate gradient acceleration of the EM algorithm. *J. Am. Statist. Ass.*, **88**, 221–228.

Jamshidian, M. and Jennrich, R. I. (1997) Acceleration of the EM algorithm by using quasi-Newton methods. *J. R. Statist. Soc.* B, **59**, 569–587.

Liu, J. (2001) *Monte Carlo Strategies in Scientific Computing*. New York: Springer.

Liu, J. and Chen, R. (1998) Sequential Monte Carlo methods for dynamic systems. *J. Am. Statist. Ass.*, **93**, 1032–1044.

Liu, J., Chen, R. and Wong, W. H. (1998) Rejection control and sequential importance sampling. *J. Am. Statist. Ass.*, **93**, 1022–1031.

Piatetsky-Shapiro, G. (1991) Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases* (eds G. Piatetsky-Shapiro and W. J. Frawley). Cambridge: American Association for Artificial Intelligence–MIT Press.

Webb, G. (2007) Discovering significant patterns. *Mach. Learn.*, **68**, 1–33.

Zaki, M. (2000) Generating non-redundant association rules. In *Proc. 6th Int. Conf. Knowledge Discovery and Data Mining*, pp. 34–43. New York: Association for Computing Machinery.