A simple algorithm for topic identification in 0–1 data

Jouni K. Seppänen, Ella Bingham, and Heikki Mannila

Helsinki University of Technology and Helsinki Institute for Information Technology

Abstract. Topics in 0–1 datasets are sets of variables whose occurrences are positively connected together. Earlier, we described a simple generative topic model. In this paper we show that, given data produced by this model, the lift statistics of attributes can be described in matrix form. We use this result to obtain a simple algorithm for finding topics in 0–1 data. We also show that a problem related to the identification of topics is NP-hard. We give experimental results on the topic identification problem, both on generated and real data.

1 Introduction

Large collections of 0–1 data occur in many applications, such as information retrieval, web browsing, telecommunications, and market basket analysis. While the dimensionality of such data sets can be large, the variables (or attributes) are seldom completely independent. Rather, it is natural to assume that the attributes are organized into (possibly overlapping) *topics*, i.e., collections of variables whose occurrences are somehow connected to each other.[1] For example, in document data the topics correspond to topics of the document: e.g., phrases "data mining", "decision trees" and "association rules" probably are included in one topic, which might be called the "data mining" topic. In supermarket market basket data, the topics could correspond to classes of products such as soft drinks, vegetables, etc. In discretized gene expression data topics could correspond to groups of genes that are expressed in similar conditions or tissues.

Finding topics from data is by no means easy: the topics can be overlapping, and a particular topic is active only for a subset of documents. For example, simple frequent set based approaches are unable to find topics, as the attributes in a topic are seldom 1 together. There has been lots of work that searches for latent structure in 0–1 data (see, e.g., [1–8]). The approaches range from simple methods based on covariance-type statistics (e.g., [9]) to full probabilistic models (e.g., [4]) and to spectral approaches [10].

In order to discover topics from 0–1 data, one first has to specify the model for topics, and then give a method that finds topics corresponding to the model. In this paper we describe a simple generative topic model, based on our previous work [11]. We prove some analytical results about the model by using the concept

[1] In the literature of information retrieval, a topic is a probability distribution on the universe of terms, typically concentrating on a few terms.

of lift [12]. We show that the lift statistics of individual attribute pairs can be described in matrix form as linear combinations of lift statistics of disjoint topics. Based on this observation, we give a simple algorithm for finding topics in 0–1 data. We also show that one form of the topic identification problem is NP-hard. We give experimental results on document data and on gene expression data, showing that the algorithm works well in practice.

First we review some other methods for finding latent structure in binary data. Many of these generative models are quite powerful and are able to describe complex situations. On the other hand, finding exact solutions for them is computationally intractable, and it is difficult to get a clear picture of the quality of the obtained estimates. Many of the methods are also symmetric with respect to 0 and 1; on the basis of the asymmetry in the data generating process, this can be viewed as a potential source of problems.

In nonnegative matrix factorization (NMF) [1], an observed data matrix is decomposed into a product of two unknown matrices. All three matrices have nonnegative entries. The observed data is regarded as a sum of latent variables. Lee and Seung give two algorithms for finding the unknown matrices; there is, however, no probabilistic interpretation of the results of NMF. Computationally, the methods seems very demanding and there are no clear results on the quality of the solutions [13].

The latent semantic analysis (LSA) method [2] uses singular-value decomposition to decompose an observed data matrix into a product of matrices. (In contrast to NMF, the matrices can have negative entries, too.) In a seminal paper by Papadimitriou et al. [3] some arguments were given to justify the performance of LSI by presenting a probabilistic corpus model. Their basic model is quite general and somewhat similar to ours.

Hofmann [4] has presented a probabilistic version of LSA, termed PLSA. His formal model is fairly close to ours and we will show comparative results on the models. For each observation vector, some topics are first selected according to some observation-specific topic probabilities; then, the topics generate attributes according to some topic-attribute probabilities. The attributes are conditionally independent given the topic. Hofmann's main interest is in good estimation of all the parameters using the EM algorithm, while we are interested in the structure of the data (that is, the probabilities of attributes belonging to topics) and also explaining why the methods would find topics.

Laten Dirichlet Allocation (LDA) [14–16] is a method in which the data model is closely similar to Hofmann's PLSA but the estimation of the parameters is computationally more demanding: a variational approximation to the data likelihood is needed prior to EM estimation of the parameters. Independent component analysis (ICA) ([8, 17, 18]) is a statistical method that expresses observed multidimensional sequences as combinations of unknown latent variables, that are statistically as independent as possible. The so called probe distances [19] of attributes can be used to find (possibly overlapping) sets of attributes that behave similarly with respect to other attributes; we studied this in an earlier paper [11]. Cooley and Clifton [9] compute the frequent sets in the data and cluster them using a hypergraph partitioning scheme, thus avoiding the problem of not having all attributes of a topic present in one data vector.

A popular method to analyze 0–1 data is the class of finite mixtures of multivariate Bernoulli distributions. However, for the Bernoulli models, the values 0 and 1 have symmetric status, while for our topic models defined in Section 2 this is not the case. Another important difference between Bernoulli (or any other) mixture model and our model is that in mixture models it is assumed that an observed 0–1 vector is only generated by one latent topic, although generation probabilities are given for all latent topics. In this paper we assume that a data vector is generated by the interaction of several latent topics. Binary generative topographic mapping [20, 21] also assumes that the data vectors are generated by one latent topic at a time.

The rest of this paper is organized as follows. We describe our model and examine some of its analytical properties in Section 2. In Section 3 we study the lift statistic and describe the simple algorithm based on it. We give experimental results in Section 4, and conclude in Section 5.

2 Topic Models

In this section we present our concept of a topic model, give the likelihood function of the model, and discuss what kinds of parameter values are realistic. This form of the model was introduced earlier by us [11].

Let U be an n-element set of *attributes* (e.g., words). A *k-topic model* $\mathcal{T}$ arranges the n attributes into k *topics*. The model has the following parameters: a k-element vector $\boldsymbol{s} = (s_1, \ldots, s_k)$ corresponding to the k topics, and a $k \times n$ matrix $\mathbf{Q}$ whose elements relate the topics to the attributes; the element corresponding to topic i and attribute A is denoted by $Q_{i,A}$. All elements of $\boldsymbol{s}$ and $\mathbf{Q}$ must be probabilities, i.e., reals in the range $[0, 1]$, but we stress that neither $\boldsymbol{s}$ nor any row or column of $\mathbf{Q}$ is required to sum up to 1.

A data vector $\boldsymbol{x}$ (e.g., a document) is sampled from $\mathcal{T}$ as follows. First, the active topics are selected by sampling a k-element binary vector $\boldsymbol{t}$ so that each component t_i is 1 with probability s_i, independently of all other components. Second, the active topics generate the attributes. For each topic i, an n-element binary vector $\boldsymbol{x}_i$ is sampled so that the component corresponding to A is 1 with probability $t_i Q_{i,j}$, independently of all other components. The data vector $\boldsymbol{x}$ is then the logical *or* (i.e., maximum) of all the vectors $\boldsymbol{x}_i$, $\boldsymbol{x} = \bigvee_{i=1}^{k} \boldsymbol{x}_i$.

It would be possible to add another layer on top of the topics, selecting the topic probabilities anew for each data vector from, e.g., a Dirichlet distribution. Many of our results could be generalized to such settings, which however fall outside the scope of this treatment. This type of approach has been taken in [3, 4, 14–16].

We next present the likelihood function of a k-topic model $\mathcal{T}$ with parameters $\boldsymbol{s}, \mathbf{Q}$. The data D consists of vectors $\boldsymbol{x}$, each considered independently of the others,

$$P(D \mid \mathcal{T}) = \prod_{\boldsymbol{x} \in D} P(\boldsymbol{x} \mid \mathcal{T}).$$

The probability of a single observation $\boldsymbol{x}$ is

$$P(\boldsymbol{x} \mid \mathcal{T}) = \sum_{\boldsymbol{t}} P(\boldsymbol{t} \mid \mathcal{T}) P(\boldsymbol{x} \mid \boldsymbol{t}, \mathcal{T}).$$

Here the sum runs over all k-element 0/1 vectors $\boldsymbol{t}$, corresponding to all 2^k possible combinations of active topics. The probability of a topic combination depends on the parameters $\boldsymbol{s}$ only,

$$P(\boldsymbol{t} \mid \mathcal{T}) = P(\boldsymbol{t} \mid \boldsymbol{s}) = \prod_{i=1}^{k} P(t_i \mid s_i) = \prod_{i=1}^{k} s_i^{t_i} (1 - s_i)^{1 - t_i}.$$

The probability of an observation given the active topics depends on the parameters $\mathbf{Q}$ only,

$$P(\boldsymbol{x} \mid \boldsymbol{t}, \mathcal{T}) = P(\boldsymbol{x} \mid \boldsymbol{t}, \mathbf{Q}) = \prod_{A \in U} P(x_A \mid \boldsymbol{t}, \mathbf{Q}),$$

where x_A denotes the element of $\boldsymbol{x}$ that corresponds to the attribute $A \in U$. A single attribute has a value of either zero or one, with distribution

$$P(x_A \mid \boldsymbol{t}, \mathbf{Q}) = p_A^{x_A} (1 - p_A)^{1 - x_A} = \begin{cases} 1 - p_A, & x_A = 0 \\ p_A, & x_A = 1, \end{cases}$$

where

$$p_A = 1 - \prod_{i=1}^{k} (1 - Q_{i,A})^{t_i}.$$

The likelihood function, if expanded fully, would have a large number of terms because of the sum over 2^k topic combinations $\boldsymbol{t}$. This hints at a high computational complexity, and indeed the task of selecting the best $\boldsymbol{t}$ is difficult, as illustrated by the following theorem, whose proof we defer to the Appendix.

Theorem 1. *The following problem is* NP*-complete: given a topic model $\mathcal{T}$, a single data vector $\boldsymbol{x}$ and a threshold ρ, decide whether there is a topic assignment $\boldsymbol{t}$ such that the probability of the data given the assignment exceeds the threshold, $P(\boldsymbol{x} \mid \boldsymbol{t}, \mathcal{T}) \geq \rho$.*

However, the models involved in the proof would best be described as contrived, so the result should not dissuade us from researching some reasonable subclass of topic models. But what kind of models are reasonable?

One assumption that we will make is that the topic probabilities s_i are small. This seems reasonable at least in the context of document data: if some words occur in most documents, in information retrieval they would be classified as stop words and not considered in searches, and it is the less common words that distinguish interesting documents.

Another question is the amount of overlap between topics—if two topics consist of almost completely the same attributes, it does not seem easy to distinguish between them. In [11] we considered a class of "ϵ-separable" models, an idea similar to that in [3]. A model is ϵ-separable if every topic has a set of primary attributes and assigns at most a fraction ϵ of its attribute-activation weight to the non-primary attributes. However, the ϵ-separability property does not perfectly capture the idea of almost-disjoint topics, as the discussion in [11, before Lemma 3] notes: for example, several topics can "conspire" against another topic i by giving high weight to one of i's primary attributes. Even if every high weight is less than a fraction ϵ of the topic's total weight, it is possible that the majority of activations of that attribute come from the conspiring topics and not the primary topic.

This leads us to define a different separability concept: a model has *θ-bounded conspiracy* if every attribute A has a primary topic i such that

$$\sum_{j \neq i} Q_{j,A} \leq \theta Q_{i,A}.$$

We conjecture that a model is discoverable from data if it has low values of s_i and conspiracy bounded by some low θ.

3 Using the Lift Statistic

We now consider a statistic commonly called called *lift* or *interest* [12, 22, 23],

$$\mathrm{lift}(A, B) = \frac{P(A \mid B)}{P(A)} = \frac{P(A, B)}{P(A)P(B)},$$

which is a kind of a relative risk factor: how much more common is it to observe A given that B is observed, compared to no information about B? Lift was chosen because it measures dependence, which is highly relevant to topic models—when two attributes belong strongly to the same topic, their co-occurrence should deviate significantly from the independence assumption. For independent A and B, $\mathrm{lift}(A, B) = 1$, and the stronger the (positive) dependence, the higher the lift. Note that our model predicts $\mathrm{lift}(A, B) \geq 1$ for all pairs $A, B \in U$; thus, one way of assessing whether the model fits a given data set is to see how $\mathrm{lift}(A, B)$ is actually distributed.

Proposition 1. *Assume that attribute A is only generated by topic i. Then for any attribute B,*

$$\mathrm{lift}(A, B) = \frac{P(t_i \mid B)}{P(t_i)} = \frac{P(t_i, B)}{P(t_i)P(B)}.$$

Proof. We factorize the probabilities: $P(A) = P(A, t_i) = P(t_i)P(A \mid t_i)$ and $P(A, B) = P(t_i, A, B) = P(t_i)P(A \mid t_i)P(B \mid t_i)$. Then

$$\text{lift}(A, B) = \frac{P(A, B)}{P(A)P(B)} = \frac{P(t_i)P(A \mid t_i)P(B \mid t_i)}{P(t_i)P(A \mid t_i)P(B)}.$$

Using Bayes' theorem $P(B \mid t_i) = P(B)P(t_i \mid B)/P(t_i)$ and canceling terms we obtain the result. ☐

What Proposition 1 says is that if A is a "core attribute" of topic i, i.e., an attribute generated by i only, then A represents i perfectly in lift calculations, even if $Q_{i,A} < 1$. Of course in practice, when the lift must be estimated from data, a small value of $Q_{i,A}$ can cause poor results. Another point to note is that the probability $P(B \mid t_i)$ appearing in the proof is *not* the model parameter $Q_{i,B}$. Instead, it is the probability that any topic will generate B conditioned on the fact that at least topic i is active. Proposition 1 has as immediate consequences two results that we used already in [11].

Corollary 1. *If attributes A and B are only generated by topic i, i.e., $Q_{j,A} = Q_{j,B} = 0$ for $j \neq i$, then* $\text{lift}(A, B) = s_i^{-1}$.

Corollary 2. *If attribute A is only generated by topic i and attribute B is only generated by topic j, then* $\text{lift}(A, B) = 1$.

Thus, the lift statistic between attributes belonging to one topic only is very simple. The interesting question is how lift behaves when an attribute belongs to several topics.

Assume that attribute A is only generated by topic i, and attribute B is generated by both topics i and j. Now $\text{lift}(A, B)$ is, after simplification,

$$\frac{P(A, B)}{P(A)P(B)} = \frac{Q_{i,B} + s_j Q_{j,B} - Q_{i,B} s_j Q_{j,B}}{s_i Q_{i,B} + s_j Q_{j,B} - s_i s_j Q_{i,B} Q_{j,B}} \approx \frac{Q_{i,B} + s_j Q_{j,B}}{s_i Q_{i,B} + s_j Q_{j,B}}$$

where in the approximation we have assumed that $Q_{i,B} s_j Q_{j,B}$ and $s_i s_j Q_{i,B} Q_{j,B}$ are small compared to the other terms. The above formula generalizes to the case where B is generated by some other topics than i and j, too: before the approximation we then have several second order terms $s_\ell Q_{\ell,B}$ corresponding to all topics ℓ that generate B, and similarly several third order terms $s_\ell Q_{i,B} Q_{\ell,B}$ (in the numerator) or fourth order terms $s_i s_\ell Q_{i,B} Q_{\ell,B}$ (in the denominator).

Assume now that all the topic probabilities are (approximately) equal, i.e., $s_\ell \approx s$ for all topics ℓ. Then we can write the above formula as $\text{lift}(A, B) \approx (s^{-1} Q_{i,B} + Q_{j,B})/(Q_{i,B} + Q_{j,B})$. Furthermore, let each topic ℓ have c_ℓ core attributes that are only generated by that topic. Then using Corollaries 1 and 2 we note that the lifts of A and all core attributes can be included in the formula as follows:

Observation. The lift between a core attribute A of topic i and an attribute B generated by topics i and j is

$$\text{lift}(A, B) \approx \sum_{A'} \text{lift}(A, A') c_i^{-1} \frac{Q_{i,B}}{Q_{i,B} + Q_{j,B}} + \sum_{D'} \text{lift}(A, D') c_j^{-1} \frac{Q_{j,B}}{Q_{i,B} + Q_{j,B}}$$

where $\sum_{A'} \text{lift}(A, A')c_i^{-1}$ is an averaged estimate of s^{-1}, $\sum_{D'} \text{lift}(A, D')c_j^{-1} = 1$ and the two sums run over the core attributes A' and D' of topics i and j, respectively. Also, we may add a third summation including $\text{lift}(A, F')$ where F' is a core attribute belonging to topic l into which B does not belong to, as then $Q_{l,B} = 0$ and the whole term vanishes. This observation again generalizes to the case where B is generated by multiple topics.

The above reasoning included approximations in discarding high-order terms and the somewhat crude assumption that all s_i are equal. In any case, it does yield an idea of how to discover topics: for an attribute B that belongs to several topics, define a vector $\boldsymbol{\alpha}$ whose length is the total number of all core attributes. Its element corresponding to A (a core attribute of topic i) is $\alpha_A = Q_{i,B}/(c_i \sum_j Q_{j,B})$, where i is the primary topic of A. Then $\text{lift}(A, B) \approx \boldsymbol{\alpha}^T \text{lift}(A, \cdot)$ for all core attributes A, where we denote by $\text{lift}(A, \cdot)$ the vector of lifts between A and all core attributes (where $\text{lift}(A, A) = 0$). This gives us an algorithm for finding the topics in which the attributes belong, and also the parameters Q:

- Identify those attributes that belong to one topic only—this can be done by looking at the lift statistics, which are always either 1 or $1/s$ for those attributes.
- Cluster those attributes using some traditional clustering algorithm; at this stage the clusters do not overlap and do not cover all attributes—if an attribute B belongs to several topics, its lifts are intermediate between 1 and $1/s$, and so B is not clustered. For A belonging to one topic i only, $Q_{i,A} = P(AA')/P(A')$ which can be averaged over all A' belonging to the same topic i as A.
- For attributes B which are not clustered, find a decomposition $\text{lift}(B, \cdot) = \boldsymbol{\alpha}^T R$, where the square symmetric matrix R has the vectors $\text{lift}(A, \cdot)$ (of already clustered attributes A) as its columns. All of the lifts in this formula are known, so the vector $\boldsymbol{\alpha}$ can be estimated straightforwardly. The elements of $\boldsymbol{\alpha}$ are nonzero for those attributes that share a topic with B, and zero for others. Also, the elements are more or less constant within attributes of a given topic. Now $Q_{i,B} = \alpha_A c_i / \sum_j Q_{j,B}$ where α_A can be averaged over all A' belonging to topic i, c_i is known, and for small and equal s_j we can approximate $P(B) \approx s \sum_j Q_{j,B}$, which gives us $\sum_j Q_{j,B}$. We can also assume $\sum_j Q_{j,B} = 1$ and scale the estimated $Q_{i,B}$ accordingly.

4 Experimental Results

4.1 Generated Data

We designed experiments to see how the conspiracy statistic θ of a model affects our clustering results. The results corroborate our conjecture that low-conspiracy models are easier to discover.

We constructed random models with θ-bounded conspiracy using the following recipe. The model has 10 topics and 100 attributes. The probability s_i

of a topic was drawn uniformly at random from the interval $[0.01, 0.8]$ in the first experiment, and from the interval $[0.01, 0.5]$ in the second experiment. Each attribute was assigned a primary topic so that each topic was primary for 10 attributes.

Vahan hassua kun exper 1 ja exper 2 mallien luonti selitetaan tassa alussa ja itse kokeet vasta myhemmin. Johtuu siita, etta mallit ovat erilaiset. -Ella

To assign the within-topic attribute probabilities $Q_{i,A}$ so that the conspiracy parameter is θ, we first drew a number p uniformly from $[0, 1]$ and let $Q_{i,A} = p$ for the primary topic i. Then we distributed the mass θp to the non-primary topics in somewhat different uneven ways in our two experiments. In the first experiment, each non-primary topic in random order received a fraction of ϕ of the remaining mass, where ϕ is chosen randomly from $[0, 1]$. The last topic received all remaining mass to make the mass sum up exactly to θp. (Thus, when $\theta > 1$, it is possible that the primary topic originally assigned for an attribute ends up having less weight than some other topics.) In the second experiment, a randomly chosen non-primary topic j_1 received a fraction ϕ_1 of the total probability mass $\theta Q_{i,A}$; another non-primary topic j_2 received a fraction ϕ_2 of the remaining unassigned probability mass; and so on. All fractions ϕ were uniformly distributed on $[0, 1]$.

This way of generating a random model includes a number of somewhat arbitrary choices that we now justify. First, the topic probabilities s_i were chosen not from $[0, 1]$ but from a smaller interval. Some lower limit is necessary so that each topic is represented in a finite data sample; and an upper limit is needed by our algorithm, which distinguishes a topic by estimating its probability and cannot discover a topic that is almost always active. In a preliminary test (not shown), our algorithm's performance was best with low upper limits, and deteriorated rapidly when the upper limit increased over about 0.85.

Second, we discuss the distribution of the within-topic attribute probabilities of non-primary topics. A more obvious strategy would be to draw the probabilities independently and then to normalize, but then the distribution would have become more even. With 9 non-primary topics, all the probabilities would center around $\theta/9$ times the primary probability, which makes the task far easier: none of the non-primary topics is likely to be confused with the primary topic. Especially in the second experiment, the procedure typically results in a few non-primary topics with non-negligible topic-attribute probabilities for each attribute. We wish to mimic the behavior of true data sets, such as text document data: a term may have several meanings, perhaps a primary meaning and one or few secondary meanings, hence it belongs primarily to one topic of discussion and secondarily to a few other topics, but not to all possible topics.

In the first experiment we study the clustering ability of our algorithm. To assess a clustering we need to compare the clustering result to the original model — we assume that the "real" model has been perturbed by the conspiring attributes, and our task is to recover the real, non-perturbed model. The comparison requires matching the clusters to the original topics, a task that has no obvious solution in less-than-perfect clusterings. Instead of fairly complex maximum matching techniques we chose the greedy approach of repeatedly pairing up the rows that are closest among all possible choices. Then we counted the proportion of attributes whose cluster was paired up with the original topic. We consider this kind of scoring appropriate when the clustering is reasonably good, but for poor clusterings it is somewhat arbitrary. For instance, combinatorial reasons preclude a score of zero: even the trivial clustering that puts every attribute in the same cluster is counted as having discovered one topic.

In this task, we compared simple hierarchical clustering (with the average linkage algorithm) to probabilistic latent semantic analysis (PLSA). We did not use the correction of attaching unclustered attributes to existing clusters described at the end of Section 3—the objective was to see how much increasing θ perturbates the clustering result. Such reconstructed $\mathbf{Q}$ matrices do indeed display the "conspiracy" effects, but the quality of the reconstruction is difficult to assess in a quantitative way: the model's likelihood function would be natural to use, but it is very slow to compute because of the exponential number of possible topic assignments. In the case of PLSA, which does not produce a clustering as such, we discretized the resulting matrix by selecting from each latent component the top 10 attributes. Figure 1 displays the results.

The "quantization" behavior observable at about $\theta = 0.25 \ldots 0.5$ is due to the definition of the score: when the algorithm sees two topics as one cluster, it loses 10% of the maximum score. For larger values of θ, the performance degrades gracefully. The vertical line marks $\theta = 1$, beyond which the task is not always well-posed: it is possible for some non-primary probability of an attribute to exceed the primary attribute so that the original cluster assignment is in fact incorrect. We show the data despite this problem, so as to better illustrate the degradation of the clustering.

In the second experiment, we estimated the topic-attribute probabilities $\mathbf{Q}$ using the lift statistic, NMF, PLSA[2] and K-means. The NMF and PLSA methods estimate Q given the observed binary data. A naive alternative is the simple K-means algorithm which clusters the attributes into non-overlapping sets; we assume that $Q_{\cdot,i}$ is equal for all attributes of topic i and sums to 1 at each topic.

Figure 2 shows the mean squared errors (MSE's) of the estimated $\mathbf{Q}$, compared to the true probabilities used to generate the data. The conspiracy parameter θ runs from 0 to 1. At each θ, the topic probabilities s are sampled anew, so there is great variability in the data models. Originally, the topic-attribute probabilities estimated by the methods do not necessarily sum to 1 at each topic — in PLSA they do, but neither in the other methods nor in the true data model — but we scale them accordingly, to be able to compare the MSE's.

In Figure 2 we see that at smaller θ, the Lift algorithm estimates the $\mathbf{Q}$ and thus the structure of the data very nicely. When θ grows very large, the data model is more difficult to estimate. The behaviors of NMF and PLSA[3] do not depend on θ, which is natural: the methods are not primarily aimed for such θ-bounded data but instead are able to estimate the structure also when the

[2] The PLSA method was kindly programmed by Mr. Teemu Hirsimäki.
[3] No simulated annealing was used in the EM algorithm of the PLSA.

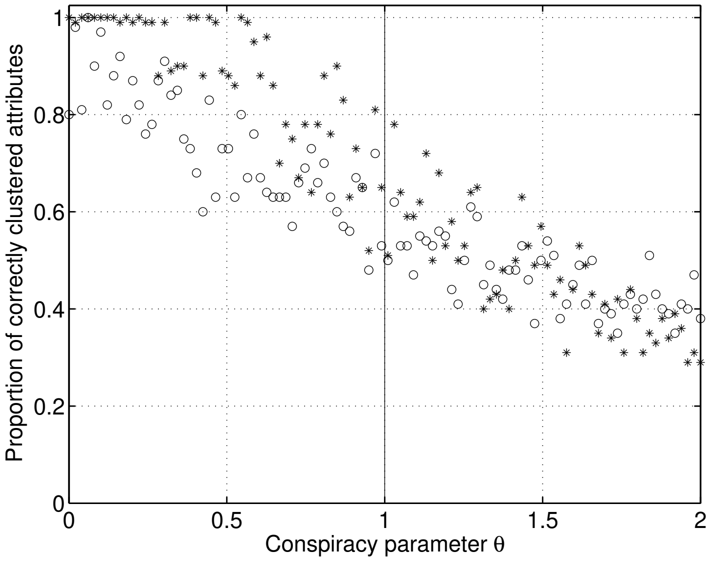

Fig. 1. Asterisks show the performance score of average linkage clustering; circles show the score of PLSA; vertical line marks $\theta = 1$.

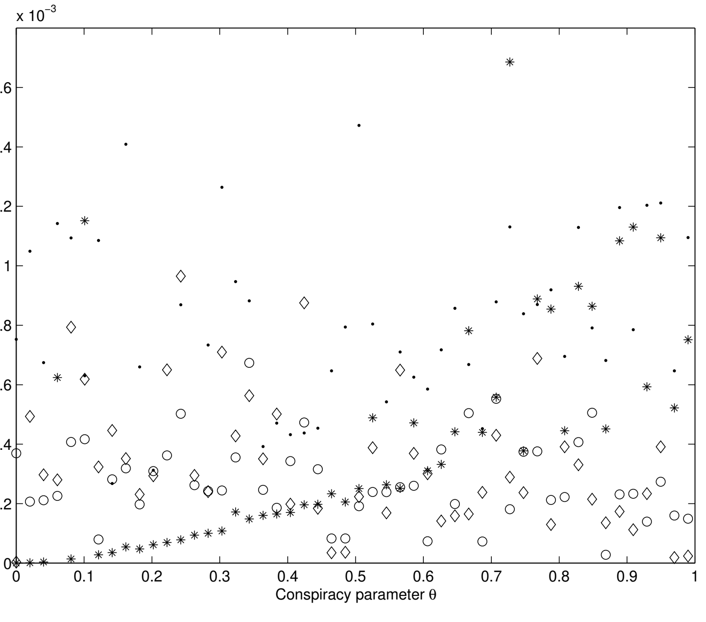

Fig. 2. Mean squared errors of $\mathbf{Q}$ at different conspiracy parameters θ. Lift $*$, NMF $\Diamond$, PLSA $\circ$, K-means $\cdot$.

topics are totally overlapping. The K-means algorithm estimates the structure of the data poorly for all θ.

4.2 Document Data

We performed experiments on bibliographical data on computer science available on the WWW[4]. We first tested the model's prediction that $\mathrm{lift}(A, B) \geq 1$ for all A, B; while it does not hold perfectly because there are negative correlations between words, the vast majority of these negative correlations are statistically insignificant (details omitted). We preprocessed the data by removing a small set of stop words and all numbers, and then selected the 100 most frequent terms for further analysis.

We computed the lift statistics between all term pairs and used hierarchical average linkage clustering based on the inverses of lifts. Table 1 shows how the terms are clustered into topics. The number of clusters (21) was chosen based on the distance between clusters being merged in the process of hierarchical clustering: until these 21 clusters, the intercluster distances were quite small but distances between the final 21 clusters were large. The structure in Table 1 is immediately familiar to a theoretical computer scientist: the topics concentrate on different fields of the science.

4.3 Gene Expression Data

We also performed topic finding on yeast gene expression data, using the same gene expression dataset as in [24] that combines the results of several different gene expression studies. The combined dataset measures the expression level of over six thousand genes in almost a hundred experiments; thus, we used the experiments as "attributes" and the genes as "measurements". The levels were discretized so that the top 5% expressed genes in each experiment were given the value 1. The results are not shown due to space constraints, but as a brief example, the discovered topics were seen to reflect cyclical behavior of the genes in the time-series experiments.

5 Concluding Remarks

We studied a simple generative topic model and showed that the lift statistics of attributes can be described in matrix form. Based on this, we obtained a simple algorithm for finding topics in 0–1 data. We also showed that a problem related to the identification of topics is NP-hard, and gave experimental results.

Several open problems remain. Our model is simple, and seems to yield good results; still, more complex models might do a better job at identifying, e.g., topics containing partly exclusive attributes. The identifiability of the model is another interesting issue: could one prove something about it? Further experimental studies are also needed.

[4] `http://liinwww.ira.uka.de/bibliography/Theory/Seiferas/`

topic	terms
1	algorithms approximation damath problems scheduling some tree two
2	analysis distributed libtr probabilistic systems
3	bounds communication complexity focs lower
4	algorithm efficient fast ipl matching problem set simple
5	design ieeetc network networks optimal parallel routing sorting
6	note tcs
7	finding graphs minimum planar polynomial sets sicomp time
8	graph number properties random tr
9	from information learning lncs theory
10	approach jacm linear new programming system
11	actainf binary search trees
12	abstract computation extended model stoc
13	automata finite languages mfcs
14	data dynamic infctrl logic programs structures using
15	applications icalp theorem
16	cacm computer computing science
17	crypto functions
18	jcss machines
19	algebraic beatcs computational geometry
20	de stacs van
21	codes dmath

Table 1. Terms in different topics. (The order of the topics is not relevant).

References

1. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401** (1999) 788–791
2. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science **41** (1990) 391–407
3. Papadimitriou, C.H., Raghavan, P., Tamaki, H., Vempala, S.: Latent semantic indexing: A probabilistic analysis. In: PODS '98. (1998) 159–168
4. Hofmann, T.: Probabilistic latent semantic indexing. In: SIGIR '99, Berkeley, CA (1999) 50–57
5. Carreira-Perpiñán, M.A., Renals, S.: Practical identifiability of finite mixtures of multivariate Bernoulli distributions. Neural Computation **12** (2000) 141–152
6. Gyllenberg, M., Koski, T., Reilink, E., Verlaan, M.: Non-uniqueness in probabilistic numerical identification of bacteria. J. Appl. Prob. **31** (1994) 542–548
7. Cadez, I.V., Smyth, P., Mannila, H.: Probabilistic modeling of transaction data with applications to profiling, visualization, and prediction. In Provost, F., Srikant, R., eds.: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Fransisco, CA (2001) 37–46
8. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley & Sons (2001)
9. Clifton, C., Cooley, R.: TopCat: Data mining for topic identification in a text corpus. In: Principles of Data Mining and Knowledge Discovery. (1999) 174–183
10. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. In: Knowledge Discovery and Data Mining. (2001) 269–274
11. Bingham, E., Mannila, H., Seppänen, J.K.: Topics in 0-1 data. In Hand, D., Keim, D., Ng, R., eds.: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD–2002), Edmonton, Alberta, Canada (2002) 450–455
12. Castelo, R., Feelders, A., Siebes, A.: MAMBO: Discovering association rules based on conditional independencies. LNCS **2189** (2001) 289–298
13. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems. (2000)
14. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. In: Neural Information Processing Systems 14. (2001)
15. Minka, T., Lafferty, J.: Expectation-propagation for the generative aspect model. In: Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence, Edmonton, Canada (2002)
16. Buntine, W.: Variational extensions to EM and multinomial PCA. In Elomaa, T., Mannila, H., Toivonen, H., eds.: Machine Learning: ECML 2002. Number LNAI 2430 in Lecture Notes in Artificial Intelligence. Springer-Verlag (2002) 23–34
17. Comon, P.: Independent component analysis — a new concept? Signal Processing **36** (1994) 287–314
18. Jutten, C., Herault, J.: Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. Signal Processing **24** (1991) 1–10
19. Das, G., Mannila, H., Ronkainen, P.: Similarity of attributes by external probes. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining. (1998) 23–29
20. Pajunen, P., Karhunen, J.: A maximum likelihood approach to nonlinear blind source separation. In: Proc. Int. Conf. Artif. Neural Networks. (1997) 541–546

21. Girolami, M.: A generative model for sparse discrete bianry data with non-uniform categorical priors. In: Proc. European Symposium on Artificial Neural Networks, Bruges, Belgium (2000) 1–6
22. Silverstein, C., Brin, S., Motwani, R.: Beyond market baskets: Generalizing association rules to dependence rules. Data Mining and Knowledge Discovery **2** (1998) 39–68
23. Tan, P.N., Kumar, V.: Interestingness measures for association patterns: A perspective. Technical Report TR00-036, University of Minnesota (2000) (KDD 2000 Workshop on Postprocessing in Machine Learning and Data Mining).
24. Mannila, H., Patrikainen, A., Seppänen, J.K., Kere, J.: Long-range control of expression in yeast. Bioinformatics **18** (2002) 482–483

Appendix

Proof of Theorem 1. That the problem is in NP is simple to see: the certificate is the topic vector $\boldsymbol{t}$, and the formula for $P(\boldsymbol{x} \mid \boldsymbol{t}, \mathcal{T})$ involves multiplying n numbers, each computable in $O(k)$ time.

To show NP-hardness, we reduce SAT to a topic assignment problem. Given a SAT instance of m clauses over n variables, we define a topic model with $2n$ topics and $n+m$ attributes. For each variable V_i, we create two topics T_i and T_i', and one attribute A_i. For each clause C_j, we create one attribute B_j. Each topic has probability 0.5, and each attribute has 0/1 within-topic probabilities as follows: attribute A_i has probability 1 in topics T_i and T_i' and probability 0 in other topics; attribute B_j has probability 1 in the topics T_i such that V_i appears positively in clause C_j and in the topics T_i' such that V_i appears negatively in clause C_j, and probability 0 in all other topics. We consider a data vector where all attributes have value 1.

Now, if the SAT problem has a satisfying truth assignment, it corresponds to a solution of the topic assignment problem where T_i is active if V_i is true and T_i' is active if V_i is false. This solution has likelihood 0.5^n, since exactly n topics are active, and the active topics explain all attributes A_i and B_j. Conversely, if a solution to the topic assignment problem exists such that the likelihood is at least 0.5^n, it must have at most n active topics. To explain attribute A_i, either T_i or T_i' must be active; thus the number of active topics is exactly n, and the solution corresponds to a truth assignment. Since the solution must also explain each attribute B_j, the truth assignment must satisfy the original problem. In summary, the SAT instance has a solution if and only if the topic assignment problem has a solution with likelihood at least 0.5^n. $\square$