Summarizing Itemset Patterns Using Probabilistic Models

Chao Wang and Srinivasan Parthasarathy

Department of Computer Science and Engineering, The Ohio State University
Contact: {srini}@cse.ohio-state.edu

ABSTRACT

Frequent itemset pattern mining is an important problem in data mining and has been studied extensively. Efficient mining algorithms exist to identify a complete set of frequent itemset patterns from data. However, sizes of the discovered patterns are usually quite large which would hinder their interpretability and application. Thus effective methods of summarizing these itemset patterns become necessary.

In this paper, we propose a novel approach to concisely represent a large number of frequent itemset patterns. In our approach, probabilistic graphical models are employed for the summarization task. More specifically, items are taken as random variables and Markov Random Field (MRF) models on these variables are constructed based on frequent itemset patterns and their support information. The summarization proceeds in a level-wise fashion. Statistics of smaller itemsets are used to construct an MRF model, and then supports of larger itemsets are inferred from the model. If the estimation error is within a user-specified tolerance, we bypass these itemsets, otherwise we use these itemsets to augment the MRF model. At the end of the process, all the itemset patterns in the resulting model afford a concise representation of the original collection of itemset patterns.

Extensive empirical study on real datasets show that the new approach can effectively summarize a large number of frequent itemset patterns. In our quantitative assessment we find that our approach compares favorably with a state-of-the-art summarization scheme in most cases.

1. INTRODUCTION

The problem of mining frequent patterns, particularly associations among items in transactional datasets, is an important one with many applications. Efficient algorithms to compute frequent patterns and rules associated with these patterns exist [2, 26, 10, 25, 8]. However, often times in many real-world problems the number of frequent patterns mined for various parametric settings is extremely large leaving the end-user swamped when it comes to interpreting and summarizing the results. As a result researchers have turned to various strategies to summarize the number of patterns the user is asked to examine. This has led to research on problems such as mining of *closed itemsets* [17], *maximal itemsets* [9], *non-derivable itemsets* [5], and more recently *profile patterns* [22].

Closed itemsets summarize itemsets by reusing their inherent representation. Basically, a frequent itemset is closed if none of its parent itemsets have exactly the same support. Similarly, a frequent itemset is maximal if none of its parent itemsets are frequent. Non-derivable itemsets (NDI) reduce the output size by eliminating redundant patterns from original set of frequent itemset patterns. More specifically, those patterns that are exactly derivable from its subset patterns using some combinatorial principle (e.g. inclusion-exclusion [5]) are pruned.

Closed itemsets and non-derivable itemsets are lossless forms of compressing frequent itemset patterns, i.e. the full list of frequent itemsets and associated frequency counts (used for computing itemset rules) can be exactly derived from the compressed representation. Note that researchers have pointed out that for some datasets and support thresholds we have $|NDI| < |Closed|$, while other datasets and support thresholds have $|Closed| < |NDI|$ [5].

Maximal itemsets allow greater compression when compared with closed patterns, but the representation is lossy – the list of frequent itemsets can be exactly computed but the exact frequency counts associated with each frequent itemset cannot be determined. There are some other lossy representations besides maximum itemsets. Top-k patterns approach by Han *et al.* [11] presents the most frequent k closed itemsets to the end-user. *Error-tolerant patterns* by Yang *et al.* [23] and Pei *et al.* [19] allow certain amount of fluctuations when evaluating supports of itemset patterns. A recent approach by Afrati *et al.* [1] uses K itemsets to recover a collection of frequent itemsets. However, it is not clear how to recover the support information in their approach.

It is important to note that the number of closed, maximal or non-derivable itemsets could still be very large, thus we need to further compress them. Furthermore, we want a summarization approach which takes into account the frequency information as well. Generally speaking, the end-user may be interested in those itemset patterns that can best represent the original collection of itemsets and their associated frequency as well. Towards this goal, recently Yan *et al.* [22] proposed an itemset summarization approach, namely *pattern profile*. The authors demonstrate that the approach can effectively summarize itemset patterns on a wide variety of datasets, resulting in good compression while retaining high accuracy.

However, the approach has several limitations in our opinion. First, from an efficiency perspective it is not clear how well this approach will scale to large datasets. Essentially the proposed strategy needs to repeatedly scanning the original dataset to achieve good summarization quality, which would become very expensive when dealing with large datasets. Although approximate profiles can be used to limit scans during the summarization process, there will be a significant effect on summarization quality [22]. Finally, the resulting pattern profiles can be quite unbalanced in terms of their size and distribution. There is no way to control this and it can result in poor interpretability.

In this paper we present an approach to summarizing itemset patterns obtained from frequent itemset mining algorithms. The objective can be stated as follows. Given a collection of frequent itemset patterns, compute a concise summary representation such that

the list of frequent itemsets as well as there associated frequency counts can be computed (reasonably) accurately. The approach we present relies on some well established ideas in probabilistic graphical models and probabilistic inference. The key idea is to derive a probabilistic graphical model summary of the data from the set of frequent non-derivable patterns. This would serve as the *profile summary* of the dataset. Subsequently the list of frequent itemsets and associated counts can be computed using probabilistic inference methods available in the literature.

The new summarization scheme yields a much more condensed representation of frequent itemset patterns. The resultant representative itemset patterns can be thought of as generalized non-derivable patterns. This is a very nice property considering that in many cases, non-derivable patterns can already give a much condensed representation of the original collection of frequent itemsets. Furthermore, there is no need to scan the original dataset during the summarization. We only rely on the information of the itemsets to summarize themselves. This is desirable for truly large datasets where repeated scans can be very expensive.

Our experimental results show that the proposed approach compares favorably with the recently proposed idea of profile patterns on the axes of accuracy, space and performance. Specifically we find on real datasets that the proposed approach is much more accurate given the same space budget, and under certain conditions (dataset properties and user-controlled parameters) faster than profile patterns.

The rest of the paper is organized as follows. We state the problem of the itemset pattern summarization and briefly go over the related probabilistic graphical model in Section 2. In Section 3 we detail our proposed probabilistic model based itemset summarization approach. We present experimental results in Section 4 and related work in Section 5. Finally, we discuss future work and conclude in Section 6.

2. PROBLEM STATEMENT AND BACKGROUND

Let I be a set of items, i_1, i_2, ..., i_d. A subset of I is called an *itemset*. The *size* of an itemset is the number of items it contains. A transactional dataset is a collection of itemsets, $D = \{t_1, t_2, \ldots, t_n\}$, where $t_i \subseteq I$. For any itemset α, we write the transactions that contain α as $D_\alpha = \{t_i | \alpha \subseteq t_i \; and \; t_i \in D\}$. In the probabilistic model context, each item corresponds to a distinct random variable[1].

Definition 1. (Frequent itemset). For a transactional dataset D, an itemset α is frequent if $|D_\alpha| \geq \sigma$, where $|D_\alpha|$ is called the support of α in D, denoted as $s(\alpha)$, and σ is a user-specified non-negative threshold.

Frequent itemsets satisfy the important *Apriori* property: any subset of a frequent itemset is also frequent. All existing frequent itemset mining algorithms rely on this property to prune the search space. Since the number of subsets of a large frequent itemset is explosive, it is more appropriate to mine closed frequent itemsets or non-derivable frequent itemsets only. We define these below.

Definition 2. (Closed frequent itemset). A frequent itemset α is closed if there does not exist an itemset β such that $\alpha \subset \beta$ and $D_\alpha = D_\beta$.

[1]In this article we use these terms – item, (random) variable – interchangeably

Definition 3. ((Non-)derivable frequent itemset). A frequent itemset α is $derivable$ if its support can be exactly inferred from its sub-itemsets and their supports based on the inclusion-exclusion principle. Otherwise it is $non-derivable$.

We refer the readers to [5] for more information about non-derivable frequent itemsets and the inclusion-exclusion principle.

2.1 Itemset Pattern Summarization

The itemset pattern summarization problem is formally stated as follows: given a collection of frequent itemset patterns, we want to find a more concise representation such that the original collection of itemset patterns and their supports information can be reasonably recovered. Additionally, the summarization should be tunable in terms of controlling the trade-off amongst often competing metrics, namely summarization quality, summary size or compactness, and efficiency.

As noted above closed itemsets and non-derivable itemsets have been shown to be two successful concise representations of a collection of frequent itemsets. In many cases they can significantly reduce the number of itemsets in the representation without information loss. However, sometimes they can be still quite large which is why new summarization schemes must be found.

2.2 Markov Random Field (MRF) Model

An MRF model is an undirected graphical model in which vertices represent variables and edges represent correlations between variables. The joint distribution associated with an undirected graphical model can be factorized as follows:

$$p(X) = \frac{1}{Z(\psi)} \prod_{C_i \in \mathcal{C}} \psi_{C_i}(X_{C_i})$$

where $\mathcal{C}$ is the set of maximal cliques associated with the undirected graph; ψ_{Ci} is a potential function over the variables of clique C_i and $\frac{1}{Z(\psi)}$ is a normalization term to ensure a valid distribution. A clique is a subset of vertices in the graph that are fully-connected. A maximal clique is a clique that cannot have more vertices added and remain a valid clique. We associate with each maximal clique a non-negative and real-valued potential function.

The MRF model fully specifies the conditional independence among variables. The Markov property states that for all disjoint vertex subsets a, b and c in the graphical model, whenever b and c are separated by a in the graph, then the random variables associated with b, c are independent given the random variables associated with a alone.

2.2.1 Using Frequent Itemset Patterns to Construct an MRF Model

The idea of using frequent itemset patterns to construct an MRF model was first described by Pavlov *et al.* [18] in their work on query selectivity estimation on sparse binary data. Essentially given a query whose selectivity is to be estimated they first identify subset patterns of the query that are frequent. Each such pattern is taken as a particular constraint on the true underlying distribution which generates the data. Among all feasible distributions satisfying these constraints, the one with the maximal entropy ("as uninformed as possible") is picked as the estimate for the true distribution. They show that this maximum entropy distribution is equivalent to an MRF model. There is a simple algorithm, called *iterative scaling* that one can use to learn an MRF model from a given set of itemset patterns. It has been shown [18] that this MRF model is very effective in estimating the selectivity of queries.

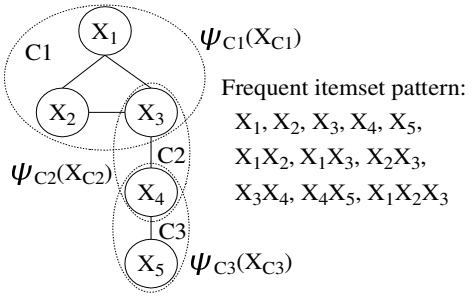


Figure 1: An MRF Example

The following gives an example of the maximal entropy distribution and the corresponding MRF model from a set of itemsets. Suppose we have collected the itemsets as x_1, x_2, x_3, x_4, x_5, x_1x_2, x_1x_3, x_2x_3, x_3x_4, x_4x_5 and $x_1x_2x_3$. Let x_Q be $\{x_1, x_2, x_3, x_4, x_5\}$. Then the maximal entropy distribution on x_Q has the following product form:

$$p(x_Q) = \mu_0 \cdot \mu_1^{I(x_1=1)} \cdot \mu_2^{I(x_2=1)} \cdot \mu_3^{I(x_3=1)} \cdot \mu_4^{I(x_4=1)} \cdot \mu_5^{I(x_5=1)} \cdot \mu_6^{I(x_1=x_2=1)} \cdot \mu_7^{I(x_1=x_3=1)} \cdot \mu_8^{I(x_2=x_3=1)} \cdot \mu_9^{I(x_3=x_4=1)} \cdot \mu_{10}^{I(x_4=x_5=1)} \cdot \mu_{11}^{I(x_1=x_1=x_3=1)}$$

where $I()$ is an indication function for the corresponding constraint and the constants μ_0, ..., μ_{11} are estimated from the data. Figure 1 shows the corresponding MRF model. In particular,

$$\psi_{C_1}(X_{C_1}) = \mu_1^{I(x_1=1)} \cdot \mu_2^{I(x_2=1)} \cdot \mu_3^{I(x_3=1)} \cdot \mu_6^{I(x_1=x_2=1)} \cdot \mu_7^{I(x_1=x_3=1)} \cdot \mu_8^{I(x_2=x_3=1)} \cdot \mu_{11}^{I(x_1=x_1=x_3=1)};$$

$$\psi_{C_2}(X_{C_2}) = \mu_4^{I(x_4=1)} \cdot \mu_9^{I(x_3=x_4=1)};$$

$$\psi_{C_3}(X_{C_3}) = \mu_5^{I(x_5=1)} \cdot \mu_{10}^{I(x_4=x_5=1)}$$

While our work is clearly inspired by the work by Pavlov *et al.* [18], there are several key differences between the two – we will outline these in the ensuing sections.

3. ALGORITHM

Our goal is to use a smaller number of itemset patterns to construct an MRF model over all involved variables and then use it to summarize a much larger collection of the frequent itemset patterns.

LEMMA 1. *Given a transactional dataset D, the MRF model M constructed based on all of its σ-frequent itemset patterns is exactly the same as M', the MRF model constructed based on all of its σ-frequent non-derivable itemset patterns only.*

Proof sketch: This is due to the universal applicability of the inclusion-exclusion principle. When we use all σ-frequent non-derivable itemsets to construct an MRF model, the model will keep the exact support information for these itemsets. Later when we use the model to infer the support for other itemset patterns, the deduction has to satisfy the inclusion-exclusion principle, thus yielding exact estimations.

LEMMA 2. *Given an itemset pattern α and all of its non-derivable sub-itemsets and their support estimations, i.e., $\hat{s_1}$, ..., $\hat{s_l}$ (true supports are s_1, ..., s_l, respectively). If these estimations are error bounded by e_1, ..., e_l, i.e., $|\hat{s_1} - s_1| \leq e_1$, ..., $|\hat{s_l} - s_l| \leq e_l$, then the support estimation for α, $\hat{s}(\alpha)$, derived from these sub-itemsets is error bounded by $e_1 + e_2 + \ldots + e_l$.*

Proof sketch: We infer the support of the itemset by applying the inclusion-exclusion principle based on its sub-itemsets' support. If a sub-itemset is non-derivable, we use its support estimation directly, otherwise we recursively apply the inclusion-exclusion principle to derive its support estimation. It's easy to see that $e_1 + e_2 + \ldots + e_l$ is the maximally possible error accumulation.

Motivated by the above lemmas, we focus on the task of summarizing non-derivable itemset patterns. These patterns capture non-redundant distribution information of the data according to Lemma 1. Furthermore, if we summarize these patterns well, the summarization quality for all other derivable patterns will be error bounded according to Lemma 2.

3.1 Summarizing Itemset Patterns Using MRF Models

The basic idea of our proposed approach is very simple. We use statistics of smaller itemset patterns to construct an MRF model, and then use this model to infer the supports of larger itemset patterns thereafter. If the estimations are accurate enough (within a user-specified error tolerance), we bypass the corresponding patterns. Otherwise we will use the extra information from these itemsets to augment the model. The summarization proceeds in a level-wise fashion. First, all 1-itemsets are collected and used to construct an MRF. Then we infer the supports for all 2-itemsets. We bypass those 2-itemsets whose supports are well estimated from the model and use the information of all skewed 2-itemsets to augment the model. Next, we move on to process all 3-itemsets and so on. This process will be repeated level by level until we process all the itemset patterns. At the end of the process, all itemsets remaining in the resulting model afford a concise representation of the original collection of itemset patterns.

Essentially we keep picking out the skewed itemset patterns and add their information to the probabilistic model. Thus we expect that the final resulting model to be able to faithfully capture the most significant dependency information in the data, and therefore summarize the original patterns well. From another angle, we try to squash the original collection of itemset patterns by eliminating redundancy from it. As we know, the MRF model fully specifies the conditional independence in the data, thus if an itemset pattern does not introduce any extra significant dependency information to the current model, it will be pruned. Furthermore, we introduce a parameter δ to tune in the granularity of the summarization. δ specifies the error tolerance during the summarization. If the estimation error is within the tolerance, we bypass the corresponding pattern. Otherwise we take it as skewed. Through specifying the error tolerance, δ provides a mechanism to trade-off between summarization accuracy and space budget.

The formal summarization algorithm is presented in Figure 2. Note that this algorithm can be easily extended to a more general summarization scheme, that can be applied to summarize any collection of itemset patterns. In our study, we do not pursue this direction and we focus on summarizing a complete collection of σ-frequent itemset patterns. The time complexity of the summarization algorithm is dominated by the MRF model learning process, which we will describe below.

3.2 Learning MRF Models

As been mentioned in Section 2, the iterative scaling algorithm can be used to learn an MRF from a set of itemsets. Figure 3 presents a high-level outline of a computationally efficient version of the algorithm given by Jelinek *et al.* [12]. During the learning process, we need to repeatedly update the model to force it to satisfy the current itemset constraint. The model updating relies on

Algorithm: Itemset Pattern Summarization Algorithm (C, δ)
Input: Collection of itemset patterns C;
Error tolerance threshold δ;
Output: Reduced collection of itemset patterns R;
1. Obtain all 1-itemset patterns in C
and their supports; Use them to initialize R;
2. $k = 2$;
3.**While** $k < MAX\_LEVEL$
4. Use itemsets in R to construct an MRF model M;
5. Obtain all k-itemset patterns in C and their supports;
6. **For** each k-itemset pattern p:
7. Estimate $s(p)$ and calculate the estimation error e;
8. **if** $e > \delta$ **then** add p to R;
7. $k++$;
8. return R;

Figure 2: Itemset pattern summarization algorithm

Algorithm: Learning MRF using itemsets (C)
Input: Collection of itemsets C;
Output: MRF model M;
1. Obtain all involved variables v
and choose an initial approximation to M
(typically uniform over v);
2. **While** (Not all constraints are satisfied)
3. **For** (each constraint c_i)
4. Update M to force it to satisfy c_i;
5. return M;

Figure 3: Iterative Scaling Algorithm

the support estimation for the current itemset constraint. Thus we need to keep making inferences on the current model. If the iterative scaling algorithm runs k iterations and there are m itemset constraints, the time complexity of the algorithm will be $O(k \times m \times t)$, where t is the average inference time over a constraint. Thus the efficient inference is crucial to the running time of the learning algorithm. In our study, we exploit two inference engines, the *Junction Tree* inference algorithm and the *Markov Chain Monte Carlo* (MCMC) inference algorithm.

3.2.1 Junction Tree Inference Algorithm

The junction tree algorithm is a general exact probabilistic inference framework. The general problem here is to calculate the marginal probability of a variable or a set of variables, given the observed values of another set of variables. In our context, there is no observed variables, and our goal is to calculate the marginal probability associated with the itemset patterns. The idea of the junction tree algorithm is to find a way to decompose a global computation on a joint probability into a linked set of local computations. The key point of this approach is the concept of locality. The junction tree is a particular data structure which fully exploits the graph-theoretic locality for efficient probabilistic inference.

Specifically, the junction tree algorithm decomposes the original model into a hyper-tree, in which each node consists of a set of variables in the original model. Two sets of variables associated with two tree nodes could overlap, and the overlapped part is called the *separator* of the two tree nodes. Particularly, each tree node corresponds to a unique maximum clique in the graph formed by triangulating the original model. Furthermore, the junction tree needs to satisfy the *running intersection* property, i.e., for every pair of cliques V and W, all cliques on the path between V and W contain $V \cap W$. Beliefs of the tree nodes propagate along all distinct paths respecting a two-phase message passing protocol in the junction tree. When the propagation terminates, the clique potentials and separator potentials are proportional to local marginal probabilities. In other words, global consistency is achieved, which implies that the inference problem within a tree node can be solved independently of the other tree nodes.

The time complexity of the junction tree algorithm is determined by the maximum number of variables a tree node contains in the junction tree (also known as the *treewidth* of the original graphical model). More specifically, the time complexity is exponential in the treewidth of the model. If the underlying MRF model is relatively simple (with a relatively low treewidth), then the junction tree algorithm can yield exact inferences very efficiently. On the other hand, when the model becomes complex (the treewidth becomes large), then the exact inference will become slow, sometimes even become intractable in which cases we have to resort to approximate inference algorithms.

3.2.2 MCMC Inference Algorithm

MCMC is a general method for simulating from complicated distributions [7]. In our study, we used a particular type of MCMC algorithm known as a *Gibbs sampler* to draw dependent samples from the joint posterior distribution from which we evaluate the marginal probabilities corresponding to the itemset patterns. Specifically, we specify a full conditional distribution $p(xi|-xi)$ for each variable x_i in our MRF model, where $-x_i$ is the set of variables in the graph not including x_i. Then we draw samples from it. Note that the Markov property indicates that it suffices to condition on the neighboring variables of x_i in the MRF model. The Gibbs sampling proceeds by sampling each hidden variable from the conditional distribution, given the current values of the other variables in the graph. Marginal probabilities can be estimated by summing over the samples. Note that the Gibbs sampling scheme yields approximate inferences. The quality of the approximate inference is usually reasonably good when the sample size is large enough. Additionally, to diminish the effect of the starting distribution, we generally discard certain amount of early iterations, referred to as *burn in* [7]. We base on the iterations later to make the inference.

3.3 Generalized Non-derivable Itemsets

The probabilistic model based summarization scheme returns a subset of the original collection of itemsets as its summary to the end-user. Similar to what pointed out in [22] that pattern profiles can be viewed as generalized closed itemsets, the resulting itemsets in our summarization approach can be viewed as generalized non-derivable itemsets. First, we construct the probabilistic models based on the non-derivable itemsets. As a result, all the itemsets in the final summary are non-derivable. Second, we allow certain error tolerance when summarizing the itemset patterns. If a particular itemset respects the currently known conditional independence structure specified by the model, we take its support as known. Note that it might be the case that we are not able to derive an itemset's support based on the inclusion-exclusion principle only, however we are able to derive it according to the further conditional independence information. In contrast, previously an itemset is derivable only when its support can be completely determined from the support of its sub-itemsets based on the inclusion-exclusion principle only. We see that essentially we relax the requirement for an itemset to be "derivable", which will significantly increase the number of derivable patterns. Furthermore, the more

	k	n	m	d
Chess	75	3196	118252	0.493
Accidents	468	340183	11500870	0.0722
Mushroom	119	8124	186852	0.193
Web	294	32711	98654	0.0102

Table 1: General characteristics of the datasets. k is the number of distinct items, n is the number of records, m is the number of total items and $d = \frac{m}{kn}$ is the density index.

relaxation we allow, the more derivable patterns will be, implying a more compressed summarization.

4. EXPERIMENTAL RESULTS

In this section, we examine the performance of our proposed approach on real datasets. We compare the probabilistic model based summarization approach (abbreviated as PM in figures presenting experimental results) against the state-of-the-art pattern profile summarization scheme (abbreviated as PP in figures presenting experimental results). The summarization algorithm is implemented in C++. The junction tree and Gibbs sampling inference algorithms are implemented based on Intel's Open-Source Probabilistic Networks Library[2]. Also, we implement the pattern profile summarization algorithm in C++ and we tune in it to achieve similar performance to that reported in [22].

4.1 Experimental Setup

All the experiments are conducted on a Pentium 4 2.66GHz machine with 1GB RAM running Linux 2.6.8. We use the implementation of apriori algorithm in [4] to collect the σ-frequent itemsets and the corresponding closed itemsets. We use the implementation in [6] to collect all σ-frequent non-derivable itemsets. Below we detail the datasets and performance metrics considered in our evaluation.

Datasets: We use four publicly available datasets in our experiments: the Chess dataset with 3196 transactions and 75 distinct items; the Accidents dataset with 340183 transactions and 468 distinct items; the Mushroom dataset with 8124 transactions and 119 distinct items; the Microsoft Anonymous Web dataset with 32711 transactions and 294 distinct items. The first three datasets are publicly available at the FIMI repository [3] and the last Web dataset is publicly available at the UCI KDD archive [4]. The main characteristics of the datasets are summarized in Table 1. As can be seen, the Chess and Mushroom datasets are relatively dense. The Web dataset is the sparsest one and the Accidents dataset lies in between.

Performance Metrics:

- Summarization accuracy.

  Definition 4. Restoration error. Given a collection of itemset patterns $\Phi = \{\alpha_1, \alpha_2, \ldots, \alpha_l\}$, the quality of a pattern summarization can be evaluated by the following average relative error (called restoration error),
  $E = \sum_{\alpha_k \in \Phi} \frac{|s(\alpha_k) - \hat{s}(\alpha_k)|}{s(\alpha_k)}$

  where s is the true support and $\hat{s}$ is the estimated support. Restoration error measures the average relative error between the estimated support of a pattern and its true support. If this measure is small enough, it means that the estimated support of a pattern is very close to its true support.

- Summary size. In order to compare fairly between different summarization schemes, we need to consider the summary size. The comparison should be made between summarizations which are of the same size. A larger summary is expected to be more accurate. Overall, we prefer the summarizations with low sizes which however yield small restoration errors. In our study we calculate the number of bytes taken by a summarization and use it to quantify its size. Specifically, an item in the summary takes 2 bytes (a short integer) and a floating point number in the summary takes 4 bytes. For example, the following itemset pattern takes 8 bytes, $\{(item_1, item_2), 0.1\}$ and the following pattern profile takes 22 bytes. $\{(item_1, item_2, item_3), (1.0, 0.8, 0.6), (0.1)\}$

- Summarization time. We consider the time taken to summarize the itemset patterns. Ideally, the summarization should be fast. We have to point out that a fair timing performance comparison between different summarization schemes is not easy. For example, both our approach and pattern profile approach are iterative processes. The running times are highly dependent of the convergence criteria, which could be rather subjective. Here we report the timing performance results of those summarizations from which we collect the summarization accuracy results.

4.2 Results on the Chess Dataset

In this section, we report the experimental results on the Chess dataset. For this set of experiments, we set $\sigma = 2000$ to collect the frequent itemset patterns. As a result there are 166581 frequent itemsets, from which 1276 itemsets are non-derivable. We also collect all the 68967 closed frequent itemsets at this support level for the pattern profile summarization scheme.

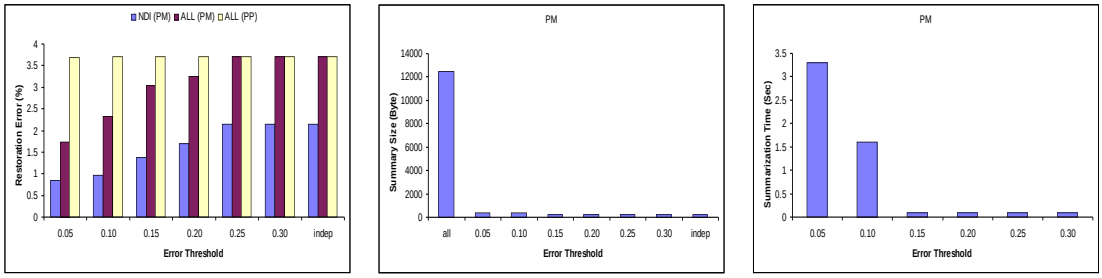
Figure 4a presents the summarization quality as we vary the error tolerance threshold used during the summarization. Specifically, for our approach we report both results on summarizing all itemset patterns and all non-derivable itemset patterns. For the pattern profile approach, we only report the results on summarizing all itemset patterns. For the reference purpose, the results based on naive independence model are also plotted in the figure.

From the figure, we see that the probabilistic model based summarization scheme effectively summarizes the itemset patterns. The restoration error for all frequent patterns is slightly worse than that for non-derivable patterns. This is as expected considering our approach particularly focuses on summarizing non-derivable patterns. It's worth pointing out that the restoration error on all frequent itemset patterns is also very small, thus supporting our claim that non-derivable patterns play a key role in representing the whole collection of frequent itemset patterns.

Furthermore, it can be clearly seen that the restoration error increases as we raise the error tolerance threshold. This is due to the fact that we will lose more distribution information with larger error tolerance thresholds. Particularly, the summarization with the threshold above 0.25 becomes equivalent to the naive independence model based summarization. The advantage of the new approach over the pattern profile approach is clearly demonstrated in the figure. For the pattern profiles of the same size, the restoration error is much higher than that of the new approach and is actually quite close to that of the naive independence model.

Table 2 presents the distribution of the skewed itemsets at different levels with respect to different error tolerance thresholds. As can be seen from the table, the numbers of skewed itemsets are very

[2]https://sourceforge.net/projects/openpnl/
[3]http://fimi.cs.helsinki.fi/
[4]http://kdd.ics.uci.edu/



(a) (b) (c)

Figure 4: Results on the Chess dataset:(a)Restoration error (b)Summary size (c)Summarization time

Itemset Size	No. of Total Itemsets	No. of Skewed Itemsets (Varying δ)					
		0.05	0.10	0.15	0.20	0.25	0.30
1	31	31	31	31	31	31	31
2	335	6	0	0	0	0	0
3	653	14	19	2	1	0	0
4	257	2	0	0	0	0	0
Sum	1276	53	50	33	32	31	31

Table 2: Skewed itemset distribution on the Chess dataset when varying error threshold

small at all error tolerance thresholds. For example, at the error tolerance threshold 0.05, there are 6, 14, 2 skewed 2, 3, 4-itemset patterns respectively. As we raise the threshold, overall the number of skewed itemsets decreases.

Figure 4b presents the summary sizes with different error tolerance thresholds. Specifically, the sizes of the original collection of patterns and the naive independence model are also plotted here for the purpose of reference. As can be seen, our summaries use a very small amount of space to summarize a much larger collection of itemset patterns. For example, the summary takes 398 bytes at the error threshold of 0.05 to summarize the itemset patterns of size 12480 bytes.

Figure 4c presents the timing performance of the new approach. As can be seen, the new approach summarizes the itemset patterns extremely fast on this dataset. In all cases, the summarization takes less than 5 seconds. In contrast, the pattern profile approach does not finish before it exhausts memory. We submit the summarization job to the supercomputer at the Ohio Supercomputer Center (OSC)[5]. The pattern profile approach takes about 40 minutes to finish there. Also, we see that the new approach takes more time when using a lower error tolerance threshold, since the models with lower error tolerance thresholds are more complex.

It's worth noting that the Chess dataset satisfies the independence assumption quite well. Thus the MRF model based summarization scheme works extremely well. A relatively simple MRF model is able to faithfully capture the conditional independence existing in the data, which results in a very low restoration error and an extremely fast summarization.

4.3 Results on the Accidents Dataset

In this section, we report the experimental results on the Accidents dataset. In this set of experiments, we set $\sigma = 150000$ to collect the frequent itemset patterns, which results in 18175 frequent itemsets, out of which 18175 are closed patterns and 5486 are non-derivable patterns.

Figure 5a presents the summarization quality as we vary the error tolerance thresholds used during the summarization process. From the figure, we see that the probabilistic model based summarization scheme works extremely well on this dataset as well. The restoration errors for both all frequent patterns and non-derivable patterns are very low. We note that the independence assumption is satisfied well on this dataset also (the naive independence model yields the error of 5.27% and 6.77% for all patterns and non-derivable patterns respectively). Furthermore, it can be clearly seen that the restoration error increases as we increase the error tolerance threshold.

Note that the pattern profile approach can not deal with this dataset due to its large size. The algorithm run out of memory after running for hours, even on the supercomputer at OSC. Repeatedly dataset scanning based summarization is very computation and memory intensive.

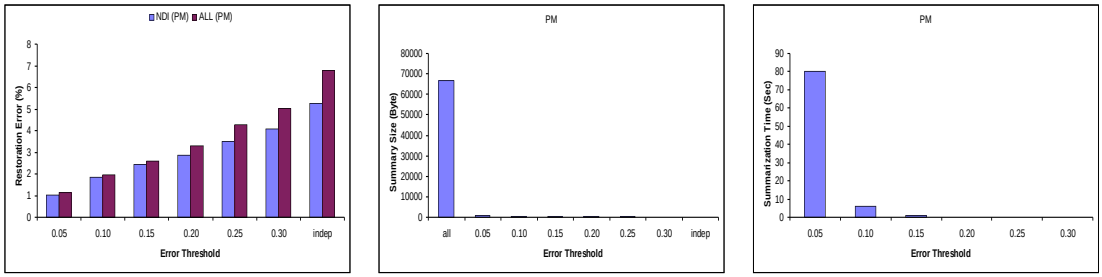
Table 3 presents the distribution of the skewed itemsets. As can be seen from the table, the numbers of skewed itemsets are also very small on this dataset, indicating that the MRF model captures the distribution information and represents all the itemsets quite well. For example, at the error tolerance threshold of 0.05, there are only 13, 54, 1, 3 skewed 2, 3, 4, 5-itemset patterns respectively. Compared with the numbers of the original non-derivable itemsets, which are 253, 1071, 2135, 1788 respectively, the numbers of skewed itemsets are much smaller. Again, as we raise the error tolerance threshold, overall the numbers of skewed itemsets decrease.

Figure 5b presents the summary sizes with different error tolerance thresholds. Again, the summary sizes are much smaller than the size of the original itemset patterns.

Figure 5c presents the timing performance of the new approach. As can be seen, the probabilistic model based approach again summarizes all the itemset patterns on this dataset very fast. Furthermore, the summarization with a smaller error tolerance threshold takes much more time. For example, the summarization takes 80 seconds when the threshold is 0.05. In contrast, it takes less than 1 second when the threshold is above 0.2.

Note that both of the Accidents dataset and the Chess dataset are relatively dense and satisfy the independence assumption well. For this kind of datasets, the MRF model based summarization scheme works extremely well. Interestingly, we note that on these two datasets, the frequent non-derivable patterns is much less than the frequent closed patterns. Take the Accidents dataset as an ex-

[5]http://www.osc.edu/



(a) (b) (c)

Figure 5: Results on the Accidents dataset:(a)Restoration error (b)Summary size (c)Summarization time

Itemset Size	No. of Total Itemsets	No. of Skewed Itemsets (Varying δ)					
		0.05	0.10	0.15	0.20	0.25	0.30
1	28	28	28	28	28	28	28
2	253	13	4	2	0	0	0
3	1071	54	9	6	4	1	0
4	2135	1	5	0	2	3	3
5	1788	3	0	1	0	0	0
6	210	0	0	0	0	0	0
7	1	0	0	0	0	0	0
Sum	5486	99	46	37	34	32	31

Table 3: Skewed itemset distribution on the Accidents dataset when varying error threshold

ample, none of its 18175 frequent patterns is closed. In contrast, only 5486 out of 18175 patterns are non-derivable. This is usually a good sign that the probabilistic model based summarization will be more efficient and effective than the pattern profile approach. Next we will examine the performance of the new approach on the skewed datasets that do not satisfy the independence assumption well.

4.4 Results on the Mushroom Dataset

In this section, we report the experimental results on the Mushroom dataset. The Mushroom dataset is also a relatively dense dataset. In this set of experiments, we set $\sigma = 2031$ (a support threshold of 25%) to collect the frequent itemset patterns, resulting in 5545 frequent itemsets, from which 688 are closed and 534 are non-derivable.

Figure 6a presents the summarization quality as we vary the error tolerance thresholds used during the summarization. From the figure, we see that the independence assumption does not hold well on this dataset. The restoration errors for all itemset patterns and non-derivable itemset patterns are 20% and 39% respectively. From the figure, we see that the probabilistic model based summarization scheme again works very well on this dataset. The restoration errors for both all frequent patterns and non-derivable patterns are reasonably low, and are much lower than that of the pattern profile summaries of the same size. Note that both approaches work much better than the naive independence model. Furthermore, we can lower the restoration error by lowering the error tolerance thresholds used during the summarization, which is at the cost of more space usage.

Table 4 presents the distribution of the skewed itemsets. Compared with the previous two datasets, the proportion of the skewed itemsets is much higher on this dataset, which signifies that the independence assumption does not hold on this dataset as well as that on the other two datasets. But overall, the numbers of skewed itemset patterns are still much less than the numbers of all the original itemset patterns. Particularly on this dataset, there is a small fluctuation when we raise the error tolerance threshold from 0.30 to 0.40, which leads to more skewed itemsets. This is because that the larger threshold 0.40 results in much more skewed 3-itemsets (46 v.s. 25), though it indeed results in less skewed 2-itemsets (11 v.s. 25). The increase of the former outweighs the decrease of the latter, resulting in more skewed itemsets overall. However, the overall trend is still that the numbers of skewed itemsets become less when we use larger error tolerance thresholds.

Figure 6b presents the summary sizes with different error tolerance thresholds. We note that the summaries take relatively more space, compared with that on the previous two datasets. Again, with a lower error tolerance threshold, the summary size is larger.

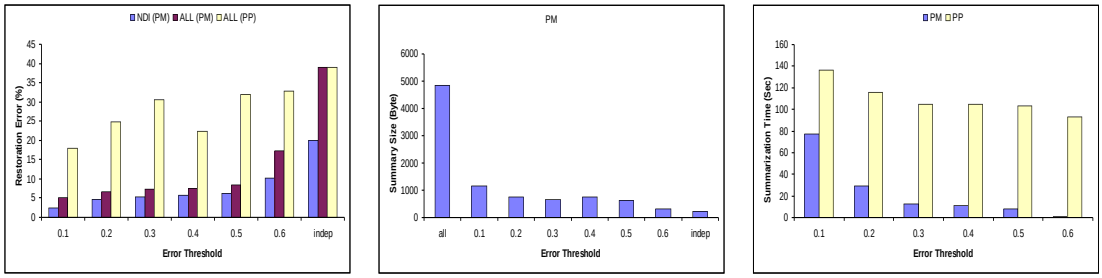
Figure 6c presents the timing performance of the two approaches. The new approach is much faster than the pattern profile approach. We see that both approaches take more time when the error tolerance threshold decreases. However, for the pattern profile approach, the increase of the running time is not as significant.

4.5 Results on the Microsoft Web Dataset

In this section, we report the experimental results on the Microsoft Web dataset, which is the sparsest dataset. For the sparse datasets, the independence assumption generally can not hold, since all single item patterns have very low support in sparse data, and any pattern containing more than one item is prone to have the support of 0 if the dataset follows independence assumption. In this set of experiments, we set $\sigma = 100$ to collect the frequent itemset patterns, resulting in 998 frequent itemsets. Specifically, all itemset patterns are closed and non-derivable, which is a common phenomena on sparse datasets. This also makes the summarization task more difficult since there does not exist much redundancy among the itemset patterns.

Figure 7a presents the summarization quality as we vary the error tolerance thresholds. Note that since that all itemsets are non-derivable, there is no separate results for non-derivable itemsets here. From the figure, we see that the independence assumption indeed does not hold on this dataset. The restoration error for all itemset patterns is 57.5%.

From the figure, we see that the probabilistic model based summarization scheme works reasonably well on this dataset. The restoration error is reasonably low. When we use 283 patterns (less

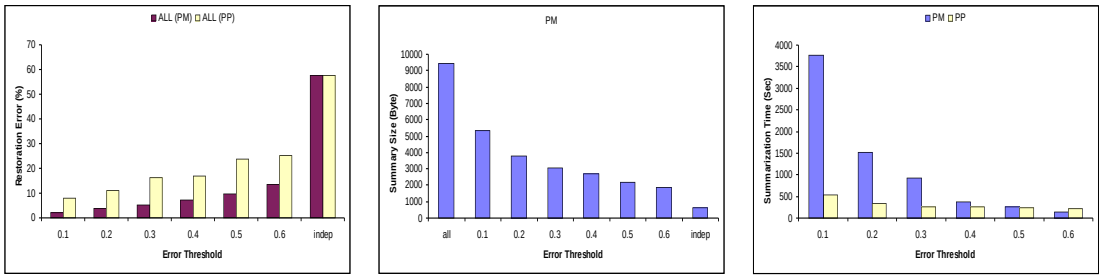


(a) (b) (c)

Figure 6: Results on the Mushroom dataset:(a)Restoration error (b)Summary size (c)Summarization time

Itemset Size	No. of Total Itemsets	No. of Skewed Itemsets (Varying δ)					
		0.10	0.20	0.30	0.40	0.50	0.60
1	35	35	35	35	35	35	35
2	207	78	42	25	11	2	1
3	269	31	19	25	46	40	8
4	23	0	0	0	0	0	1
Sum	534	144	96	85	92	77	45

Table 4: Skewed itemset distribution on the Mushroom dataset when varying error threshold



(a) (b) (c)

Figure 7: Results on the Web dataset:(a)Restoration error (b)Summary size (c)Summarization time

Itemset Size	No. of Total Itemsets	No. of Skewed Itemsets (Varying δ)					
		0.10	0.20	0.30	0.40	0.50	0.60
1	104	104	104	104	104	104	104
2	329	287	249	202	160	123	102
3	346	192	108	72	66	47	29
4	168	39	8	7	13	9	11
5	49	1	1	2	0	0	0
6	2	0	0	0	0	0	0
Sum	998	623	470	387	343	283	246

Table 5: Skewed itemset distribution on the Web dataset when varying error threshold

than one third of all patterns, $\delta = 0.5$), the restoration error is 9.62%. When we use 470 patterns (less than half of all patterns, $\delta = 0.2$), the restoration error is reduced to 3.72%. Furthermore, the new approach consistently outperforms pattern profile approach in terms of restoration error. Again, both approaches work much better than the naive independence model.

Table 5 presents the distribution of the skewed itemsets. The proportion of the skewed itemsets is relatively high on this dataset, which is similar to that on the Mushroom dataset. Still we are able to reduce the numbers of skewed itemset patterns by raising the error tolerance threshold.

Figure 7b presents the summary sizes with different error tolerance thresholds. We see that the summarizations take relatively much more space on this dataset than that on the Chess and Accidents datasets.

Figure 7c presents the timing performance of the two approaches. We see on this dataset, the new approach is overall much slower than the pattern profile approach. This is due to the complexity of the underlying MRF models. We know that for this dataset, the proportion of skewed patterns is relatively high, resulting in more complex models, especially when the error tolerance threshold is small. Consequently, the summarization becomes much slower. For example, when the error tolerance threshold of 0.1 is used, the summarization takes more than 1 hour. In contrast, the pattern profile approach takes less than 10 minutes.

4.6 Results on Approximate Inference Based Summarization

In this section, we report the results on the approximate inference based summarization. We focus on the comparison of the summarization quality between the exact inference based summarization and the approximate inference based summarization. To this end, we use the approximate inference based approach to summarize the same collections of frequent itemset patterns as the previous sets of experiments. We report the results on the Mushroom dataset. The other results are similar and are thus omitted. Specifically, we set the sample size to be 4000 and the first 10% of the sample is used as burn in data when we use the Gibbs sampling inference algorithm.

Figure 8a presents the numbers of itemset patterns in the resulting summarizations. As can be seen, the approximate inference based summarization scheme usually yields more patterns at the end for the same parametric setting. This is as expected since its support estimating is not as accurate as that of the exact inference based summarization scheme. Consequently there will be more skewed itemsets identified and placed into the model during the summarization. But we see that the difference is not significant.

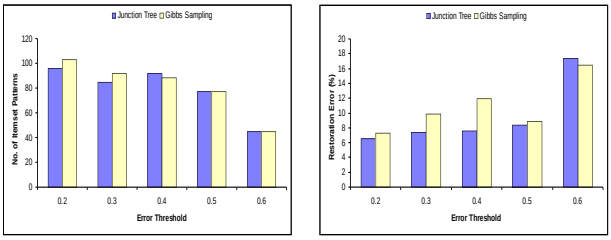
Figure 8b presents the restoration errors of the approximate inference based summarizations. As seen, the approximate inference based scheme usually yields larger restoration errors, which is as expected. But overall, the approximate inference based scheme still yields reasonably good summarizations. It significantly outperforms the pattern profile approach.

The approximate inference based summarization scheme takes hours to finish. It's worth pointing out that on these datasets, there is no need to use the approximate inference based summarization scheme. It yields worse summarization using much more time. However, we just want to show that the approximate inference based summarization scheme can yield comparable summarizations as that of the exact inference based summarization scheme. When the underlying MRF model becomes more complex (the treewidth becomes larger), we have to use the approximate inference based summarization scheme, and we are currently pursuing different approximate inference algorithms besides the Gibbs sampling approach.

4.7 Result Summary and Discussion

The experimental results have shown that the probabilistic model based summarization scheme is overall very efficient and effective in summarizing itemset patterns. In most cases, it outperforms the pattern profile summarization scheme.

We know that our approach is to summarize itemset patterns by eliminating redundancy from them. When datasets are dense and largely satisfy the conditional independence assumption, there usually exists a large amount of redundancy in the corresponding itemset patterns in which case our approach will be extremely efficient and effective. On the other hand, when datasets become sparser and do not satisfy conditional independence assumption well, the summarization task will become more difficult for our approach. As a result, we have to spend more space and time on summarizing the corresponding itemset patterns.



(a) (b)

Figure 8: Approximate inference based summarization on the Mushroom dataset (a)No. of patterns after summarization (b)Restoration error (all patterns)

5. RELATED WORK

There have been significant work on reducing the output size of frequent itemset mining algorithms. Lossless methods can recover the exact information of all of the original itemsets. Two representative methods are frequent closed itemset patterns and frequent non-derivable itemset patterns, proposed by Pasquier *et al.* [17] and Calders *et al.* [5] respectively. Lossy methods were developed in parallel. Gunopulos *et al.* [9] proposed mining maximum itemset patterns. Yang *et al.* [23] and Pei *et al.* [19] proposed mining error-tolerant patterns. Han *et al.* [11] proposed mining Top-k patterns. These methods can further reduce the output size at the cost of some information loss. Mielikainen and Mannila [14] proposed an approximation solution to ordering patterns according to their informativeness. Another related work is a pattern approximation approach which relies on k frequent (or border) itemsets to cover the original collection of frequent itemsets [1]. However, it can only recover the itemset patterns, not their supports.

Recently, Yan *et al.* [22] proposed a pattern summarization approach for frequent itemsets. The key notion is *pattern profile*, which essentially can be taken as a generalization of closed itemsets. Specifically, a pattern profile is a triple $< a, b, c >$ where a is a set of items, b is a distribution vector on these items and c is the support of the whole pattern profile. A frequent itemset is a special pattern profile where the distribution vector entirely consists of 1.0. Essentially, a pattern profile is a compressed representation of similar itemset patterns and can be used for summarizing the itemset patterns. In their proposed summarization scheme, pattern profiles are compared based on their Kullback-Leibler (KL) divergence between their distribution vectors. The first principle is that the pattern profiles having smaller KL divergence are more correlated than that having larger KL divergence. Based on this

similarity measure, the traditional k-means clustering algorithm is applied to cluster the itemsets into K groups. Then a representative profile pattern will be identified for each group and used as a compressed representation for that group of itemsets.

Pavlov *et al.*'s work on query selectivity estimation on binary transactional data [18] is also closely related to the work presented in this paper. They also construct MRF models based on a set of itemset patterns. However, there are significant distinctions between the two pieces of work. First, their goal is to estimate query selectivity. When a query Q with variables x_Q is posed in real-time, all itemset patterns whose variables are subsets of x_Q are picked up as the distribution constraints. Then an MRF model on x_Q is built in an online fashion. Once the model is ready, any conjunctive query whose variables are subset of x_Q can be answered, including x_Q itself as well. When a new query is posed, a new model has to be constructed from scratch. This approach is inherently online and local. In contrast, our goal is to summarize the itemset patterns. Our MRF model is global in that it contains all the conditional independence information known so far. This global character benefits the accurate support estimations for the itemsets to be summarized. However, the global character also makes the model learning and inference much more difficult than that in [18]. For example, learning global models requires approximate inference engines when the models become complex.

In parallel there have been significant work on probabilistic graphical models and approximate inference. Besides the sampling techniques used in this paper, variational methods [13, 20, 24, 3, 21, 16] for approximate inference is a very active research topic. Specifically, the variational methods yield approximations to marginal probabilities via the solution to an optimization problem derived from the corresponding inference problem that generally exploits some of the graphical structure. Mean field methods [13, 20, 21] and Pearl's belief propagation (BP) algorithm [15, 24] (when applied to loopy graphs) are both belonging to this category.

6. CONCLUSIONS

In this paper, we have presented a novel approach for summarizing itemset patterns using probabilistic graphical models – Markov Random Fields. Our approach relies on probabilistic graphical models, which exploit conditional independence relations between the different items in the transactional data to allow a compact representation of all itemset patterns. We have tested our algorithm on several real-world datasets. The success of our approach on all of these datasets indicates that the conditional independence structure exploited by our approach is very common, and this is particularly true when dealing with relatively dense datasets, whose dense structure leads to significant redundancy in mined itemset patterns. As a result, our approach is a viable option for many real-world cases.

Our approach has several important advantages. First, the resulting itemset patterns are very easy to interpret and use, since all of them are frequent itemset patterns as well. Second, the summarization only relies on the information of the itemset patterns themselves, these is no need to go back and rescan the original dataset. Finally, interestingly we note that our summarization approach yields generalized non-derivable itemset patterns.

In the future, we will study the following issues: First, we would like to summarize truly large-scale collections of itemset patterns which will result in very complex MRF models. The Gibbs sampling based inference is time consuming, and the detection of convergence is not easy. We would like to borrow some of the ideas from the probabilistic inference field for our itemset summarization purpose. For example, we would like to exploit mean field algorithms and generalized belief propagation algorithms for our purpose. Second, we would like to integrate both exact and approximate inference engines in the summarization. During the beginning iterations, the exact inference engine will be employed since the MRF models are not very complex. Once the models reach to some complexity (by estimating its treewidth), we will switch to the approximate inference engine. Third, we would like to exploit summarizing itemsets under the streaming environment, which requires incrementally maintaining the model. Finally, we would like to look for some real applications on the condensed representation of itemset patterns.

7. REFERENCES

[1] F. N. Afrati, A. Gionis, and H. Mannila. Approximating a collection of frequent sets. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 12–19, 2004.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, 1994.

[3] C. M. Bishop, D. J. Spiegelhalter, and J. M. Winn. Vibes: A variational inference engine for bayesian networks. In *NIPS 2002*, pages 777–784, 2002.

[4] C. Borgelt. Efficient implementations of apriori and eclat. In *Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations*, 2003.

[5] T. Calders and B. Goethals. Mining all non-derivable frequent itemsets. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, 2002.

[6] T. Calders and B. Goethals. Depth-first non-derivable itemset mining. In *Proceedings of the SIAM 2005 International Conference on Data Mining*, 2005.

[7] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian data analysis*. Chapman & Hall/CRC, 2004.

[8] K. Gouda and M. J. Zaki. Genmax: An efficient algorithm for mining maximal frequent itemsets. *Data Mining and Knowledge Discovery*, 11(3):223–242, November 2005.

[9] D. Gunopulos, R. Khardon, H. Mannila, and H. Toivonen. Data mining, hypergraph transversals, and machine learning. In *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 209–216, 1997.

[10] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pages 1–12, 2000.

[11] J. Han, J. Wang, Y. Lu, and P. Tzvetkov. Mining top-k frequent closed patterns without minimum support. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, pages 211–218, 2002.

[12] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, MA, 1998.

[13] M. I. Jordan, M. J. Kearns, and S. A. Solla. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

[14] T. Mielikainen and H. Mannila. The pattern ordering problem. In *Knowledge Discovery in Databases: PKDD 2003, 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003, Proceedings*, pages 327–338, 2003.

[15] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 467–475, 1999.

[16] I. Murray and Z. Ghahramani. Bayesian learning in undirected graphical models: Approximate mcmc algorithms. In *Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence*, 2004.

[17] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Database Theory - ICDT '99, 7th International Conference, Jerusalem, Israel, January 10-12, 1999, Proceedings*, pages 398–416, 1999.

[18] D. Pavlov, H. Mannila, and P. Smyth. Beyond independence: probabilistic models for query approximation on binary transaction data. *IEEE Transactions on Knowledge and Data Engineering*, 15(6):1409–1421, November 2003.

[19] J. Pei, A. K. H. Tung, and J. Han. Fault-tolerant frequent pattern mining: Problems and challenges. In *2001 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, 2001.

[20] W. Wiegerinck. Variational approximations between mean field theory and the junction tree algorithm. In *Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence*, pages 626–633, 2000.

[21] E. Xing, M. Jordan, and S. Russell. A generalized mean field algorithm for variational inference in exponential families. In *Uncertainty in Artificial Intelligence*, pages 583–591, 2003.

[22] X. Yan, H. Cheng, J. Han, and D. Xin. Summarizing itemset patterns: a profile-based approach. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 314–323, 2005.

[23] C. Yang, U. M. Fayyad, and P. S. Bradley. Efficient discovery of error-tolerant frequent itemsets in high dimensions. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 194–203, 2001.

[24] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In *IJCAI*, 2001.

[25] M. J. Zaki and C.-J. Hsiao. Charm: An efficient algorithm for closed itemset mining. In *Proceedings of the Second SIAM International Conference on Data Mining*, 2002.

[26] M. J. Zaki, S. Parthasarathy, and W. L. Mitsunori Ogihara. New algorithms for fast discovery of association rules. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 283–286, 1997.