Learning Approximate MRFs From Large Transaction Data

Chao Wang and Srinivasan Parthasarathy

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

Abstract. In this paper we consider the problem of learning approximate Markov Random Fields (MRFs) from large transaction data. We rely on frequent itemsets to learn MRFs on the data. Since learning exact large MRFs is generally intractable, we resort to learning approximate MRFs. Our proposed modeling approach first employs graph partitioning to cluster variables into balanced disjoint partitions, and then augments important interactions across partitions to capture interdependencies across them. A novel treewidth based augmentation scheme is proposed to boost performance. We learn an exact local MRF for each partition and then combine all the local MRFs together to derive a global model of the data. A greedy approximate inference scheme is developed on this global model. We demonstrate the use of the learned MRFs on the selectivity estimation problem. Extensive empirical evaluation on real datasets demonstrates the advantage of our approach over extant solutions.

1 Introduction

In this paper we address the problem of learning approximate *Markov Random Fields* (MRF) from large transaction data. Examples of such data include market basket data, web log data, and document or text analysis data. Such data can be represented by a high-dimensional data matrix, with each row corresponding to a particular market basket (web session), and each column corresponding to a particular item (web page). Each entry takes a value of “1” if the corresponding item is in the corresponding basket, otherwise it takes a value of “0”. The data matrix is binary, and very often in such applications, highly sparse in that the number of non-zero entries is small.

To model such data effectively in order to answer queries about the data efficiently, we consider the use of probabilistic models. Probabilistic models capture association or causal correlations among attributes in data and have been successfully applied in applications such as selectivity estimation in query optimization [1–3], link analysis/recommender systems [4, 5] and bioinformatics [6].

Specifically to tackle the selectivity estimation problem, Pavlov *et al.* [3] propose an *Maximum Entropy* (ME) model based on frequent itemsets. Particularly, the ME model has been shown to be equivalent to an MRF and is effective in estimating query selectivity. However, a key limitation of their approach is that it needs to learn a *local model* over query variables on the fly for every query. Due to the fact that inferring an ME model is an iterative process and is usually very expensive, such a just-in-time model construction approach is not appropriate in settings where online estimation time is crucial.

The alternative is to learn a *global model* offline and to use the global model to answer queries on the fly. The advantages are a more accurate model (relies on complete information from all the data) and online performance. The critical challenge is that a global model may be prohibitive to compute for large datasets of high dimension (offline learning cost) . To address this problem, in this paper, we consider the problem of learning approximate global MRFs from large transaction data – the approximation is used to make the offline learning cost tractable. To summarize, the main contributions of this paper are highlighted below.

1. We introduce a novel divide-and-conquer style approach based on graph partitioning to learning approximate MRFs from large transaction data.

2. We introduce a novel interaction importance and treewidth based augmentation scheme to capture interdependencies across partitions.

3. We conduct an extensive empirical study on real datasets to show the efficiency and effectiveness of the new approach.

2 Background

Let $\mathcal{I}$ be a set of items, i_1, i_2, ..., i_d. A subset of $\mathcal{I}$ is called an *itemset*. The *size* of an itemset is the number of items it contains. An itemset of size k is a k-itemset. A transaction 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 transaction dataset D, an itemset α is frequent if $|D_\alpha| \geq \sigma$, where $|D_\alpha|$ is called the support of α in D, and σ is a user-specified non-negative threshold.*

Definition 2 *(Markov Random Field). An Markov Random Field (MRF) 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.

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

2.1 Using Frequent Itemsets to Learn an MRF

The idea of using frequent itemsets to learn an MRF was first proposed by Pavlov *et al.* [3]. A k-itemset and its support represents a k-way statistic and can be taken as a constraint for the true underlying distribution which generates the data. Given a set of itemset constraints, a maximum entropy distribution satisfying all these constraints is picked up as the estimate for the true underlying distribution. It has been shown in [3] that a maximum entropy distribution specifies an MRF.

A simple iterative scaling algorithm can be used to learn an MRF from a set of itemsets. Figure 1 presents a high-level outline of a computationally efficient version of the algorithm given by Jelinek *et al.* [7]. It has been shown that the iterative process will converge if all the constraints are consistent, which naturally holds in our context. If the iterative scaling algorithm runs k iterations to converge 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 an itemset constraint. Thus efficient inference is crucial to the running time of the learning algorithm. We call models learned through exact inference procedures exact.

```
Iterative-Scaling(C)
 Input: $C$, collection of itemsets;
 Output: MRF $\mathcal{M}$;
 Obtain all involved variables $v$ and
   initialize parameters of $\mathcal{M}$;
   //typically uniform over $v$;
 while (Not all constraints are satisfied)
     for (each constraint $C_i$)
         Update $\mathcal{M}$ to force it to satisfy $C_i$;
 return $\mathcal{M}$;
```

Fig. 1. Iterative scaling algorithm

2.2 Junction Tree Inference Algorithm

The junction tree algorithm is a commonly used exact inference engine for probabilistic models. The time complexity of the junction tree algorithm is exponential in the treewidth of the underlying model. For the real world models, it's quite common that the treewidth will be well above 20, making learning exact models intractable. Correspondingly, we have to resort to learning approximate models. We note that there are two approaches to derive approximate models. The first approach is to plug in some approximate inference engine during the model learning process. The second approach is to learn a simplified model which is feasible to learn exactly and is close to the true exact model.

For the first approach, it is not clear whether or not the model learning process will still converge when subjected to approximate inference engines. In our empirical study, we found sometimes the model learning process will not converge when we plug in some MCMC inference engines. Parameters such as sample size and burn-in data size seem to affect the convergence behavior of the learning algorithm. As a result, we conjecture that we have to guarantee certain accuracy of approximate inference engines to ensure the convergence of the model learning process. Whether or not the convergence property still holds when we apply approximate inference algorithms in its own right is an interesting research problem and is beyond the scope of this paper. In this paper we pursue a variant of the second approach.

It's worth noting that Pavlov *et al.* [3] did not solve the problem of learning MRFs on large transaction data. The MRFs they generate target specifically the query variables and are therefore quite lightweight. Actually, they have noted the difficulty of learning a global model on the whole data in [3].

3 Learning Approximate MRFs

Before discussing our proposed approach, let us consider an extreme case in which the overall graph consists of a set of disjoint non-correlated components. Then the joint distribution can be obtained in a straightforward fashion according to the following lemma.

Lemma 1. *Given an undirected graph G subdivided into disjoint components D_1, D_2, ..., D_n (not necessarily connected components), and there is no edge across any two components, then the probability distribution associated with G is given by:*

$$p(X) = \prod_{i=1}^{n} p(X_{Di})$$

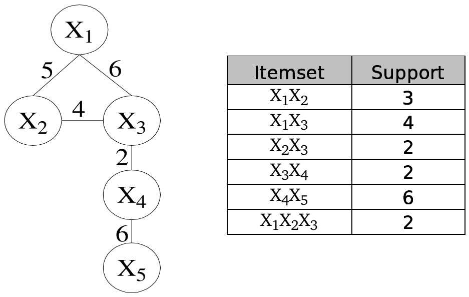
This conclusion follows immediately from the *global Markov property* of the MRF.

3.1 Clustering Variables Based on Graph Partitioning

The basic idea of our proposed divide-and-conquer style approach comes directly from the above observation. Specifically, the variables are clustered into groups according to their correlation strengths. We call the group *variable-cluster*. Then a local MRF is defined on each *variable-cluster*. In the end we aggregate the local models to obtain a global model. From Lemma 1, we see that if we have a perfect partitioning of an MRF in which there is no correlations across different partitions, the divide-and-conquer style approach gives the exact estimate of the full model. Even for a non-perfect partitioning, if the correlations across partitions are not strong, we still expect a reasonable approximation of the full model. Correspondingly, the first problem we face is how to cluster the variables such that the correlations across partitions is minimized?

***k*-MinCut** The k-MinCut problem is defined as follows [8]: Given a graph $G = (V, E)$ with $|V| = n$, partition V into k subsets, $V_1, V_2, \dots, V_k$ such that $V_i \cap V_j = \emptyset$ for $i \neq j$, $|V_i| = \frac{n}{k}$, and $\cup_i V_i = V$, and the number of edges of E whose incident vertices belong to different subsets is minimized. Given a partitioning P, the number of edges whose incident vertices belong to different partitions is called the *edge-cut* of the partitioning. In the case of weighted graphs, we minimize the sum of weights of all edges across different partitions. Correspondingly, the edge-cut is the sum of weights of all edges across different partitions.

Therefore the k-MinCut can serve our purpose of clustering variables. Each graph partition corresponds to a *variable-cluster*. Intuitively, we want to maximize correlations among variables within *variable-clusters*, and minimize correlations among variables across *variable-clusters*. So we should make the weight of edges reflect the strength of correlations between variables. We have the collection of all frequent itemsets. In particular, itemsets of size 2 specify the connectedness structure of the graph, and their associated supports indicate the strength of pairwise correlations between variables. We can use their supports as the edge weights directly. However, we also have higher-order statistics available, i.e., the larger itemsets. We expect that taking into consideration the information of all itemsets will yield a better weighting scheme. To this end, we propose an accumulative weighting scheme as follows: for each itemset, we add its support to all related edges, whose two vertices are contained by the itemset. Intuitively, we strengthen the graph regions which involve closely related itemsets in the hope that the edges within these regions will not be broken in the partitioning. Figure 2 illustrates the weighting scheme using a simple example. The collection of frequent itemsets and their supports are given in the figure.



Itemset	Support
X_1X_2	3
X_1X_3	4
X_2X_3	2
X_3X_4	2
X_4X_5	6
$X_1X_2X_3$	2

Fig. 2. Accumulative graph weighting scheme

An advantage of the k-MinCut partitioning scheme is that the resulting clustering is forced to be balanced. This is desirable for the sake of efficient model learning, since we will not encounter very large *variable-clusters* which might result in very complex local models. We need to specify k a priori. By choosing k one can examine trade offs involving model complexity, accuracy and online estimation time.

The k-MinCut partitioning scheme yields disjoint partitions. However, there exist edges across different partitions. In other words, different partitions are correlated to each other. So how do we account for the correlations across different partitions?

3.2 Interaction Importance and Treewidth Based Variable-Cluster Augmentation

The balanced *variable-clusters* produced by the k-MinCut partitioning scheme are disjoint. Intuitively, there is significant correlation information that is lost during the partitioning. The loss could be more severe considering that we force the balanced clusters, thus somehow we have to eliminate some edges with relative high weights. To compensate for this loss, we propose an interaction importance based *variable-cluster* augmenting scheme to recover the damaged correlation information. The idea is that for each *variable-cluster*, we let it grow outward. More specifically, it attracts and absorbs most significant (important) interactions (edges) incident to its vertices from outside to itself. As a result, some extra variables are pulled into the *variable-cluster*. We control the augmentation through the number of extra vertices pulled into the cluster (called *growth factor*). One can use the same growth factor for all *variable-clusters* to preserve their balance.

As an optimization, we account for the model complexity during the augmentation. We keep augmenting a partition until its complexity reaches a user-specified threshold. More specifically, we keep track of the growth of the treewidth during the augmenting process for this purpose. Additionally, 1-hop neighboring vertices are first considered by the augmentation, followed by 2-hop neighboring vertices and so on. Meanwhile, we still stick to the interaction importance criteria. As a result, the augmented partitions are likely to become unbalanced in terms of their size. The partitions with a small treewidth will grow more significantly than those with a large treewidth. However, these partitions are balanced in terms of their complexity. A benefit of this scheme is that usually more interactions across different partitions will be accounted for in a computationally controllable manner, leading to a more accurate global model. After the augmentation, we obtain overlapped *variable-clusters*. Figure 3 presents a sketch of the augmented *variable-clusters*.

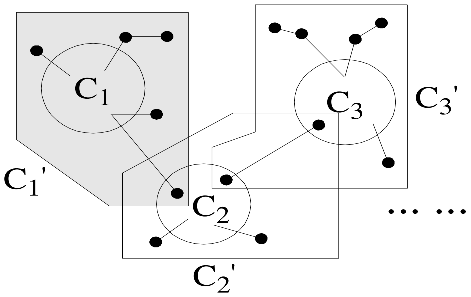


Fig. 3. Augmented *variable-clusters*

3.3 Approximate Global MRFs and A Greedy Inference Algorithm

For each augmented *variable-cluster*, we collect all of its related itemsets and use the iterative scaling algorithm to learn an exact local model. This is computationally feasible since the local model corresponding to each *variable-cluster* is much simpler than the original model. Two local models are correlated to each other if they share variables. The collection of all local models forms a global model of the original transaction data. We note that this global model is an approximation of the exact global MRF, since we lose dependency information by breaking edges in the exact graphical model. However, most of the lost strong correlations are compensated during the *variable-cluster* augmentation. As such, we believe that the proposed global model reasonably approximates the exact model. Figure 4 provides the formal algorithm for learning an approximate global MRF.

```
LearnMRF(F, k, g)
 $Input:\ F,\ collection\ of\ frequent\ itemsets;$
     $k,\ number\ of\ partitions\ for\ MinCut\ partitioning;$
     $g,\ growth\ factor;$
 $Output: \mathcal{M},\ global\ MRF;$
 $Construct\ a\ weighted\ graph\ G\ from\ F;$
   $//G\ specifies\ graphical\ structure\ of\ the\ exact\ MRF;$
 $k - MinCut\ G;$
 for $each\ graph\ partition\ G_i$
     $G'_i \leftarrow\ augment(G_i, g);$
     $Pick\ itemsets\ F_i\ related\ to\ G'_i$
     $M_i\ \leftarrow\ LearnLocalMRF(F_i);$
     $add\ M_i\ to\ \mathcal{M};$
 $return\ \mathcal{M};$
```

Fig. 4. Learning approximate global MRF algorithm

Given the global model consisting of a set of local MRFs, how do we make inferences on this model efficiently? In the first case where all query variables are subsumed by a single local MRF, we just need to calculate the marginal probability within the local model. In the second case where query variables span multiple local models, we use a greedy decomposition scheme to compute. First, we pick the local model which intersects most with the current query (covers most query variables). Then we pick the next local model which covers most uncovered variables in the query. This covering process will be repeated until we cover all variables in the query. Simultaneously, all intersections between the above local models and the query are recorded. In the end, we derive an overlapped decomposition of the query. We notice that locally the dependency among small pieces in the decomposition often exhibits a tree-like structure, and we use Lemma 2 to compute the marginal probabilities.

Lemma 2. *Given an undirected graph G subdivided into n overlapped components, if there exists an enumeration of these n components, i.e., C_1, C_2, …, C_n, s.t., for any*

$2 \leq i \leq n$, the separating set, $s(C_i, \cup_{j=1}^{i-1} C_j) \subseteq (C_i \cap (\cup_{j=1}^{i-1} C_j))$, then the probability distribution associated with G is given by:

$$p(X) = \frac{\prod_{i=1}^{n} p(X_{C_i})}{\prod_{i=2}^{n} p(X_{C_i} \cap (\cup_{j=1}^{i-1} X_{C_j}))}$$

Proof: We follow the order C_1, C_2, $\ldots$, C_n to deduce the full joint distribution as follows (repeatedly apply Lemma 3):

$$\begin{aligned} p(X_{C_1} \cup X_{C_2}) &= \frac{p(X_{C_2}) \cdot p(X_{C_1})}{p(X_{C_2} \cap X_{C_1})} \\ p(X_{C_1} \cup X_{C_2} \cup X_{C_3}) = p(X_{C_1} \cup X_{C_2}) &\cdot \frac{p(X_{C_3})}{p(X_{C_3} \cap (X_{C_1} \cup X_{C_2}))} \\ &\vdots \\ p(X_{C_1} \cup \ldots \cup X_{C_n}) &= \frac{\prod_{i=1}^{n} p(X_{C_i})}{\prod_{i=2}^{n} p(X_{C_i} \cap (\cup_{j=1}^{i-1} X_{C_j}))} \end{aligned}$$

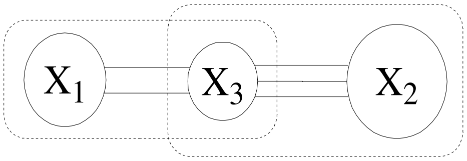


Fig. 5. Conditional independence

Lemma 3. *Let X_1, X_2, X_3 be three disjoint sets of variables in an undirected graph G, such that $X = X_1 \cup X_2 \cup X_3$. Additionally, there is no edge across X_1 and X_2 (we only allow edges across X_1 and X_3, X_2 and X_3 (see Figure 5), i.e., the separating set for X_1 and X_2, $s(X_1, X_2) \subseteq X_3$, then the probability distribution associated with G is given by:*

$$p(X) = \frac{p(X_1, X_3) \cdot p(X_2, X_3)}{p(X_3)}$$

Proof:

$$\begin{aligned} p(X_1, X_2, X_3) &= p(X_3) \cdot p(X_1|X_3) \cdot p(X_2|X_1, X_3) \\ &(\mathbf{X_1 \text{ and } X_2 \text{ are independent given } X_3}) \\ &= p(X_3) \cdot p(X_1|X_3) \cdot p(X_2|X_3) \\ &= \frac{(p(X_3) \cdot p(X_1|X_3)) \cdot (p(X_2|X_3) \cdot p(X_3))}{p(X_3)} \\ &= \frac{p(X_1, X_3) \cdot p(X_2, X_3)}{p(X_3)} \end{aligned}$$

□

To use the above formula, we require that there is no cyclic dependency among components, The overall dependency among components has a tree-like structure. Essentially Lemma 2 specifies a junction tree-like structure. Given any model and one of its such decomposition, we can use the above formula to make exact inferences.

Discussion: We note that the greedy inference scheme is a heuristic since it is possible to have a cyclic dependency among the decomposed pieces. Also, we note that our global model is not globally consistent in that there exists inconsistency across the local models. However, we expect that the global model is nearly consistent since two correlated local models contain exactly the same evidence information (itemsets) regarding their shared variables. A belief propagation style approach is currently under investigation to force the local consistency across the local models, thereby offering a globally consistent model.

4 Experimental Results

In this section, we examine the performance of our proposed approach on real datasets. Particularly, we focus on the application of the proposed model on the selectivity estimation problem. We compare the new model against the previous approach in [3] where a local MRF over query variables is learned for every query in an online fashion. We call this approach online local MRF approach (abbreviated as *OLM* in figures presenting experimental results). The MRF learning algorithm is implemented in C++. The junction tree inference algorithm is implemented based on Intel's Open-Source Probabilistic Networks Library[2]. We use *apriori* (a well-known efficient frequent itemset pattern mining algorithm) to collect frequent itemsets and *Metis* [8, 10] to obtain a k-MinCut of the exact graphical model.

4.1 Experimental Setup

All the experiments were conducted on a Pentium 4 2.66GHz machine with 1GB RAM running Linux 2.6.8. Below we detail the datasets, query workloads and performance metrics considered in our evaluation.

Datasets: We used two publicly available datasets in our experiments: the Microsoft Anonymous Web dataset (publicly available at the UCI KDD archive, kdd.ics.uci.edu) with 32711 transactions (Web site visitors) and 294 distinct attributes (Web pages); the BMS-Webview1 dataset (publicly available from the FIMI repository, fimi.cs.helsinki.fi), which is a web click-stream dataset from a web retailer company, Gazelle.com. The dataset contains 59602 transactions (Web sessions) and 497 distinct attributes (product detailed pages).

Query Workloads: In our experiments we considered the workloads consisting of conjunctive queries of different sizes. Following the same practice in [3], we first specified the number of query variables n (varied from 4, 6, 8, 10 to 12), then we picked n variables according to the probability of the variable taking a value of ”1” and generated a value for each selected variable by its univariate probability distribution.

[2] https://sourceforge.net/projects/openpnl/

Performance Metrics:
Time. We considered the *online time* cost, the time taken to answer the queries using the model. We also considered the *offline time* cost, the time taken to learn the model. Our objective is to have a fast and accurate online answer at the expense of potentially much higher offline time cost.
Error. We quantified the *accuracy* of estimations using the *average absolute relative error* over all queries in the workload. The absolute relative error is defined as $|\sigma - \hat{\sigma}|$ / σ, where σ is the true selectivity and $\hat{\sigma}$ is the estimated selectivity.

In the experiments, we varied k, the number of clusters; g, the number of vertices used to augment *variable-clusters* (the larger g is, the more overlapped the *variable-clusters* are, in the special case where $g = 0$, the *variable-clusters* are disjoint); the treewidth threshold tw when the treewidth based augmentation optimization is used and query size (4, 6, 8, 10, 12).

4.2 Results on the Microsoft Web Data

In this section, we report the experimental results on the Microsoft Web Data. We use the support threshold of 20 to collect the frequent itemsets, which results in 9901 frequent itemsets. According to the *Maximum Cardinality Search* (MCS)-ordering heuristic [11], the treewidth of the resulting MRF is 28 for which learning the exact model is intractable.

Figure 6a presents the estimation accuracy when k is varied (g is fixed as 5) for queries of different sizes. As can be seen, the new approach works very well compared with the online local MRF model. The global model based approach gives very close or even better estimations compared with the online local model based approach. These results are not surprising since for the online model, we only use the information of the itemsets whose variables are subsets of the online query to estimate the selectivity for the sake of a more efficient model construction. However, for the offline global model, we rely on more complete information to make the estimation. Even though the graph partitioning phase gives rise to information loss, since the model is global in nature, in many cases it is still able to yield better estimations. Furthermore, an obvious trend that stands out is that as the query size increases, the quality of the estimations degrades. This is as expected since for larger sized queries, estimation errors grow for both approaches. Another observation is that the estimations are more accurate when we use less *variable-clusters*. This is because with less *variable-clusters*, the information loss due to the graph partitioning is less, thus we capture better the correlations between partitions.

Figure 6b illustrates how the online times depend on the number of *variable-clusters* for queries of different sizes. It can be clearly seen the significant growth of the online times taken by the online model (note the Y-axis scale). The extreme online timing efficiency of the offline model can be clearly seen from the results. In most cases, it outperformed the online model by two to three orders of magnitude. Further, we see that the smaller number of *variable-clusters* results in higher online estimation time. This is as expected since the smaller k is, the larger each local model will be, which explains the slower estimation. In the extreme case where k is 1, we revert to learning the exact global MRF, which has been shown to be computationally infeasible.

Figure 6c presents the offline learning times of the offline model when varying k. An obvious trend is that as we increase k, overall the learning cost of the offline model decreases significantly. This is as expected since the larger k results in less complex local models.

Figure 7a presents the estimation accuracy when varying g (k is fixed as 20). As can be seen from the results, the error decreases steadily with increasing g. When g is 0 (disjoint *variable-clusters*), the estimations are most inaccurate. In contrast, the estimations are much more accurate when g is 5. The results clearly show the effects of the interaction importance based *variable-cluster* augmenting scheme. The offline model approximates the exact global model better when more correlations across the local models are compensated.

Figure 7b presents the online times when varying g. We see from the results that the model with the larger g takes more online time to answer the query. This is also as expected since the larger g results in more complex models (similar to the case of the smaller number of *variable-clusters*).

Figure 7c presents the offline learning times of the offline model when varying g. An obvious trend is that as we increase g, the time cost increases significantly. This is again as expected.

Figure 8a-c present the estimation accuracy, the online times and the offline learning times of the offline global model when the treewidth based augmentation optimization is used (k is fixed as 25). As can be seen, the optimization can further boost the estimation performance. For example, the average relative estimation errors are 0.29%, 0.97%, 2.01%, 3.66% and 4.81% on the workloads consisting of queries of size 4, 6, 8, 10 and 12, respectively. In contrast, the corresponding errors of the online local MRF approach are 0.99%, 2.76%, 4.45%, 7.82% and 10.9%, respectively. Furthermore, the offline model is faster by about two orders of magnitude in terms of online estimating time. Another obvious trend is that as we raise the treewidth threshold, the estimations will become more accurate, at a higher cost of online estimating and offline learning times.

4.3 Results on the BMS-Webview1 Data

In this section, we report the experimental results on the BMS-Webview1 data. We use the support threshold of 50 to collect the frequent itemsets, which results in 8191 frequent itemsets. The treewidth of the resulting exact MRF is 44 according to the MCS heuristic, which also makes learning the exact model intractable. The results on varying k and g are similar to that on the Microsoft Web data and are thus omitted in the interest of space. We only report the results when the treewidth based *variable-cluster* augmentation scheme is used.

Figure 9a-c present the estimation accuracy, the online times and the offline learning times of the offline global model when the treewidth based augmentation optimization is used (k is fixed as 70). As can be seen, the offline model is able to achieve estimations close to the online model, when the treewidth threshold is 10. When we increase the threshold to 12, the offline model generates more accurate estimations. Furthermore, the offline model provides better online timing performance than the online model, though the difference is not as significant as that on the Microsoft Web data. The reason is that the BMS-Webview1 data contains much more items than the Microsoft Web data.

As a result, the random queries generated are more likely to contain more uncorrelated items. As such, we have to use more local MRFs to cover one query when we estimate its selectivity, slowing down the estimation. In contrast, learning an online local MRF becomes easier in this case. However, if we consider correlations between items when we generate random workloads, in other words, more correlated items are more likely to occur in the same query, we expect that the offline global model will be significantly faster.

5 Related Work

Pavlov *et al.* [12, 3] have done significant work on exploiting probabilistic models for selectivity estimation on transaction data. They examined several models for this purpose. Besides the online local MRF learning based approach, they also examined the Chow-Liu tree model, the Bernoulli mixture model, the ADTree model and Bayesian networks. They showed that the online local MRF approach yields best performance on sparse data.

Goldenberg *et al.* [5] proposed an approach (SNBS) of using frequent itemsets to learn large Bayesian networks from sparse data. Further, they augmented the learned Bayesian networks with edges of high mutual information for variables that have not co-occurred in the data, since such dependencies are not captured by the frequent itemsets. The same technique can be adopted to enhance our proposed model.

Also, there has been significant work on approximate inference. Besides the MCMC techniques mentioned in this paper, variational methods [13–18] for approximate inference is a very active research field. Specifically, 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, 14, 17] and Pearl's belief propagation (BP) algorithm [15] (when applied to loopy graphs) are both belonging to this category.

6 Conclusion

In this paper, we have described a new approach to learning an approximate MRF on large sparse data. The new approach is shown to be very effective and efficient in solving the selectivity estimation problem. In the future, we would like to exploit a belief propagation style approach to force consistency of the model. Further, we would like to exploit approximate inference techniques, such as generalized belief propagation and generalized mean field algorithms in our model learning process. Finally, we would like to exploit the use of the models on link analysis tasks.

References

1. Getoor, L., Taskar, B., Koller, D.: Selectivity estimation using probabilistic models. In: SIGMOD Conference 2001. (2001) 461–472
2. Deshpande, A., Garofalakis, M.N., Rastogi, R.: Independence is good: Dependency-based histogram synopses for high-dimensional data. In: SIGMOD Conference 2001. (2001) 199–210

3. Pavlov, D., Mannila, H., Smyth, P.: Beyond independence: probabilistic models for query approximation on binary transaction data. IEEE Transactions on Knowledge and Data Engineering **15** (2003) 1409–1421
4. Breese, J.S., Heckerman, D., Kadie, C.M.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence. (1998) 43–52
5. Goldenberg, A., Moore, A.: Tractable learning of large bayes net structures from sparse data. In: Proceedings of the twenty-first international conference on Machine learning. (2004)
6. Friedman, N.: Inferring cellular networks using probabilistic graphical models. Science **303** (2004) 799–805
7. Jelinek, F.: Statistical Methods for Speech Recognition. MIT Press, Cambridge, MA (1998)
8. Karypis, G., Kumar, V.: Multilevel k-way partitioning scheme for irregular graphs. J. Parallel Distrib. Comput. **48** (1998) 96–129
9. Wang, C., Parthasarathy, S.: Learning approximate mrfs from large transaction data. In: The Ohio State University, Technical Report. (2006)
10. Karypis, G., Kumar, V.: Metis: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. http://www-users.cs.umn.edu/∼karypis/metis/metis/files/manual.ps (1998)
11. Tarjan, R.E., Yannakakis, M.: Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. SIAM Journal of Computing **13** (1984)
12. Pavlov, D., Smyth, P.: Probabilistic query models for transaction data. In: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. (2001) 164–173
13. Jordan, M.I., Kearns, M.J., Solla, S.A.: An introduction to variational methods for graphical models. Machine Learning **37** (1999) 183–233
14. Wiegerinck, W.: Variational approximations between mean field theory and the junction tree algorithm. In: Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence. (2000) 626–633
15. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Understanding belief propagation and its generalizations. In: IJCAI. (2001)
16. Bishop, C.M., Spiegelhalter, D.J., Winn, J.M.: Vibes: A variational inference engine for bayesian networks. In: NIPS 2002. (2002) 777–784
17. Xing, E., Jordan, M., Russell, S.: A generalized mean field algorithm for variational inference in exponential families. In: Uncertainty in Artificial Intelligence. (2003) 583–591
18. Geiger, D., Meek, C.: Structured variational inference procedures and their realizations. In: AI and Statistics 2005. (2005)

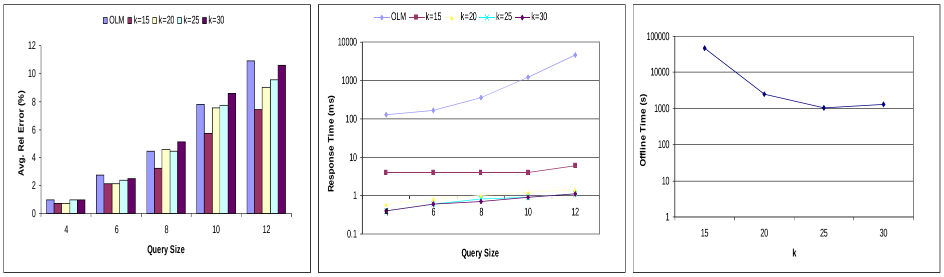


Fig. 6. Varying k ($g = 5$): (a) estimation accuracy (b) online time (c) offline time

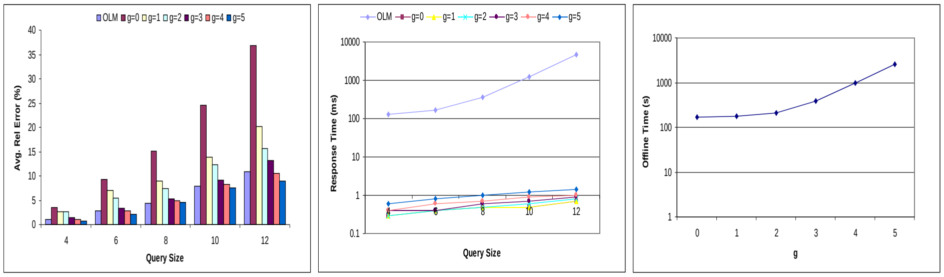


Fig. 7. Varying g ($k = 20$): (a) estimation accuracy (b) online time (c) offline time

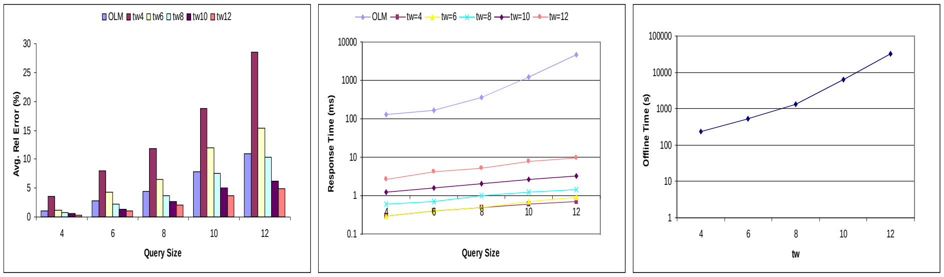


Fig. 8. Varying tw ($k = 25$): (a) estimation accuracy (b) online time (c) offline time

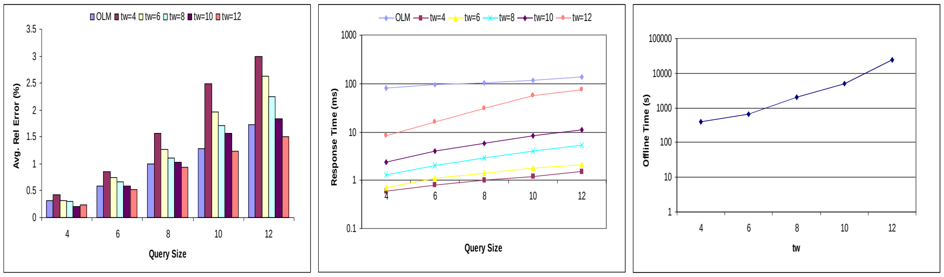


Fig. 9. Varying tw ($k = 70$): (a) estimation accuracy (b) online time (c) offline time