An approximate MRF model for querying large sparse binary 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 estimating the selectivity of conjunctive queries on large sparse binary data. Recently, an approach based on generating a maximum entropy model from frequent itemset patterns was shown to out-perform other extant approaches such as mixture modeling using Bernoulli distributions, inclusion-exclusion ADTree model, etc. However, a key limitation of the maximum entropy model, which can also be viewed as a Markov Random Field (MRF), is that it needs to build a model on the fly for every query. This localized approach to estimation can be quite expensive for complex queries, thus limiting its use in settings where response time is crucial.

We address this limitation through the design of a global model that takes into account all the variables in the data. The model is constructed offline once. Subsequently answers to ad-hoc queries can be estimated from the global model, thus avoiding online model construction. The modeling approach first employs graph partitioning to cluster variables into balanced disjoint partitions, and then augments important edges across partitions to capture interdependencies across them. A local MRF model is generated for each partition, and then all local MRF models are combined together to form a global model of the data.

A probabilistic estimation procedure is developed to estimate the selectivity of ad-hoc queries from the model summary. Extensive empirical evaluations on real datasets demonstrate the viability of the approach when compared to extant strategies in terms of overall accuracy and especially in terms of efficiency.

1. INTRODUCTION

In this paper we address the problem of constructing probabilistic models to estimate the selectivity of conjunctive queries on large binary data. Examples of such data are market basket data, web log data, etc. 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". Thus the data is a binary matrix. An important characteristic of this matrix is that it is often sparse in that the number of entries per row that take the value of "1" is small when compared to the number of columns. In other words, a large portion of the matrix takes the value of "0".

Selectivity estimation is important for query optimization, since the query optimizer routinely relies on the selectivity estimations to evaluate optimal query plans. Selectivity estimation techniques have been extensively examined in the context of relational database queries [5, 28, 27], boolean string queries [20, 16, 7, 8] and semi-structured data queries [6, 1, 26, 25]. Furthermore, reasonably accurate selectivity estimation is also desirable in interactive settings and for approximate querying. For instance, an end user can interactively refine her query if she knows that the current query will return an overwhelming result set. Similarly, the estimated value can be returned as an approximate answer to aggregate queries using the COUNT primitive.

Estimating selectivity usually relies on some summary structure of the original large data. Ideally, the summary structure should be much smaller than the original data (usually in memory), such that the selectivity estimator can efficiently process the summary structure and yield the estimation fast. Furthermore, in order to yield reasonably accurate estimations, the summary structure has to be able to faithfully capture important statistics of the data.

An important class of such summary structures are probabilistic models of the data. Specifically for large binary data, probabilistic models can effectively capture association or causal correlations among attributes. An important benefit of probabilistic models is that one is able to make predictions, for example, how often a particular combination of items are likely to be purchased together in the future. A direct application of this is in recommender systems whose goal is to predict what items that users would buy next according to their purchase history. Another use of probabilistic models is to define interestingness of itemset patterns [17]. Itemset patterns whose count deviates significantly from its expected value are considered interesting. Probabilistic models also find applications in bioscience as well. Friedman *et al.* [13] describe gene regulatory networks using probabilistic graph models.

We rely on probabilistic models to tackle the selectivity estimation problem on large binary data. First a probabilistic model is learned from the data, and then query selectivity is estimated based on the model. Specifically, our proposed approach is based on the Maximum Entropy (ME) model proposed by Pavlov *et al.* [22]. Their approach is based on generating a maximum entropy model from frequent itemset patterns, and has been shown to yield more accurate estimations than other approaches, such as Bernoulli mixture model, inclusion-exclusion ADTree model and so forth on large sparse binary data. It has been shown that the maximum entropy model defines a Markov Random Field (MRF) [12], which specifies the class of probabilistic models we are inferring using this approach.

However, a key limitation of the maximum entropy model is that it needs to build a model on the fly for every query. A local model on query variables is constructed in an online fashion to estimate the query's selectivity. 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, especially for processing complex queries.

We address this limitation through the design of a global model that takes into account all the variables in the data. The model is constructed offline once. Subsequently answers to ad-hoc queries can be estimated from the global model thus avoiding just-in-time model construction. The modeling approach relies on graph partitioning to cluster variables into balanced disjoint partitions, and then augmenting important edges across partitions to capture inter-dependencies across them. A local MRF model is generated for each augmented partition, and then all local MRF models are combined together to form a global model of the data.

A probabilistic estimation procedure is developed to estimate the selectivity of ad-hoc queries from the model summary. Extensive empirical evaluations on real datasets demonstrate the viability of the proposed approach when compared to the previous approaches in terms of overall accuracy and efficiency. The main contributions of this paper are highlighted below.

- We introduce a novel divide-and-conquer style approach based on graph partitioning to learning a global MRF model from large binary data. Such a model is an approximation of the exact global MRF model, which is usually computationally expensive to learn.
- We introduce an efficient mechanism to estimate a query's selectivity based on the above model.
- We conduct extensive empirical evaluations on real datasets to show the efficiency and effectiveness of the new approach.

The rest of the paper is organized as follows. We briefly go over the maximum entropy model in Section 2. In Section 3 we detail our proposed probabilistic model and the corresponding selectivity estimating mechanism. We present experimental results in Section 4 and related work in Section 5. Finally, we discuss the future work and conclude in Section 6.

2. BACKGROUND AND PROBLEM STATEMENT

Definition 1. An *item* associated with the binary table r (with header R) is a single attribute in R.

Within the probabilistic model context, each item corresponds to a distinct random variable. Throughout the rest of the paper, item and variable will be used interchangeably.

Definition 2. [22] An *itemset* associated with the binary table r (with header R) is defined to be either a single positively initialized attribute or a conjunction of mutually exclusively positively initialized attributes from R. We call an itemset T-frequent if its count in the table r is at least T, where T is some pre-specified non-negative threshold. The size of an itemset is the number of conjuncts it is defined on.

Formally, borrowing the precise terminology from Pavlov *et al.* [22], the query selectivity estimation problem on binary data can be defined as follows. Let $R = A_1, \ldots, A_k$ be a table header with k 0/1 valued attributes and r be a table of n rows over R. We assume that $k \ll n$, and that the data are sparse, i.e., the average number of 1's per row is substantially smaller than the number of attributes. By definition, a row of the table r satisfies a conjunctive query Q if and only if the corresponding attributes in the query and in the row have equal values. We are interested in finding the number of rows in the table r satisfying a given conjunctive query Q defined on a subset of its attributes.

To tackle this problem, Pavlov *et al.* [22] used the collection of T-frequent itemset patterns from the binary data as the statistics of the data. They took each itemset pattern as a particular constraint on the true joint distribution which generates the data. Among all distributions satisfying all the constraints, they picked the distribution with the maximal entropy ("as uninformed as possible") as the estimate for the true joint distribution. It has been shown that this ME based approach is very effective in estimating the selectivity of queries, especially on the sparse data. Specifically, they showed that the ME based approach is more accurate than models such as the Chow-Liu tree model, the Bernoulli mixture model and the inclusion-exclusion ADTree model, etc. The usefulness of frequent itemset patterns in constructing probabilistic models is supported by an observation that positive correlations are much stronger than negative correlations in the case of sparse data [14]. The frequent itemset patterns capture exactly the positive correlations between items.

Specifically this approach works as follows. First all T-frequent itemset patterns are collected offline. 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. Next, a full distribution on x_Q is built in an online fashion based on the ME principle. Once the model is ready, any conjunctive query whose variables are subset of x_Q can be answered, including Q itself as well. Due to its inherent online and local character, we call this approach the online local MRF approach.

The following gives an example of the maximal entropy distribution. Suppose we have collected the itemset patterns 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 $\mu_0, \ldots, \mu_{11}$ are estimated from the data.

Importantly, it has been shown that the ME based approach defines an MRF model for the original data [22]. This precisely tells us the class of probabilistic model that is inferred from the data using this approach. The MRF model is an undirected graph in which vertices represent variables and edges represent correlations between variables. The joint distribution associated with the undirected graph 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 for the elements 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. The maximal cliques of a graph are the cliques 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. Figure 1 shows the MRF model defined by the maximal entropy distribution in the previous example. In particular, $\psi_{C_1}(X_{C_1}) = \mu_1^{I(x_1=1)} \cdot \mu_2^{I(x_2=1)} \cdot \mu_3^{I(x_3=1)} \cdot \mu_6^{I(x_1=x_2=1)} \cdot \mu_7^{I(x_1=x_3=1)} \cdot \mu_8^{I(x_2=x_3=1)} \cdot \mu_{11}^{I(x_1=x_1=x_3=1)}$; $\psi_{C_2}(X_{C_2}) = \mu_4^{I(x_4=1)} \cdot \mu_9^{I(x_3=x_4=1)}$ and $\psi_{C_3}(X_{C_3}) = \mu_5^{I(x_5=1)} \cdot \mu_{10}^{I(x_4=x_5=1)}$.

Note that the online local MRF approach has several limitations. First the MRF model is built online. It has been shown that the ME algorithm for inferring an MRF model has worst-case time com-

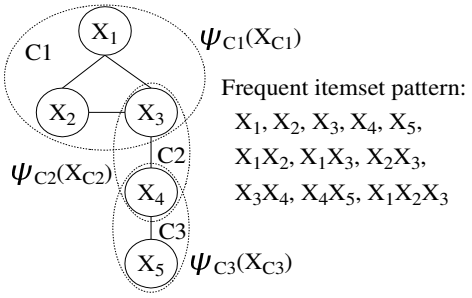


Figure 1: An MRF Example

plexity exponential in the number of query variables, thus it is not feasible to process complex queries in real time. Further, the model constructed is a local model in the sense that it targets specifically the query variables, thus cannot be reused for the queries thereafter. However, selectivity estimation usually requires very prompt response. Thus the online local MRF approach can not be applied for processing complex queries. In our empirical study, we found that the approach became inefficient when processing queries of size approaching or above 10.

The goal of our study is to tackle the crucial limitations of the online local MRF approach. Specifically, we want the new model to be an offline model. Thus we do not need to sacrifice precious online time for model construction. Another benefit of offline model is that we are able to afford a more fine-grained refinement of the model, since this can be done offline. Additionally, in order to cope with complex queries, we expect the new model to be a global model which specifies the joint distribution on all variables. Correspondingly, another closely related requirement is that given the model satisfying the above requirements, we should be able to have an efficient online estimating mechanism.

3. APPROXIMATE OFFLINE GLOBAL MRF MODEL

Towards the goal of a global model, our first attempt is to account for all the T-frequent itemset patterns, then build a complete MRF model on all variables offline directly. We call this model the exact global MRF model. When a query is posed online, we make an estimation based on this model. However, one problem with this approach is that the iterative algorithm for inferring an MRF model on the variables has worst case time complexity exponential in the number of variables, as already pointed out in Section 2. In [22], Pavlov *et al.* propose several optimizations which rely on the graph structure to speedup model construction. The time complexity of iterative algorithm has been reduced to being exponential in the induced width of the MRF graph, which however is still computational prohibitive for inferring large complex MRF models. Moreover, even suppose we were able to construct the exact global model, there is another difficulty when doing selectivity estimation for online queries. Essentially we need calculate marginal distributions (estimation) corresponding to the query variables. Making the estimation efficiently for large graph models is not an easy task. Therefore, we resort to approximate MRF models. Now let us take a look at the MRF model and its important properties.

3.1 MRF Model and Its Properties

The MRF models fully specify the conditional independence among variables. The *global Markov property* tells us that for all disjoint

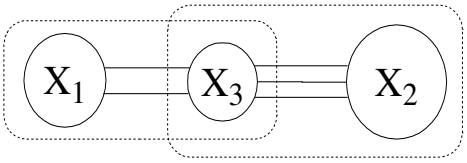


Figure 2: Conditional independence

vertex subsets a, b and c in the graph 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. Based on this property, we can derive several useful properties.

Let's 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.

From Lemma 1, we see that if the graph model consists of a set of disjoint non-correlated components, then a divide-and-conquer style approach gives the exact estimate for the full joint distribution. Specifically, we focus on the variables belonging to a single graph component, and construct the joint distribution over these variables. Essentially we construct a local model specific to this graph component. We do this for all graph components. In the end, we combine these local models together into a global model according to Lemma 1.

However, we know that in practice, different components of the graph usually interact with one another. So how do we take into consideration the correlations between different components?

LEMMA 2. *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 2), 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:

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

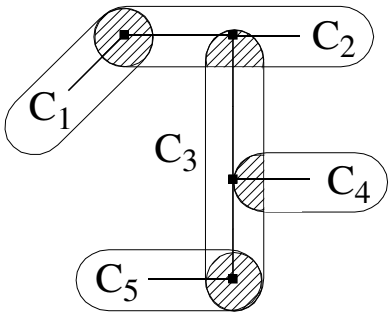


Figure 3: Generalized conditional independence

□

From Lemma 2, we see that even when there exist correlations between graph components, the divide-and-conquer approach can still be used. In particular, we make use of the conditional independence structure of the graph, and divide the whole graph into overlapped partitions. The correlations between partitions are captured by their shared common part. Next we construct a local model for each partition, then we combine the local models together to obtain a global model according to Lemma 2. For the example in Figure 2, two overlapped partitions $X_1 \cup X_3$ and $X_2 \cup X_3$ give exact estimate for the full joint distribution. Lemma 2 can be generalized to handle the case where we have multiple overlapped partitions.

LEMMA 3. *Given an undirected graph G subdivided into n overlapped components, if there exists an enumeration of these n components, i.e., $C_1, C_2, \ldots, C_n$, 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 Sketch: We follow the order $C_1, C_2, \ldots, C_n$ to deduce the full joint distribution as follows (repeatedly apply Lemma 2):

$$p(X_{C_1} \cup X_{C_2}) = \frac{p(X_{C_2}) \cdot p(X_{C_1})}{p(X_{C_2} \cap X_{C_1})}$$

$$p(X_{C_1} \cup X_{C_2} \cup X_{C_3}) = p(X_{C_1} \cup X_{C_2}) \cdot \frac{p(X_{C_3})}{p(X_{C_3} \cap (X_{C_1} \cup X_{C_2}))}$$

$$\vdots$$

$$p(X_{C_1} \cup \ldots \cup X_{C_n}) = \frac{\prod_{i=1}^{n} p(X_{C_i})}{\prod_{i=2}^{n} p(X_{C_i} \cap (\cup_{j=1}^{i-1} X_{C_j}))}$$

Essentially, we require that there is no cyclic dependence among components, which would hinder the applicability of Lemma 2. The overall dependence among components has a tree-like structure. Figure 3 illustrates an example of this. Specifically for this example, we have:

$$p(X) = \frac{p(X_{C_1}) \cdot p(X_{C_2}) \cdot p(X_{C_3}) \cdot p(X_{C_4}) \cdot p(X_{C_5})}{p(X_{C_1} \cap X_{C_2}) \cdot p(X_{C_2} \cap X_{C_3}) \cdot p(X_{C_3} \cap X_{C_4}) \cdot p(X_{C_3} \cap X_{C_5})}$$

□

3.2 Approximate Offline Global MRF Model by Graph Partitioning

The basic idea of our proposed divide-and-conquer style approach comes directly from the above theoretical analysis and an important observation. On large sparse high-dimensional binary data, most variables have strong correlations to only a few other variables, rather than many variables. In other words, the number of vertices with low degrees in the graph model is much higher than that of vertices with high degrees. This is called *scale-free* property in social network literature [2, 21]. Let us consider a real binary transaction data, in which each item is an author and each transaction is an author list for a particular academic paper. It's easy to show that the graph of the corresponding MRF model on all variables (authors) is essentially a co-authorship network, which has been shown to be scale-free [3].

Specifically in the proposed approach, the variables are clustered into groups according to their correlation strengths. We call the group *variable-cluster*. Then a local MRF model is defined on each *variable-cluster*. In the end we aggregate the local models to obtain a global model. Correspondingly, the first problem we face is how to cluster the variables and how to construct local models for these *variable-clusters*? Furthermore, how do we aggregate these local models to form a global model? Finally, how to make online selectivity estimations efficiently based on the global model?

3.2.1 Clustering Variables Based on Graph Partitioning

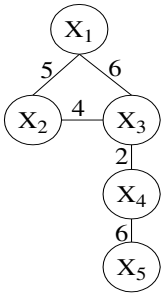
In our study, we evaluated different variable clustering schemes. All the schemes can be presented within the context of graph partitioning.

3.2.2 k-MinCut

The k-MinCut problem is defined as follows [19]: Given a graph $G = (V, E)$ with $|V| = n$, partition V into k subsets, $V_1, V_2, \ldots, V_k$ 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.

k-MinCut is useful in factorizing the full joint distribution. Each graph partition corresponds to a *variable-cluster*. Intuitively, we want to maximize correlations between variables within clusters, and minimize the correlations between variables across clusters. So we should make the weight of edges to some extent reflect the strength of the correlation between variables. We have the collection of all T-frequent itemset patterns. In particular, itemsets of size 2 specify the connectedness structure of the graph, and their associated counts indicate the strength of pairwise correlations between variables. We can use the counts as the edge weights directly. However, we also have higher-order statistics available, i.e., the larger itemset patterns. We expect that taking into consideration the information of all itemset patterns will yield a better weighting scheme. To this end, we propose an accumulative weighting scheme as follows: for each itemset pattern, we add its count to all related edges, whose incident vertices are contained by the itemset. Intuitively, we strengthen the graph regions which involve many closely related itemset patterns in the hope that the edges within these regions will not be broken in the partitioning. Figure 4 illustrates the weighting result for the previous example in Section 2. The collection of frequent itemset patterns and their associated counts are given in the figure.

A significant advantage of the k-MinCut scheme is that the resulting clustering is forced to be balanced. This is desirable for the sake of efficient estimation, since we will not encounter very large clusters which would result in complex local models. We need to



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

Figure 4: The accumulative graph weighting scheme

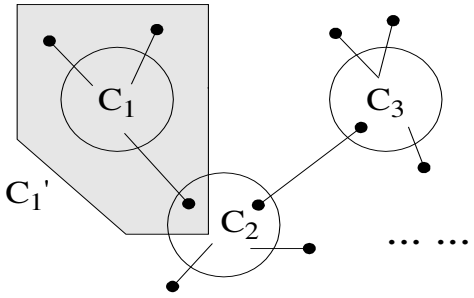


Figure 5: Augmented *variable-clusters*

specify k a priori. By choosing k one can examine trade offs involving model complexity, accuracy and online estimation time.

In our study, we used Metis [19, 18] to obtain k-MinCut of the original exact graph model. Metis is a multilevel partitioning scheme and has been shown to be successful in producing balanced clusters of good quality. The multilevel partitioning scheme has three major phases: coarsening, initial partitioning and uncoarsening. In the coarsening phase, the original graph is gradually transformed into a smaller graph. An initial two-way partitioning of the smaller graph that satisfies the balancing constraint while minimizing the edge cut is obtained in the next phase. During the final phase, the partitioning is successively refined as it is projected back to the original graph by traversing intermediate partitions.

3.2.3 *Pattern Profile and Flow-based Graph Partitioning*

Yan *et al.* [32] recently proposed a pattern summarization approach for frequent itemsets. The key notion is *pattern profile*, which essentially corresponds to an item cluster. Specifically, a pattern profile is a triple $< a, b, c >$ where a is a set of items, b is a distribution vector on the items in a and c is the count of the whole pattern profile. 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. In our study, we apply this technique to summarize all the T-frequent itemset patterns and use the resulting pattern profiles to generate the *variable-clusters*. An advantage of this approach is that it is natural to obtain overlapped clusters, since the same variables can belong to multiple pattern profiles simultaneously. Pattern profile based clustering scheme can be presented using graph partitioning framework. A particular pattern profile corresponds to an induced subgraph of the complete MRF model.

However, a disadvantage of this approach is that there is no balancing constraint on the resulting clusters. It is possible that the resulting clusters are quite unbalanced, since general MRF models are typically scale-free networks when the dimensionality is high. As for partitioning scale-free networks without forcing any balancing constrain on the clusters, it has been shown that extremely unbalanced clusters will be formed [4, 33]. Specifically, a giant core consisting of hub nodes and their neighboring nodes will dominate the whole network. Obviously this is not what we want since we prefer balanced clusters for the sake of efficient estimation thereafter.

In addition, we also evaluated flow-based graph clustering algorithms on the MRF model. In particular, we used MCL [29], which has been shown to be able to identify natural cluster structures of a graph, and has been widely used in biological data analysis [11, 24]. An important feature of MCL is that there is no need to specify the number of clusters a priori. MCL faces the same problem as the pattern profile, since the natural cluster structures of a scale-free network are probably extremely unbalanced. The drawbacks of the pattern profile and the flow-based clustering algorithm have been verified by our empirical study, we do not pursue these two ideas further.

3.2.4 *Edge Importance Based Variable Cluster Augmentation*

The balanced 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 edge importance based *variable-cluster* augmenting scheme to recover the damaged correlation information. The idea is that for each cluster, we let it grow outward. In other words, it attracts and absorbs most significant (important) edges incident to its vertices from outside to itself. As a result, some extra variables are pulled into the cluster. We control the augmentation through the number of extra vertices pulled into the cluster (called *growth factor*). We usually use the same growth factor for all clusters to preserve their balance. We want to keep the important correlation information that is lost during the graph partitioning. Consequently, we are obtaining overlapped *variable-clusters* by this scheme. Figure 5 gives a sketch of the augmented *variable-clusters*.

3.2.5 *Approximate Offline Global MRF Model*

For each augmented cluster, we construct a local MRF model for which we use the same process as the online model construction. Two local models are correlated to each other if they share variables. The collection of all local MRF models forms a global model of the original binary data. We note that this global model is an approximation of the exact global MRF model, since we lose dependency information by breaking edges in the exact graph model. However, most of the important correlation information loss will be compensated during the cluster augmentation. Further, the augmenting scheme implicitly gives higher priority to more frequent variables such that they tend to be shared among *variable-clusters*. Interestingly, we note that this is beneficial for preserving the basic topological structure of the exact graph model. We know in scale-free networks, hub nodes sustain the whole network. The most frequent variables play roles of hub nodes in the exact graph model. Keeping the information of these variables in the approximate graph model enables it to preserve the basic topological struc-

Algorithm: BuildMRFModel (F, k, g)
Input: F, collection of T-frequent itemset patterns;
k, number of partitions for MinCut algorithm;
g, growth factor;
Output: $\mathcal{X}$, global MRF model;
1. Construct a weighted graph G from F;
//G has the same graphical structure as the exact MRF model;
2. k-MinCut G;
3. **for** each graph partition G_i
G_i'= augment(G_i, g);
Pick itemset patterns F_i related to G_i'
M_i = BuildLocalMRF(F_i);
add M_i to $\mathcal{X}$;
4. return $\mathcal{X}$;

Figure 6: Building Global MRF Model Algorithm

ture of the exact graph model. As such, we believe that the proposed global graph model reasonably approximates the exact graph model. Figure 6 provides the formal algorithm for building approximate offline global MRF model.

3.3 Model-Aware Decomposition Scheme for Online Estimation

Given the global model consisting of a set of local MRF models, how do we do the online estimation efficiently? In the first scenario, where all query variables are subsumed by a single local MRF model, we just need to estimate the selectivity within the local model. In the second scenario, where query variables span multiple local models, we use a greedy decomposition scheme to estimate. 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 dependencies among small chunks in the decomposition often exhibit a tree-like structure, thus we use Lemma 3 to calculate the selectivity estimation. Strictly speaking, this is a heuristic since it is possible to have cyclic dependencies.

The intuition is that such a greedy approach will produce near optimal estimations without sacrificing estimation time. The goal is to find a minimum size set of local MRF models whose union contains all query variables. As can be seen, decomposition proceeds in a model-aware fashion. Compared to a naive decomposition scheme in which we simply subdivide the query into small enough chunks without taking the model into consideration, this scheme is able to give much more accurate estimations.

Additionally, due to the overlap between local models, it is possible to have multiple feasible decompositions for a single query, which results in multiple estimations for that query. In the present study, we limit it to one particular decomposition for selectivity estimation. A potential followup work is to evaluate all feasible decompositions and corresponding estimations, and then apply some voting scheme to get even better estimations. This is more worthwhile when there are parallel computing resources available, since evaluation of different decompositions can be made independently. Figure 7 gives the formal model-aware decomposition algorithm. In particular, $eval()$ is a function that calculates the selectivity estimation within a local MRF model.

Algorithm: Estimate (Q, M)
Input: Q, online query;
M, offline global MRF model;
Output: s, selectivity estimation for Q;
1. Initialize $remaining$ by Q;
2. Initialize $covered$ to be $\emptyset$;
3. **While** $remaining \neq \emptyset$
4. Choose a local model M_i which covers $remaining$ most;
5. $newCovered = M_i \cap remaining$;
6. $D_i = M_i \cap covered$;
7. $C_i = newCovered \cup D_i$;
8. add C_i to C;
9. add D_i to D;
10. $remaining = remaining - C_i$;
11. $covered = covered \cup C_i$;
12. $s = \frac{\prod_i eval(C_i, M_i)}{\prod_i eval(D_i, M_i)}$;
13. return s;

Figure 7: Online Estimation Algorithm

4. EXPERIMENTAL RESULTS

In this section, we examine the performance of our proposed approach on real datasets. We compare the new offline global MRF model against the previous online local MRF model (abbreviated as OLM in figures presenting experimental results).

4.1 Experimental Setup

All the experiments were conducted on a Pentium 4 2.66GHz machine with 1GB RAM running Linux 2.6.8. We used *apriori* (a well-known efficient frequent itemset pattern mining algorithm) to collect the T-frequent itemset patterns. In particular, we imposed the same precision (0.01) on the two models during their iterative construction for the sake of an impartial comparison between the two approaches. 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 [22], we first specified the number of query variables n (varied from 4, 6, 8 to 10), 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. As pointed out in [22], the variables are prone to take the zero values in sparse data, thus using purely random queries (variables are randomly chosen with randomly chosen values) would result in a preponderance of queries with zero count.

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 construct the model. Our objective is to have a fast and accurate online answer at the expense of potentially much higher offline time cost.
- Memory. We considered the *memory consumption* of the models, which to some extent reflects the complexity of the

models. We prefer the models with low memory consumption which however yield fast and accurate answers.

- 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 $|\sigma - \hat{\sigma}|$ / max(10, σ), where σ is the true selectivity , $\hat{\sigma}$ is the estimated selectivity and a sanity bound of 10 is used to avoid the artificially high percentages of low selectivity queries.

In the experiments, we varied C, the number of clusters; k, the number of edges used to augment *variable-clusters* (the larger k is, the more overlapped the *variable-clusters* are, in the special case where $k = 0$, the clusters are disjoint); query size (4, 6, 8, 10) and the threshold T used to collect the frequent itemset patterns from the data.

4.2 Results on the Microsoft Web Data

In this section, we report the experimental results on the Microsoft Web Data. In the first set of experiments, we set $T = 200$ to collect the frequent itemset patterns, which results in 405 frequent itemsets. Figure 8a presents the graphical structure of the exact complete MRF model. We applied Metis to partition the graph. Specifically, we did not consider the 24 isolated vertices in the partitioning process, since it is safe to take each of such vertices as an independent cluster and then apply the independence formula on these *variable-clusters* according to Lemma 1. As a result, we had the 42 vertices for partitioning. Note that all the isolated vertices will be considered in the model construction.

Figure 9a presents the estimation accuracy when C is varied (k is fixed as 4) for queries of different sizes. As can be seen, the offline global MRF model works very well compared with the online local MRF model. The offline model gives very close or even better estimations as that given by the online model. In particular, when C is 5, 10 and 15, the offline model yields more accurate estimations than the online model. When C is 20, the two models give very close estimations, though the online model is slightly better. These results are not surprising since in the online model, we only use the information of the itemset patterns whose variables are subsets of the online query to estimate the selectivity for the sake of a more efficient model construction. However, in the offline model, we are able to rely on the information of all related itemset patterns whose variables intersect with the query variables 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 size of the query 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 clusters, the information loss due to the graph partitioning is less severe, thus we capture better the correlations between items. However, this will affect the online estimation time. Now let us examine the timing performance when varying the number of clusters.

Figure 9b illustrates how the online time depends on the number of clusters for queries of different sizes. It can be clearly seen the exponential growth of online time taken by the online model. The online model is very fast when processing rather small queries (query size less than 6), but is extremely slow when processing complex queries (query size larger than 8). In contrast, the offline model has much flatter curves here. In some cases, the curve is actually going down as the size of the queries increases. For example, when C is 5 and 10, the timing curve keeps going down as we process more complex queries. This shows the extreme efficiency of the offline model for processing complex queries. The model-aware decomposition scheme can efficiently decompose the complex query, and estimations within local models are fast. Further, we see that the smaller number of clusters results in higher online estimation time. This is as expected since the smaller C is, the larger each *variable-cluster* will be. Correspondingly, we will have a larger local MRF model for each cluster, which results in slower estimation. In the extreme case where C is 1, we revert to the exact complete MRF model, which has been shown to be computationally infeasible.

Figure 9c presents the offline time cost of the offline approach when varying C. An obvious trend is that as we increase the value of C, the time cost of the offline model decreases significantly. This is as expected since the larger C results in less complex models.

Figure 10a presents the estimation accuracy when varying k (C is fixed as 20). As can be seen from the results, the error decreases steadily with increasing k. When k is 0 (disjoint *variable-clusters*), the estimations are most inaccurate. In contrast, the estimations are much more accurate when k is 6. The results clearly show the effects of the edge importance based cluster augmenting scheme. The offline model approximates the exact complete model better when more correlations are compensated.

Figure 10b presents the online times when varying k. We see from the results that the model with the larger k takes more online time to answer the query. This is also as expected since the larger k results in more complex models (similar to the case of smaller number of clusters).

Figure 10c presents the offline time cost of the offline approach when varying k. An obvious trend is that as we raise the value of k, the time cost of the offline model increases significantly. This is again as expected.

We also examined the memory usage taken by models under different conditions. Figure 11 presents the memory usage of different models with varied k and C. For the comparison with the online model, we also plot its memory usage. Note that the online model only has one column. As can be seen, when we raise k, the models take more memory. Interestingly, we note that when k is small (less than 3), the models with smaller C values take more memory. In contrast, when k is large, the models with larger C values take more memory. This is because that when k is small, the memory usage is primarily caused by recording the within-cluster correlations, rather than the cross-cluster correlations, thus the models with smaller C values need more memory. In contrast, when k becomes larger, the memory usage caused by recording the cross-cluster correlations becomes more significant, therefore the models with larger C values need more memory, since usually they need to record more cross-cluster correlations.

Furthermore, we see that the models with k less than 4 take roughly the same amount of memory as the online model. We know from the previous results that the models with $k = 4$ are able to generate fairly accurate estimations. Moreover, if we keep increasing k, the models will generate even more accurate estimations at the cost of reasonably more memory. Overall, the offline model does not incur much more memory usage, compared to the online model. We believe that the optimal values of C and k are application dependent and we will exploit that in the future work.

We varied the threshold T from 200 to 100, thus collected more frequent itemset patterns. Specifically, we had 998 patterns collected when $T = 100$. Figure 8b plots the graphical structure of the corresponding exact complete MRF model. As can be seen, the graph is much more complex than that with $T = 200$. Correspondingly, we partition the graph into more clusters to maintain small local models. We had 65 vertices for partitioning.

Figures 12a-c present the estimation accuracy, the online times and the offline times of the models with varied C (k is fixed as 4). As illustrated, the offline model also works very well compared with the online model. It yields very close or even better estimations, while exhibiting a much better timing performance. The offline model is faster than the online model when query size exceeds 6. Moreover, its timing curve is very flat. Additionally, the larger C results in less complex models which take less time to construct.

Figure 13a-c present the estimation accuracy, the online times and the offline times of the models with varied k (C is fixed as 20). As before, we see that the models with larger k values yield better estimations, while taking more online time to estimate and more offline time to construct.

Figure 14 presents the memory usage by different models with varied k and C. The memory usage of the online model is also plotted for comparison. As can be seen, the models with larger k values take more memory. When k is small, the models with smaller C take more memory. In contrast, when k is large, the models with larger C take more memory. Specifically, the offline models take less or roughly the same memory usage up to the point where k is 4.

4.3 Results on the BMS-Webview1 Data

In this section, we report the experimental results on the BMS-Webview1 data. Again, we considered two different threshold values, $T = 200$ and $T = 100$. The resulting exact complete MRF models had 63 and 146 vertices respectively, without counting the 146 and 140 isolated vertices. Figures 15a-b present the graphical structure of the two exact complete models.

First, we report the results on the threshold 200. Figures 16a-c present the estimation accuracy, the online times and the offline times of the models with varied C while fixing k. Figures 17a-c present the performance of the models with varied k while fixing C. Figure 18 presents the memory usage of the models. Overall, the results are very similar to that on the Microsoft Web data. The offline models works very well on this dataset as well. They yield very close or even better estimations while exhibiting a much better timing performance at a reasonable offline time cost. Moreover, the edge importance based augmenting scheme can effectively compensate the correlation loss due to the graph partitioning. Furthermore, we can tradeoff the estimation accuracy for online efficiency, model complexity by tuning C and k. For a smaller C and a larger k, the model yields more accurate estimations while taking more online time to estimate and more offline time to construct. In contrast, the model with a larger C and a smaller k takes less online time and less offline time, while yielding reasonable estimations. The results on the threshold 100 are very similar to that on the threshold 200 and are thus omitted in the interest of space.

4.4 Results on Weighting Scheme and Model Refinement

Weighting scheme. We examined the effect of the accumulative weighting scheme. We compared three schemes. 1) No weighting scheme at all; 2) Use the count of itemsets of size 2; 3) Accumulative weighting scheme (consider all itemsets). Figure 19 presents the results on the Microsoft web data ($T = 100$, $C = 20$, $k = 3$). The other results are similar and are thus omitted.

The advantage of the accumulative weighting scheme over the other two schemes can be clearly seen from the results. The models derived from the graph with the accumulative weighting scheme yield more accurate estimations. The results show that the accumulative weighting scheme can effectively enhance the local structures where the strong correlations exist, thereby benefiting the following partitioning process. From the figure, we can also see that the weighting scheme of using the count of itemsets of size 2 is better than the no-weighting scheme.

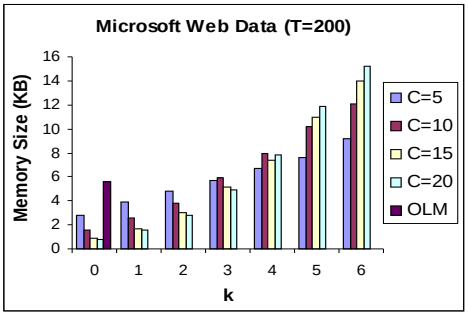


Figure 11: Memory size

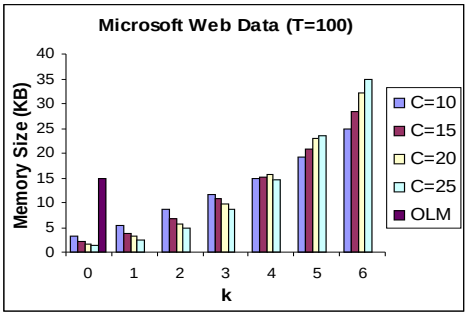


Figure 14: Memory size

Refining the offline models. Since our proposed model is built offline, we are able to afford a further model refinement. In this experiment, we raised the accuracy precision of the iterative model constructing algorithm from 0.01 to 0.001. Figure 20a-b present the results before and after the model refinement on the Microsoft web data and the BMS-Webview1 data respectively. We see that we can improve the estimation quality by further refining the offline models, and this can be done without affecting any online efficiency and memory usage. This is a significant advantage of the offline model over the online model.

5. RELATED WORK

Pavlov *et al.* [23, 22] have done significant work on exploiting probabilistic models for query approximation (selectivity estimation) on binary transaction data. They examine several models for this purpose. Besides the online local MRF model on which

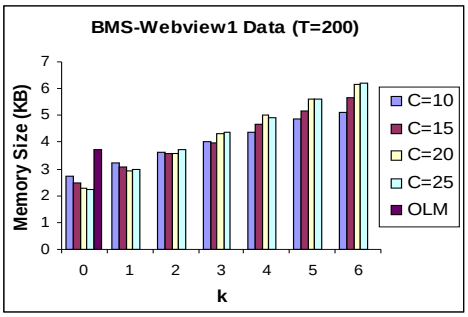


Figure 18: Memory size

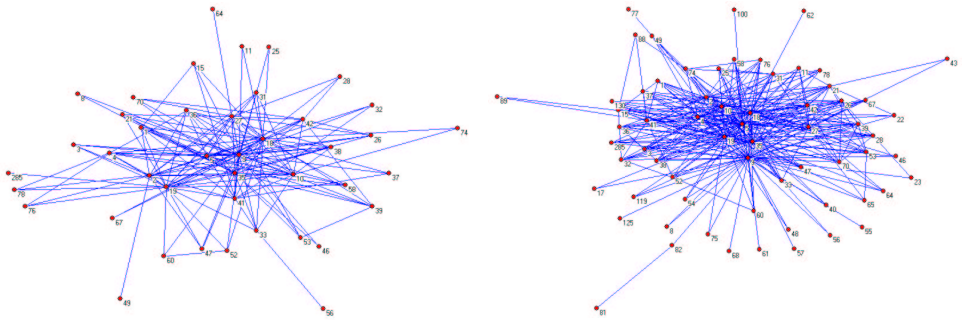


Figure 8: The graphical structures of exact MRF models (Microsoft Web): (a) threshold=200 (b) threshold=100

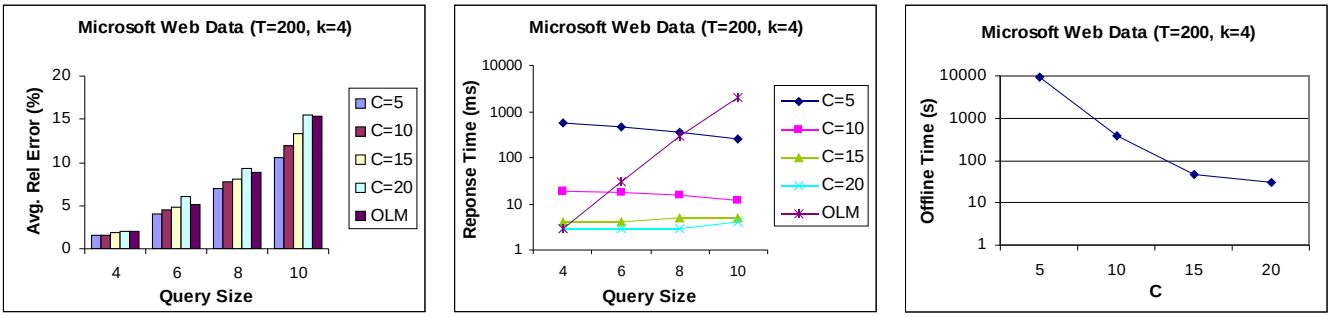


Figure 9: Varying C ($k = 4$): (a) estimation accuracy (b) online time (c) offline time

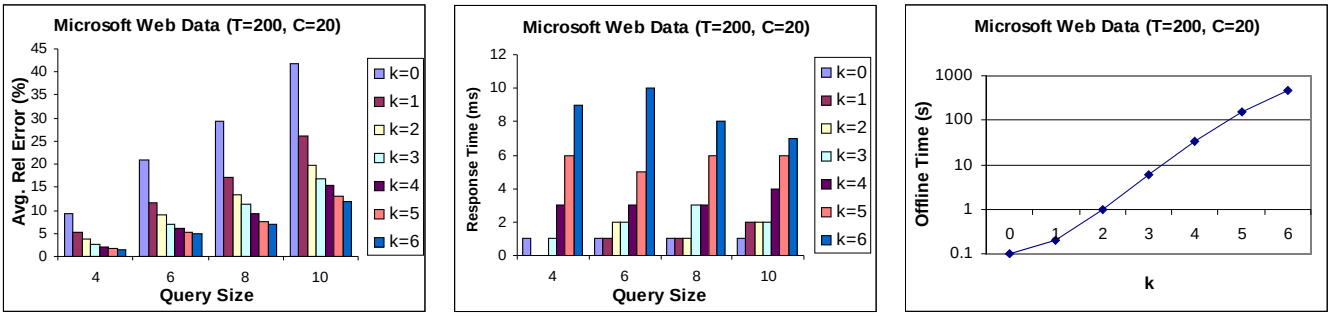


Figure 10: Varying k ($C = 20$): (a) estimation accuracy (b) online time (c) offline time

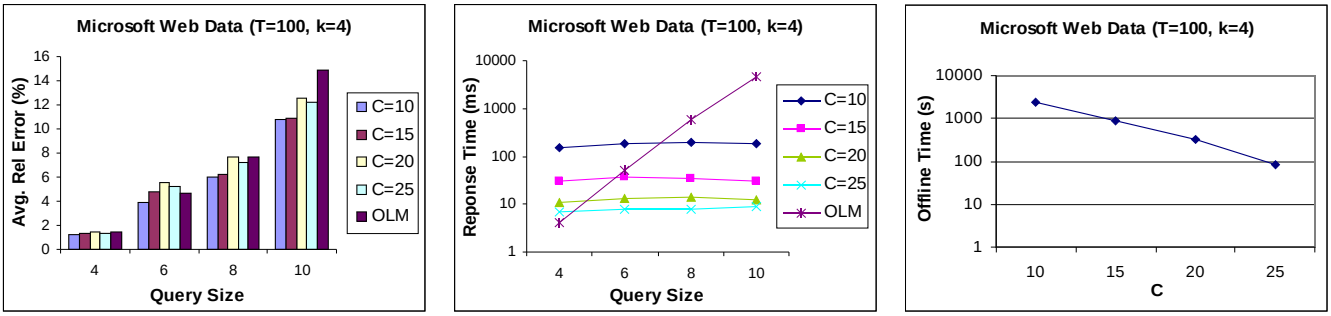


Figure 12: Varying C ($k = 4$): (a) estimation accuracy (b) online time (c) offline time

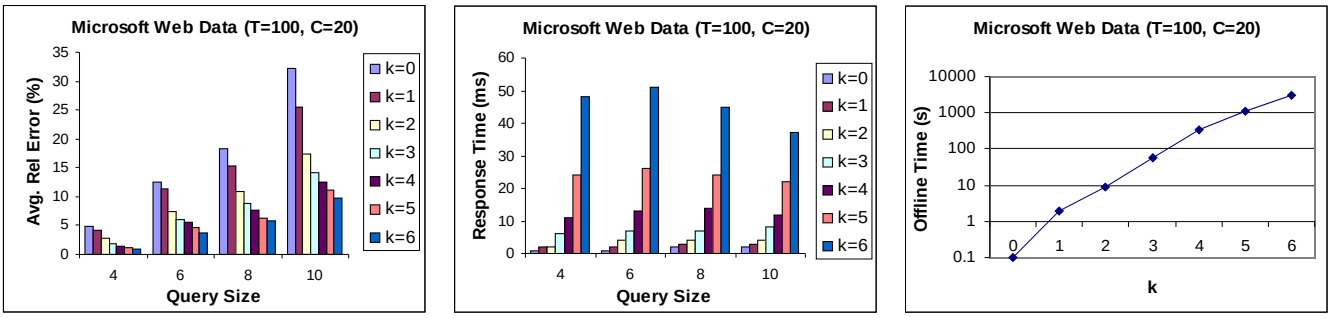


Figure 13: Varying k ($C = 20$): (a) estimation accuracy (b) online time (c) offline time

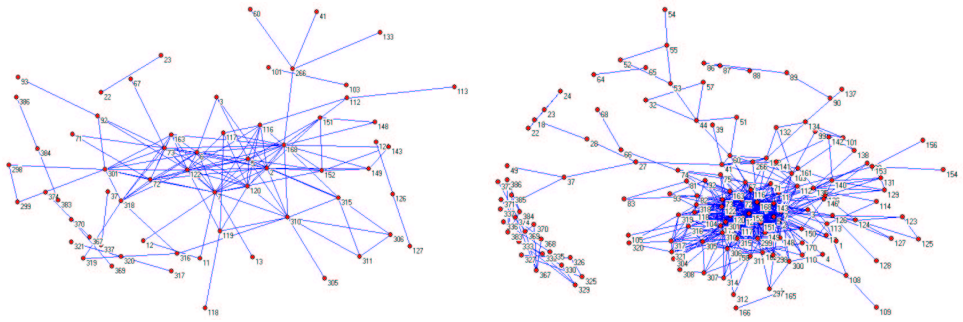


Figure 15: The graphical structure of exact MRF models (BMS-Webview1): (a) threshold=200 (b) threshold=100

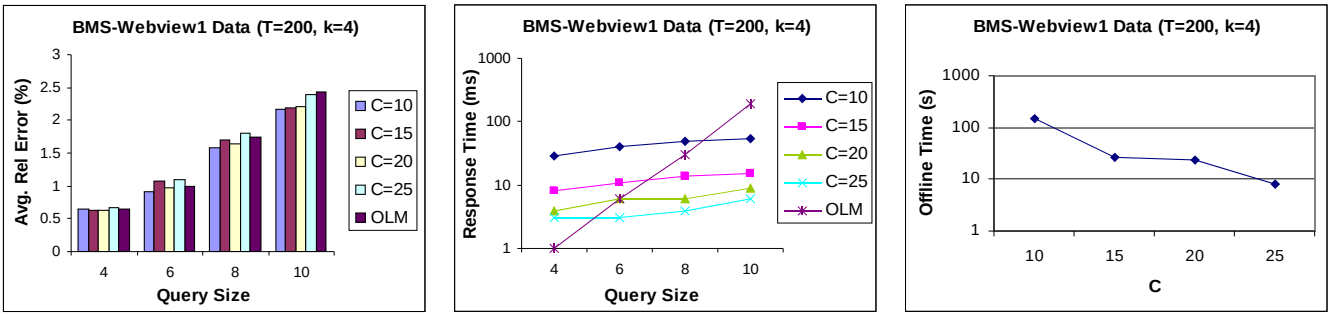


Figure 16: Varying C ($k = 4$): (a) estimation accuracy (b) online time (c) offline time

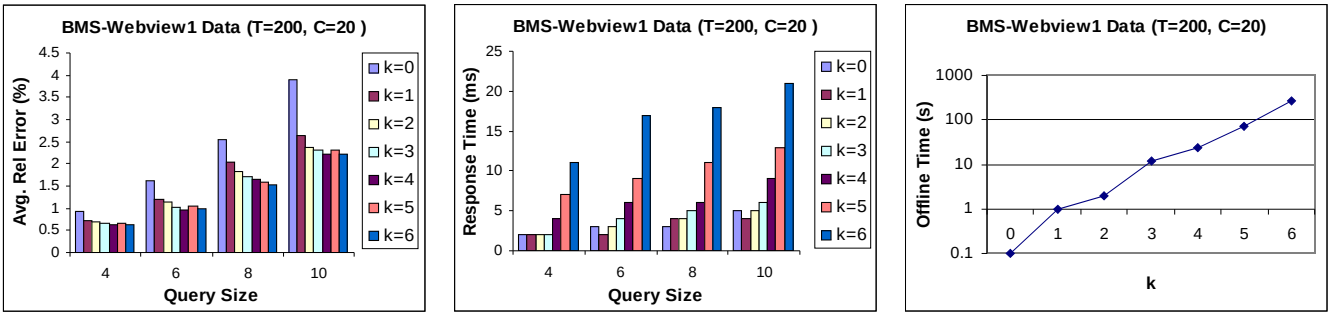


Figure 17: Varying k ($C = 20$): (a) estimation accuracy (b) online time (c) offline time

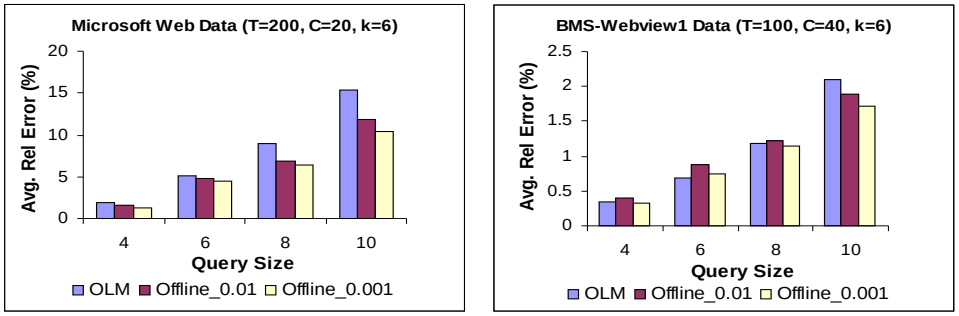


Figure 20: Refining offline models: (a) Microsoft Web (threshold=200) (b) BMS-Webview1 (threshold=100)

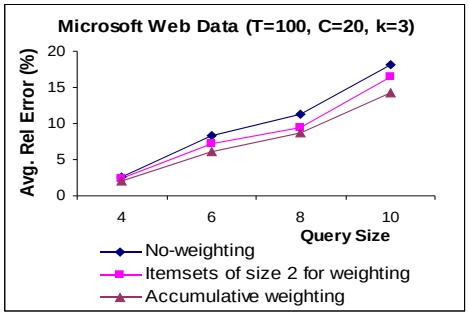


Figure 19: Weighting scheme

our study based, they also examine the Chow-Liu tree model, the Bernoulli mixture model, the ADTree model and Bayesian networks. They show that the online local MRF model gives the most accurate estimates on sparse data under similar conditions. They also employ several optimizations to speed up the online model construction, namely bucket elimination and clique tree (also called *junction tree*) algorithms. As a result, the time complexity of the online model construction can be reduced from being exponential in the number of query variables to being exponential in the induced width of the model graph.

Hollmen *et al.* [15] proposed a mixture model in which the transaction data are clustered using EM clustering algorithm. For each data chuck, the frequent itemset patterns are identified and are used to construct a probabilistic model based on the ME principle. They combine local models of all data chucks together to form a global model of the original data. They show that this mixture model yields good performance. However, due to the fact that the data are only row-wise clustered, the number of variables a local graph model must account for remains unchanged. As a result, the global model on all variables cannot be constructed. In their paper, the authors only modeled the most frequent variables. The same idea can also be applied to our approach. Essentially, the graph partitioning scheme clusters the data column-wise. We can enhance it with row-wise clustering, which in spirit is very similar to the *co-clustering* or *bi-clustering* approaches present in the literature [9, 10]. In the near future, we plan to examine if this approach yields benefits for our approach.

Goldenberg *et al.* [14] proposed an approach (SNBS) of using frequent itemset patterns to learn large Bayesian networks from sparse data. The resulting model can be used in recommender systems. Specifically, the frequent itemset patterns are collected as the evidence for constructing the Bayesian network. Further, they augment the constructed Bayesian network with edges of high mutual information for variables that have not co-occurred in the data, considering that such dependencies are not captured by the frequent itemset patterns. The same technique can again be adopted to enhance our proposed model as well.

Xing and Jordan *et al.* describe an algorithm for variational mean field inference in probabilistic graph models [30, 31]. Their approach shares some characteristics with what we propose in that they partition the original global model into disjoint clusters and subsequently a local model is constructed for each cluster based on its own evidence and the expected sufficient statistics obtained from its neighboring clusters. Local models are constructed in an iterative fashion until convergence. However, there are significant distinctions between the two approaches. First, our work is targeting on constructing an approximate global probabilistic graph model for the large binary data, while their goal is probabilistic inference. Second, in our study, we do not have the global model available all the time. We only have a partial information of the global model, i.e., its graphical structure. Moreover, we do not have any information of any potential functions associated with the graph. In contrast, in their study they have the global model available from the beginning, and they need the potential information during the iterative model construction process. Third, their modeling of neighboring interactions is significantly more complex, and moreover, the iterative phase can take a long time to converge. Both these factors significantly limit the efficiency of their approach.

6. CONCLUSION

In this paper, we have described a new approach, the offline global MRF model, to estimate the selectivity of conjunctive queries on large sparse binary data. The new approach is shown to yield comparable or even better selectivity estimations than the previous online local MRF approach. Moreover, our new approach is also significantly faster in terms of online estimation time, especially for complex queries. Interestingly, we note that our proposed offline global MRF model is an approximation of the exact global MRF model.

In the future, we will study the following issues. We would like to adopt the optimization techniques to speed up model construction and online estimation. Particularly, we would like to employ bucket elimination and clique tree algorithms in constructing local MRF models. Additionally, we can also rely on them to speed up the online estimation process. With these optimizations, we expect that the graph model can be partitioned into fewer clusters without affecting the online efficiency, which would result in better estimations. Finally, we would also like to exploit the temporal behavior of such probabilistic models as the underlying data evolve. How to incrementally maintain the models, identify interesting temporal patterns on the data through the models are both interesting and

challenging problems we would like to pursue in the future.

7. REFERENCES

[1] A. Aboulnaga, A. R. Alameldeen, and J. F. Naughton. Estimating the selectivity of xml path expressions for internet scale applications. In *Proceedings of 27th International Conference on Very Large Data Bases*, pages 591–600, 2001.

[2] A. L. Barabasi, R. Albert, H. Jeong, and G. Bianconi. Power-law distribution of the world wide web. *Science*, 287(5461):2115, March 2000.

[3] A. L. Barabasi, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, and T. Vicsek. Evolution of the social network of scientific collaborations. *Physica A*, 311:590–614, 2002.

[4] B. Bollobas. *Random Graphs*. Academic Press, London, 1985.

[5] C. M. Chen and N. Roussopoulos. Adaptive selectivity estimation using query feedback. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, pages 161–172, 1994.

[6] Z. Chen, H. V. Jagadish, F. Korn, N. Koudas, S. Muthukrishnan, R. T. Ng, and D. Srivastava. Counting twig matches in a tree. In *Proceedings of the 17th International Conference on Data Engineering*, pages 595–604, 2001.

[7] Z. Chen, F. Korn, N. Koudas, and S. Muthukrishnan. Selectivity estimation for boolean queries. In *Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 216–225, 2000.

[8] Z. Chen, F. Korn, N. Koudas, and S. Muthukrishnan. Generalized substring selectivity estimation. *J. Comput. Syst. Sci.*, 66(1):98–132, February 2003.

[9] Y. Cheng and G. Church. Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 93–103, 2000.

[10] I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 89–98, 2003.

[11] A. J. Enright, V. Kunin, and C. A. Ouzounis. Protein families and tribes in genome sequence space. *Nucleic Acids Research*, 31(15):4632–4638, 2003.

[12] B. J. Frey. *Graphical Models for Machine Learning and Digital Communication*. MIT Press, Cambridge, MA, 1998.

[13] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799–805, February 2004.

[14] A. Goldenberg and A. Moore. Tractable learning of large bayes net structures from sparse data. In *Proceedings of the twenty-first international conference on Machine learning*, 2004.

[15] J. Hollmen, J. K. Seppanen, and H. Mannila. Mixture models and frequent sets: Combining global and local methods for 0-1 data. In *Proceedings of the Third SIAM International Conference on Data Mining*, 2003.

[16] H. V. Jagadish, R. T. Ng, and D. Srivastava. Substring selectivity estimation. In *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 249–260, 1999.

[17] S. Jaroszewicz and D. A. Simovici. Interestingness of frequent itemsets using bayesian networks as background knowledge. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 178–186, 2004.

[18] G. Karypis and V. Kumar. 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.

[19] G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.*, 48(1):96–129, 1998.

[20] P. Krishnan, J. S. Vitter, and B. R. Iyer. Estimating alphanumeric selectivity in the presence of wildcards. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 282–293, 1996.

[21] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[22] 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.

[23] D. Pavlov and P. Smyth. Probabilistic query models for transaction data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 164–173, 2001.

[24] J. Pereira-Leal, A. Enright, and C. Ouzounis. Detection of functional modules from protein interaction networks. *Proteins*, 54(1):49–57, January 2004.

[25] N. Polyzotis, M. N. Garofalakis, and Y. E. Ioannidis. Approximate xml query answers. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 263–274, 2004.

[26] N. Polyzotis, M. N. Garofalakis, and Y. E. Ioannidis. Selectivity estimation for xml twigs. In *Proceedings of the 20th International Conference on Data Engineering*, pages 264–275, 2004.

[27] V. Poosala and Y. E. Ioannidis. Selectivity estimation without the attribute value independence assumption. In *Proceedings of 23rd International Conference on Very Large Data Bases*, pages 486–495, 1997.

[28] V. Poosala, Y. E. Ioannidis, P. J. Haas, and E. J. Shekita. Improved histograms for selectivity estimation of range predicates. In *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 294–305, 1996.

[29] S. van Dongen. A cluster algorithm for graphs. *Technical Report INS-R0010, National Research Institute for Mathematics and Computer Science in the Netherlands, Amsterdam*, May 2000.

[30] 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.

[31] E. Xing, M. Jordan, and S. Russell. Graph partition strategies for generalized mean field inference. In *Uncertainty in Artificial Intelligence*, pages 602–610, 2004.

[32] 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.

[33] S. H. Yook, Z. N. Oltvai, and A. L. Barabasi. Functional and topological characterization of protein interaction networks. *PROTEOMICS*, 4(4):928–942, February 2004.