

Learning Approximate MRFs From Large Transactional Data ^{*}

Chao Wang and Srinivasan Parthasarathy

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

Abstract. In this paper we consider the problem of learning approximate Markov Random Fields (MRFs) from large transactional 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. Empirical evaluation on real data demonstrates the advantage of our approach for the selectivity estimation problem over extant solutions.

1 Introduction

In this paper we address the problem of learning approximate *Markov Random Fields* (MRF) from large transactional data. Examples of such data include market basket data, web log data, etc. Such data can be represented by a high-dimensional binary data matrix, with an entry (i, j) takes a value of one (zero) if the item j is (not) in the basket i . 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 and causal correlations among attributes in data and have been successfully applied in applications such as selectivity estimation in query optimization [1, 2], link analysis/recommender systems [3, 4] and bioinformatics [5].

Pavlov *et al.* [2] propose a *Maximum Entropy* (ME) model based on frequent itemsets to tackle the selectivity estimation problem. The ME model is essentially 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 expensive iterative process, such a just-in-time model construction approach is not appropriate in settings where online estimation time is crucial. The alternative is to first learn a *global model* offline. Subsequently, queries can be answered on the fly using standard probabilistic inference methods [6–8]. The advantages are a more accurate model (relies on complete information from all the data) and huge online performance gains. The critical

^{*} This work is supported by DOE Award No. DE-FG02-04ER25611 and NSF CAREER Grant IIS-0347662.

challenge is that a global model may be prohibitive to compute from large high dimensional transactional data. To address this problem, in this paper, we consider employing frequent itemsets to learn approximate global MRFs on large transactional data. Frequent itemsets capture important distribution information of the data. Hollmen *et al.* [9] proposed to use frequent itemsets to learn mixture models from transactional data on the local scale. Goldenberg *et al.* [4] proposed an approach (SNBS) of using frequent itemsets to learn large Bayesian networks from transactional data. We conduct an empirical study on real datasets to show the efficiency and effectiveness of our model on solving the selectivity estimation problem. In certain cases, it can outperform a state-of-the-art solution by up to three orders of magnitude in terms of online estimating time while delivering similar estimations.

2 Background

Let \mathcal{I} be a set of items, i_1, i_2, \dots, 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 transactional dataset is a collection of itemsets, $D = \{t_1, t_2, \dots, t_n\}$, where $t_i \subseteq \mathcal{I}$. For any itemset α , we write the transactions that contain α as $D_\alpha = \{t_i | \alpha \subseteq t_i \text{ and } t_i \in D\}$. In the probabilistic model context, each item is modeled as a random variable¹.

Definition 1 (*Frequent itemset*): For a transactional dataset D , an itemset α is frequent if $|D_\alpha| \geq \sigma$, where $|D_\alpha|$ is called the support of α in D , 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_{C_i} is a non-negative potential function over the variables of clique C_i and $\frac{1}{Z(\psi)}$ is a normalization term.

Using Frequent Itemsets to Learn an MRF: The idea of using frequent itemsets to learn an MRF was first proposed by Pavlov *et al.* [2]. A k -itemset and its support represents a k -way statistic and can be viewed as a constraint on the true underlying distribution that generates the data. Given a set of itemset constraints, a *Maximum Entropy* (ME) distribution satisfying all these constraints is selected as the estimate for the true underlying distribution. This ME distribution is essentially equivalent to 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 version of the algorithm given by Jelinek [10]. Efficient inference is crucial to the running time of the learning algorithm. We call those models learned through exact inference procedures *exact* models. The *junction tree* algorithm is a commonly-used exact inference engine for probabilistic models. The time complexity of the algorithm is exponential in the treewidth of the underlying model. For real-world models, it is quite common that the treewidth will be well above 20, making learning exact models intractable. As a result, we have to resort to learning approximate models.

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

```

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

```

Fig. 1. Iterative scaling algorithm

3 Learning Approximate MRFs

Before discussing our proposed approach, let us consider an extreme case in which the whole graphical model consists of a set of disjoint non-correlated components. Then the joint distribution can be obtained in a straightforward fashion according to Lemma 1.

Lemma 1. ² *Given an undirected graphical model G subdivided into disjoint components D_1, D_2, \dots, 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: $p(X) = \prod_{i=1}^n p(X_{D_i})$*

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 such a group a *variable-cluster*. Then a local MRF is inferred on each *variable-cluster*. In the end we aggregate all 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 partitions, the divide-and-conquer style approach gives the exact estimate of the full model. Even for an imperfect 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 [11]: 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 partitions.

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*. To accomplish this we ensure that the weight of edges reflect the strength of correlations between variables. We have the collection of all frequent itemsets. In particular, 2-itemsets specify the connectedness structure of the graph, and their

² This follows immediately from the *global Markov property* of the MRF.

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. Our hypothesis is that taking into consideration all the 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 tend to strengthen the graph regions that involve closely related itemsets in the hope that the edges within these regions will not be broken in the partitioning. 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.

3.2 Interaction Importance & Treewidth Based Variable-Cluster Augmentation

The *variable-clusters* produced by the k -MinCut partitioning scheme are disjoint. Intuitively, there can be correlation information that is lost during the partitioning. To compensate for this loss, we propose an interaction importance based *variable-cluster* augmenting scheme. The idea is that we allow each *variable-cluster* to grow outward. More specifically, it attracts and absorbs most 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 augmentation. 1-hop neighboring vertices are first considered for the augmentation, followed by 2-hop neighboring vertices and so on. Meanwhile, we still follow the interaction importance criteria. The resultant 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 is that more interactions across partitions will be accounted for in a computationally controllable manner, leading to a more accurate global model.

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 transactional 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 strong correlations are compensated for during the *variable-cluster* augmentation. As such, we believe that the proposed global model reasonably approximates the exact model. Figure 2 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 : M, global MRF;
1. Construct a weighted graph G from F;
   //G specifies graphical structure of the exact MRF;
2. k-MinCut G;
3. for each graph partition  $G_i$ 
4.    $G'_i \leftarrow \text{augment}(G_i, g)$ ;
5.   Pick itemsets  $F_i$  related to  $G'_i$ ;
6.    $M_i \leftarrow \text{LearnLocalMRF}(F_i)$ ;
7.   add  $M_i$  to M;
8. return M;

```

Fig. 2. 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 that has the largest intersection with the current query (i.e., covers most query variables). Then we pick the next local model that covers most uncovered variables in the query. This covering process will be repeated until we cover all query variables. 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 2.³ *Given an undirected graphical model G subdivided into n overlapped components, if there exists an enumeration of these n components, i.e., C_1, C_2, \dots, 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: $p(X) = \frac{\prod_{i=1}^n p(X_{C_i})}{\prod_{i=2}^n p(X_{C_i \cap (\cup_{j=1}^{i-1} C_j)})}$*

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. However, it is possible to have cyclic dependencies among the decomposed pieces. Therefore, the greedy inference scheme is a heuristic. Also, we note that our global model is not globally consistent in that there can exist inconsistencies across the local models. However, we expect that the global model is *nearly consistent* since two correlated local models support the same evidence (itemsets) regarding their shared variables.

4 Experimental Results

In this section, we examine the performance of our proposed model for the selectivity estimation problem on real datasets. We compare our proposed model against the previous approach [2] where a local MRF over query variables is learned on the fly for every query, referred to as the online local MRF approach (abbreviated as *OLM*).

³ The complete proof can be found in the full version of this paper [12].

Experimental Setup: All the experiments were conducted on a Pentium 4 2.66GHz machine with 1GB RAM running Linux 2.6.8. The MRF learning algorithm was implemented in C++. We used *apriori* [13] to collect frequent itemsets and *Metis* [11] to obtain a k -MinCut of the exact graphical model.

Datasets: We used two publicly available datasets: the Microsoft Anonymous Web dataset (kdd.ics.uci.edu) with 32711 transactions and 294 items; the BMS-Webview1 dataset (fimi.cs.helsinki.fi) with 59602 transactions and 497 items. **Query Workloads:** We considered the workloads consisting of conjunctive queries (following the same practice in [2]) of different sizes. **Performance Metrics:** 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. 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 $|\sigma - \hat{\sigma}| / \sigma$, where σ is the true selectivity and $\hat{\sigma}$ is the estimated selectivity.

Results on the Microsoft Web Data: We used the support threshold of 20 to collect the frequent itemsets, which resulted in 9901 frequent itemsets. According to the *Maximum Cardinality Search* (MCS)-ordering heuristic [14], the treewidth of the resulting MRF is 28, for which learning the exact model is considered intractable.

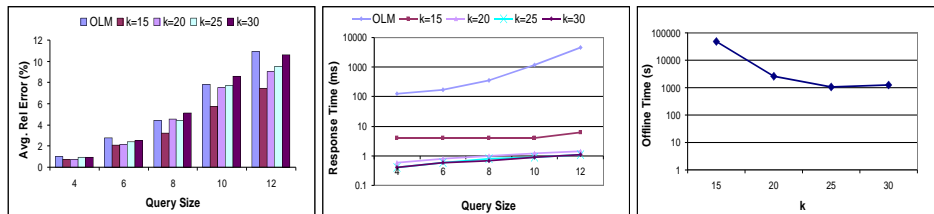


Fig. 3. Varying k ($g = 5$): (a) estimation accuracy (b) online time (c) offline time

Figure 3a presents the estimation accuracy when k is varied (g is fixed as 5) for queries of different sizes. As seen, our approach gives very close or even better estimations compared with OLM. These results are not surprising since for OLM, we only use the local information to estimate the selectivity. However, for our model, we rely on the global information to make the estimation. An obvious trend that stands out is that as the query size increases, the quality of the estimations degrades. This is expected since for larger sized queries, estimation errors grow for both approaches. Another observation is that the estimations are more accurate when we use fewer *variable-clusters*. This is because with fewer *variable-clusters*, the information loss due to the graph partitioning is smaller, thus we capture better the correlations between partitions. Figure 3b illustrates how the online times depend on k . It can be clearly seen the significant growth of the online times taken by OLM (note the Y-axis scale). The extreme online timing efficiency of our model can be clearly seen from the results. In most cases, it outperforms OLM by two to three orders of magnitude. Further, we see that the smaller k results in

higher online estimating time. This is because the smaller k results in more complex local models. 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 3c presents the offline learning times of our model when varying k . An obvious trend is that as we increase k , overall the learning cost of our model decreases significantly. This is because the larger k results in less complex local models.

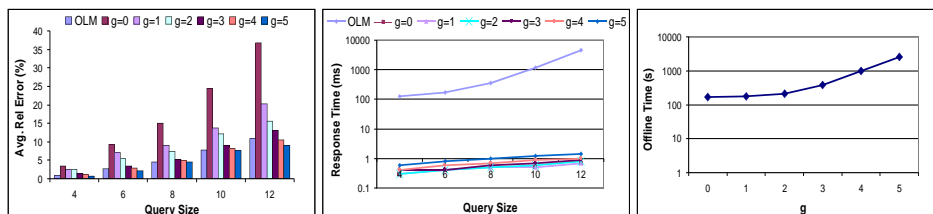


Fig. 4. Varying g ($k = 20$): (a) estimation accuracy (b) online time (c) offline time

Figure 4a presents the estimation accuracy when varying g (k is fixed as 20). As one can see, 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. Our model approximates the exact model better when more correlations across the local models are compensated for. Figure 4b 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 expected since the larger g results in more complex models (similar to the case of the smaller k). Figure 4c presents the offline learning times of our model when varying g . An obvious trend is that as we increase g , the time cost increases significantly, which is again expected.

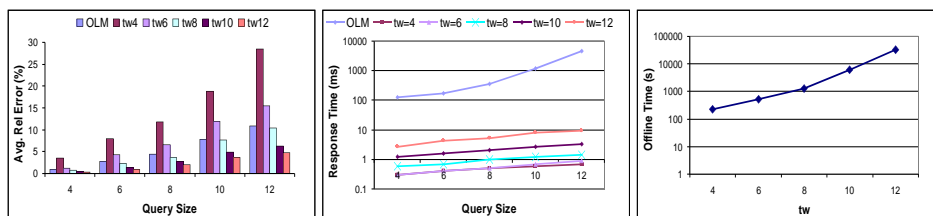


Fig. 5. Varying tw ($k = 25$): (a) estimation accuracy (b) online time (c) offline time

Figure 5a-c present the estimation accuracy, the online estimating times and the offline learning times of our model when the treewidth based augmentation optimization is used (k is fixed as 25). As 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 OLM are 0.99%, 2.76%, 4.45%, 7.82% and 10.9%, respectively.

The results on the BMS-Webview1 dataset overall are quite similar to that on the Microsoft Web dataset and can be found in [12].

5 Conclusion

In this paper, we have described a new approach to learning an approximate MRF on large transactional data. Our proposed approach has been 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 the consistency of the model. Furthermore, we would like to investigate the use of the approximate inference techniques during the model learning process. Finally, it would be interesting to exploit the learned models on various link analysis tasks.

References

1. Getoor, L., Taskar, B., Koller, D.: Selectivity estimation using probabilistic models. In: Proceedings of the SIGMOD Conference. (2001) 461–472
2. 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
3. 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
4. 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)
5. Friedman, N.: Inferring cellular networks using probabilistic graphical models. *Science* **303** (2004) 799–805
6. Lauritzen, S., Spiegelhalter, D.: Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B (Methodological)* **50** (1988) 157224
7. Jordan, M.I., Kearns, M.J., Solla, S.A.: An introduction to variational methods for graphical models. *Machine Learning* **37** (1999) 183–233
8. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Understanding belief propagation and its generalizations. In: Technical report, Mitsubishi Elect. Research Labs., Inc. (2001)
9. Hollmen, J., Seppanen, J.K., Mannila, H.: 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)
10. Jelinek, F.: *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, MA (1998)
11. Karypis, G., Kumar, V.: Multilevel k-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.* **48** (1998) 96–129
12. Wang, C., Parthasarathy, S.: Learning approximate mrfs from large transaction data. In: The Ohio State University, Technical Report. (2006)
13. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases. (1994) 487–499
14. 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)