Two-level Mixtures of Markov Trees

François Schnitzler and Louis Wehenkel

Université de Liège, Department of EECS and GIGA-Research,
Grande Traverse, 10 - B-4000 Liège - Belgium,
fschnitzler@ulg.ac.be,L.Wehenkel@ulg.ac.be

Abstract. We study algorithms for learning *Mixtures of Markov Trees* for density estimation. There are two approaches to build such mixtures, which both exploit the interesting scaling properties of Markov Trees. We investigate whether the maximum likelihood and the variance reduction approaches can be combined together by building a two level Mixture of Markov Trees. Our experiments on synthetic data sets show that this two-level model outperforms the maximum likelihood one.

Keywords: mixture models, Markov Trees, bagging, EM algorithm

1 Introduction

A Bayesian Network is a probabilistic graphical model that represents a joint probability density over a finite set $\mathcal{X}$ of n variables $\{X_1, X_2, ..., X_n\}$ by exploiting independence relationships between them, thus reducing the number of parameters needed [10]. Each variable of the problem is a node of a Directed Acyclic Graph (DAG) $\mathcal{G}$ whose structure encodes the set of conditional independence relationships between variables, and defines a factorization of the joint probability density as a product

$$P(\mathcal{X}) = \prod_{i=1}^{n} P(X_i | Pa_{\mathcal{G}}(X_i)) \ , \tag{1}$$

where $P(X_i|Pa_{\mathcal{G}}(X_i))$ is the probability density of variable X_i conditionally to the set of its parents $Pa_{\mathcal{G}}(X_i)$ in the graph.

Such a model can be defined using expert knowledge due to the graphical decomposition of the factorization, or estimated by machine learning algorithm using a data set $\mathbf{D}$ containing p realizations or samples of the probability density. In the latter case, learning the parameters for a given structure is straightforward (and will not be discussed here), but learning the DAG $\mathcal{G}$ is NP-hard when the class of structures considered is unconstrained [4], and in practice not possible over a few thousands variables [2, 5]. Inference, i.e. using the model to answer a query such as computing the likelihood of a problem instance or estimating the conditional probability density of a subset of variables given observed values of another subset of variables, is NP-hard too [8].

Reducing the set of candidate structures leads to more scalable algorithms, see e.g. [5]. In that sense, a rather interesting set of structures is the class of Markov Trees, composed of $n-1$ edges and where each variable has at most one parent. Indeed, inference on such a graph is of linear complexity in n, and learning a Markov Tree maximizing the data likelihood can be done by the Chow-Liu algorithm [3] for a computational cost essentially quadratic in n.

Mixtures of Markov Trees have been developed to improve Markov Trees while preserving their algorithmic scalability. Such a model is composed of a set $\hat{\mathcal{T}} = \{T_1, \ldots, T_m\}$ of m elementary Markov Tree densities and a set $\{\mu_i\}_{i=1}^m$ of weights so that $\mu_i \in [0,1]$ and $\sum_{i=1}^m \mu_i = 1$, defining a convex combination :

$$P_{\hat{\mathcal{T}}}(\mathcal{X}) = \sum_{k=1}^{m} \mu_k P_{T_k}(\mathcal{X}) \ . \tag{2}$$

Algorithms for learning this model can be divided into two categories. In the first one, learning the mixture is viewed as a global optimization problem aiming at maximizing the data likelihood. It can be (localy) solved for m fixed in advance using the EM algorithm to partition the data and to learn a tree for each subset [9]. In the second approach, the mixture model aims at variance reduction and can be viewed as an approximation of Bayesian learning in the space of Markov Tree structures. A sequence of trees is generated by a procedure ranging from purely random structures to learning on bootstrap replicas [1].

A combination of those two approaches was developed in [6] in the form of an approximation of Bayesian averaging (through a MCMC walk) in the space of mixtures of Markov Trees, and of cubic complexity.

In this work, we propose another way of combining those two frameworks. We introduce a two-level meta-algorithm for learning mixtures of Markov Trees, that consists in pluging the second class of methods (model averaging) into the first one (maximum likelihood mixtures). The former replaces in the latter the Chow-Liu algorithm by a procedure for learning a mixture of m_k trees based on each subset k of the data set. In other words, we propose to use model averaging for variance reduction in the fitting of each subset of the data set created by the EM algorithm. The proposed model is thus:

$$P_{\hat{\mathcal{T}}}(\mathcal{X}) = \sum_{k=1}^{m} \mu_k P_{\hat{\mathcal{T}}_k}(\mathcal{X}) \tag{3}$$

$$P_{\hat{\mathcal{T}}_k}(\mathcal{X}) = \sum_{j=1}^{m_k} \lambda_{k,j} P_{T_{k,j}}(\mathcal{X}) \quad \forall k \ , \tag{4}$$

where m is the number of terms in the upper maximum likelihood level optimization of the algorithm, and $T_{k,j}$ is the jth Markov Tree in the lower level mixture learned on subset k of the data set, and $\lambda_{k,j}$ denotes its weight.

In order to test this meta-algorithm, we instance it in Sec. 3, resulting in a new algorithm for learning mixtures of Trees. We use as building blocks the upper and lower level methods which we describe in Sec. 2. Finaly, we test this algorithm on synthetic problems in Sec. 4 and conclude.

2 Base Methods

Many learning algorithms for mixtures of Markov Trees, including those we will use as building blocks, require the computation of a Markov Tree that maximizes the likelihood of a data set $\mathbf{D}$. The Chow-Liu algorithm [3] learns such a tree by estimating a matrix of mutual informations between each pair of variables based on that data set, extracting a maximum weight spanning tree from this matrix, and learning the parameters of the corresponding Bayesian network by counting frequencies in the data set. We will denote it by **Chow-Liu**($\mathbf{D}$).

2.1 EM Learning of Mixtures of Markov Trees

The expectation-maximization (EM) algorithm can be used to find the maximum likelihood estimate of a mixture of m Markov Trees, with m fixed in advance [9]. The weights of the mixture are considered hidden parameters, and the procedure alternates between optimizing those parameters, which (softly) partition the data set, and optimizing a tree on each data subset. A soft partition of $\mathbf{D}$ is a a set of k new datasets $\mathbf{D}'_k$ where a weight $\gamma_k(i)$ is associated to each sample $\mathbf{D}'_k[i] = \mathbf{D}[i]$.

Algorithm 1 (EM algorithm for mixtures of Markov Trees (EM-MT)).

1. *Initialize* $\hat{\mathcal{T}}$ *and* μ*;*
2. *Repeat until convergence:*
   (a) $\gamma_k(i) = \dfrac{\mu_k P_{T_k}(\mathbf{D}[i])}{\sum_{k=1}^{m} \mu_k P_{T_k}(\mathbf{D}[i])} \ \forall i, k;$
   (b) $\mu_k = \dfrac{\sum_{i=1}^{p} \gamma_k(i)}{p} \ \forall k;$
   (c) *Repeat for* $k = 1, \cdots, m$*:*
      i. $\mathbf{D}'_k = \mathit{weightDataSet}(\mathbf{D}, \{\gamma_k\}_{i=1}^{p});$
      ii. $T_k = \boldsymbol{\mathit{Chow\text{-}Liu}}(\mathbf{D}'_k).$

2.2 Mixtures of Bagged Markov Trees

Bagging is a meta-algorithm that combines the randomized results produced by applying an optimal algorithm on m different bootstrap replicas of an original data set. A bootstrap replica $\mathbf{D}'$ is obtained by uniformly and independently drawing p samples from $\mathbf{D}$ and copying them in $\mathbf{D}'$. The result of the bagging algorithm is an average between the m models learnt from the m replicas that typically exhibits a lower variance than a model learned directly from the original data set.

In the context of mixtures of Markov Trees, bagging was shown to outperform a tree obtained by the Chow-Liu algorithms on small data sets. It was also demonstrated that learning the parameters of each tree in $\hat{\mathcal{T}}$ on $\mathbf{D}$ leads to better accuracy than on the replica $\mathbf{D}'$ used to generate its structure, and we will therefore use this approach [11].

Algorithm 2 (Mixture of bagged Markov Trees (Bag-MT)).

1. $\hat{\mathcal{T}} = \{\}$; $\mu = \{1/m, \cdots, 1/m\}$;
2. *Repeat for* $j = 1, \cdots, m$:
   (a) $\mathbf{D}' =$ *bootstrap*($\mathbf{D}$);
   (b) $\hat{\mathcal{T}} = \hat{\mathcal{T}} \cup \{\textbf{Chow-Liu}(\mathbf{D}')\}$.

3 Two-level Mixture of Markov Trees

In this section, we propose a new algorithm for building a mixture of Markov Trees by combining the two methods described in the previous section.

The EM algorithm learns m maximum likelihood Markov Trees on a (soft) partition of size m of the data set, hence the number of samples considered by each tree learning algorithm is smaller than the size of the original data set $\mathbf{D}$. On the other hand, a mixture of bagged Markov Trees has been shown to be increasingly better than a single maximum likelihood Markov Tree when the number of samples decreases [11]. Therefore it seems quite natural to replace the maximum likelihood Markov Tree in the EM algorithm by a mixture of bagged Markov Trees. This leads to a two-level mixture of Markov Trees, as described in eq. 4.

Replacing each optimal Chow-Liu tree by a bagged mixture of trees might either slow down the convergence of the EM algorithm or lead to premature convergence. We consequently decided to learn a mixture of bagged Markov Trees on each data set $\mathbf{D}'_k$ only after the convergence of the EM algorithm with single Chow-Liu trees, as described in Algorithm 3.

Algorithm 3 (EM-bagging mixture of Markov Trees (EM-Bagg-MT)).

1. $\hat{\mathcal{T}} = \textbf{EM-MT}(\mathbf{D})$;
2. $\gamma_k(i) = \dfrac{\mu_k P_{T_k}(\mathbf{D}[i])}{\sum_{k=1}^{m} \mu_k P_{T_k}(\mathbf{D}[i])} \ \forall i, k$;
3. *Repeat for* $k = 1, \cdots, m$:
   (a) $\mathbf{D}'_k = \textit{weightDataSet}(\mathbf{D}, \gamma_k)$;
   (b) $\hat{\mathcal{T}}_k = \textbf{\textit{Bag-MT}}(\mathbf{D}'_k)$.

4 Simulations

We compared algorithms **EM-MT** and **EM-Bagg-MT** on 5 synthetic densities over 200 binary variables times 6 data sets, for different sample sizes p ranging from 120 to 8000. Each density is encoded by a Bayesian network whose structure is generated by uniformly drawing the number of parents for each X_i in $[0, max(5, i-1)]$ and by randomly selecting these parents in $\{X_1, ..., X_{i-1}\}$. Parameters of the densities are drawn from uniform Dirichlet distributions [11]. Monte-Carlo estimation (for computational reasons) of the Kullback-Leibler divergence [7] was used for quantifying the performance of the models learned, and we report here an average of the results on 6×5 datasets for each sample size.

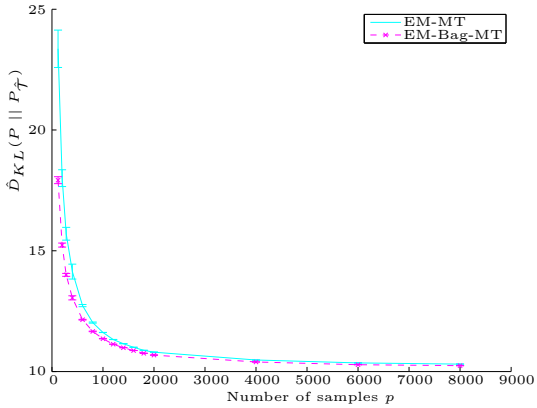


Fig. 1. A comparison between **EM-MT** and **EM-Bag-MT** shows the superiority of the latter for $n = 200$, $m = 2$ ($m_k = 10\ \forall k$) and p ranging from 120 to 8000. Horizontal axis: sample size p; vertical axis KL divergence to the target density estimated by Monte-Carlo and averaged over 5 target densities and 6 training sets.

4.1 Results

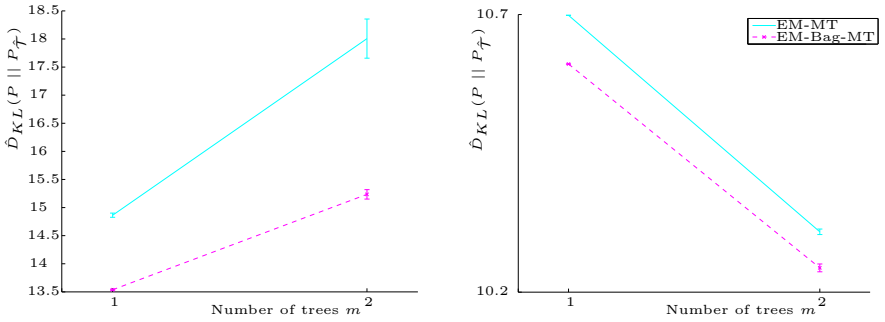
Figure 1 displays a comparison of the EM algorithm (m=2) with and without bagging (with $m_k = 10$). **EM-Bagg-MT** is always superior, but its advantage shrinks when the sample size p increases. This is to be expected, since larger sample sizes reduce the interest of bagging.

The EM algorithm is however not always interesting. In Fig. 2 it can be seen that increasing m from 1 (equivalent to the Chow-Liu algorithm) to 2 in **EM-MT** is counterproductive when samples are scarce. But $m = 2$ becomes increasingly better when p rises. At 1800 samples (not shown) both are equivalent.

5 Conclusion

We have proposed a new meta-algorithm for learning mixtures of Markov Trees combining the maximum likelihood approach and the Bayesian one, by replacing the Chow-Liu algorithm in a maximum likelihood approach by a Bayesian-inspired mixture of Markov Trees, effectively building a two stage mixture of trees. We applied this approach to the EM algorithm and the bagged mixture of Markov Trees and demonstrated its interest on synthetic densities over 200 variables. We also noted that at low sample size an EM mixture of two terms is worse than a single term; and that with a fixed number of terms the advantage of the EM-bagged mixture over the EM one reduces when the sample size p grows, which is a consequence of the increasing size of the bootstrap replicas.

However, the optimal number of terms m of the maximum likelihood mixture increases in practice with the sample size p, which will slow down the increase in size of the bootstrap replicas



(a) 200 samples. (b) 8000 samples.

Fig. 2. The number of samples influences the relevance of the EM algorithm. When samples are few, a single Chow-Liu tree is better than a mixture of 2 maximum likelihood trees, but the opposite is true for larger samples.

Acknowledgments. François Schnitzler is supported by a F.R.I.A. scholarship. This work was also funded by the Biomagnet IUAP network of the Belgian Science Policy Office and the Pascal2 network of excellence of the EC. The scientific responsibility is the authors'.

References

1. Ammar, S., Leray, P., Schnitzler, F., Wehenkel, L.: Sub-quadratic Markov tree mixture learning based on randomizations of the Chow-Liu algorithm. In: The Fifth European Workshop on Probabilistic Graphical Models. pp. 17–24 (2010)
2. Auvray, V., Wehenkel, L.: Learning inclusion-optimal chordal graphs. In: Proceedings of 24th Conference on Uncertainty in Artificial Intelligence. pp. 18–25 (2008)
3. Chow, C., Liu, C.: Approximating discrete probability distributions with dependence trees. IEEE Trans. Inf. Theory 14, 462–467 (1968)
4. Cooper, G.: The computational complexity of probabilistic inference using bayesian belief networks. Artificial Intelligence 42(2-3), 393–405 (1990)
5. Elidan, G., Gould, S.: Learning bounded treewidth bayesian networks. JMLR 9, 2699–2731 (2008)
6. Kirshner, S., Smyth, P.: Infinite mixtures of trees. In: ICML '07: Proceedings of the 24th international conference on Machine learning. pp. 417–423. (2007)
7. Kullback, S., Leibler, R.: On information and sufficiency. ANN MATH STAT 22(1), 79–86 (1951)
8. Kwisthout, J.H., Bodlaender, H.L., van der Gaag, L.: The necessity of bounded treewidth for efficient inference in bayesian networks. In: the 19th European Conference on Artificial Intelligence. pp. 623–626 (2010)
9. Meila, M., Jordan, M.: Learning with mixtures of trees. JMLR 1, 1–48 (2001)
10. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann (1988)
11. Schnitzler, F., Leray, P., Wehenkel, L.: Towards sub-quadratic learning of probability density models in the form of mixtures of trees. In: 18th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2010). pp. 219–224. (2010)