

Learning Structure in Directed Graphical Models

Edward Jay Kreps
jay@cs.ucsc.edu

June 11, 2004

Abstract

Graphical models have become a popular method for creating models of complex phenomena. Efficient algorithms have been developed for learning parameter values and carrying out inference when the structure of the graph in question is known *a priori*. The more difficult problem of learning a suitable graphical structure from data has proved more resistant. In this paper I discuss the difficulties that inhibit structure learning, and survey the more popular methods for surmounting them.

1 Introduction

Directed graphical models consist of a directed acyclic graph, \mathcal{G} , with nodes \mathcal{V} and edges \mathcal{E} , and a set of random variables $X = \{X_v : v \in \mathcal{V}\}$. The variable X_i corresponds to the i th node of \mathcal{G} , and X_{π_i} denotes the variables corresponding to the parents of i in \mathcal{G} . The graph encodes a set of conditional independence statements about the distribution of the random variables in X , specifically that it factors as follows

$$P(x_1, x_2, \dots, x_n) = \prod_{i=1}^n P(x_i | x_{\pi_i})$$

where x_i is a particular value for the variable X_i .

Given a set of l training examples, $D = (x^{(1)}, y^{(1)}), \dots, (x^{(l)}, y^{(l)})$, our goal is to find the graph G and the conditional probabilities θ that will minimize the error in the prediction of future values of y . $x^{(i)}$ is the set of observed values for random variables in X on the i example; but it is important

to note that in general it is not necessarily the case that all variables will be observed. The case where no variables are hidden leads to simplifications and is often treated separately.

2 The Challenges of Learning Structure

The difficulties in learning graphical structure focus on both the computational demands and the accuracy of the structure learned. The first of these difficulties is the number of possible graphs. Considering the presence or absence of every possible edge over a fixed set of n vertices yields a set of possible graphs that is super-exponential in n .

In addition to the number of possible graphs, the number of parameters for a given graph is also quite large. In general, to represent a discrete joint distribution between n variables each of which takes k values, requires a table of size k^n , which is clearly unfeasible both due to the storage requirements for so many parameters and the sample size that would be required to learn them. One of the main advantages of the graphical model is that the Markov property associated with X means that the table of size k^n can be replaced with n tables where the i th table has size $k^{|X_{\pi_i}|}$. If the graph is sparse this can represent a large savings, but it should be noted that the storage requirements for the individual tables are still exponential in the number of parents of the node that the given table represents.

This abundance of parameters aggravates the common problem of over-fitting the model to the test data during the learning phase. In particular, if we attempt to find \mathcal{G} that maximizes $P(\mathcal{G}|\mathcal{D}) \propto \mathcal{P}(\mathcal{G})\mathcal{P}(\mathcal{D}|\mathcal{G})$ by maximizing the likelihood $P(\mathcal{D}|\mathcal{G})$, it becomes apparent that assuming independencies can never increase the likelihood. In the case that two variables are actually independent the assumption will have no effect, whereas accounting for any artifact of the sample, no matter how small, by adding dependence, will always improve the empirical likelihood. Thus the graph that maximizes the likelihood of the data is always the complete graph, which wholly defeats the purpose of using a graphical model in the first place. This tendency towards model complexity is made particularly damaging by the fact that each added dependence increases the number of parameters in one of the conditional probability tables by a multiplicative factor. Thus, the cost in terms of model complexity, when using standard conditional probability tables is quite coarse;

it is not possible to represent partial dependencies using fewer parameters. There are, however, a number of alternate representations that make the representation somewhat more fine grained; I discuss these in a latter section.

An additional difficulty is encountered in cases where some variables are not observed in the data and their value must be inferred. When all variables are observed and there is a known graph G with parameters $\theta = (\theta_1, \dots, \theta_n)$, the probability of the observed data is

$$P(D|\theta) = \prod_{i=1}^n \prod_{x_i} P(x_i|x_{\pi_i}, \theta_i),$$

which is the product over all nodes in the graph and all possible values for each node. And so the log likelihood is,

$$l(\theta|D) = \log \prod_{i=1}^n \prod_{x_i} P(x_i|x_{\pi_i}, \theta_i) = \sum_{i=1}^n \sum_{x_i} \log P(x_i|x_{\pi_i}, \theta_i)$$

But this is a sum of independent sums, and has a closed form maximization when θ_i is the relative frequency of occurrence of x_i in D . Thus structure has only local effect, and the problem of maximizing the likelihood of a graph can be treated locally. In the case where there are unobserved variables, however the log likelihood does not decompose, and the θ 's cannot be considered locally.

3 The Search-And-Score Approach

Structure learning algorithms attempt to deal with these limitations in a multitude of ways, but the vast majority follow a common template; they choose to define a way of scoring models that accounts for the deficiencies of maximum likelihood by punishing model complexity and then search the space of possible models for one which scores highly.

A few common assumptions are often invoked to limit the search space. The first of these is to assume a known ordering for the vertices in the graph such that no vertex has an edge to a vertex that precedes it in the ordering [CGK⁺02] [FMR]. This reduces the search space to $2^{n(n-1)}$ possible graphs for a given ordering. It also has the benefit of eliminating the burden of cycle detection during the learning process. For many if not most domains, however, an ordering over the random variables is much less obvious than a graph itself, and it is not clear that such a thing can easily

be found for practical problems.

Another common approach is to assume a bound on the in-degree of possible graphs so that $|\pi_i| \leq k$ for each node i in the graph. However even with a bound of $k = 2$ finding the best possible network and parameters is known to be NP-Complete [Chi95].

A number of methods for scoring models have been proposed. The most commonly used are the Bayesian Information Criteria (BIC), the Minimum Description Length principle (MDL), and the BDe.

$$\text{BIC: } 2 \log P(D|\mathcal{G}, \theta) - d \log n$$

$$\text{DL: } DL(\mathcal{G}) + \sum_i DL(\theta_i, x_{p_i}) + d \sum_i H(x_i|x_{p_i}) \text{ where } H \text{ is the conditional entropy}$$

$$\text{BDe: } \log P(D|\mathcal{G}, \theta) - \frac{d}{2} \log n$$

Here d is the number of parameters in the model. The close relationship between these scoring functions is immediately apparent; minimizing the description length is roughly equivalent to maximizing the BIC if the encoding is good, and maximizing the BIC differs only by constant factors from maximizing the BDe. In all cases the salient point is that we now are considering the likelihood with an additional punishment added for the complexity of the model.

Given a scoring function we can then maximize the score of our model and parameters rather than just the likelihood. In the case of fully observed model, the likelihood decomposes and we can find local extrema in our scoring function by choosing successive modifications to \mathcal{G} that increase the score. This procedure is repeated until no gain is made from any local modification.

Models with hidden variables are more difficult, however. Here local changes in the graph structure can effect the entire maximal parameter settings throughout the graph. Naive approaches force us to consider the graph in its entirety, and lead to intractable algorithms. Friedman's Structural E.M. algorithm addresses this problem [Fri97]. It works by choosing a random graph, and then maximizing θ with respect to this fixed graph; then θ is held fixed in place of the partial observation, and \mathcal{G} is improved. Thus the decomposition property of the fully observed model is essential to both methods of learning. Provided that neither \mathcal{G} nor θ are ever changed to a lower scored value, the Structural E.M. algorithm is guaranteed to converge to a local extremum with respect to the given scoring function.

In practice the weakness of this approach is that there are no guarantees as to the quality of the extremum, and no guarantees on the speed of convergence. Furthermore re-estimating the values of θ on every step is computationally expensive, and so the algorithm performs quite slowly.

Algorithm 1 The Structural E.M. Algorithm

Choose $G^{(0)}$ and $\theta^{(0)}$ *randomly*
for $t = 1, \dots$ until convergence **do**
 1. Find the model $G^{(t+1)}$ that maximizes $Q(G^{(t)}, \theta^t)$
 2. Set $\theta^{(t+1)} = \arg \max_{\theta} Q(G^{(t+1)}, \theta^{(t)})$
end for

4 Other Strategies

Numerous alternatives to the search and score paradigm have been attempted. Among the more attractive of these is a Bayesian approach to model selection: namely, not to select a model at all but rather average over all models. Thus rather than derive a deterministic set of edges in the graph, \mathcal{E} , we instead would like to compute,

$$P(e \in \mathcal{E} | D) = \sum_{\mathcal{E}} P(\mathcal{E} | D) I(e \in \mathcal{E}).$$

While this is theoretically desirable from a Bayesian point of view, this is attempting to solve a hard problem by making it harder. Whereas before we needed to find a single good model from the space of possible models, now we must use the whole model space for inference. However, this difficulty can be skirted by defining a Markov chain over graphs whose stationary distribution is $P(\mathcal{G} | \mathcal{D})$ and then sampling from this chain using MCMC [FK]. Heckerman shows that in many common learning situations the model that scores highest does so by a large factor and so the Bayesian approach is well approximated with only a single model [Hec97].

One special case where structure learning is tractable is when the class of models is known to consist only of trees. The Chow-Liu algorithm allows for likelihood maximization in quadratic time in the number of random variables. Meila et al. generalize this to arbitrary mixtures of trees; and derive a polynomial time version of E.M. for learning hidden variables [MJ00] [BJ01].

A final method of lowering the parameter space and allowing for more efficient learning is so-called “Context-Specific Independence” introduced by Boutilier et al [BFGK]. Rather than storing conditional probability in a tabular form which requires a fixed, exponential number of parameters, they use a linked form which, while no better in the worst case, can have substantial savings on distributions for which the dependencies are sparse. Their method is to explicitly store the conditional probability of variables that show strong correlation, and to implicitly assign a uniform conditional probability to all other values. Friedman generalizes this technique to use local decision trees for representing the tables, where decisions in the tree represent the conditioning variables [FG]. This allows for even more specific pruning in exchange for increased implementation complexity.

The primary advantage of context specific independence is that the trade off between model complexity and test error becomes quite fine-grained. In the tabular representation each dependence requires a multiplicative increase in the size of the local conditional probability distribution table; using context-specific independence there is an explicit trade off in the relative entropy from our model to the empirical distribution and the number of parameters required by our model. Friedman and Boutilier give strong empirical evidence that this increases the learning rate and decreases overfitting problems, even when the resulting graph is more complete. Due to the requirement for sparse dependencies necessary to realize these benefits, no hard bounds are possible.

5 Conclusion

Methods for learning structure in directed graphical models have not met with the same advances that the parameter learning algorithms have. The basic techniques used remain the search and score techniques that represented some of the earliest attempts at structure learning. None of the techniques that allow arbitrary models can achieve anything better than local maximization of heuristic model scoring criteria. In practical terms, both the computational resources required and the large amount of data required to learn high dimensional models puts learning larger networks out of reach. Nonetheless, in the absence of algorithmic methods, the graphical models must be created by informed-guess and checked by cross-validation or other model-scoring techniques. This

is little better than the haphazard methods used for automated learning. In short graphical models represent an excellent solution to the second half of a two part problem; and thus structure learning methods will remain an appealing if difficult target.

References

- [BFGK] Craig Boutilier, Nir Friedman, Moises Goldszmidt, and Daphne Koller. Context-specific independence in bayesian networks. In *UAI-96*, pages 115–123.
- [BJ01] F. Bach and M. Jordan. Thin junction trees. In *NIPS, 2001*, 2001.
- [CGK⁺02] Jie Cheng, Russell Greiner, Jonathan Kelly, David Bell, and Weiru Liu. Learning bayesian networks from data: an information-theory based approach. *Artif. Intell.*, 137(1-2):43–90, 2002.
- [Chi95] D. Chickering. Learning bayesian networks is np-complete. In *Proceedings of AI and Statistics, 1995*, 1995.
- [FG] Nir Friedman and Moises Goldszmidt. Learning Bayesian networks with local structure. In *UAI,96*, pages 252–262.
- [FK] N. Friedman and D. Koller. Being bayesian about network structure: A bayesian approach to structure discovery in bayesian networks. In *UAI00*.
- [FMR] Nir Friedman, Kevin Murphy, and Stuart Russell. Learning the structure of dynamic probabilistic networks. In *Fourteenth Conf. on Uncertainty in Artificial Intelligence (UAI)*, pages 139–147.
- [Fri97] Nir Friedman. Learning belief networks in the presence of missing values and hidden variables. In *Proc. 14th International Conference on Machine Learning*, pages 125–133. Morgan Kaufmann, 1997.
- [Hec97] David Heckerman. Bayesian networks for data mining. *Data Mining and Knowledge Discovery*, 1(1):79–119, 1997.

- [MJ00] Marina Meila and Michael I. Jordan. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1:1–48, 2000.