Discrete Graphical Model Structure Learning

Vishvas Vasuki

August 6, 2010

1 Abstract

- This report examines the discrete graphical model structure learning algorithm specified in a paper by Ravikumar etal.
- Purpose is to check if the algorithm works, and gain intuition into the conditions under which it works. This report details early experiments in this direction.
- Future work will involve theoretical analysis and more detailed experiments to determine conditions under which the algorithm is guaranteed to recover the graph structure.

2 Introduction

- Outline.

2.1 Undirected graphical models and structure learning

- Introduction to probabilistic graphical models. The problem of learning undirected graphs associated with graphical models.

2.1.1 Pairwise graphical models.

- Definition.

2.1.2 The structure learning problem.

- Motivate the problem using a small example.

2.1.3 Discrete graphical models.

- Definition and probability distribution.
- We consider only pairwise discrete graphical models: doing so does not result in any loss of generality.
- Remark about needing many parameters per edge.
- Specify the conditional probability distribution.

2.2 Ising models

- Definition and probability distribution.
- Remark about needing only one parameter per edge.
- Specify the conditional probability distribution.

3 Structure learning algorithm

- Overview.

3.1 Neighborhood learning algorithm

- Describe algorithm structure: you take one node at a time, and learn its neighborhood.

3.1.1 For general discrete models

- l1/l2 regularized parameter estimation.
- Formulation as a constrained optimization problem.
- Meaning of 'regularization parameter'.

3.1.2 For Ising models

- l1 regularized parameter estimation.

3.1.3 Resolving inconsistencies

- From the sparsity pattern of the parameters $\theta_{i,:,:,:}$ learned using logistic regression, the algorithm deduces the neighbors of each node in G. The algorithm can encounter the following inconsistency when deducing graph structure: The neighborhood set learned for node u may include v, but the neighborhood set learned for v may not include u.
- For the Ising model case, this is usually not a problem, as the analysis by Ravikumar etal guarantees consistent signed edge recovery when certain conditions are met.
- In the case of general discrete graphical models, during our early experimetns, we resolve inconsistencies using the OR rule.

3.2 Related work

- Mention some related work in order to contrast with other approaches to structure learning.
- Chow-Liu: assume tree structure.
- Learning graphical models using contingency tables.
- Koller paper which learns graphical models using l1 regularization.
- Learning conditional random fields using l1l2 regularization.

4 Experiment Setup

- Overview.

4.1 Test distributions.

- Picking a topology. We have tried star graphs and chain structured graphs.
- Range of each random variable.
- Selecting the parameters: For a given number of nodes and for a given topology, the parameter array is drawn uniformly at random from [0, k]; k is usually 1 or 3.
- Importance of picking k correctly.

4.2 Drawing samples.

- Sampling from the distribution: As we are sampling from tree structured graphical models, data is sampled exactly: MCMC is not used. Cite Mark Schmidt software.
- MCMC will be used for non-tree structured graphical models.

4.3 Goals and Evaluation

- One of the goals of our experiments are to understand how many examples are needed for the algorithm to learn the structure of a pairwise discrete graphical model with p nodes with high probability.
- Another of the goals of our experiments are to understand the dependency between the number of samples n, the number p of nodes in G, and the parameter λ used by the neighborhood learning algorithm.
- In order to address the first goal, we draw multiple samples sets of increasing size ($n \in [2^5, 2^{14}]$) from the distribution D and do the following for each choice of n: We fix a suitable λ either manually or using validation, and finally we empirically determine the probability of success of the structure learning algorithm.
- In order to address the second goal, we plot the λ used during the previous process either against n or against $\sqrt{\frac{\log p}{n}}$.
- Procedure used in solving the logistic regression problem in its contrained optimization form.

4.4 Choosing the regularization parameter

- Picking the parameter c while solving the logistic regression problem in its constrained optimization form.
- This is one of the most vexing problems during experimentation.
- Dependence of the sparsity in the structure learned on the regularization parameter.

- Dependence of the ideal regularization parameter on p and n, according to the Obozinsky paper.

4.4.1 The validation procedure.

- One can manually pick the regularization parameter - by picking the one which yields a model of expected sparsity.
- One can also use validation to automatically pick regularization parameters. This process involves assigning scores to regularization parameters based on how good it is. The goodness of a given regularization parameter depends on the goodness of the model parameters learned using that regularization parameter. This raises the question: How good is a given model parameter array θ?
- Goodness of the model parameters can either be judged based on its sparsity, or based on its likelihood given some observations.
- We will consider the latter first. As determining the likelihood of θ requires the computation of the log partition function, we often find it better to compute the pseudolikelihood of the parameter. However, experiments revealed that this way of picking parameters is not reliable.
- Rate model parameters based on its ability to yield the desired sparsity. For our experiments, we use this method - not only is it more reliable, it is also faster. We use binary search here.[**Check**]Hopefully, analysis will help us identify the parameters even without knowing beforehand the sparsity structure of the graphical model generating the observations.
- Describe k-fold validation. Draw k samples-sets. Learn the model parameters corresponding to a given regularization parameter using each one of these sample sets, and determine the average goodness of the model parameters thus learned. This is used to rate the corresponding regularization parameter.

4.5 Solving the optimization problem efficiently

- Van Greer algorithm to solve the problem in its Lagrangian formulation. This formulation is generally more convenient for the purpose of analysis.
- Projected Quasi-Newton algorithm to solve the problem in its constrained optimization formulation.

5 Results and discussion

- Figures corresponding to star graph experiments using the constrained optimization problem. They form a sort of *proof of concept* for the structure learning algorithm.
- Figures corresponding to star graph experiments using the lagrangian formulation of the problem. They show that using sparsity-based validation to learn the best λ succeeds, but other experiments show that the k in k-fold validation must be larger. Explain the small number of nodes tested.

- Failure of chain graph experiments.
- For a given p, charactarize the dependence of the regularization parameter on n.
- For a given n, charactarize the dependence of the regularization parameter on p.

6 Conclusion

- Repeat the abstract in past tense.

6.1 Future work

- Analysis.
- Application to a problem of practical interest.

7 Acknowledgements

1. Acknowledge Pradeep, Ali Jalali, Sujay.