Bayesian Structure Learning

R. B. Alexander

Bayesian structure learning is a combined structure learning and parameter learning task that involves learning a Bayesian network (or graph) G from a dataset D. The dataset contains n discrete random variables $\mathbf{X}_{1:n}$. Each of the variables has $\mathbf{r}_i$ possible instantiations and for a given graph G, each of the variables has $\mathbf{q}_i$ possible instantiations of its parents π_{ij}. The number of times $\mathbf{X}_i = \mathbf{k}$ given π_{ij} occurs in the dataset is $\mathbf{m}_{ijk}$ and the associated probability of $\mathbf{X}_i = \mathbf{k}$ given π_{ij} is $\mathbf{P}(\mathbf{X}_i = \mathbf{k} \mid \pi_{ij}) = \theta_{ijk}$. Using the Bayesian score function, we can estimate the likelihood of a graph structure given the dataset. Once we have computed the Bayesian score, we must search the space of all Bayesian networks G to find the graph that maximizes the Bayesian score and is thus, the most probable graph.

Bayesian-Dirichlet Score Function

For this project, the Bayesian-Dirichlet scoring function was used, which assumes a Dirichlet prior over the Bayesian network parameters ($P(\boldsymbol{\theta}) \sim \mathrm{Dir}(\boldsymbol{\theta} \mid \boldsymbol{\alpha})$). The Bayesian-Dirichlet scoring function can be shown to take the following form, where Γ is the gamma function, $\Gamma(n) = (n-1)!$.

$$\ln P(G \mid D) = \ln P(G) + \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left[\ln \left(\frac{\Gamma(\alpha_{ij0})}{\Gamma(\alpha_{ij0} + m_{ij0})} \right) + \sum_{k=1}^{r_i} \ln \left(\frac{\Gamma(\alpha_{ijk} + m_{ijk})}{\Gamma(\alpha_{ijk})} \right) \right]$$

$$\alpha_{ij0} = \sum_{k=1}^{r_i} \alpha_{ijk} \qquad m_{ij0} = \sum_{k=1}^{r_i} m_{ijk}$$

Since we have no information about which graph structures are more or less probable, we use a uniform graph prior, $P(G) = 1$. Under weak assumptions, we can assume a uniform Dirichlet prior over the Bayesian network parameters where $\alpha_{ijk} = \alpha$. Here, we set $\alpha = 1$, which gives the K2 scoring function:

$$\ln P(G \mid D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \left[\ln \Gamma(r_i) - \ln \Gamma(r_i + m_{ij0}) + \sum_{k=1}^{r_i} \ln \Gamma(1 + m_{ijk}) \right]$$

Graph Search Algorithms

The space of directed acyclic graphs is superexponential with the number of nodes, so an efficient search strategy is critical in finding an optimal Bayesian network. Several algorithms for graph search exist and two relevant classes of these algorithms are directed graph search algorithms and partially-directed graph search algorithms. Two prominent algorithms for directed graph search are K2 search, which greedily adds parents to a node until no higher-scoring graphs are found, and local search, which starts from a graph structure and moves to the highest-scoring graph in its neighborhood (defined by elementary graph operations). Algorithms for partially-directed graph search search the space of Markov equivalence classes, which is smaller than the space of directed graphs. Some algorithms exploit the *score equivalence* of the scoring function to minimally search the space of partially-directed graphs, yielding robust searches.

In our implementation, we used K2 search and added randomized starts to generate several most-probable graphs that could

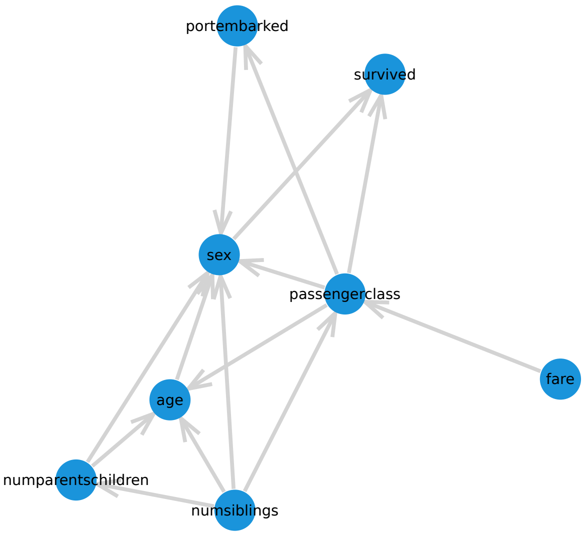

Figure 1 Bayesian network learned from the `small` dataset (8 variables) using a K2 search of the space of directed acyclic graphs with 1000 randomized starts. ($\ln P(G \mid D) \approx -3795$)

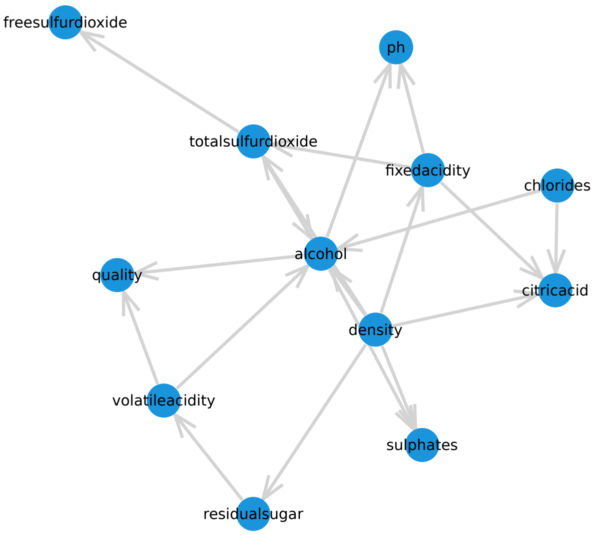

Figure 2 Bayesian network learned from the `medium` dataset (12 variables) using a K2 search of the space of directed acyclic graphs with 100 randomized starts. ($\ln P(G \mid D) \approx -41961$)

K2 Search

K2 Search with Randomized Start

Seeded a random number generator for reproducibility.

Did a K2 search algorithm that iterated over random permutations of variable orderings. Limited the maximum number of parents to 8. Did not do efficient caching or efficient recomputation of the m_{ijk} counts, so the runtimes were long.

describe the strategy you used for your search (e.g. K2) and what modifications (if any) you made to the algorithm. Include any drawbacks that your modifications may have introduced.

provide the timing of how long it took to generate each graph

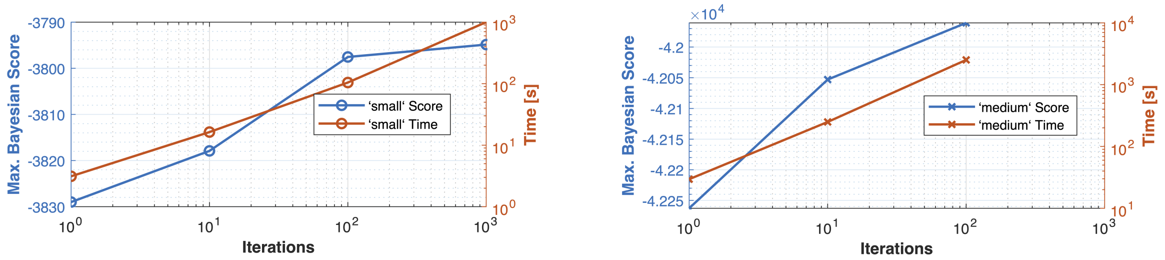

Figure 3 Bayesian score improvement using K2 search iterated over randomly-permuted variable orderings.

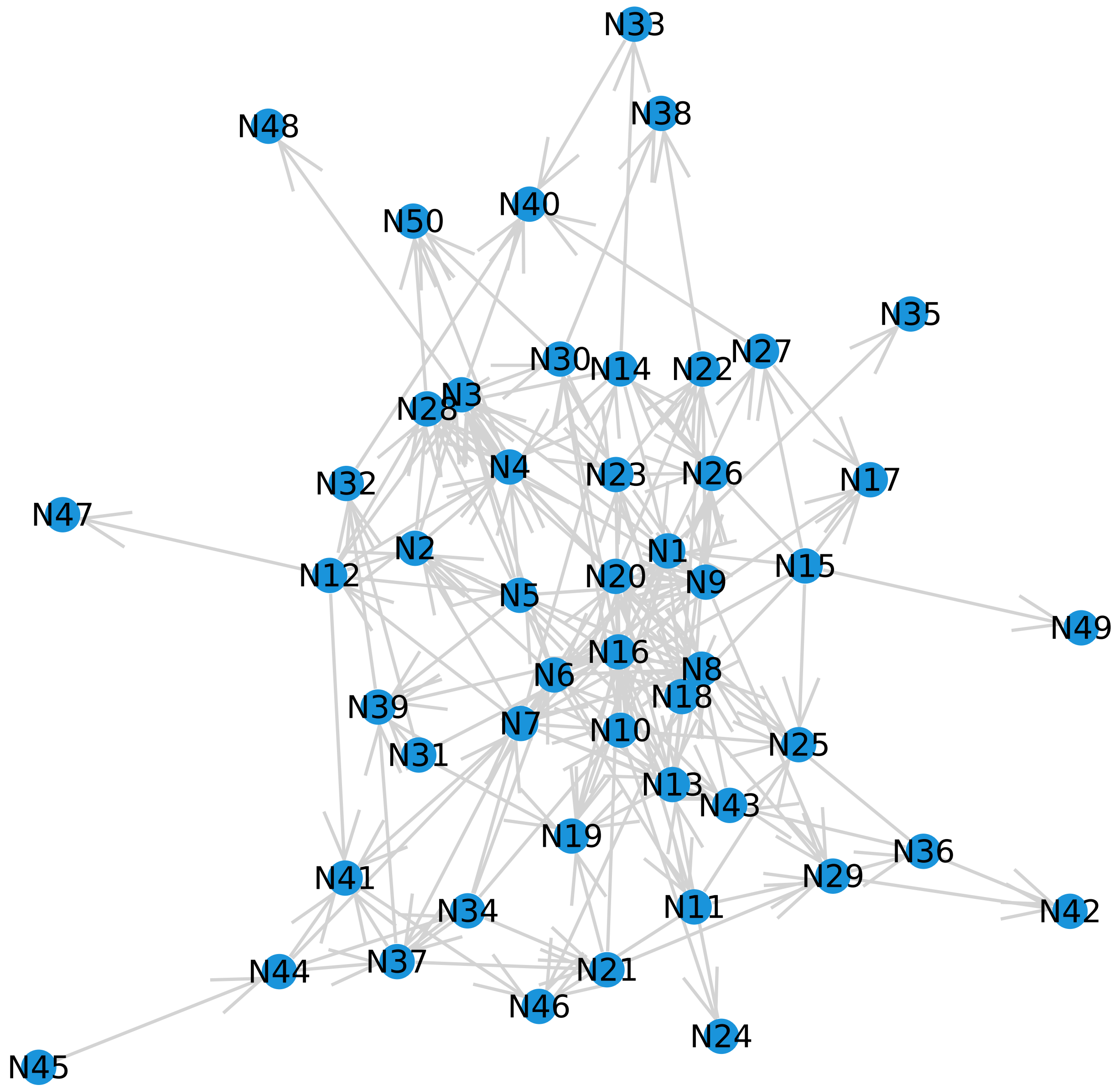

Figure 4 Bayesian network learned from the `large` dataset (50 variables) using a K2 search of the space of directed acyclic graphs with 1 randomized start. ($\ln P(G \mid D) \approx -427612$)