Notes for Different Graphical Gaussian Models

Guiyuan Lei*

September, 2006

1 HdBCS: Bayesian Covariance Selection in High-Dimensions

The model of Bayesian Covariance Selection in High-Dimensions [1] is designed to perform covariance selection for datasets with tens or possibly hundreds of thousands of variables.

1.1 The algorithm

The main idea of HdBCS is to identify good regression models for each variable and to combine these models in a joint multivariate normal distribution using the chain rule (which is essentially equivalent to constructiong a DAG or Bayesian network). The algorithm starts with dependency network which is built from sets of regressions seperately, then generates an appropriate ordering of the variables to underlie a compositional network. With this ordering, variable selection for each linear model for the ordered variables produces a DAG. Finally, by removing arrows and marrying all parents of any node in this DAG, an undirected graph is delivered.

For analysis and selection of regressions, the model utilizes the inverse Wishart prior, applies for forward/backward varialbe selection with respect to the defined set of candidate predictors, and score each regression by the posterior model probabilities. See section 3 in the paper [1] for detail.

*Guiyuan.Lei@ncl.ac.uk

Once the predictors of each variable are obtained, the order of all variables should be generated to construct compositional network. The strategy is to: the ordering aims to maximize the resulting posterior probability on the resulting model in an iterative process that ,at each step, decreases the overall score (equation 5 in the paper) the least. See section 4.1 for fully described construction.

1.2 The software

There is one software implemented for this model. Codes can be downloaded from http://www.stat.duke.edu/∼adobra/hdbcs.html

The search procedure has three separate steps: filtering, generating good starting models and improving the starting models until convergence.

The fisrt step is to select a relatively rich set of possible predictors for each variable. The search strategy is forward/backward procedure. This procedure is for each variable seperately. Starting with random selected predictors for each variable x_i, consider the edges between all other variables $x_j, j = [-i]$ with this variable, if x_j is already predictor of variable x_i, propose to delete x_j as predictor of x_i, otherwise, propose to add x_j as its predictor. Calculate the ratio of posterior probability between proposed model and current model (one predictor difference between these two models), and accept/reject the proposed model according to the accept propobility. Repeat the above process for many iterations to get set of possible predictors for each variable. The pseudo code for the search procedure can be describled as following:

For each variable x_i,

1. Initial model, randomly select predictors of x_i from all other variables $x_j, j = \{1, \cdots, p\} \backslash \{i\}$
2. Iteration step: repeat the following process for a fixed times:

   For all other variables $x_j, j = \{1, \cdots, p\} \backslash \{i\}$, randomly select x_j (each variable can be selected once)

   i. Propose new model from current model

      If (x_j is not currently predictor of x_i), calculate the log posterior probability with x_j is predictor of x_i

else, calculate the log posterior probability calculate the log posterior probability with x_j is not predictor of x_i

ii. Calculate accept propobility and accept/reject proposed model, if x_j is a predictor in new model, update the frequency of x_j as predictor of x_i

3. For all other variable $x_j, j = \{1, \cdots, p\} \backslash \{i\}$, if the frequency x_j as predictor of x_i is greater than threshold, set x_j as predictor of x_i in filtered model (one dependency network)

In second step, first generate dependency network which is based the filtered model in step1, the possible predictors for initial model are selected from the final model in step1 (in step1, all other variables $[-i]=\{1, \cdots, p\} \backslash \{i$ $\}$ are possible predictors of x_i), in addition, the best model is recorded (in step1, final model is average model); then DAG is generated from the best model. In this step, generate several DAGs by running several times of above procedure. For generating compositional network (DAG) from dependency network, see section 4.1.2 in the paper [1].

In third step, the starting points/models (which are generated in step2) are improved in a procedure (simulated annealing) that converges to local models of the posterior distribution over the space of DAGs.

Given possible set of predictors S_i for each variable x_i and DAG generated in step2. With order of variables for given DAG, switch variable x_i in position *equation* with variable x_j in position *equation*$+1$, reconsider the regression equations of these two varialbes. The search for regression equation of variable x_i is similar as that of in step1, but the predictors are restricted to the intersection of set S_i and successor of x_i. Calculate the posterior probabilies for variable x_i and x_j, also calculate the difference δ between these posterior probabilies and that of former regression equations for variable x_i and x_j. Calculate δ for each varialbe $x_i, i = 0, 1, \cdots, NumberOfGenes - 1$ anb sort these δ in a list eq.

Next, select which equation (and its adjacent equation+1) to be switched order. The weights $w[i]$ of each pair of order switching are normalized as following:

1. $w[i] = w[i] - w[nmax - 1]$, where $nmax$ is number of pair switching,

the weights are in increasing order

2. $w[i] = pow(exp(w[i]), anneal)$, where $anneal$ is annealing parameter, the minimal value is 1, maximum valuse is set to be 3
3. $cumw[i] = \sum_{j=0}^{i-1} x[j], i = 1, 2, \cdots, nmax$
4. $cumw[i] = cumw[i]/cumw[nmax]$

The pseudo code for selecting equation by weights $cumw[i]$ is as following:

1. Initialize: $q1 = 0; q2 = nmax$; and generate a random number s
2. While$(q1 + 1! = q2)$

   $q = (q1 + q2)/2;$
   If$(cumw[q] < s)$, $q1 = q;$
   else, $q2 = q$

The returned $q1$ is the selected equation, denote it as $chosenEquation$ (this is the position of selected equation/variable in the order) and update order and regression equations for x_i and x_j in position $chosenEquation$ and $chosenEquation + 1$ in the order.

Then, for generate next DAG, only need to calculate δ for three pairs of ordering switch, that is $chosenEquation - 1$ (and its adjacent position $chosenEquation$), $chosenEquation$ (and its adjacent position $chosenEquation$+ 1) and $chosenEquation + 1$ (and its adjacent position $chosenEquation + 2$). After updating the δ for new order, select equation to switched with variable in its adjacent position, update order and regression equation for next new DAG.

Repeat the above procedure for $nGraphsToGenerate$ times and record these DAGs/models every $whenToSave$ iterations.

After all these three steps, an average network is generated from all statistically significant models during the simulated annealing procedure. Find the model with best posterior probability w, all other recorded (for example, recorde every 1000 iterations) models with porstiror probability w_i

$$\exp(w_i - w) > \epsilon \tag{1}$$

will be considered in averaging network. When add one graph to averaged graph, the weight of each edge w_i(if this edge exists in current graph) is added as $b + \log(1 + \exp(a - b))$ (assuming $a < b$, otherwise just exchange the value of a and b) if the weight of average edge is a (initialized as 0) and the weight of added graph is b. Note that a and b are log posterior probability and $b + \log(1 + \exp(a - b)) = \log(\exp(a) + \exp(b))$. Meanwhile, the weight of average graph $GrandTotal$ is added as same formula $b + \log(1 + \exp(a - b))$ but the value of a is set to be very small number, like -99999999999999999.0. At last, when all statistically significant graphs are considered, the weight of each edge is $\exp(w_i - GrandTotal)$.

References

[1] Adrian Dobran, Chris Hans, Beatrix Jones, Joseph R. Nevins, Guang Yao & Mike Wes. Sparse graphical models for exploring gene expression data *Journal of Multivariate Analysis* **90** (2004), 196–212.