10-708: Probabilistic Graphical Models 10-708, Spring 2014

25 : Graphical Induced Structured Input-Output Methods

Lecturer: Eric P. Xing *Scribes: Alok Kothari, Haoyu Wang*

So far we have been building up the theory and applications of Probabilistic Graphical Models. In this lecture, we dig into peculiar ways of using them. Particularly, we move away from the view of graphical models as modeling dependencies of a distribution or as minimizations of a loss function on graphs. We will look into how graphical methods can be used to model constraints.

1 Genetic Basis of Disease

Though humans are 99.9% same in our genetic make up, there are number of genotype sites we differ at. Such sites are called Single Nucleotide Polymorphisms (SNP). These sites reflect the differences between us, e.g. difference in hair color.

Such differences may also be responsible for the cause of hereditary disease. Genetic Association Hypothesis testing then is the study to find which SNP's are causal (or associated) vis-a-vis a hereditary disease. Going site to site we try and associate the genotype with a phenotype. Usually these genetic association mappings are very sparse, (1 in 1000) and may represent basic differences in body physiology. So far, a number of standard approaches have been applied to find the causal SNP's. Some examples of standard approaches are: using the linkage analysis of selected markers, quantitative trait locus (QTL) mapping conducted over one phenotype and one marker genotype at a time, which are then corrected for multiple hypothesis testing. Primitive data mining methods have also been employed, such as the clustering of gene expressions and the high-level descriptive analysis of molecular networks. Such approaches yield crude, usually qualitative characterizations of the study subjects. However, many complex disease syndromes, such as asthma and cancer, consist of a large number of highly related, rather than independent, clinical or molecular phenotypes. They are the effect of many sites of mutation - multiple causal SNP's. Attempting to find the top k SNP's for complex diseases, using the standard approaches, is not ideal due to interactive effects between the two SNPS's. Often the correlation between two SNP's may have been forced as they are bound by physical constraints, (if one mutates the other one has to mutate too). So the way forward is by multiple hypothesis testing. Studying gene expression networks, may tell us what a complex disease like cancer means at a molecular level.

Thus for understanding genetic variations causal for a disease requires the analysis of multi-correspondence mapping between multiple SNP's and (multiple) symptoms/clinical phenotypes.

2 Structured Association

Summarizing earlier section, standard approaches have been trying to map one phenotype to one genotype, via statistical tests, like cross-entropy etc. Now we consider multiple correlated phenotypes and genotypes jointly i.e. multiple signals in genotype causing multiple things in the phenotype. We want to thus extract out the meaningful and thus the sparse associations and mappings between the two. Technically we want to find the Pleotropic and Epistatic effects. Here, we do sparse learning via sparse linear regression.

$$\arg\min_{\beta} f(\beta) = \arg\min_{\beta} \sum_{k} (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k)^T (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k) + \lambda \sum_{j} |\beta_j| \tag{1}$$

Here $\beta \in \mathbb{R}^J$, could be a few million dimensions. The sparse solution is enforced by the L1-regularization. In multi-task learning β may be considered to be a matrix: $\beta \in \mathbb{R}^{J \times K}$.

$$\arg\min_{\beta} f(\beta) = \arg\min_{\beta} \sum_{k} (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k)^T (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k) + \lambda \sum_{j} \sum_{k} |\beta_{jk}| \tag{2}$$

3 Graph Guided Fused Lasso

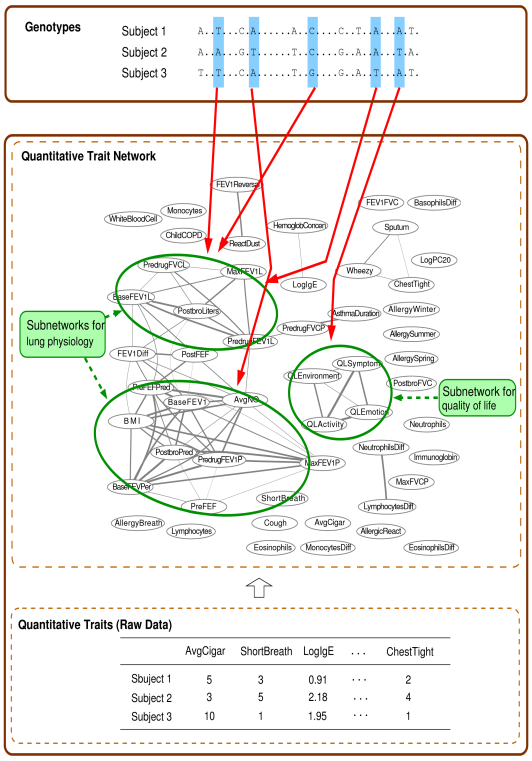

Figure 1: An illustration of association analysis using the QTN for asthma dataset

We want to leverage the domain knowledge we might have about how the various symptoms/phenotypes are correlated, e.g. Figure 1. (showing a Quantitative Trait Network (QTN)), demonstrates how these traits are related for asthma. This has been taken from the actual domain knowledge of a doctor.

In order to estimate the association strengths jointly for multiple correlated traits, while maintaining sparsity, we introduce another penalty term called graph-guided fusion penalty into the lasso framework. This novel penalty makes use of the complex correlation pattern among the traits represented as a QTN, and encourages the traits which appear highly correlated in the QTN to be influenced together. Thus, this new approach estimate of the regression coefficients reveals joint associations of each SNP with the correlated traits in the entire subnetwork as well as associations with each individual trait.

Accordingly, the following two approaches are proposed to enforce these constraints:

Model G_cF Lasso : This model uses the graph without considering the edge-weights. The objective function, consisting of two regularizers (lasso and graph) and a squared error loss term, has the form:

$$\boldsymbol{B}^{\hat{\boldsymbol{G_c}}} = \arg\min_{\beta} \sum_k (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k)^T (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k) + \lambda \sum_j \sum_k |\beta_{jk}| + \gamma \sum_{(m,l)\in E} \sum_j |\beta_{jm} - sign(r_{ml})\beta_{jl}| \quad (3)$$

E is the set of edges.

Model G_wF Lasso : An advanced version of this model is G_wF Lasso. Here, besides exploiting the graph topology of QTN, the lasso also exploits the edge weights. It weights the fusion penalty by the amount of correlation between the two traits being fused, so that the amount of correlation controls the amount of fusion for each edge. More generally,it weights each term in the fusion penalty with a monotonically increasing function of the absolute values of correlations, and finds an estimate of the regression coefficients, as follows:

$$\boldsymbol{B}^{\hat{\boldsymbol{G_w}}} = \arg\min_{\beta} \sum_k (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k)^T (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k) + \lambda \sum_j \sum_k |\beta_{jk}| + \gamma \sum_{m,l\in E} f(r_{ml}) \sum_j |\beta_{jm} - sign(r_{ml})\beta_{jl}| \quad (4)$$

In the work that is referenced in the lecture, G_w^2 Lasso is with $f(r) = r^2$ and G_w^1 Lasso is with $f(r) = \|r\|$. G_cF Lasso is then a special case of G_wF Lasso with $f(r) = 1$.

The regularization parameters (λ and γ) in both above models may be determined by cross-validation or by using a validation set.

The optimization problems in Equation 3 and Equation 4 are convex. They may be formulated as a quadratic programming problem (QPP). There are many publicly available software packages that efficiently solve the QPP. However, the issues with using software packages for the above QPP are:

- These approaches do not scale in terms of computation time to a large problem involving hundreds or thousands of traits, as is the case in a typical multiple-trait association study.
- Difficulty arises in directly optimizing Equation 3 and Equation 4, as this is a non-smooth function of the L1-norm.

We thus transform this problem to an equivalent form that involves only smooth functions, and use a fast coordinate-descent algorithm to find the estimates of regression coefficients.

We restate the problem as follows:

$$\boldsymbol{B}^{\hat{\boldsymbol{G_{QPP}}}} = min_{\beta_k, d_{jk}, d_{jml}} \sum_k (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k)^T (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k) + \lambda \sum_{j,k} \frac{\beta_{jk}^2}{d_{jk}} + \gamma \sum_{m,l\in E} f(r_{ml})^2 \sum_j \frac{(\beta_{jm} - sign(r_{ml})\beta_{jl})^2}{d_{jml}} \quad (5)$$

subject to:

$$\sum_{j,k} d_{jk} = 1 \quad (6)$$

$$\sum_{m,l \in E} \sum_{j} d_{jml} = 1 \tag{7}$$

Here, $d_{jk} \geq 0$, for all j,k and $d_{jml} \geq 0$, for all j,$(m,l) \in E$.

We solve the above equation by coordinate-descent, iteratively updating β_k, d_{jk} and d_{jml}, until there is little improvement in the value of the objective function. Using this approach, we first fix values of d_{jk} and d_{jml}, and then update for β_{jk} by differentiating the objective function in Equation 5 with respect to each βjk and setting it to 0.

The following update equations result:

$$\beta_{jk} = \frac{\sum_i x_{ij}(y_{ik} - \sum_{j \neq t} x_{it}\beta_{tk}) + \gamma(\sum_{k,l \in E} f(r_{kl})^2 \frac{sign(r_{kl})\beta_{jl}}{d_{jkl}} + \sum_{m,k \in E} f(r_{mk})^2 \frac{sign(r_{mk})\beta_{jm}}{d_{jmk}})}{\sum_i x_{ij}^2 + \frac{\lambda}{d_{jk}} + \gamma(\sum_{k,l \in E} \frac{f(r_{kl})^2}{d_{jkl}} + \sum_{m,k \in E} \frac{f(r_{mk})^2}{d_{jmk}})} \tag{8}$$

$$d_{jk} = \frac{\|\beta_{jk}\|}{\sum p, q \|\beta_{pq}\|} \tag{9}$$

$$d_{jml} = \frac{f(r_{ml})|\beta_{jm} - sign(r_{ml})\beta_{jl}|}{\sum_{p,q \in E} f(r_{pq}) \sum_t |\beta_{tp} - sign(r_{pq})\beta_{tq}|} \tag{10}$$

4 Tree-guided Group Lasso

4.1 Background

In a univariate-output regression setting, sparse regression methods that extend lasso have been proposed to allow the coefficients to reflect the underlying structural information among the inputs. Particularly, if the coefficients are positive they signify an underlying mapping. Group lasso apply an L1 norm of the lasso penalty over groups of inputs, while using an L2 norm for the input variables within each group. This L1/L2 norm for group lasso has been extended to a more general setting to encode prior knowledge on various sparsity patterns, where the key idea is to allow the groups to have an overlap. However, the overlapping groups in their regularization methods can cause an imbalance among different outputs, because the regression coefficients for an output that appears in large number of groups are more heavily penalized than for other outputs with memberships to fewer groups. So the tree-guided group lasso for multi-task regression with structured sparsity has been proposed.

Since tree represents a hierarchical clustering structure, it is important to have methods to to recover the common set of relevant inputs for each output cluster. Comparing a graph with $O(|V|^2)$ edges, a tree has only $O(|V|)$ edges which makes it scalable to a very large number of phenotypes. In tree-guided group lasso, we consider the problem of learning a sparse multi-task regression, where the structure in the outputs can be represented as a tree with leaf nodes as outputs and internal nodes as clusters of the outputs.

4.2 Sparse Regression and Multi-task Learning

The multi-task sparse regression is similar to the linear regression problem and it can be treated as multiple linear regression at the same time. Here is the definition of the Multi-task sparse regression using the similar

setting of variables as earlier:

$$\boldsymbol{y}_k = \boldsymbol{X}\boldsymbol{\beta}_k + \boldsymbol{\epsilon}_k \tag{11}$$

where β_k is a vector of J regression coefficients $(\beta_k^1, ..., \beta_k^J)T$ for the k-th output, and ϵ_k is a vector of N independent error terms having mean 0 and a constant variance.

Let $\boldsymbol{B} = (\beta_1, ..., \beta_k)$, $\boldsymbol{B}^{\hat{lasso}}$ is obtained by solving the following optimization problem:

$$\boldsymbol{B}^{\hat{lasso}} = \arg\min_{\beta} \sum_k (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k)^T (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k) + \lambda \sum_j \sum_k |\beta_{jk}| \tag{12}$$

While the general linear regression solves a vector of coefficients, this optimization function for multi-task sparse regression is to solve the coefficients matrix. In the optimization function above, the parameter λ controls the sparsity of the solution. Larger λ will lead to a smaller number of non-zero regression coefficients.

4.3 Examples of Embedding Prior Knowledge

The $L1$-penalized regression assumes that all of the outputs in the problem share the common set of relevant input variables. However, in many real-world applications, different outputs are related in a complex manner. Having this prior knowledge of the complex structure of the data, we can definitely improve the learning performance. Here is an example of such prior knowledge. Suppose we know the hierarchical tree clustering of the genes as a-priori, we can use this prior knowledge as follows. In a simple case of two genes, low clustering height from nodes to their parents means a higher correlation between the genes, while a large height means a weak correlation between them which is shown in the following figure.

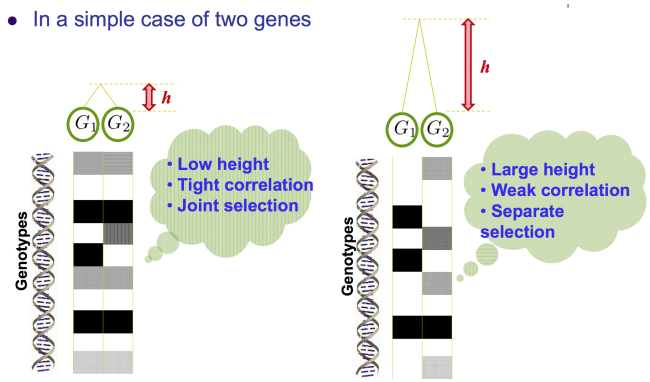

Figure 2: Two genes example of selecting β_1 and β_2 in different ways accoding to the height

According to the figure above, it is intuitive to think that we can jointly select the β_{jk}s of child nodes if the genes are in tight correlation and separately if the genes are in weak correlation. Thus, selecting β_{jk}s according to the difference in height is important prior knowledge we want to embed into our optimization function. Then how do we take into account this prior knowledge into our optimization problem? Here is a intuitive modification on the lasso.

$$\boldsymbol{B}^{\hat{Tree}} = \arg\min_{\beta} \sum_k (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k)^T (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k) + \lambda \sum_j [h(|\beta_{j1}| + |\beta_{j2}|) + (1-h)(\sqrt{\beta_{j1}^2 + \beta_{j2}^2})] \tag{13}$$

In the optimization formula above, we have made some changes as compared to the normal L_1-penalization. The first part of the penalization $|\beta_{j1}|+|\beta_{j2}|$ is from the original L_1 penalty while the second part $\sqrt{\beta_{j1}^2+\beta_{j2}^2}$ represents the L_2 penalty. It is important to notice that the L_1 penalty means we select the β_{jk}s for two nodes separately while the L_2 penalty means we select the β_{jk}s jointly. By adding the height information h as the balance parameter, the penalization then represents what we wanted to embed in the optimization problem.

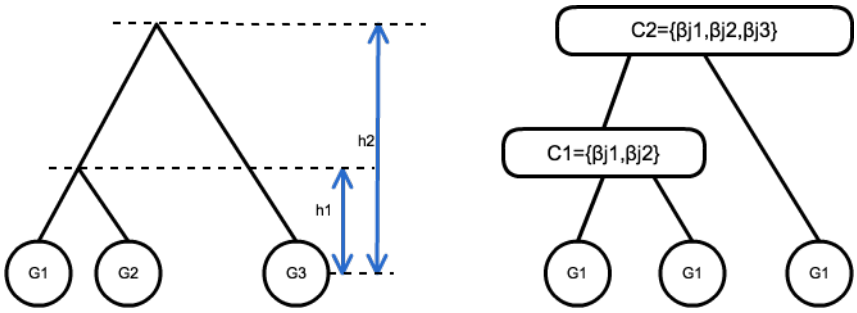

Figure 3: A general tree for penalty extension

As for a more general tree in the figure above, we can easily extend our penalty using the same form.

$$\boldsymbol{B}^{\hat{Tree}} = \arg\min_{\beta} \sum_{k} (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k)^T (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k) + \lambda \sum_{j} [h_2(|C1| + |\beta_{j2}|) + (1-h_2)(\sqrt{\beta_{j1}^2+\beta_{j2}^2+\beta_{j3}^2})] \quad (14)$$

In the optimization formula above, the $C1$ can be extended recursively as follows:

$$C1 = h_1(|\beta_{j1}| + |\beta_{j2}|) + (1-h_1)(\sqrt{\beta_{j1}^2+\beta_{j2}^2}) \quad (15)$$

4.4 Definition of Tree-guided Group Lasso

We now discuss the tree-guided group lasso. Given this tree T over the outputs, we generalize the L_1/L_2 regularization to a tree regularization as follows. We expand the L_2 part of the L_1/L_2 penalty into a group-lasso penalty, where the group is defined based on tree T. Each node $v \in V$ of tree T is associated with group G_v, whose members consist of all of the output variables (or leaf nodes) in the subtree rooted at node v. Given these groups of outputs that arise from tree T, tree-guided group lasso can be written as follows:

$$\boldsymbol{B}^{\hat{Tree}} = \arg\min_{\beta} \sum_{k} (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k)^T (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k) + \lambda \sum_{j} \sum_{v \in V} \omega_v ||\beta_{jG_v}||_2 \quad (16)$$

In the formula above, β_{jG_v} is a vector of regression coefficients. Each group of regression coefficients β_{jG_v} is weighted with ω_v that reflects the strength of correlation within the group.

In order to define the weights of ω_v, we first associate each internal node v of the tree T with two quantities s_v and g_v that satisfy the condition $s_v + g_v = 1$. Comparing to the examples in the previous section, the s_v is similar to the height value h representing the weight for selecting the output variables associated with each of the children of node v separately, and the g_v, is similar to $(1-h)$, representing the weight for selecting them jointly.

Given an arbitrary tree T, we recursively apply a similar operation starting from the root node towards the leaf nodes as follows:

$$\sum_j \sum_{v \in V} \omega_v ||\beta_{jG_v}||_2 = \lambda \sum_j W_j(v_{root}) \tag{17}$$

where

$$W_j(v) = \begin{cases} s_v \times \sum\limits_{c \in Children(v)} |W_j(c)| + g_v \times \omega_v ||\beta_{jG_v}||_2 & \text{if v is an internal node} \\ \sum\limits_{m \in G_v} |\beta_{jm}| & \text{if v is a leaf node} \end{cases} \tag{18}$$

4.5 Parameter Estimation

In order to estimate the regression coefficients in tree-guided group lasso, an alternative formulation of the problem can be used.

$$\boldsymbol{B}^{\hat{Tree}} = \arg\min_{\beta} \sum_k (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k)^T (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k) + \lambda \left(\sum_j \sum_{v \in V} \omega_v ||\beta_{jG_v}||_2 \right)^2 \tag{19}$$

Since the L_1/L_2 norm in the original optimization equation is a non-smooth function, it is not trivial to optimize it directly. We make use of the fact that the variational formulation of a mixed-norm regularization is equal to a weighted L_2 regularization as follows:

$$\left(\sum_j \sum_{v \in V} \omega_v ||\beta_{jG_v}||_2 \right)^2 \leq \sum_j \sum_{v \in V} \frac{\omega_v^2 ||\beta_{jG_v}||_2^2}{d_{j,v}} \tag{20}$$

where $\sum_j \sum_v d_{j,v} = 1, d_{j,v} \geq 0$

Thus we can re-write problem so that it contains only smooth functions:

$$\boldsymbol{B}^{\hat{Tree}} = \arg\min_{\beta} \sum_k (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k)^T (\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k) + \lambda \sum_j \sum_{v \in V} \frac{\omega_v^2 ||\beta_{jG_v}||_2^2}{d_{j,v}} \tag{21}$$

In this parameter estimation method, we actually take the advantage of the additional variables $d_{i,j}$ to smooth our function. This optimization problem can be solved by optimizing β_k and $d_{i,j}$ alternately until convergence. In each iteration, we first fix the values for β_k, and update $d_{i,j}$, where the update equations for $d_{i,j}$ are given below.

$$d_{j,v} = \frac{w_v ||\beta_{j,v}||_2}{\sum_j \sum_{v \in V} w_v ||\beta_{j,v}||_2} \tag{22}$$

Then, we hold $d_{i,j}$ as constant, and optimize for β_k. We differentiate the objective in Equation (21) with respect to β_k, set it to zero, and solve for β_k to obtain the following update equation:

$$\boldsymbol{\beta}_k = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{D})^{-1}\boldsymbol{X}^T\boldsymbol{y}_k \tag{23}$$

where $\boldsymbol{D}$ is a $J \times J$ diagonal matrix with $\sum_{v \in V} w_v^2/d_{j,v}$ in the jth element along the diagonal.

Disclaimer: Some portions of the content have been taken directly/paraphrased from the required reading papers[1][2] and last years scribe notes[3]. The authors do not claim the scribe to be a completely original work.

[1] https://www.cs.cmu.edu/~epxing/papers/2009/kim_xing_PlosG2009.pdf
[2] http://arxiv.org/pdf/1005.4717.pdf
[3] https://www.cs.cmu.edu/~epxing/Class/10708-13/lecture/scribe25.pdf