10-708: Probabilistic Graphical Models 10-708, Spring 2016

25 : Graphical induced structured input/output models

Lecturer: Eric P. Xing *Scribes: Raied Aljadaany, Shi Zong, Chenchen Zhu*

Disclaimer: A large part of the content are taken from the references listed.

For this section, we will extend the concept of traditional graphical models from modeling dependencies of a distribution or minimizing the loss function on graphs to modeling constraints.

1 Genetic Basis of Diseases

A single nucleotide polymorphism, often abbreviated to SNP, is a variation in a single nucleotide that occurs at a specific position in the genome[1]. Genetic Association Hypothesis testing aims at finding which SNP's are causal (or associated) vis-a-vis a hereditary disease. In other words, we hope to find out the mapping between genotype and phenotype. Until now, most popular approaches for genetic and molecular analysis of diseases are mainly based on classical statistical techniques, such as 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 include 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, consist of a large number of highly related, rather than independent, clinical or molecular phenotypes. This raises a new technical challenge in identifying genetic variations associated simultaneously with correlated traits.

In this lecture, we will see several methods to analyze the multi-correspondence mapping between multiple SNP's and (multiple) symptoms phenotypes.

2 Basics

2.1 Sparse Learning

Assume we now have a linear model,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{N \times J}$ is the input matrix, $\mathbf{y} \in \mathbb{R}^{N \times 1}$ is the output matrix, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_{N \times N})$ is an error term of length N with zero mean and a constant variance.

Then the lasso can be formulated as:

$$\arg\min f(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1 \tag{2}$$

[1] https://en.wikipedia.org/wiki/Single-nucleotide_polymorphism

where $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{J} |\beta_j|$ is defined to be the sum of all the absolute values of elements in $\boldsymbol{\beta}$.

Figure 1 provides a geometric view of lasso.

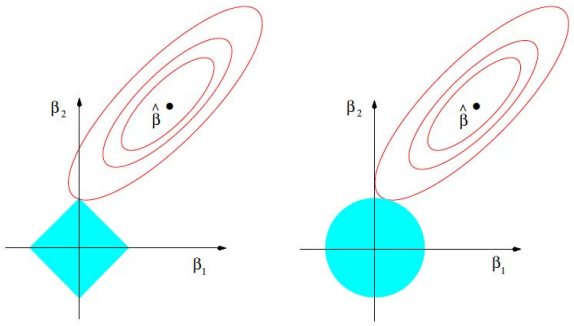

Figure 1 Geometric view of lasso. *left*: ℓ_1 regularization (lasso); *right*: ℓ_2 regularization (ridge regression). Image taken from [Hastie et al., 2009].

2.2 Multi-Task Learning

The basic linear model for multi-task regression can be written as:

$$\mathbf{y}_k = \mathbf{X}\boldsymbol{\beta}_k + \boldsymbol{\epsilon}_k, \forall k = 1, 2, ..., K \tag{3}$$

where $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_J] \in \mathbb{R}^{N \times J}$ denotes the input matrix, $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_K] \in \mathbb{R}^{N \times K}$ denotes the output matrix, $\boldsymbol{\beta}_k = [\beta_{1k}, ..., \beta_{Jk}] \in \mathbb{R}^J$ is a regression parameter term of length J for the k-th output, and $\boldsymbol{\epsilon}_k$ is an error term of length N with zero mean and a constant variance.

Denote $\mathbf{B}$ as a combination of all $\boldsymbol{\beta}_k$, which is

$$\mathbf{B} = (\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K) = \begin{pmatrix} \beta_{11} & \beta_{12} & ... & \beta_{1K} \\ \beta_{21} & \beta_{22} & ... & \beta_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{J1} & \beta_{J2} & ... & \beta_{JK} \end{pmatrix} \tag{4}$$

We can use lasso to solve the equation 3, which is to solve the following optimization problem:

$$\hat{\mathbf{B}}^{\text{lasso}} = \arg\min \sum_k (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)^T (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k) + \lambda \sum_j \sum_k |\beta_{jk}| \tag{5}$$

where λ is a tuning parameter that controls the level of sparsity. A larger λ would lead to a sparser solution.

In multi-task learning, the goal is to select input variables that are relevant to at least one task. Thus, an ℓ_1/ℓ_2 penalty has been proposed. Here ℓ_2 penalty comes from taking the ℓ_2 norm of the regression coefficients $\boldsymbol{\beta}^j$ for all outputs for each input j, and ℓ_1 penalty comes from taking sum of the above J ℓ_2 norms, which could encourage sparsity across input variables. The ℓ_1/ℓ_2 penalized multi-task regression is

defined as follows:

$$\hat{\mathbf{B}}^{\ell_1/\ell_2} = \arg\min \sum_k (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)^T (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k) + \lambda \sum_j \|\boldsymbol{\beta}^j\|_2 \tag{6}$$

Here ℓ_1 part enforces sparsity, and ℓ_2 part combines information across tasks. As all of the elements of $\boldsymbol{\beta}^j$ would take non-zero values if the j-th input is selected, the estimation $\hat{\mathbf{B}}^{\ell_1/\ell_2}$ is sparse only across inputs but not across outputs.

2.3 Structure Association

Specific to the genetic basis of diseases example, we can formulate it as a regression problem. That is, given multivariate input $\mathbf{X}$ (the SNPs) and multivariate output $\mathbf{Y}$ (the phenotypes), we hope to identify the association $\mathbf{B}$ between $\mathbf{X}$ and $\mathbf{Y}$. This matrix $\mathbf{B}$ encodes the structure and strength of the association, e.g. the parameter β_{jk} represents the association strength between SNP j and trait k.

Here the output covariates $\mathbf{Y}$ can be a graph connecting the phenotypes, a tree structure connecting genes etc. We will mainly consider three types of structure association.

- Association to a graph-structured phenome: Graph-guided fused lasso [Kim and Xing, 2009]
- Association to a tree-structured phenome: Tree-guided group lasso [Kim and Xing, 2010]
- Association between a subnetwork of genome and a subnetwork of phenome: Two-graph guided multi-task lasso [Chen et al., 2012]

3 Structure Association I: Graph-guided Fused lasso

3.1 Motivation

To capture correlated genome associations to a Quantitative Trait Network (QTN), we employ a multivariate linear regression model as the basic model for trait responses given inputs of genome variations such as SNPs, with the addition of a sparsity-biasing regularizer to encourage selection of truly relevant SNPs in the presence of many irrelevant ones. 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 by a common set of genetic markers. Thus, the GFlasso 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.

Figure 2 provides a visualization about two different choices of the fusion scheme, which leads to two variants of GFlasso: Graph-constrained Fused lasso (G_cFlasso) and Graph-weighted Fused lasso (G_wFlasso) .

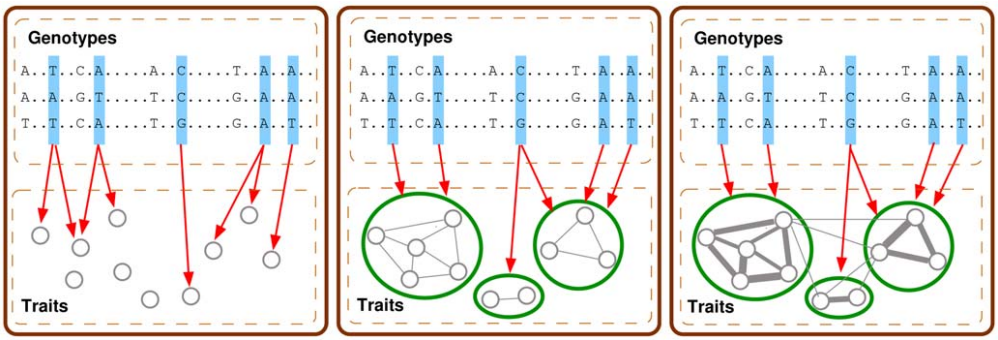

Figure 2 Illustrations for association analysis with multiple quantitative traits using various regression methods. *left*: original lasso; *middle*: Graph-constrained Fused lasso (G_cFlasso); *right*: Graph-weighted Fused lasso (G_wFlasso). Image taken from [Kim and Xing, 2009].

3.2 Model I: Graph-constrained Fused lasso

As shown in the middle subfigure of Figure 2, G_cFlasso model considers the graph structure without edge-weights. Formally, G_cFlasso can be formulated as:

$$\hat{\mathbf{B}}^{\mathrm{GC}} = \arg\min \sum_k (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)^T (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k) + \lambda \underbrace{\sum_k \sum_j |\beta_{jk}|}_{\text{lasso penalty}} + \gamma \underbrace{\sum_{(m,l)\in E} \sum_j |\beta_{jm} - \mathrm{sign}(r_{ml})\beta_{jl}|}_{\text{Graph-constrained fusion penalty}} \quad (7)$$

where E is the set of edges. Here the last term (which we refer to as a fusion penalty or a total variation cost) encourages (but does not strictly enforce) β_{jm} and $\mathrm{sign}(r_{ml})\beta_{jl}$ to take the same value by shrinking the difference between them toward zero. γ is a tuning parameter and a larger value for γ leads to a greater fusion effect, or in other words, a sparser result.

3.3 Model II: Graph-weighted Fused lasso

As shown in the right subfigure of Figure 2, G_wFlasso model not only considers the graph structure, but also considers the edge weights. Formally, G_wFlasso can be formulated as:

$$\hat{\mathbf{B}}^{\mathrm{GW}} = \arg\min \sum_k (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)^T (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k) + \lambda \underbrace{\sum_k \sum_j |\beta_{jk}|}_{\text{lasso penalty}} + \gamma \underbrace{\sum_{(m,l)\in E} f(r_{ml}) \sum_j |\beta_{jm} - \mathrm{sign}(r_{ml})\beta_{jl}|}_{\text{Graph-weighted fusion penalty}} \quad (8)$$

The G_wFlasso method weights each term in the fusion penalty in equation 8 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, G_wFlasso 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.

3.4 Optimization Problem

The optimization problem in equation 7 and equation 8 are convex and thus can be formulated as a quadratic programming problem. There are several existing tools for solving the quadratic programming problem. But there are some issues:

- 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 7 and equation 8, as they are non-smooth function of the ℓ_1 norm.

Here, take G_wFlasso as an example, we may reformulate it into an equivalent form that only involves smooth functions, which is:

$$\begin{aligned}\min_{\beta_k, d_{jk}, d_{jml}} \sum_k (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)^T(\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k) + \lambda \sum_{jk} \frac{\beta_{jk}^2}{d_{jk}} + \gamma \sum_{(m,l)\in E} f(r_{ml})^2 \sum_j \frac{(\beta_{jm} - \text{sign}(r_{ml})\beta_{jl})^2}{d_{jml}} \\ \text{subject to} \quad \sum_{j,k} d_{jk} = 1 \\ \sum_{(m,l)\in E} \sum_j d_{jml} = 1 \\ d_{jk} \geq 0 \quad \forall j,k \\ d_{jml} \geq 0 \quad \forall j, (m,l) \in E\end{aligned} \tag{9}$$

We can solve the above problem by coordinate-descent algorithm, that is we iteratively update β_k, d_{jk} and d_{jml}, until there is little improvement in the value of the objective function. By taking derivatives with respect to a specific variable and set it to be zero while keeping other variables fixed, we could get the following update rules:

$$\beta_{jk} = \frac{\sum_i x_{ij}\left(y_{jk} - \sum_{j'\neq j} x_{ij'}\beta_{j'k}\right) + \gamma\left(\sum_{(k,l)\in E} \frac{f(r_{kl})^2 \text{sign}(r_{kl})\beta_{jl}}{d_{jkl}} + \sum_{(m,k)\in E} \frac{f(r_{mk})^2 \text{sign}(r_{mk})\beta_{jm}}{d_{jmk}}\right)}{\sum_i x_{ij}^2 + \frac{\lambda}{d_{jk}} + \gamma \sum_{(k,l)\in E} \frac{f(r_{kl})^2}{d_{jkl}} + \gamma \sum_{(m,k)\in E} \frac{f(r_{mk})^2}{d_{jmk}}} \tag{10}$$

$$d_{jk} = \frac{|\beta_{jk}|}{\sum_{j',a} |\beta_{j'a}|} \tag{11}$$

$$d_{jml} = \frac{f(r_{ml})|\beta_{jm} - \text{sign}(r_{ml}\beta_{jl})|}{\sum_{(a,b)\in E} \sum_{j'} f(r_{ab})|\beta_{j'a} - \text{sign}(r_{ab})\beta_{j'b}|} \tag{12}$$

4 Structure Association II: Tree-guided Group lasso

4.1 Motivation

In a univariate-output regression setting, sparse regression methods that extend lasso have been proposed to allow the recovered relevant inputs to reflect the underlying structural information among the inputs.

Group lasso achieved this by applying an ℓ_1 norm of the lasso penalty over groups of inputs, while using an ℓ_2 norm for the input variables within each group. This ℓ_1/ℓ_2 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 a large number of groups are more heavily penalized than for other outputs with memberships to fewer groups. Thus, a tree-guided lasso for multi-task regression with structured sparsity has been proposed.

Considering tree-guided lasso has several advantages, such as:

- A tree structure would naturally represent a hierarchical structure;
- Compared to 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.

4.2 Examples of Constructing Penalties with Tree Structure

The ℓ_1-penalized regression assumes that all outputs in the problem share the common set of relevant input variables. But that is not always the case in practice. Here we consider a simple case of two genes. As shown in Figure 3, low height from nodes to their parents leads to a tight correlation and we need to select them jointly, while high height leads to a weak correlation and we need to select them separately.

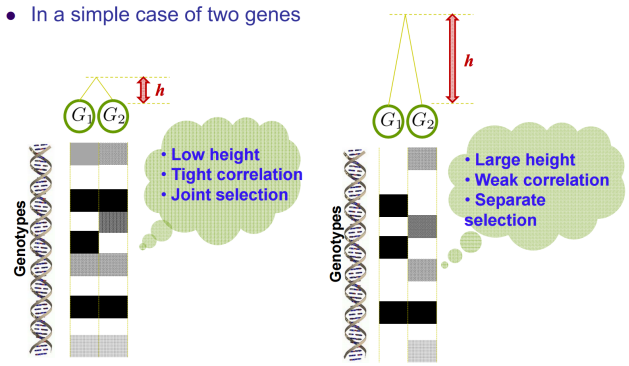

Figure 3 Two genes example for tree-structured penalty construction.

Based on the above intuition, we can revise the original ℓ_1 penalty to fit this tree structure. One possible way to formulate penalty is as follows:

$$\text{penalty} = \lambda \sum_j \left[h \underbrace{(|\beta_1^j| + |\beta_2^j|)}_{\ell_1 \text{ regularization}} + (1-h) \underbrace{\sqrt{(\beta_1^j)^2 + (\beta_2^j)^2}}_{\ell_2 \text{ regularization}} \right] \tag{13}$$

Here, ℓ_1 penalty means selecting β_{jk}s for two nodes separately while ℓ_2 penalty means selecting them jointly. h is the tuning parameter that can control the level of balance.

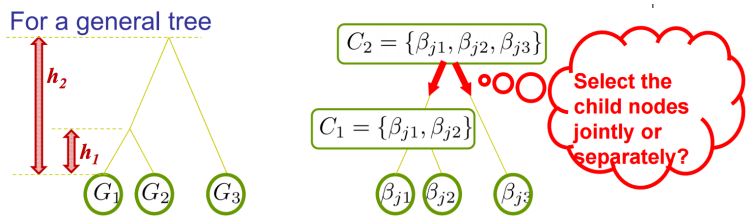

Figure 4 General tree for tree-structured penalty construction.

For a general tree as shown in Figure 4, the penalty can be formulated as:

$$\hat{\mathbf{B}}^{\text{Tree}} = \arg\min \sum_k (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)^T (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k) + \lambda \sum_j \left[h_2 \underbrace{(|C_1| + |\beta_2^j|)}_{\text{separate selection}} + (1-h_2) \underbrace{\sqrt{(\beta_1^j)^2 + (\beta_2^j)^2 + (\beta_3^j)^2}}_{\text{joint selection}} \right] \tag{14}$$

where C_1 can be recursively extended as:

$$C_1 = h_1(|\beta_1^j| + |\beta_2^j|) + (1-h_1)\sqrt{(\beta_1^j)^2 + (\beta_2^j)^2} \tag{15}$$

4.3 Definition

Now we could formally formulate tree-guided group lasso.

Given the 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. In this tree T, each node $v \in V$ 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:

$$\hat{\mathbf{B}}^{\text{Tree}} = \arg\min \sum_k (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)^T (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k) + \lambda \sum_j \sum_{v \in V} w_v \|\boldsymbol{\beta}_{G_v}^j\|_2 \tag{16}$$

where $\boldsymbol{\beta}_{G_v}^j$ is a vector of regression coefficients. Each group of regression coefficients $\boldsymbol{\beta}_{G_v}^j$ is weighted with w_v that reflects the strength of correlation within the group.

In order to define the weights of w_v, we first associate each internal node v of the tree T with two quantities s_v and g_v that satisfy the condition $s_v + g_v = 1$. Here the s_v represents the weight for selecting the output variables associated with each of the children of node v separately, and the g_v represents the weight for selecting them jointly.

Section 4.2 has given out an example of constructing tree-structured penalty. Here we give out a formal formulation. Given an arbitrary tree T , by recursively applying the similar operation starting from the root node towards the leaf nodes, we could get as follows:

$$\sum_j \sum_{v \in V} w_v \|\boldsymbol{\beta}_{G_v}^j\|_2 = \lambda \sum_j W_j(v_{\text{root}}) \tag{17}$$

where

$$W_j(v) = \begin{cases} s_v \cdot \sum_{c \in \text{Children}(v)} |W_j(c)| + g_v \cdot \|\boldsymbol{\beta}^j_{G_v}\|_2 & \text{if } v \text{ is an internal node,} \\ \sum_{m \in G_v} |\beta^j_m| & \text{if } v \text{ is a leaf node.} \end{cases} \tag{18}$$

4.4 Parameter Estimation

We use an alternative formulation in order to estimate the regression coefficients in tree-guided group lasso.

$$\hat{\mathbf{B}}^{\text{Tree}} = \arg\min \sum_k (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)^T (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k) + \lambda \underbrace{\left(\sum_j \sum_{v \in V} w_v \|\boldsymbol{\beta}^j_{G_v}\|_2 \right)^2}_{\text{relaxation needed on this term}} \tag{19}$$

As ℓ_1/ℓ_2 norm is a non-smooth function, we need to make some relaxation using the fact that the variational formulation of a mixed-norm regularization is equivalent to a weighted ℓ_2 regularization, which is:

$$\left(\sum_j \sum_{v \in V} w_v \|\boldsymbol{\beta}^j_{G_v}\|_2 \right)^2 \leq \sum_j \sum_{v \in V} \frac{w_v^2 \|\boldsymbol{\beta}^j_{G_v}\|_2^2}{d_{j,v}} \tag{20}$$

where $\sum_j \sum_v d_{j,v} = 1, d_{j,v} \geq 0, \forall j, v$, and the equality holds for

$$d_{j,v} = \frac{w_v \|\boldsymbol{\beta}_{j,v}\|_2}{\sum_j \sum_{v \in V} w_v \|\boldsymbol{\beta}_{j,v}\|_2} \tag{21}$$

Thus, the optimization problem listed in equation 19 can be rewritten as:

$$\begin{aligned} \min \sum_k (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)^T (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k) + \lambda \sum_j \sum_{v \in V} \frac{w_v^2 \|\boldsymbol{\beta}^j_{G_v}\|_2^2}{d_{j,v}} \\ \text{subject to} \quad \sum_j \sum_v d_{j,v} = 1, \\ d_{j,v} \geq 0, \forall j, v \end{aligned} \tag{22}$$

Here, additional variables $d_{j,v}$ are introduced for smoothing. We solve the problem in equation 22 by optimizing $\boldsymbol{\beta}$ and $d_{j,v}$ alternatively over iterations until convergence. For each iteration, we first fix the values for $\boldsymbol{\beta}_k$, and update $d_{j,v}$, where the update equations for $d_{j,v}$ are given in equation 21. Then, we treat $d_{j,v}$ as constant, and optimize for $\boldsymbol{\beta}_k$. It would lear to a closed-form solution, which is:

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

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

5 Structure Association III: Two-graph Guided Multi-task lasso

5.1 Motivation

Two-graph guided multi-task lasso tries to answer the question that how multiple genetic variants in a biological process or pathway, by forming a subnetwork, jointly affect a subnetwork of multiple correlated traits. It is motivated by graph structures in both genome and phenome and tries to take advantage of the two side information simultaneously.

Figure 5 gives out a illustration of two-graph guided multi-task lasso. Here we can see that two traits connected in trait network are coupled though paths between the two nodes; and two SNPs connected in genome network are coupled through paths between the two nodes.

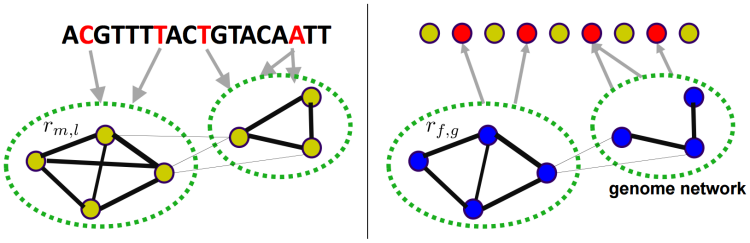

Figure 5 Illustration of two-graph guided multi-task lasso.

5.2 Parameter Estimation

The two-graph guided multi-task lasso is defined as:

$$\hat{\mathbf{B}}^{\text{TCML}} = \arg\min \sum_k (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k)^T (\mathbf{y}_k - \mathbf{X}\boldsymbol{\beta}_k) + \lambda \|\mathbf{B}\|_1 + \gamma_1 \underbrace{\text{pen}_1(E_1, \mathbf{B})}_{\text{Trait network}} + \gamma_2 \underbrace{\text{pen}_2(E_2, \mathbf{B})}_{\text{Genome network}} \tag{24}$$

where pen_1 and pen_2 are two penalty functions measuring the discrepancy between the prior label and feature graphs and the association pattern. Specifically, they can be defined as:

$$\begin{aligned} \text{pen}_1(E_1, \mathbf{B}) &= \sum_{e_{m,l} \in E_1} w(e_{m,l}) \sum_{j=1}^{J} |\beta_{jm} - \text{sign}(r_{m,l})\beta_{jl}| \\ \text{pen}_2(E_2, \mathbf{B}) &= \sum_{e_{f,g} \in E_2} w(e_{f,g}) \sum_{k=1}^{K} |\beta_{fk} - \text{sign}(r_{f,g})\beta_{gk}| \end{aligned} \tag{25}$$

where $w(e_{m,l})$ and $w(e_{f,g})$ are the weights assigned to the edge $e_{m,l}$ in graph E_1 and E_2, respectively. $r_{m,l}$ and $r_{f,g}$ are the correlations between $\mathbf{y}_m$ and $\mathbf{y}_l$, $\mathbf{y}_f$ and $\mathbf{y}_g$, respectively.

It is easy to see that the objective function in equation 24 is non-differentiable. Thus, its optimization is achieved by transforming it to a series of smooth functions that can be efficiently minimized by the coordinate-descent algorithm. Detailed update rules can be found in [Chen et al., 2012].

References

[Chen et al., 2012] Chen, X., Shi, X., Xu, X., Wang, Z., Mills, R., Lee, C., and Xu, J. (2012). A two-graph guided multi-task lasso approach for eqtl mapping. In Lawrence, N. D. and Girolami, M. A., editors, *AISTATS*, volume 22 of *JMLR Proceedings*, pages 208–217. JMLR.org.

[Hastie et al., 2009] Hastie, T. J., Tibshirani, R. J., and Friedman, J. H. (2009). *The elements of statistical learning : data mining, inference, and prediction*. Springer series in statistics. Springer, New York. Autres impressions : 2011 (corr.), 2013 (7e corr.).

[Kim and Xing, 2009] Kim, S. and Xing, E. P. (2009). Statistical estimation of correlated genome associations to a quantitative trait network. *PLoS Genet*, 5(8):1–18.

[Kim and Xing, 2010] Kim, S. and Xing, E. P. (2010). Tree-guided group lasso for multi-task regression with structured sparsity. In Frnkranz, J. and Joachims, T., editors, *ICML*, pages 543–550. Omnipress.