Supplementary Notes for Multi-Task Feature Selection on Multiple Networks via Maximum Flows

Mahito Sugiyama* Chloé-Agathe Azencott* Dominik Grimm*†
Yoshinobu Kawahara‡ Karsten M. Borgwardt*†

A Network Transformation for Maximum Flow Algorithms

We explain how our single-task formulation

$$\operatorname*{argmax}_{S \subset V} f(S) - g(S), \tag{A.1}$$

can be solved by the maximum flow algorithm. Given a network $G = (V, E)$, let us introduce a source node s and a sink node t and construct a transformed s/t-network $M(G) = (V', E')$ such that $V' = V \cup \{s, t\}$ and $E' = E \cup S \cup T$, where

$$S = \left\{ \{s, v\} \mid v \in V,\, q(v) > \eta \right\},$$
$$T = \left\{ \{t, v\} \mid v \in V,\, q(v) < \eta \right\}.$$

In addition, we set the capacity $c : E' \to \mathbb{R}^+$ of the edges in $M(G)$ to

$$c(\{v, u\}) = \begin{cases} |\, q(u) - \eta \,| & \text{if } u \in \{s, t\} \text{ and } v \in V, \\ \lambda w(\{v, u\}) & \text{otherwise.} \end{cases}$$

Figure S1 shows a simple example of a network G and its s/t-network.

Theorem A.1. ([S1]) *Let G be a network. The minimum s/t cut of the network $M(G)$, which gives the solution of the maximum flow problem from the source s to the sink t, coincides with the solution of the problem* (A.1) *on G.*

The proof is available in [S1] and can be derived similarly as in [S2]. Thus a maximum flow algorithm can be directly applied to solve the problem (A.1).

B Conditions for the Parametric Maximum Flow Algorithm

The anti-monotonicity of η, that is,

$$S(\eta) \subset S(\eta') \text{ if and only if } \eta > \eta',$$

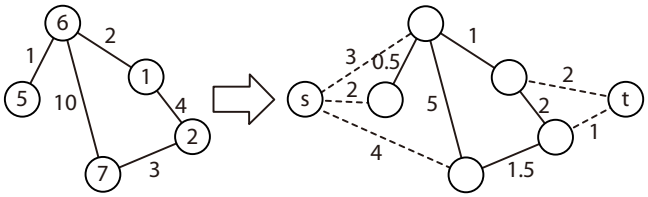

Figure S1: Example of a network (left) and its corresponding s/t-network (right) for the maximum flow problem. Numbers in circles denote values q and those on edges denote weights (left) and capacities (right). In this example, $\lambda = 0.5$ and $\eta = 3$.

is derived from noticing that η only affects the capacities on edges connected the source or the sink on the s/t network $M(G)$.

Moreover, the following three properties hold:

1. The capacity $c(\{s, v\})$ is a non-decreasing function of $-\eta$ for all $v \in V$.
2. The capacity $c(\{t, v\})$ is a non-increasing function of $-\eta$ for all $v \in V$.
3. The capacity $c(\{v, u\})$ is constant for all $v, u \in V$.

These properties exactly coincide with the assumptions necessary to the application of the parametric maximum flow algorithm.

C Multi-Task Lasso and Multi-Task Grace

Given a design matrix $\boldsymbol{X} \in \mathbb{R}^{N \times |V|}$ and K response vectors $\boldsymbol{y}_k$, the multi-task Lasso solves

$$\operatorname*{argmin}_{\boldsymbol{B} \in \mathbb{R}^{|V| \times K}} \frac{1}{2N} \sum_{k=1}^{K} ||\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k||_2^2 + \lambda\, ||\boldsymbol{B}||_{\ell_1/\ell_2}\,, \tag{C.2}$$

where the k-th column of $\boldsymbol{B}$ is the parameter vector $\boldsymbol{\beta}_k$ of the corresponding task and the ℓ_1/ℓ_2-norm of $\boldsymbol{B}$ is given by

$$||\boldsymbol{B}||_{\ell_1/\ell_2} = \sum_{v=1}^{|V|} \sqrt{\sum_{k=1}^{K} \beta_{k,v}^2} = \sum_{v=1}^{|V|} \|\boldsymbol{\beta}_v\|_2\,.$$

*Machine Learning and Computational Biology Research Group, Max Planck Institutes Tübingen, Germany
†Zentrum für Bioinformatik, Eberhard Karls Universität Tübingen, Germany
‡The Institute of Scientific and Industrial Research (ISIR), Osaka University, Japan

Grace can be extended to multi-task learning with a single network over the features as follows. Given a design matrix $\boldsymbol{X} \in \mathbb{R}^{N\times|V|}$, K response vectors $\boldsymbol{y}_k$, a network over the $|V|$ features described by its Laplacian L, and two parameters $\lambda_1, \lambda_2 \in \mathbb{R}$. We formulate Multi-Grace for K tasks as

(C.3)
$$\underset{\boldsymbol{B}\in\mathbb{R}^{|V|\times K}}{\operatorname{argmin}} \sum_{k=1}^{K} \left(\begin{array}{l} ||\boldsymbol{y}_k - \boldsymbol{X}\boldsymbol{\beta}_k||_2^2 + \lambda_1 ||\boldsymbol{B}||_{\ell_1/\ell_2} \\ +\lambda_2 \boldsymbol{\beta}_k^\top \boldsymbol{L}\boldsymbol{\beta}_k \end{array} \right).$$

Following the reasoning from Lemma 1 of [S3], this is equivalent to the following multi-task Lasso problem:

$$\underset{\boldsymbol{B}^*\in\mathbb{R}^{|V|\times K}}{\operatorname{argmin}} \sum_{k=1}^{K} \left(||\boldsymbol{y}_k^* - \boldsymbol{X}^*\boldsymbol{\beta}_k^*||_2^2 + \gamma ||\boldsymbol{B}^*||_{\ell_1/\ell_2} \right),$$

where $\gamma = \lambda_1/\sqrt{1+\lambda_2}$ and for each task, $(\boldsymbol{y}_k^*, \boldsymbol{X}^*)$ is an artificial dataset defined by

$$\boldsymbol{X}^* = (1+\lambda_2)^{-1/2} \begin{pmatrix} \boldsymbol{X} \\ \sqrt{\lambda_2}\boldsymbol{S}^\top \end{pmatrix}, \; \boldsymbol{y}_k^* = \begin{pmatrix} \boldsymbol{y}_k \\ \boldsymbol{0}_{|\mathbf{V}|} \end{pmatrix},$$

where $\boldsymbol{S}$ is such that $\boldsymbol{S}\boldsymbol{S}^\top = \boldsymbol{L}$. If $(\hat{\boldsymbol{\beta}}_k^*)_{k=1,\ldots,K}$ is the solution to this multi-task Lasso problem, then the solution to Equation C.3 is given by $\hat{\boldsymbol{\beta}}_k = \hat{\boldsymbol{\beta}}_k^*/\sqrt{1+\lambda_2}$.

Li and Li [S3] proposed to use a singular value decomposition to obtain $\boldsymbol{S}$, but the Lemma also holds if $\boldsymbol{S}$ is replaced by the *incidence matrix* of the network. As it can be constructed in linear time in the number of vertices and edges, this makes for a much faster implementation of Grace, aGrace and Multi-Grace. We used this implementation in our experiments.

If the different tasks have different networks, this derivation does not apply and solving a multi-task version of Grace is not straightforward any more.

D Generation of Synthetic Data

There are $2,200$ features in total composed of 200 transcription factors (TFs) and $2,000$ genes. Each TF is connected to 10 regulatory target genes. That is, we have a network $G = (V, E)$ such that

$$V = \bigcup_{i=1}^{200} \{t_i\} \cup G_i \text{ with } |G_i| = 10, \text{ and}$$
$$E = \bigcup_{i=1}^{200} E_i \text{ with } E_i = \{\{t_i, v\} \mid v \in G_i\},$$

which includes 200 connected subnetworks. Moreover, for each TF, an expression level x is generated from the normal distribution $N(0,1)$, and the expression levels for its regulatory genes are generated from $N(0.7x, 0.51)$. Thus the correlation between a TF and its regulatory genes is 0.7. Finally, we simulate a response vector from the linear model $y = \boldsymbol{X}\beta + \epsilon$ with i.i.d. noise $\epsilon \sim N(0, \sigma^2)$ and $\sigma^2 = \sum_i \beta_i^2/4$.

We prepare four models with different feature weights β and use their combination in the multi-task setting. In model 1,

$$\beta = \Big(5, \frac{5}{\sqrt{10}}, \ldots, \frac{5}{\sqrt{10}}, -5, \frac{-5}{\sqrt{10}}, \ldots, \frac{-5}{\sqrt{10}}, 3, \frac{3}{\sqrt{10}}, \ldots, \frac{3}{\sqrt{10}}, -3, \frac{-3}{\sqrt{10}}, \ldots, \frac{-3}{\sqrt{10}}, 0, \ldots, 0\Big).$$

Thus the first four TFs and their regulatory genes, that is to say 44 features in total, are causal to the response. Note that there is no edge between causal and non-causal features. Model 2 differs from model 1 in that the signs of the first three target genes in each subnetwork are flipped to their opposites, that is, For example, $\beta_2, \beta_3, \beta_4 = -5/\sqrt{10}$ and $\beta_5, \beta_6, \ldots, \beta_{11} = 5/\sqrt{10}$. This models a negatively correlated network. Model 3 (resp. model 4) is identical to model 1 (resp. model 2), except that all $\sqrt{10}$ in β are replaced with 10. Therefore the connection between TF and genes in models 3 and 4 is weaker than in models 1 and 2.

References

[S1] Azencott, C.A., Grimm, D., Sugiyama, M., Kawahara, Y., Borgwardt, K.M.: Efficient network-guided multi-locus association mapping with graph cuts. Bioinformatics 29(13), i171–i179 (2013)

[S2] Greig, D.M., Porteous, B.T., Seheult, A.H.: Exact maximum a posteriori estimation for binary images. Journal of the Royal Statistical Society: Series B 51(2), 271–279 (1989)

[S3] Li, C., Li, H.: Network-constrained regularization and variable selection for analysis of genomic data. Bioinformatics 24(9), 1175–1182 (2008)