
Structured Sparse Canonical Correlation Analysis

Xi Chen

Carnegie Mellon University
Pittsburgh, PA, USA
xichen@cs.cmu.edu

Han Liu

Johns Hopkins University
Baltimore, MD, USA
hanliu@cs.jhu.edu

Jaime G. Carbonell

Carnegie Mellon University
Pittsburgh, PA, USA
jgc@cs.cmu.edu

Abstract

In this paper, we propose to apply sparse canonical correlation analysis (sparse CCA) to an important genome-wide association study problem, eQTL mapping. Existing sparse CCA models do not incorporate structural information among variables such as pathways of genes. This work extends the sparse CCA so that it could exploit either the pre-given or unknown group structure via the *structured-sparsity-inducing penalty*. Such structured penalty poses new challenge on optimization techniques. To address this challenge, by specializing the excessive gap framework, we develop a scalable primal-dual optimization algorithm with a fast rate of convergence. Empirical results show that the proposed optimization algorithm is more efficient than existing state-of-the-art methods. We also demonstrate the effectiveness of the structured sparse CCA on both simulated and genetic datasets.

1 Introduction

A fundamental problem in genome-wide association study (GWA study or GWAS) is to understand associations between genetic variations and phenotypes. An important special case of GWA study is the *expression quantitative trait loci (eQTLs) mapping*. More specifically, the eQTL mapping discovers genetic associations between genotype data of single nucleotide polymorphisms (SNPs) and phenotype data of gene expression levels to provide insights into gene regulation, and potentially, controlling factors of a disease.

Appearing in Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX. Copyright 2012 by the authors.

More formally, we have two datasets \mathbf{X} (e.g. SNPs data) and \mathbf{Y} (e.g. gene expression data) of dimensions $n \times d$ and $n \times p$ collected on the same set of n observations. Both p and d could be much larger than n in an eQTL study. Our goal is to investigate the relationship between \mathbf{X} and \mathbf{Y} .

A popular approach for eQTL mapping is to formulate the problem into a sparse multivariate regression (Lee *et al.*, 2010; Kim *et al.*, 2009). These methods treat \mathbf{X} as input, \mathbf{Y} as output and try to identify a small subset of input variables that are simultaneously related to all the responses. Despite the promising aspects of these models, such multivariate-regression approaches are not symmetric in that the regression coefficients are only put on the \mathbf{X} side. There is no clear reason why one wants to regress \mathbf{Y} on \mathbf{X} but instead of \mathbf{X} on \mathbf{Y} for an association study. Also, in eQTL mapping, it is difficult to find a small subset of SNPs which can explain the expression levels for all the involved genes.

In contrast to sparse regression approach, sparse canonical correlation analysis (sparse CCA) (Witten *et al.*, 2009; Witten and Tibshirani, 2009) provides a more “symmetric” solution in which it finds two *sparse* canonical vectors \mathbf{u} and \mathbf{v} to maximize the correlation between $\mathbf{X}\mathbf{u}$ and $\mathbf{Y}\mathbf{v}$. Although sparse CCA has been successfully applied to some genomic datasets (e.g. CGH data (Witten and Tibshirani, 2009)), it has not well been studied for eQTL mapping. In a study of eQTL mapping, it is of great interest for biologists to seek for a subset of SNP genotypes and a subset of gene expression levels that are closely related. To address this problem, we apply sparse CCA to eQTL mapping; and show that by incorporating the proper structural information, sparse CCA can be a useful tool for GWAS.

It is well known that when dealing with high-dimensional data, prior structural knowledge is crucial for the analysis, which facilitates model’s interpretability. For example, a biological pathway is a *group* of genes that participate in a particular biological process to perform certain functionality in a cell. To find

the controlling factors related to a disease, it is more meaningful to study the genes by considering their pathways. However, the existing sparse CCA models use the ℓ_1 -regularization and do not incorporate the rich structural information among variables (e.g. genetic pathways). In this paper, we propose a *structured sparse CCA framework* that can naturally incorporate the group structural information. In particular, we consider two scenarios: (1) when the group structure is pre-given, we propose to incorporate such prior knowledge using the *overlapping-group-lasso penalty* (Jenatton *et al.*, 2009, 2010b). Compare to the standard group lasso (Yuan and Lin, 2006), we allow arbitrary overlaps among groups which reflects the fact that a gene may belong to multiple pathways. We refer to this model as the *group-structured sparse CCA*; (2) if such structural information is not available as a *priori*, we propose the *group pursuit sparse CCA* using a *group pursuit penalty*, which simultaneously conducts variable selection and structure estimation.

We formulate the structured sparse CCA into a bi-convex problem and adopt an alternating optimization strategy. However, unlike in Witten *et al.* (2009) where the simple ℓ_1 -norm penalty is used, our formulation involves the non-separable *overlapping-group-lasso* penalty and *group pursuit* penalty. Such non-separability poses great challenge on optimization techniques. Although many methods have been proposed (Duchi and Singer, 2009; Jenatton *et al.*, 2009, 2010a; Mairal *et al.*, 2010; Yuan *et al.*, 2011; Chen *et al.*, 2012; Argyriou *et al.*, 2011; Qin and Goldfarb, 2011) for solving the related sparse learning problems, as we surveyed in Section 3.1, they either (1) cannot be applied to our problem, or (2) suffer from slow convergence rate, or (3) have no convergence guarantees or convergence rate analysis. In this paper, we propose an efficient optimization algorithm under the excessive gap framework (Nesterov, 2003), which solves the sparse CCA with various *structured-sparsity-inducing penalties*. Since it is a first-order method, the per-iteration time complexity is very low (e.g. linear in the sum of group sizes) and the method can scale up to millions of variables. It is a primal-dual approach which diminishes the primal-dual gap over iterations. For each subproblem in the alternating optimization procedure, the algorithm provably converges to an ϵ accurate solution (i.e. the duality gap is less than ϵ) in $O(1/\sqrt{\epsilon})$ iterations.

2 Preliminaries

Given two datasets \mathbf{X} and \mathbf{Y} of dimensions $n \times d$ and $n \times p$ on the same set of n observations, we assume that each column of \mathbf{X} and \mathbf{Y} is normalized to have mean zero and standard deviation one. The sparse

CCA (Witten *et al.*, 2009) takes the form:

$$\begin{aligned} \max_{\mathbf{u}, \mathbf{v}} \quad & \mathbf{u}^T \mathbf{X}^T \mathbf{Y} \mathbf{v} \\ \text{s.t.} \quad & \|\mathbf{u}\|_2 \leq 1, \|\mathbf{v}\|_2 \leq 1, P_1(\mathbf{u}) \leq c_1, P_2(\mathbf{v}) \leq c_2, \end{aligned} \quad (1)$$

where the constraints $\|\mathbf{u}\|_2 \leq 1, \|\mathbf{v}\|_2 \leq 1$ are the convex relaxations of the equality constraints $\|\mathbf{u}\|_2 = 1, \|\mathbf{v}\|_2 = 1$ which ensure that the correlation is normalized. P_1 and P_2 are convex and non-smooth sparsity-inducing penalties that yield sparse \mathbf{u} and \mathbf{v} . We note that since the focus of the paper is on structured sparsity, we only consider the first pair of canonical vectors. Using the technique from Section 3 in Witten and Tibshirani (2009), we could easily extend our method to estimate multiple canonical vectors.

Witten *et al.* (2009) only studied two specific forms of the penalty P (either P_1 or P_2) with relatively simple structure: ℓ_1 -norm penalty and the chain-structured fused lasso penalty. In this work, we extend the sparse CCA to more general forms of P to incorporate the group structural information. In eQTL mapping, the structural knowledge among genes on \mathbf{Y} side is often of more interest. For the ease of illustration, we assume that $P_1(\mathbf{u}) = \|\mathbf{u}\|_1$ and mainly focus on $P_2(\mathbf{v})$, which incorporates the structural information. Since Eq. (1) is biconvex in \mathbf{u} and \mathbf{v} , a natural optimization strategy is the alternating approach: fix \mathbf{u} and optimize over \mathbf{v} ; then fix \mathbf{v} and optimize over \mathbf{u} ; and iterate over these two steps. In our setting, the optimization with respect to \mathbf{u} with $P_1(\mathbf{u}) = \|\mathbf{u}\|_1$ is relatively simple and the closed-form solution has been obtained in Witten *et al.* (2009). However, due to the complicated structure of $P_2(\mathbf{v})$, the optimization with respect to \mathbf{v} cannot be easily solved and we will address this challenge in the following section.

3 Group-structured Sparse CCA

In this section, we study the problem in which the group structural information among variables in \mathbf{Y} is pre-given from the domain knowledge; and our goal is to identify a small subset of relevant groups under the sparse CCA framework. More formally, let us assume that the set of groups of variables in \mathbf{Y} : $\mathcal{G} = \{g_1, \dots, g_{|\mathcal{G}|}\}$ is defined as a subset of the power set of $\{1, \dots, p\}$, and is available as prior knowledge. Note that the members (groups) of \mathcal{G} are allowed to overlap. Inspired by the *group-lasso penalty* (Yuan and Lin, 2006) and the *elastic-net penalty* (Zou and Hastie, 2005), we define our penalty $P_2(\mathbf{v})$ as follows:

$$P_2(\mathbf{v}) = \sum_{g \in \mathcal{G}} w_g \|\mathbf{v}_g\|_2 + \frac{c}{2} \mathbf{v}^T \mathbf{v}, \quad (2)$$

where $\mathbf{v}_g \in \mathbb{R}^{|g|}$ is the subvector of \mathbf{v} in group g , w_g is the predefined weight for group g ; c is the tuning

parameter. The ℓ_1/ℓ_2 mixed-norm penalty in $P_2(\mathbf{v})$ plays the role of group selection. Since some gene expression levels are highly correlated, the ridge penalty $\frac{\epsilon}{2}\mathbf{v}^T\mathbf{v}$ addresses the problem of the collinearity, enforcing strongly correlated variables to be in or out of the model together for better interpretability (Zou and Hastie, 2005). In addition, according to Zou and Hastie (2005), the ridge penalty is crucial to ensure stable variable selection performance when $p \gg n$, which is a typical setting of eQTL mapping.

Rather than solving the constraint form of $P_2(\mathbf{v})$, we solve the regularized problem using the Lagrange form:

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{v}} & -\mathbf{u}^T \mathbf{X}^T \mathbf{Y} \mathbf{v} + \frac{\tau}{2} \mathbf{v}^T \mathbf{v} + \theta \sum_{g \in \mathcal{G}} w_g \|\mathbf{v}_g\|_2 \\ \text{s.t.} & \|\mathbf{u}\|_2 \leq 1, \|\mathbf{v}\|_2 \leq 1, \|\mathbf{u}\|_1 \leq c_1, \end{aligned} \quad (3)$$

where there exists a one to one correspondence between (θ, τ) and (c, c_2) . We refer to this model as the *group-structured sparse CCA*.

3.1 Optimization Algorithm

The main difficulty in solving Eq. (3) arises from optimizing with respect to \mathbf{v} . Let the domain of \mathbf{v} be denoted as $Q_1 = \{\mathbf{v} \mid \|\mathbf{v}\|_2 \leq 1\}$, $\beta = \frac{1}{\tau} \mathbf{Y}^T \mathbf{X} \mathbf{u}$ and $\gamma = \frac{\theta}{\tau}$, the optimization of Eq. (3) with respect to \mathbf{v} can be written as:

$$\min_{\mathbf{v} \in Q_1} f(\mathbf{v}) \equiv l(\mathbf{v}) + P(\mathbf{v}), \quad (4)$$

where $l(\mathbf{v}) = \frac{1}{2} \|\mathbf{v} - \beta\|_2^2$ is the Euclidean distance loss function and $P(\mathbf{v})$ is the overlapping-group-lasso penalty: $P(\mathbf{v}) = \gamma \sum_{g \in \mathcal{G}} w_g \|\mathbf{v}_g\|_2$.

Related First-order Methods When \mathbf{v} is unconstrained and the groups are non-overlapped, the *closed-form optimal* solution can be easily obtained as in Duchi and Singer (2009). In contrast, when the groups are overlapped, the sub-gradient over each group becomes very complicated and hence there is no *closed-form* solution. The traditional interior point method and iterated reweighted least squares (Argyriou *et al.*, 2008) suffer from the high computational cost of solving a large linear system.

A number of first-order methods have recently been developed for solving variants of overlapping-group-lasso problem. The methods in Jenatton *et al.* (2010a) and Mairal *et al.* (2010) can only be applied to the tree-structured or ℓ_1/ℓ_∞ -regularized groups but not to ℓ_1/ℓ_2 -regularized overlapping group structure. The forward-backward splitting method in Duchi and Singer (2009) and smoothing proximal-gradient method in Chen *et al.* (2012) achieve slow convergence rate of $O(1/\epsilon)$. An algorithm for ℓ_1/ℓ_2 -regularized

overlapping group structure was proposed very recently in Yuan *et al.* (2011). However, due to the additional constraint $\|\mathbf{v}\|_2 \leq 1$, the key theorem in Yuan *et al.* (2011) no longer holds. Although it is still possible to apply a modification of this algorithm, the convergence rate is unknown. Other possible methods, including alternating direction augmented Lagrangian method (Qin and Goldfarb, 2011) and the fixed-point method (Argyriou *et al.*, 2011), also lack of the convergence rate.

In this section, by specializing a general *excessive gap* framework of Nesterov (2003), we present a first-order approach with a fast convergence rate of $O(1/\sqrt{\epsilon})$ for solving Eq. (4).

Reformulation of the Penalty As shown in Chen *et al.* (2012), the overlapping-group-lasso penalty $P(\mathbf{v})$ can be reformulated into a maximization form as follows. Using the dual norm, we have $\|\mathbf{v}_g\|_2 = \max_{\|\alpha_g\|_2 \leq 1} \alpha_g^T \mathbf{v}_g$ where α_g is a vector of length $|g|$. Let $\alpha = [\alpha_{g_1}^T, \dots, \alpha_{g_{|\mathcal{G}|}}^T]^T$ be the concatenation of the vectors $\{\alpha_g\}_{g \in \mathcal{G}}$. We denote the domain of α as $Q_2 \equiv \{\alpha \mid \|\alpha_g\|_2 \leq 1, \forall g \in \mathcal{G}\}$. The penalty $P(\mathbf{v})$ can be reformulated as:

$$P(\mathbf{v}) = \gamma \sum_{g \in \mathcal{G}} w_g \max_{\|\alpha_g\|_2 \leq 1} \alpha_g^T \mathbf{v}_g = \max_{\alpha \in Q_2} \alpha^T C \mathbf{v}, \quad (5)$$

where $C \in \mathbb{R}^{(\sum_{g \in \mathcal{G}} |g|) \times p}$ is defined as follows. The rows of C are indexed by all pairs of $(i, g) \in \{(i, g) \mid i \in g, i \in \{1, \dots, p\}\}$, the columns are indexed by $j \in \{1, \dots, p\}$, and $C_{(i, g), j} = \gamma w_g$ if $i = j$ and 0 otherwise.

Here, we provide some insights of this reformulation by showing its connection with *Fenchel Conjugate* (Borwein and Lewis, 2000). Let f^* denote the Fenchel Conjugate of a general function f . $P(\mathbf{v})$ can be viewed as Fenchel Conjugate of the indicator function

$$\delta_{Q_2}(\mathbf{x}) = \begin{cases} 0 & \mathbf{x} \in Q_2, \\ +\infty & \mathbf{x} \notin Q_2. \end{cases} \text{ at } C\mathbf{v}:$$

$$P(\mathbf{v}) = \max_{\alpha \in Q_2} \alpha^T C \mathbf{v} = \delta_{Q_2}^*(C\mathbf{v}).$$

Smoothing the Penalty With the penalty $P(\mathbf{v})$ in the form of $\max_{\alpha \in Q_2} \alpha^T C \mathbf{v}$, we construct a smooth approximation of $P(\mathbf{v})$ using the Nesterov's smoothing technique (Nesterov, 2005) as follows:

$$P_\mu(\mathbf{v}) = \max_{\alpha \in Q_2} \alpha^T C \mathbf{v} - \mu d(\alpha), \quad (6)$$

where μ is a positive smoothness parameter and $d(\alpha)$ is defined as $\frac{1}{2} \|\alpha\|_2^2$. The relationship between the smooth approximation $P_\mu(\mathbf{v})$ and original penalty $P(\mathbf{v})$ can be characterized by the following inequality:

$$P(\mathbf{v}) - \mu D \leq P_\mu(\mathbf{v}) \leq P(\mathbf{v}), \quad (7)$$

where $D = \max_{\alpha \in Q_2} d(\alpha)$. In our case, $D = |\mathcal{G}|/2$, where $|\mathcal{G}|$ is the number of groups.

According to Theorem 1 (Nesterov, 2005) as below, $P_\mu(\mathbf{v})$ is a smooth function for any $\mu > 0$.

Theorem 1 For any $\mu > 0$, $P_\mu(\mathbf{v})$ is convex and continuously-differentiable in \mathbf{v} with the gradient:

$$\nabla P_\mu(\mathbf{v}) = C^T \alpha_\mu(\mathbf{v}), \quad (8)$$

where $\alpha_\mu(\mathbf{v})$ is the optimal solution to Eq. (6):

$$\alpha_\mu(\mathbf{v}) = \operatorname{argmax}_{\alpha \in Q_2} \alpha^T C \mathbf{v} - \mu d(\alpha). \quad (9)$$

Performing some algebra, we obtain the closed-form equation for $\alpha_\mu(\mathbf{v})$ as in the next proposition.

Proposition 1 The $\alpha_\mu(\mathbf{v})$ in Eq. (9) is the concatenation of subvectors $\{\alpha_\mu(\mathbf{v})\}_g$ for all $g \in \mathcal{G}$. For any g , $[\alpha_\mu(\mathbf{v})]_g = S_2\left(\frac{\gamma w_g \mathbf{v}_g}{\mu}\right)$, where S_2 is the projection operator (to the ℓ_2 -ball) defined as follows: $S_2(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$ if $\|\mathbf{x}\|_2 > 1$ and $S_2(\mathbf{x}) = \mathbf{x}$ otherwise.

We substitute $P(\mathbf{v})$ in the original objective function $f(\mathbf{v})$ with $P_\mu(\mathbf{v})$ and construct the smooth approximation of $f(\mathbf{v})$: $f_\mu(\mathbf{v}) \equiv l(\mathbf{v}) + P_\mu(\mathbf{v})$. According to Eq. (7), for any $\mu > 0$:

$$f(\mathbf{v}) - \mu D \leq f_\mu(\mathbf{v}) \leq f(\mathbf{v}). \quad (10)$$

Therefore, $f_\mu(\mathbf{v})$ is a uniformly smooth approximation of the objective $f(\mathbf{v})$ with the maximum gap of μD , and μ controls the gap between $f_\mu(\mathbf{v})$ and $f(\mathbf{v})$.

Fenchel Dual of $f(\mathbf{v})$ The fundamental idea of the excessive gap method is to diminish the duality gap between the objective $f(\mathbf{v})$ and its Fenchel dual over iterations. According to Theorem 3.3.5 in Borwein and Lewis (2000), the Fenchel dual problem of $f(\mathbf{v})$, $\phi(\alpha)$, takes the following form:

$$\phi(\alpha) = -l^*(-C^T \alpha) - \delta_{Q_2}(\alpha), \quad (11)$$

where l^* is the Fenchel Conjugate of l and $-l^*(-C^T \alpha) = \min_{\mathbf{v} \in Q_1} \mathbf{v}^T C^T \alpha + \frac{1}{2} \|\mathbf{v} - \beta\|_2^2$. By a direct consequence of Theorem 1 in Nesterov (2005), we obtain the next theorem.

Theorem 2 The gradient of $\phi(\alpha)$ is as follows:

$$\nabla \phi(\alpha) = C \mathbf{v}(\alpha), \quad (12)$$

where $\mathbf{v}(\alpha) = \operatorname{argmin}_{\mathbf{v} \in Q_1} \mathbf{v}^T C^T \alpha + \frac{1}{2} \|\mathbf{v} - \beta\|_2^2$. Moreover, $\nabla \phi(\alpha)$ is Lipschitz continuous with the Lipschitz constant $L(\phi) = \frac{1}{\sigma} \|C\|^2$, where $\sigma = 1$ is the strongly convex parameter for function $l(\mathbf{v})$ and $\|C\|$ is the matrix spectral norm of C : $\|C\| \equiv \max_{\|\mathbf{x}\|_2=1} \|C\mathbf{x}\|_2$.

Utilizing Proposition 1 in Chen *et al.* (2012), we present the closed-form equations for $\mathbf{v}(\alpha)$ and $\|C\|$.

Proposition 2 $\mathbf{v}(\alpha)$ takes the following form: $\mathbf{v}(\alpha) = S_2(\beta - C^T \alpha)$, where S_2 is the projection operator (to the ℓ_2 -ball). $\|C\| = \gamma \max_{j \in \{1, \dots, p\}} \sqrt{\sum_{g \in \mathcal{G} \text{ s.t. } j \in g} (w_g)^2}$.

According to Proposition 2, the Lipschitz constant for $\nabla \phi(\alpha)$ is:

$$L(\phi) = \|C\|^2 = \gamma^2 \max_{j \in \{1, \dots, p\}} \sum_{g \in \mathcal{G} \text{ s.t. } j \in g} (w_g)^2. \quad (13)$$

Excessive Gap Method According to the Fenchel duality theorem (Borwein and Lewis, 2000), under certain mild conditions which hold for our problem, we have $\min_{\mathbf{v} \in Q_1} f(\mathbf{v}) = \max_{\alpha \in Q_2} \phi(\alpha)$, and

$$\phi(\alpha) \leq f(\mathbf{v}), \quad \forall \mathbf{v} \in Q_1, \alpha \in Q_2. \quad (14)$$

For each iteration t , the excessive gap method (Nesterov, 2003), simultaneously updates \mathbf{v} and α to guarantee that: $f_{\mu_t}(\mathbf{v}^t) \leq \phi(\alpha^t)$. According to Eq. (10) and (14):

$$f_{\mu_t}(\mathbf{v}^t) \leq \phi(\alpha^t) \leq f(\mathbf{v}^t) \leq f_{\mu_t}(\mathbf{v}^t) + \mu_t D. \quad (15)$$

To guarantee the convergence of the algorithm, the excessive gap method diminishes the value of the smoothness parameter μ_t over iterations: $\mu_{t+1} \leq \mu_t$ and $\lim_{t \rightarrow \infty} \mu_t = 0$. From Eq. (15), when $\mu_t \rightarrow 0$, we have $f(\mathbf{v}^t) \approx \phi(\alpha^t)$, which are hence the optimal primal and dual solution.

Moreover, in the excessive gap method, a gradient mapping operator $\psi: Q_2 \rightarrow Q_2$ is defined as follows,

$$\psi(\mathbf{z}) = \operatorname{argmax}_{\alpha \in Q_2} \left\{ \langle \nabla \phi(\mathbf{z}), \alpha - \mathbf{z} \rangle - \frac{1}{2} L(\phi) \|\alpha - \mathbf{z}\|_2^2 \right\}, \quad (16)$$

where \mathbf{z} is any point in Q_2 . In our problem, the gradient mapping operator ψ can also be computed in a closed-form as shown in the next proposition.

Proposition 3 For any $\mathbf{z} \in Q_2$, $\psi(\mathbf{z})$ in Eq. (16) is the concatenation of subvectors $[\psi(\mathbf{z})]_g$ for all groups $g \in \mathcal{G}$. For any g , $[\psi(\mathbf{z})]_g = S_2\left(\mathbf{z}_g + \frac{[\nabla \phi(\mathbf{z})]_g}{L(\phi)}\right)$, where $[\nabla \phi(\mathbf{z})]_g = \gamma w_g [\mathbf{v}(\mathbf{z})]_g$ is the g -th subvector of $\nabla \phi(\mathbf{z})$.

With Propositions 1, 2, 3 in place, all the essential ingredients of the excessive gap framework can be computed in a closed-form. We present the excessive gap method for solving Eq. (4) in Algorithm 1.

Convergence Rate and Time Complexity The Lemma 7.4 and Theorem 7.5 in Nesterov (2003) guarantee that both the starting points, \mathbf{v}^0 and α^0 , and

Algorithm 1 Excessive Gap for Solving Eq. (4)

Input: $\beta, \gamma, \mathcal{G}$ and $\{w_g\}_{g \in \mathcal{G}}$
Initialization: (1) Construct C ; (2) Compute $L(\phi)$ as in Eq. (13) and set $\mu_0 = 2L(\phi)$; (3) Set $\mathbf{v}^0 = \mathbf{v}(\mathbf{0}) = S_2(\beta)$; (4) Set $\alpha^0 = \psi(\mathbf{0})$
Iterate For $t = 0, 1, 2, \dots$, until convergence of \mathbf{v}^t :

1. Set $\tau_t = \frac{2}{t+3}$.
2. Compute $\alpha_{\mu_t}(\mathbf{v}^t)$ as in Proposition 1.
3. Set $\mathbf{z}^t = (1 - \tau_t)\alpha^t + \tau_t\alpha_{\mu_t}(\mathbf{v}^t)$.
4. Update $\mu_{t+1} = (1 - \tau_t)\mu_t$.
5. Compute $\mathbf{v}(\mathbf{z}^t) = S_2(\beta - C^T\mathbf{z}^t)$.
6. Update $\mathbf{v}^{t+1} = (1 - \tau_t)\mathbf{v}^t + \tau_t\mathbf{v}(\mathbf{z}^t)$.
7. Update $\alpha^{t+1} = \psi(\mathbf{z}^t)$ as in Proposition 3.

Output: \mathbf{v}^{t+1} .

the sequences, $\{\mathbf{v}^t\}$ and $\{\alpha^t\}$ in Algorithm 1 satisfy the key condition $f_{\mu_t}(\mathbf{v}^t) \leq \phi(\alpha^t)$. Using Eq. (15), the duality gap can be bounded by:

$$f(\mathbf{v}^t) - \phi(\alpha^t) \leq f_{\mu_t}(\mathbf{v}^t) + \mu_t D - \phi(\alpha^t) \leq \mu_t D. \quad (17)$$

From Eq. (17), we can see that the duality gap which characterizes the convergence rate is reduced at the same rate at which μ_t approaches to 0. According to Step 4 in Algorithm 1, the closed-form equation of μ_t can be written as:

$$\begin{aligned} \mu_t &= (1 - \tau_{t-1})\mu_{t-1} = \frac{t}{t+2} \cdot \frac{t-1}{t+1} \cdots \frac{2}{4} \cdot \frac{1}{3} \cdot \mu_0 \\ &= \frac{2}{(t+1)(t+2)}\mu_0 = \frac{4\|C\|^2}{(t+1)(t+2)}. \end{aligned} \quad (18)$$

Combining Eq. (17) and (18), we immediately obtain the convergence rate for duality gap of Algorithm 1.

Theorem 3 (Nesterov, 2003) *The duality gap between the primal solution $\{\mathbf{v}^t\}$ and dual solution $\{\alpha^t\}$ generated from Algorithm 1 satisfies: $f(\mathbf{v}^t) - \phi(\alpha^t) \leq \mu_t D = \frac{4\|C\|^2 D}{(t+1)(t+2)}$, where $D = \max_{\alpha \in Q_2} d(\alpha)$. In other words, if we require that the duality gap is less than ϵ , Algorithm 1 needs at most $\left\lceil 2\|C\| \sqrt{\frac{D}{\epsilon}} - 1 \right\rceil$ iterations.*

According to Theorem 3, Algorithm 1 converges in $O(1/\sqrt{\epsilon})$ iterations. The per-iteration time complexity of Algorithm 1 is linear in $p + \sum_{g \in \mathcal{G}} |g|$.

4 Group Pursuit in Sparse CCA

When the group information is not given *a priori*, it is of desire to automatically group the relevant variables into clusters under the sparse CCA framework. For this purpose, we propose the group pursuit sparse CCA in this section. Our group pursuit approach is based on pairwise comparisons between v_i and v_j for all $1 \leq i < j \leq p$: when $v_i = v_j$, the i -th and

j -th variables are grouped together. We identify all subgroups among p variables by conducting pairwise comparisons and applying transitivity rule, i.e. $v_i = v_j$ and $v_j = v_k$ implies that the i -th, j -th, and k -th variables are clustered into the same group. The pairwise comparisons can be naturally encoded in the *fusion penalty* (Tibshirani and Saunders, 2005; Kim *et al.*, 2009) $\sum_{i < j} |v_i - v_j|$, where the ℓ_1 -norm will enforce $v_i - v_j = 0$ for closely related (i, j) pairs.

In practice, instead of using the simple penalty $\sum_{i < j} |v_i - v_j|$ which treats each pair of variables equally, we could add the weight w_{ij} to incorporate the prior knowledge that how likely the i -th and j -th variables are in the same group. Moreover, the ℓ_1 -norm of \mathbf{v} is also incorporated in the penalty to enforce sparse solution as in the fused lasso model (Tibshirani and Saunders, 2005). Then, the *group pursuit penalty* takes the following form

$$P_2(\mathbf{v}) = \sum_{i < j} w_{ij} |v_i - v_j| + c' \|\mathbf{v}\|_1 + \frac{c}{2} \mathbf{v}^T \mathbf{v}, \quad (19)$$

where c' is the tuning parameter to balance the ℓ_1 -norm and the fusion penalty. This penalty function will simultaneously select the relevant variables and cluster them into groups in an automatic manner. A natural way for assigning w_{ij} is to set $w_{ij} = |r_{ij}|^q$, where r_{ij} is the correlation between the i -th and j -th variable; q models the strength of the prior: a larger q results in a stronger belief of the correlation based group structure. For the purpose of simplicity, we set $w_{ij} = |r_{ij}|$ with $q = 1$ in this paper; while in principle, any prior knowledge of the possibility of being in the same group can be incorporated into w . Note that the group pursuit penalty can be viewed as a variant of graph-guided fusion penalty (Kim *et al.*, 2009).

In some cases, we have the prior knowledge that the i -th and j -th variables do not belong to the same group; then the term $|v_i - v_j|$ should not appear in the group pursuit penalty Eq. (19). Therefore, rather than having $|v_i - v_j|$ for all (i, j) pairs which forms a complete graph, we generalize the group pursuit penalty with the fusion penalty defined on an *arbitrary graph* with the edge set E . In summary, the *group pursuit sparse CCA* is defined as follows:

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{v}} & -\mathbf{u}^T \mathbf{X}^T \mathbf{Y} \mathbf{v} + \frac{\tau}{2} \mathbf{v}^T \mathbf{v} + \theta_1 \|\mathbf{v}\|_1 + \theta_2 \sum_{(i,j) \in E} w_{ij} |v_i - v_j| \\ \text{s.t.} & \|\mathbf{u}\|_2 \leq 1, \|\mathbf{v}\|_2 \leq 1, P_1(\mathbf{u}) \leq c_1. \end{aligned} \quad (20)$$

It is straightforward to specialize excessive gap method for solving the group pursuit sparse CCA with a similar approach as in Section 3.1. Note that another non-convex group pursuit penalty has recently been proposed in (Shen and Huang, 2010). However, it is computationally very expensive due to the non-convexity of the penalty and could be easily trapped by local minima.

$ \mathcal{G} = 40, p = 40, 100$			
$\gamma = 0.4$	CPU	Primal Obj	Rel_Gap
ExGap	1.9767E-02	8.8682E+03	2.6663E-15
AG	3.5394E-01	8.8682E+03	—
ADAL	2.1506E+00	8.8682E+03	—
IPM	9.6420E+02	8.8682E+03	7.0490E-09
$\gamma = 4$	CPU	Primal Obj	Rel_Gap
ExGap	1.7907E-01	8.8851E+03	1.6285E-08
AG	3.0946E+00	8.8851E+03	—
ADAL	1.8290E+01	8.8851E+03	—
IPM	2.9520E+03	8.8851E+03	3.8840E-08
$ \mathcal{G} = 500, p = 500, 100$			
$\gamma = 5$	CPU	Primal Obj	Rel_Gap
ExGap	4.1094E+00	1.1211E+05	4.9669E-08
AG	1.0723E+01	1.1211E+05	—
ADAL	1.8374E+01	1.1211E+05	—
$\gamma = 10$	CPU	Primal Obj	Rel_Gap
ExGap	6.2159E+00	1.1219E+05	2.8000E-07
AG	9.0650E+00	1.1219E+05	—
ADAL	1.5146E+01	1.1219E+05	—
$ \mathcal{G} = 5,000, p = 5,000, 100$			
$\gamma = 10$	CPU	Primal Obj	Rel_Gap
ExGap	9.8349E+01	1.1240E+06	9.0019E-07
AG	8.5362E+02	1.1240E+06	—
ADAL	7.7661E+02	1.1240E+06	—
$\gamma = 20$	CPU	Primal Obj	Rel_Gap
ExGap	1.7264E+02	1.1245E+06	8.7364E-07
AG	1.0585E+03	1.1245E+06	—
ADAL	7.3510E+02	1.1245E+06	—

Table 1: Computational Efficiency Comparisons

5 Experiment

5.1 Computational Efficiency

We evaluate the scalability and efficiency of our excessive gap method (ExGap) for solving

$$\arg \min_{\mathbf{v}: \|\mathbf{v}\|_2 \leq 1} f(\mathbf{v}) = \frac{1}{2} \|\mathbf{v} - \boldsymbol{\beta}\|_2^2 + \gamma \sum_{g \in \mathcal{G}} \|\mathbf{v}_g\|_2,$$

where $\boldsymbol{\beta}$ is given. We compare ExGap with several state-of-the-art algorithms: (1) A modification of the accelerated gradient method (AG) (Yuan *et al.*, 2011) by adding the constraint; (2) Alternating direction augmented Lagrangian (ADAL) method (Qin and Goldfarb, 2011); (3) Interior point method (IPM) for second order cone programming by CVX (Grant and Boyd, 2011). All of the experiments are performed on a PC with Intel Core 2 Quad Q6600 2.4GHz CPU and 4GB RAM. The software is written in MATLAB. We terminate ExGap and IPM when the relative duality gap (Rel_Gap) is less than 10^{-6} : $\text{Rel_Gap} = \frac{|f(\mathbf{v}^t) - \phi(\boldsymbol{\alpha}^t)|}{1 + |f(\mathbf{v}^t)| + |\phi(\boldsymbol{\alpha}^t)|} \leq 10^{-6}$. For ADAL and (modified) AG, the dual solutions cannot be easily derived. We stop AG and ADAL when its objective is less than 1.00001 times the objective of ExGap.

We generate the simulated data with an overlapping group structure imposed on $\boldsymbol{\beta}$ as follows. Assuming that inputs are ordered and each group is of size 1000, we define a sequence of groups of 1000 adjacent inputs with an overlap of 100 variables between two successive groups, i.e. $\mathcal{G} = \{\{1, \dots, 1000\}, \{901, \dots, 1900\}, \dots, \{p - 999, \dots, p\}\}$ with $p = 1000|\mathcal{G}| + 100$. We set the support of $\boldsymbol{\beta}$ to the first half of the variables and set the values of $\boldsymbol{\beta}$ in the support to be 1 and otherwise 0.

We vary the number of the groups $|\mathcal{G}|$ and report the CPU time in *seconds* (CPU), primal objective (Primal Obj), relative duality gap (Rel_Gap) in Table 1. For each setting of $|\mathcal{G}|$, we use two levels of regularization parameter γ . Note that when $|\mathcal{G}| \geq 50$ ($p \geq 50, 100$), we are unable to collect results for IPM, because they lead to out-of-memory errors due to the large storage requirement for solving the Newton linear system. From Table 1, we can see ExGap is more efficient than both AG and ADAL in terms of CPU time. In addition, when p is smaller, AG is more efficient than ADAL; while for large p , ADAL seems to be more efficient. In addition, ExGap can easily scale up to high-dimensional data with millions of variables. Another interesting observation is that: for smaller γ which leads to smaller $\|C\|$ and $L(\phi)$, the convergence of ExGap is much faster. This observation suggests that ExGap is more efficient when the non-smooth part plays less important role in the optimization problem.

5.2 Simulation Study

In this and next subsection, we use simulated data and a real eQTL dataset to investigate the performance of the overlapping group-structured and group pursuit sparse CCA. All the regularization parameters are chosen from 0.01 to 10 and set using the permutation-based method in Witten and Tibshirani (2009). Instead of tuning all the parameters on a multi-dimensional grid which is computationally heavy, we first train the ℓ_1 -regularized sparse CCA (i.e. $P_1(\mathbf{u}) = \|\mathbf{u}\|_1$, $P_2(\mathbf{v}) = \|\mathbf{v}\|_1$) and the tuned regularization parameter c_1 in Eq. (1) is used for all structured models. For the overlapping-group-lasso penalty in Eq. (3), all the group weights $\{w_g\}$ are set to 1. In addition, we observe that the learned sparsity pattern is insensitive to the parameter τ ; and therefore we set it to 1 for simplicity. For all algorithms, we use 10 random initializations of \mathbf{u} and select the results that lead to the largest correlation.

Given Overlapping Group Structure In this section, we conduct the simulation where the overlapping group structure in \mathbf{v} is given as *a priori*. We generate the data \mathbf{X} and \mathbf{Y} with $n = 50$, $d = 100$ and $p = 82$ as follows. Let \mathbf{u} be a vector of length d with

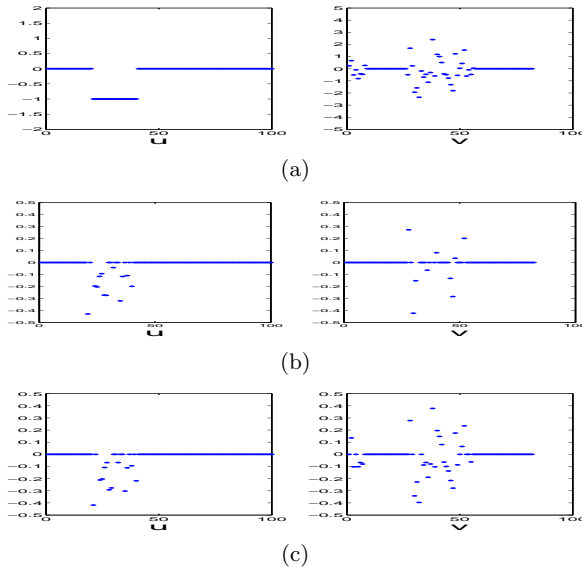


Figure 1: (a) True \mathbf{u} and \mathbf{v} ; (b) Estimated \mathbf{u} and \mathbf{v} using the ℓ_1 -regularized sparse CCA; (c) Estimated \mathbf{u} and \mathbf{v} using the group-structured sparse CCA.

20 0s, 20 -1s, and 60 0s. We construct \mathbf{v} with $p = 82$ variables as follows: assuming that \mathbf{v} is covered by 10 groups; each group has 10 variables with 2 variables overlapped between every two successive groups, i.e. $\mathcal{G} = \{\{1, \dots, 10\}, \{9, \dots, 18\}, \dots, \{73, \dots, 82\}\}$. For the indices of the 2nd, 3rd, 8th, 9th and 10th groups, we set the corresponding entries of \mathbf{v} to be zeros and the other entries are sampled from i.i.d. $N(0, 1)$. In addition, we randomly generate a latent vector \mathbf{z} of length n from $N(\mathbf{0}, \mathbf{I}_{n \times n})$ and normalize it to unit length.

We generate the data matrix \mathbf{X} with each $X_{ij} \sim N(z_i u_j, 1)$ and \mathbf{Y} with each $Y_{ij} \sim N(z_i v_j, 1)$. The true and estimated vectors for \mathbf{u} and \mathbf{v} are presented in Figure 1. For the group-structured sparse CCA, we add the regularization $\sum_{g \in \mathcal{G}} \|\mathbf{v}_g\|_2$ on \mathbf{v} where \mathcal{G} is taken from the prior knowledge. It can be seen that the group-structured sparse CCA recovers the true \mathbf{v} much better while the simple ℓ_1 -regularized sparse CCA leads to an over-sparsified \mathbf{v} vector.

Group Pursuit In this simulation, we assume that the group structure over \mathbf{v} is unknown and the goal is to uncover the group structure using the group pursuit sparse CCA. We generate the data \mathbf{X} and \mathbf{Y} with $n = 50$ and $p = d = 100$ as follows. Let \mathbf{u} be a vector of length d with 20 0s, 20 -1s, and 60 0s as in the previous simulation study; and \mathbf{v} be a vector of length p with 10 3s, 10 -1.5s, 10 1s, 10 2s and 60 0s. In addition, we randomly generate a latent vector \mathbf{z} from $N(\mathbf{0}, \mathbf{I}_{n \times n})$ and normalize it to unit length. We generate \mathbf{X} with each sample $\mathbf{x}_i \sim N(z_i \mathbf{u}, 0.1 \mathbf{I}_{d \times d})$; and \mathbf{Y} with each sample

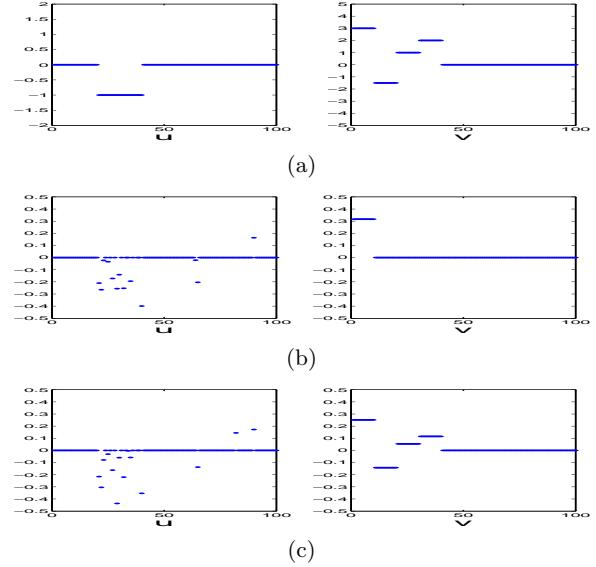


Figure 2: (a) True \mathbf{u} and \mathbf{v} ; (b) Estimated \mathbf{u} and \mathbf{v} using the ℓ_1 -regularized sparse CCA; (c) Estimated \mathbf{u} and \mathbf{v} using the group pursuit sparse CCA.

$\mathbf{y}_i \sim N(z_i \mathbf{v}, 0.1 \Sigma_y)$ where $(\Sigma_y)_{jk} = \exp^{-|v_j - v_k|}$. We conduct the group pursuit sparse CCA in Eq. (20), where we add the fusion penalty for each pair of variables in \mathbf{v} , i.e. E is the edge set of the complete graph. The estimated vector \mathbf{u} and \mathbf{v} are presented in Figure 2 (c). It can be easily seen that the group pursuit sparse CCA correctly captures the group structure in the \mathbf{v} vector.

5.3 Real eQTL Data Analysis

Pathway Selection via Group-structured Sparse CCA We analyze a yeast eQTL data in Zhu *et al.* (2008)¹: \mathbf{X} contains $d = 1260$ SNPs from the chromosomes 1–16 for $n = 114$ yeast strains. \mathbf{Y} is the gene expression data of $p = 1,155$ genes for the same 114 yeast strains. All these genes are in the KEGG database (Kanehisa and Goto, 2000) and belong to 92 pathways. We treat each pathway as a group. To achieve more refined resolution of gene selection, besides the 92 pathway groups, we also add in $p = 1,155$ groups where each group only has one singleton gene so that the solutions could be sparse at both the group and individual feature levels as in Friedman *et al.* (2010).

Using the group-structured sparse CCA, we select 121 SNPs and 47 genes. These 47 genes spread over 32 pathways. Using the tool ClueGO (Bindea *et al.*, 2009), we perform KEGG enrichment on the selected genes and the overview chart is presented in Figure

¹The full dataset could be downloaded from <http://blogs.ls.berkeley.edu/bremlab/data>

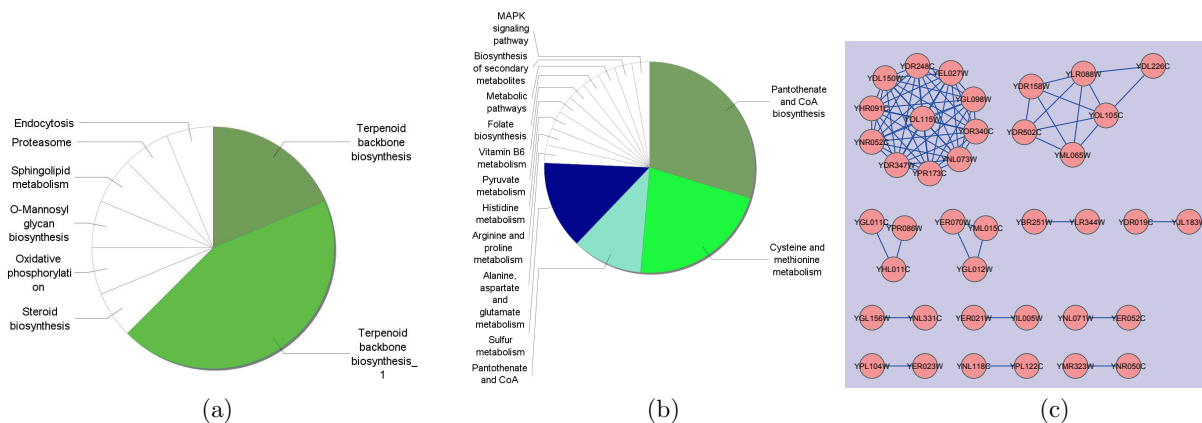


Figure 3: Overview chart of KEGG functional enrichment using (a) the group-structured sparse CCA; (b) ℓ_1 -regularized sparse CCA; (c) learned group structure using the group pursuit sparse CCA. In (a) and (b), the proportion of the pie chart represents the number of the selected genes with this annotation over the total number of the selected genes. Different colors are used to visually discriminate different functional groups. See Bindea *et al.* (2009) for more details of the presentation.

3(a). From Figure 3(a), we see that most pathways involve in the functional group *Terpenoid backbone biosynthesis*, which is a large class of natural products consisting of isoprene (C5) units. As a comparison, the ℓ_1 -regularized sparse CCA selects 173 SNPs and 71 genes and these 71 genes belong to 50 pathways. The enrichment of the selected genes by ℓ_1 -regularized sparse CCA is presented in Figure 3(b). We further perform GO enrichment analysis on the selected genes. Most Go terms obtained from group-structured sparse CCA has much less p -values than that from ℓ_1 -regularized sparse CCA. In addition, the overlapping-group-lasso penalty on genes will also affect the selection of SNPs. An interesting observation is that, most of the selected SNPs using the group-structured sparse CCA are concentrated on Chromosome 12 and 13.

In addition to the group-structured sparse CCA with the group extracted from KEGG pathways, we also perform the *tree-structured sparse CCA*, where we first run the hierarchical agglomerative clustering of the $p \times p$ correlation matrix of \mathbf{Y} ; define the groups by each tree node and then run the group-structured sparse CCA. The tree-structured sparse CCA selects 123 SNPs and 66 genes. The functional enrichment results show that most of the functions are identical to those learned by the group-structured sparse CCA with groups from KEGG, e.g. *Terpenoid backbone biosynthesis*, *Steroid biosynthesis*, *O-Mannosyl glycan biosynthesis*, *Sphingolipid*, etc. This experiment suggests that, even without any prior knowledge of the group structure, the correlation based tree-structured sparse CCA can also select the relevant genes and pro-

vide the similar enrichment results.

Group Pursuit Sparse CCA Now, we do not assume any prior information of the group structure among the genes. Our goal is to simultaneously select the relevant genes and group them into clusters. Using the group pursuit sparse CCA in Eq. (20) and thresholding the absolute value of the pair-wise correlation at 0.8 to construct the edge set E , we selected 61 genes in total. We present the obtained clusters among these 61 genes in Figure 3 (c) where two selected genes are connected if the absolute value of difference between the estimated parameters ($|v_i - v_j|$) is less than $1E-3$ (singleton nodes are not plotted due to space limitations). We observe that there are two obvious clusters. With the learned clustering structure, we can study the functional enrichment of each cluster separately, which could lead to more elaborate enrichment analysis as compared to the analysis of the selected genes all together.

6 Conclusions

In this paper, we propose a structured sparse CCA framework that can exploit either the pre-given or unknown group structural information. We also provide an efficient optimization algorithm that can scale up to very large problems.

7 Acknowledgement

We would like to thank Dr. Seyoung Kim for pointing out and helping us prepare the yeast data. We also thank Qihang Lin for helpful discussions on related first-order methods. Han Liu is supported by NSF grant IIS-1116730.

References

- Argyriou, A., Evgeniou, T., and Pontil, M. (2008). Convex multi-task feature learning. *Machine Learning*, **73**(3), 243–272.
- Argyriou, A., Micchelli, C. A., Pontil, M., Shen, L., and Xu, Y. (2011). Efficient first order methods for linear composite regularizers. arXiv:1104.1436v1.
- Bindea, G., Mlecnik, B., and et. al., H. H. (2009). Cluego: a cytoscape plug-in to decipher functionally grouped gene ontology and pathway annotation networks. *Bioinformatics*, **25** (8), 1091–1093.
- Borwein, J. and Lewis, A. S. (2000). *Convex Analysis and Nonlinear Optimization: Theory and Examples*. Springer.
- Chen, X., Lin, Q., Kim, S., Carbonell, J., and Xing, E. (2012). Smoothing proximal gradient method for general structured sparse learning. *Annals of Applied Statistics (to be appeared)*.
- Duchi, J. and Singer, Y. (2009). Efficient online and batch learning using forward backward splitting. *Journal of Machine Learning Research*, **10**, 2899–2934.
- Friedman, J., Hastie, T., and Tibshirani, R. (2010). A note on the group lasso and a sparse group lasso.
- Grant, M. and Boyd, S. (2011). CVX: Matlab software for disciplined convex programming, version 1.21. <http://cvxr.com/cvx>.
- Jenatton, R., Audibert, J., and Bach, F. (2009). Structured variable selection with sparsity-inducing norms. Technical report, INRIA.
- Jenatton, R., Mairal, J., Obozinski, G., and Bach, F. (2010a). Proximal methods for sparse hierarchical dictionary learning. In *ICML*.
- Jenatton, R., Obozinski, G., and Bach, F. (2010b). Structured sparse principal component analysis. In *AISTATS*.
- Kanehisa, M. and Goto, S. (2000). Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, **28**, 27–30.
- Kim, S., Sohn, K.-A., and Xing, E. P. (2009). A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics*, **25**(12), 204–212.
- Lee, S., Zhu, J., and Xing, E. P. (2010). Adaptive multi-task lasso: with applications to eqtl detection. In *NIPS*.
- Mairal, J., Jenatton, R., Obozinski, G., and Bach, F. (2010). Network flow algorithms for structured sparsity. In *NIPS*.
- Nesterov, Y. (2003). Excessive gap technique in non-smooth convex minimization. Technical report, Universit catholique de Louvain, Center for Operations Research and Econometrics (CORE).
- Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Math Programming*, **103**(1), 127–152.
- Qin, Z. and Goldfarb, D. (2011). Structured sparsity via alternating direction methods. Technical report, Columbia University.
- Shen, X. and Huang, H.-C. (2010). Grouping pursuit through a regularization solution surface. *Journal of the American Statistical Association*, **105**(490), 727–739.
- Tibshirani, R. and Saunders, M. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, **67**(1), 91–108.
- Witten, D. and Tibshirani, R. (2009). Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical Applications in Genetics and Molecular Biology*, **8** (1), 1–27.
- Witten, D., Tibshirani, R., and Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, **10**, 515–534.
- Yuan, L., Liu, J., and Ye, J. (2011). Efficient methods for overlapping group lasso. In *NIPS*.
- Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, **68**, 49–67.
- Zhu, J., Zhang, B., Smith, E. N., Beck, D., Drees, R. B. B., Kruglyak, L., Bumgarner, R. E., and Schadt, E. E. (2008). Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. *Nature Genetics*, **40**, 854–861.
- Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, **67**, 301–320.