Joint Estimation of Structured Sparsity and Output Structure in Multiple-Output Regression via Inverse-Covariance Regularization

Kyung-Ah Sohn, Seyoung Kim
School of Computer Science, Carnegie Mellon University

Abstract

We consider the problem of learning a sparse regression model for predicting multiple related outputs given high-dimensional inputs, where related outputs are likely to share common relevant inputs. Most of the previous methods for learning structured sparsity assumed that the structure over the outputs is known *a priori*, and focused on designing regularization functions that encourage structured sparsity reflecting the given output structure. In this paper, we propose a new approach for sparse multiple-output regression that can jointly learn both the output structure and regression coefficients with structured sparsity. Our approach reformulates the standard regression model into an alternative parameterization that leads to a conditional Gaussian graphical model, and employes an inverse-covariance regularization. We show that the orthant-wise quasi-Newton algorithm developed for L_1-regularized log-linear model can be adopted for a fast optimization for our method. We demonstrate our method on simulated datasets and real datasets from genetics and finances applications.

1 Introduction

In a regression estimation with inputs lying in a high-dimensional space, lasso [18] has been widely used to obtain a sparse estimate of parameters. Lasso optimizes the squared error loss function with an L_1 regularization to select only few input variables relevant to outputs with non-zero regression coefficients. While lasso was originally proposed for univariate-output regression, in this paper, we consider a sparse estimation of multiple-output regression, assuming that related outputs are likely to be affected by a common set of relevant inputs. Most of the previous approaches for combining statistical strength across multiple regression tasks to estimate such *structured sparsity* patterns in regression coefficients were based on introducing different types of regularization functions that reflect the prior knowledge on output structure representing how multiple outputs or tasks are related [15, 22, 10, 9, 7]. For example, when all of the outputs are believed to be related, mixed-norm penalties (e.g., L_1/L_2, L_1/L_∞ norms) have been used to recover a union support or the set of inputs that are relevant to all of the outputs jointly [15]. In order to incorporate a more complex output structure, these mixed-norm penalties have been further extended by allowing for multiple overlapping groups of related outputs [22, 10, 7].

While most of these previous works share the limitation that the output structure needs to be known *a priori*, in this paper, we propose a new approach for a sparse multiple-output regression estimation that can learn both the structured sparsity in regression coefficients and the output structure jointly. We assume that the output structure can be represented as a graph, where each node corresponds to an individual output variable and related outputs are connected with an edge. Then, the outputs that are related according to the graph are encouraged to share a common set of relevant inputs, leading to a structured sparsity pattern in regression coefficients. Although the similar problem setting has been considered in graph-guided fused lasso (GFlasso) [10], GFlasso was restrictive in that the output graph structure needs to be available as prior knowledge. In addition, rather than directly decoding the relationship between correlated outputs and their common relevant inputs, GFlasso used the heuristic approach that made the values of the regression coefficients for a given input to be the same for correlated outputs, which is a somewhat arbitrary assumption that may not hold in reality.

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

Instead of using the standard formulation for regression estimation that minimizes the squared error loss with a specific penalty function for structured sparsity, our proposed approach is based on a maximum-likelihood estimation with inverse-covariance regularization. More specifically, we first establish the connection between sparse inverse-covariance estimation (widely known as graphical lasso [6, 14]) and sparse multiple output regression, and propose an alternative and more powerful formulation for learning a sparse multiple-output regression via inverse-covariance regularization that can recover both the output structure and regression coefficients with structured sparsity. This alternative formulation leads to a convex optimization problem of semi-definite program. Instead of relying on the computationally inefficient interior-point method [3], we take advantage of the problem structure and show how the orthant-wise limited-memory quasi-Newton (OWL-QN) algorithm [1] for learning L_1-regularized log-linear model can be adopted for efficient optimization.

Although the connection between a single-output lasso and inverse-covariance regularization was first noticed by Witten and Tibshirani [20] in their method called Scout procedure, the approach we propose provides a significantly deeper insight on the full connection among multiple-output regression, inverse-covariance regularization, and structured sparsity. Scout procedure used inverse-covariance regularization to learn a single-output regression and the input structure, which we believe is counter-intuitive because a regression model is primarily concerned with the predictive model for the output given inputs rather than modeling the inputs themselves. In contrast, we show that in our formulation, inverse-covariance regularization can be used for predictive modeling in multiple-output setting. Furthermore, we show that this approach is equivalent to learning a sparse conditional Gaussian graphical model (CGGM) analogous to a conditional random field (CRF) [12] for structured-output prediction for discrete outputs. This connection between the regression model and graphical model allows us to recover sparsity pattern in both conditional and marginal distributions and to extract much richer information on sparsity patterns in parameters.

Another work that is closely related to ours is a multivariate regression with covariance estimation (MRCE) [16] that performs a joint estimation of sparse regression coefficients and covariance matrix for correlated noise across multiple outputs, where the noise covariance matrix can be viewed as output correlation structure. MRCE uses the standard formulation of regression model and optimizes the loss function with an L_1 regularization for both regression coefficients and noise covariance matrix. Although both MRCE and our proposed method address the problem of joint estimation of regression coefficients and output structure, they have three major differences. First, our formulation is convex with a single global optimum, whereas MRCE is only bi-convex. Second, as we show in our experiments, the optimization method for our approach is significantly faster than that of MRCE even with the approximate method suggested by Rothman et al. [16]. Third, our method explicitly recovers a shared sparsity pattern in regression coefficients for multiple related outputs, whereas MRCE does not provide any mechanism for enforcing structured sparsity.

The rest of the paper is organized as follows. In Section 2, we provide a brief review of the standard formulation of regression and various structured-sparsity-inducing penalties. In Section 3, we present our new formulation for multiple-output regression via inverse-covariance regularization and an efficient optimization algorithm. We evaluate our method on simulated and real datasets in Section 4, and conclude in Section 5.

2 Background on Sparse Multiple-Output Regression

Given J-dimensional inputs $\mathbf{x}_i = (x_{i1}, \ldots, x_{iJ})^T \in \mathbb{R}^J$ and K-dimensional outputs $\mathbf{y}_i = (y_{i1}, \ldots, y_{iK})^T \in \mathbb{R}^K$ for the ith sample, the functional mapping from the inputs to the outputs is often modeled as a linear regression model:

$$\mathbf{y}_i = \mathbf{B}\mathbf{x}_i + \boldsymbol{\epsilon}_i, \text{ for } i = 1, \ldots, N, \tag{1}$$

where $\mathbf{B}$ is the $K \times J$ matrix of regression coefficients β_{kj}'s, N is the number of samples, and $\boldsymbol{\epsilon}_i = (\epsilon_{i1}, \ldots, \epsilon_{iK})^T$ is the vector of length K for noise distributed as mean 0 and covariance $\boldsymbol{\Psi}$. The model in Eq. (1) contains a set of K linear regressions for predicting the K outputs given the common input space. Assuming that the input data are standardized to have mean 0 and unit variance and that the output data are centered, we consider the model without an intercept.

Let $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)^T$ and $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)^T$ denote the input and output data matrices. When $J >> N$ or $|S| << J$, where $S = \{(j,k) \,|\, \beta_{kj} \neq 0\}$ is the support of $\mathbf{B}$, the lasso [18] obtains a sparse estimate of $\mathbf{B}$ by solving the following optimization problem:

$$\arg\min_{\mathbf{B}} \quad \mathrm{tr}\big((\mathbf{Y} - \mathbf{X}\mathbf{B})(\mathbf{Y} - \mathbf{X}\mathbf{B})^T\big) + \lambda||\mathbf{B}||_1, \tag{2}$$

where $||\mathbf{B}||_1 = \sum_{k,j} |\beta_{kj}|$ is a matrix L_1 norm, and λ is the regularization parameter that controls the amount of sparsity in $\mathbf{B}$. A large value of λ leads to a sparser estimate with a greater number of zero elements in $\mathbf{B}$. The λ can be determined using cross-validation. Eq.

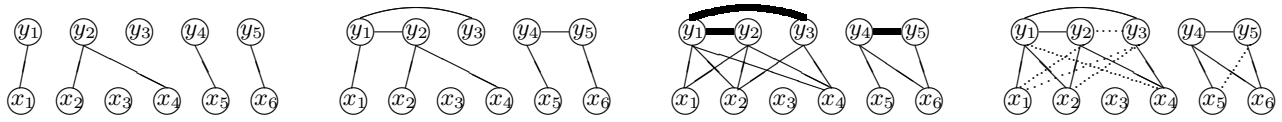

Figure 1: Illustration of different sparse multiple-output regression methods. (a) Lasso. (b) MRCE. (c) GFlasso. The graph structure over the outputs (shown as thick edges) is assumed to be known *a priori*. (d) Our proposed method. Solid edges represent direct influence, and the dotted edges indirect influence.

(2) is convex and efficient algorithms such as pathwise coordinate descent are available [5] for optimization.

Solving Eq. (2) amounts to treating the K regression problems as independent and performing K separate regression analyses, while assuming that the noise terms ϵ_{ik}'s in Eq. (1) are uncorrelated with zeros in the off-diagonal elements of $\mathbf{\Psi}$. In order to combine the statistical strengths across multiple regression tasks through the correlated outputs, MRCE proposed to estimate the full noise covariance matrix $\mathbf{\Psi}$ [16]. MRCE minimizes the negative log-likelihood function with an L_1 penalization for both $\mathbf{B}$ and $\mathbf{\Omega} = \mathbf{\Psi}^{-1}$:

$$\underset{\mathbf{B},\mathbf{\Omega}}{\operatorname{argmin}} \; -N \log |\mathbf{\Omega}| + \operatorname{tr}\big((\mathbf{Y} - \mathbf{XB})\mathbf{\Omega}(\mathbf{Y} - \mathbf{XB})^T\big) + \lambda_1 ||\mathbf{B}||_1 + \lambda_2 ||\mathbf{\Omega}||_1, \quad (3)$$

where λ_1 and λ_2 are the regularization parameters. Eq. (3) is not convex but bi-convex in $\mathbf{B}$ and $\mathbf{\Omega}$, as fixing either $\mathbf{B}$ or $\mathbf{\Omega}$ and solving for the other is a convex problem. Although Rothman et al. [16] proposed an alternate optimization of $\mathbf{B}$ and $\mathbf{\Omega}$ over iterations, they also noted that this method often does not converge, and suggested to use an approximate method that solves each of lasso and graphical lasso [6] once, followed by another round of optimization for $\mathbf{B}$ with $\mathbf{\Omega}$ fixed. This approximate method is also less costly in terms of computational time than the alternate optimization, as it is equivalent to terminating the alternate optimization prematurely after a single iteration.

Although MRCE considers the correlation structure in outputs through the noise model $\mathbf{\Psi}$, MRCE essentially selects relevant inputs for each output independently and does not learn structured sparsity in $\mathbf{B}$. In other words, MRCE does not have any mechanism that leverages the learned output structure to encourage related outputs to share relevant input variables.

In a different body of work on structured-sparsity-inducing norms, methods for encouraging shared relevant inputs for related outputs, assuming that the output structure is known, have been proposed. For example, when all of the outputs are believed to have the same relevant inputs, a mixed-norm penalty such as $J(\mathbf{B}) = \sum_j \sqrt{\sum_k \beta_{kj}^2}$ has been used [15]. For a more complex group structure, extensions of the mixed-norm penalty to overlapping groups have been proposed [22, 10, 7]. When the output structure is represented as a graph, GFlasso used the graph-guided fusion penalty $J(\mathbf{B}) = \sum_j \sum_{m,k} r_{mk} |\beta_{mj} - \operatorname{sign}(r_{mk})\beta_{kj}|$ to encourage the correlated outputs to share the same relevant inputs [9].

MRCE and the methods with structured-sparsity-inducing norms address complementary aspects of the problem of sparse multiple-output regression estimation. MRCE can learn the output structure but does not achieve structured sparsity, while the structured-sparsity-inducing norms can recover structured sparsity but have the limitation that the output structure must be known. Recently an integer programming method has been proposed that learns the grouping structure over outputs and shared relevant inputs for each group of outputs [8], assuming a predefined number of output groups. In the next section, we propose a new method for sparse multiple-output regression that combines the advantages of MRCE and structured-sparsity-inducing norms, assuming graph-structured outputs, which is a more expressive representation of structure than a simple grouping. The behaviors of different methods are illustrated in Figure 1.

3 Multiple-Output Regression with Inverse-Covariance Regularization

In order to perform a joint estimation of structured sparsity and output structure in multiple-output regression, instead of learning the regression model in the standard parameterization in Eq. (1), we introduce an alternative parameterization of the model derived as a conditional probabaility model $p(\mathbf{Y}|\mathbf{X})$ from the joint distribution $p(\mathbf{Y}, \mathbf{X})$, and propose to perform a maximum-likelihood estimation with inverse-covariance regularization. Our approach sheds new insights on the relationship among the standard regression model, Gaussian graphical model, and structured sparsity learning.

3.1 Multiple-Output Regression as Conditional Gaussian Graphical Model

In order to introduce our alternative formulation of the regression model in Eq. (1), we start by assuming a joint probability distribution for $\mathbf{x}_i$ and $\mathbf{y}_i$ as follows:

$$\begin{pmatrix} \mathbf{x}_i \\ \mathbf{y}_i \end{pmatrix} \sim N\left(\begin{pmatrix} \mathbf{0}_J \\ \mathbf{0}_K \end{pmatrix}, \mathbf{\Sigma}\right), \quad (4)$$

where $\mathbf{0}_D$ is a vector of D 0's, and $\mathbf{\Sigma} = \begin{pmatrix} \mathbf{\Sigma_{xx}} & \mathbf{\Sigma_{xy}} \\ \mathbf{\Sigma_{xy}}^T & \mathbf{\Sigma_{yy}} \end{pmatrix}$, $\mathbf{\Theta} = \mathbf{\Sigma}^{-1} = \begin{pmatrix} \mathbf{\Theta_{xx}} & \mathbf{\Theta_{xy}} \\ \mathbf{\Theta_{xy}}^T & \mathbf{\Theta_{yy}} \end{pmatrix}$. It is well-known that the $\mathbf{\Sigma}$ corresponds to the marginal independencies in pairs of variables after marginalizing over all the other variables, whereas the inverse covariance matrix $\mathbf{\Theta}$ encodes the conditional independence information for each pair of variables given all the other variables [11]. Thus, the $\mathbf{\Theta}$ represents a Gaussian graphical model, where a zero value in the (i, j)th entry in $\mathbf{\Theta}$ implies no edge between the ith and jth variables in the undirected graphical model [6].

From the above joint distribution, we derive the conditional distribution of $\mathbf{y}_i$ given $\mathbf{x}_i$ as follows:

$$\mathbf{y}_i|\mathbf{x}_i \sim N(\mathbf{\Sigma_{xy}}^T\mathbf{\Sigma_{xx}}^{-1}\mathbf{x}_i,\ \mathbf{\Sigma_{yy}} - \mathbf{\Sigma_{xy}}^T\mathbf{\Sigma_{xx}}^{-1}\mathbf{\Sigma_{xy}}), \quad (5)$$

and equivalently, using the inverse covariance matrix $\mathbf{\Theta}$ and the partitioned inverse formula [13],

$$\mathbf{y}_i|\mathbf{x}_i \sim N(-\mathbf{\Theta_{yy}}^{-1}\mathbf{\Theta_{xy}}^T\mathbf{x}_i,\ \mathbf{\Theta_{yy}}^{-1}). \quad (6)$$

We notice that Eqs. (1), (5), and (6) are different parameterizations of the same regression model, since $\mathbf{B} = \mathbf{\Sigma_{xy}}^T\mathbf{\Sigma_{xx}}^{-1} = -\mathbf{\Theta_{yy}}^{-1}\mathbf{\Theta_{xy}}^T$ and $\mathbf{\Psi} = \mathbf{\Sigma_{yy}} - \mathbf{\Sigma_{xy}}^T\mathbf{\Sigma_{xx}}^{-1}\mathbf{\Sigma_{xy}} = \mathbf{\Theta_{yy}}^{-1}$.

The equivalence of Eq. (1) and Eq. (6) shows that the standard regression estimation of $\mathbf{B}$ and $\mathbf{\Psi}$ is related to the problem of covariance estimation. This connection was first noticed by Witten and Tibshirani [20] and exploited in the case of a sparse univariate-output regression in their Scout procedure. However, we argue that this connection becomes significantly more valuable when we consider the multiple-output case, because of its equivalence to a graphical model for discriminative modeling that is analogous to conditional random field (CRF) [12] for structured-output prediction for discrete outputs. In order to see this, we further expand the quadratic term in the Gaussian distribution in Eq. (6) to obtain what we call a conditional Gaussian graphical model (CGGM):

$$p(\mathbf{y}_i|\mathbf{x}_i) = \exp\big(-1/2\,\mathbf{y}_i^T\mathbf{\Theta_{yy}}\mathbf{y}_i - \mathbf{x}_i^T\mathbf{\Theta_{xy}}\mathbf{y}_i\big) / Z(\mathbf{\Theta_{xy}}, \mathbf{\Theta_{yy}}, \mathbf{x}_i), \quad (7)$$

where

$$\begin{aligned} &Z(\mathbf{\Theta_{xy}}, \mathbf{\Theta_{yy}}, \mathbf{x}_i) \\ &= \int \exp\big(-\frac{1}{2}\,\mathbf{y}_i^T\mathbf{\Theta_{yy}}\mathbf{y}_i - \mathbf{x}_i^T\mathbf{\Theta_{xy}}\mathbf{y}_i\big)d\mathbf{y}_i \\ &= \sqrt{(2\pi)^K / \det\mathbf{\Theta_{yy}}}\,\exp(1/2\mathbf{x}_i^T\mathbf{\Theta_{xy}}\mathbf{\Theta_{yy}}^{-1}\mathbf{\Theta_{xy}}^T\mathbf{x}_i) \quad (8) \end{aligned}$$

is the partition function. As Eq. (7) is equivalent to the Guassian distribution in Eq. (6), the partition function can be obtained in a closed-form by directly comparing Eq. (7) with Eq. (6). If $\mathbf{\Theta_{yy}}$ is positive definite, the integral in the partition function is finite and the probability distribution is well-defined.

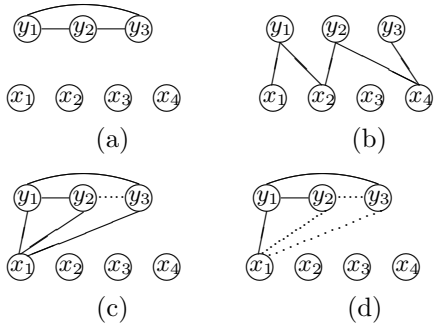

Figure 2: Illustration of the behavior of sparse CGGM. (a) When $\lambda_1 \to \infty$, we have the graphical lasso only on the output data. (b) When $\lambda_2 \to \infty$, it approaches the lasso. (c) Direct influence of input x_1 on outputs y_1, y_2, and y_3. (d) Direct influence of input x_1 on output y_1, and indirect influence on y_2 and y_3.

We argue that when our goal is to borrow statistical strength across multiple outputs for structured-output prediction, Eq. (7) provides a more natural representation of multiple output regression than the widely-used one in Eq. (1). This is because $\mathbf{\Theta_{yy}}$ directly parameterizes the conditional independence relationship among outputs. Then, $\mathbf{\Theta_{xy}}$ represents the direct influence of inputs on outputs that propagates through the output network $\mathbf{\Theta_{yy}}$ to influence other outputs indirectly. We point out that just as in CRF, Eq. (7) does not model the input distribution, since it includes parameters only for output interactions $\mathbf{\Theta_{yy}}$ and input-output interactions $\mathbf{\Theta_{xy}}$. This is in contrast with the work by Witten and Tibshirani [20], who used the representation in Eq. (6) for a different purpose of modeling input structure in a single-output regression.

3.2 Inverse-Covariance Regularization for Conditional Gaussian Graphical Models

Although in all of Eqs. (1), (6), and (7), the maximum-likelihood estimates lead to the same predictive model for outputs given inputs, enforcing sparsity in these different parameterizations leads to different sparsity patterns in $\hat{\mathbf{B}}$. In this section, instead of sparsifying $\mathbf{B}$ and $\mathbf{\Psi}$, we propose to regularize the inverse-covariance parameters, $\mathbf{\Theta_{yy}}$ and $\mathbf{\Theta_{xy}}$, in Eq. (6). While lasso and MRCE that regularize $\mathbf{B}$ and $\mathbf{\Psi}$ do not result in shared relevant inputs for multiple related outputs, we show that the inverse-covariance regularization in CGGM representation naturally leads to structured sparsity and extracts much richer and intuitively more appealing information on the sparse interactions between inputs and outputs as well as among outputs.

Our new approach minimizes the negative log-

likelihood for Eq. (6) or equivalently Eq. (7) with an L_1 regularization of the inverse-covariance parameters as follows:

$$\begin{aligned}\text{argmin} \quad & -L(\mathbf{X},\mathbf{Y};\mathbf{\Theta_{xy}},\mathbf{\Theta_{yy}})+\lambda_1||\mathbf{\Theta_{xy}}||_1+\lambda_2||\mathbf{\Theta_{yy}}||_1 \\ & \text{subject to } \mathbf{\Theta_{yy}} \succ 0, \end{aligned} \tag{9}$$

where λ_1 and λ_2 are the regularization parameters that control the amount of sparsity, and

$$\begin{aligned}L(\mathbf{X},\mathbf{Y};\mathbf{\Theta_{xy}},\mathbf{\Theta_{yy}}) = & -1/2\,\mathrm{tr}\big(\mathbf{Y}\mathbf{\Theta_{yy}}\mathbf{Y}^T\big) \\ & -\mathrm{tr}\big(\mathbf{X}\mathbf{\Theta_{xy}}\mathbf{Y}^T\big) - \sum_i \log Z(\mathbf{\Theta_{xy}},\mathbf{\Theta_{yy}},\mathbf{x}_i)\end{aligned}$$

is the log-likelihood based on Eq. (7), or equivalently from Eq. (6)

$$\begin{aligned}L(\mathbf{X},\mathbf{Y};\mathbf{\Theta_{xy}},\mathbf{\Theta_{yy}}) = -\frac{1}{2}\Big[& -N\log\det\mathbf{\Theta_{yy}} \\ & +\mathrm{tr}\big[(Y+X\mathbf{\Theta_{xy}}\mathbf{\Theta_{yy}}^{-1})\mathbf{\Theta_{yy}}(Y+X\mathbf{\Theta_{xy}}\mathbf{\Theta_{yy}}^{-1})^T\big]\Big].\end{aligned} \tag{10}$$

It is straightforward to show that the optimization problem above is convex. When $\lambda_1 >> \lambda_2$, the model effectively disregards the inputs by setting all of the elements of $\mathbf{\Theta_{xy}}$ to 0, and tries to explain the output variability across samples only through the other outputs connected to it with edges in $\mathbf{\Theta_{yy}}$. As illustrated in Figure 2(a), this is equivalent to learning a sparse Gaussian graphical model only for the outputs using the graphical lasso [6]. On the other hand, if $\lambda_1 << \lambda_2$, the model treats the outputs as independent of each other given the inputs by setting all of the off-diagonal entries in $\mathbf{\Theta_{yy}}$ to 0, and tries to explain the output variability only through the inputs. As illustrated in Figure 2(b), this is equivalent to lasso. The optimal values for λ_1 and λ_2 that strike the right balance between these two extreme cases can be found by cross-validation.

While the output structure is recovered in $\hat{\mathbf{\Theta}}_{\mathbf{yy}}$, the shared sparsity pattern among the related outputs with respect to $\hat{\mathbf{\Theta}}_{\mathbf{yy}}$ is recovered in $\hat{\mathbf{B}}$ after applying the transformation to the standard representation $\hat{\mathbf{B}} = -\hat{\mathbf{\Theta}}_{\mathbf{yy}}^{-1}\hat{\mathbf{\Theta}}_{\mathbf{xy}}^{T}$. If $\hat{\mathbf{\Theta}}_{\mathbf{yy}}$ is diagonal representing independent outputs, $\hat{\mathbf{B}}$ shows no shared relevant inputs across outputs. On the other hand, if the output structure in $\hat{\mathbf{\Theta}}_{\mathbf{yy}}$ contains multiple connected components, these connected components play the role of groups as in the group lasso [21], and the outputs in the same connected component have the same relevant inputs in $\hat{\mathbf{B}}$. Unlike in [8], the number of such groups does not need to be pre-defined by the user, but is discovered automatically.

In addition, our approach allows us to distinguish between direct and indirect influence of inputs on ouputs. More specifically, the inputs with non-zero entries in $\hat{\mathbf{\Theta}}_{\mathbf{xy}}$ can be seen as having direct influence on the corresponding outputs, whereas the inputs that are not relevant in $\hat{\mathbf{\Theta}}_{\mathbf{xy}}$ but become relevant only in $\hat{\mathbf{B}}$ can be viewed as having only indirect influence on the given output. As illustrated in Figures 2(c) and (d), in general, $\mathbf{\Theta_{xy}}$ corresponds to the direct influence on the output variables, which then propagate through the output network $\mathbf{\Theta_{yy}}$ to indirectly influence other output variables. Computing $\hat{\mathbf{B}}$ from $\hat{\mathbf{\Theta}}_{\mathbf{xy}}$ and $\hat{\mathbf{\Theta}}_{\mathbf{yy}}$ via $\hat{\mathbf{B}} = -\hat{\mathbf{\Theta}}_{\mathbf{yy}}^{-1}\hat{\mathbf{\Theta}}_{\mathbf{xy}}^{T}$ is equivalent to performing inference or marginalization on the given graphical model to infer these indirect influences of inputs on outputs. Thus, our approach learns both the sparsity in $\mathbf{\Theta_{xy}}$ in the conditional distribution and the structured sparsity in $\mathbf{B}$ in the marginal distribution.

The optimization problem for our method in Eqs. (9) and (10) allows for a direct comparison with MRCE in Eq. (3). Although both MRCE and our approach represent the output structure using the same parameter (i.e., $\mathbf{\Omega} = \mathbf{\Theta_{yy}}$), the output structure in MRCE does not play the role of propagating the influence of relevant inputs to other related outputs, and thus, does not learn a shared sparsity pattern for related outputs.

3.3 Optimization

The problem in Eq. (9) for our approach is a semidefinite program with a positive-definite constraint, and can be solved with an interior-point method [3]. However, it is well-known that the interior-point method is computationally slow and does not scale to a high-dimensional problem. In order to develop a more efficient method, we notice that because Eq. (10) contains $\log\det\mathbf{\Theta_{yy}}$ term, which acts as a log-barrier function for the positive-definite constraint, we do not need to consider the constraint explicitly. Taking advantage of this property of our problem, we further notice that efficient optimization methods for a general L_1-regularized log-linear model are directly applicable. In this paper, we adopt the OWL-QN [1] for learning an L_1-regularized log-linear model.

Motivated by the observation that the L_1 norm is smooth within any orthant, the OWL-QN iteratively optimizes the L_1-regularized negative log-likelihood by constructing a quadratic approximation of the objective and finding the minimum of the approximation within the orthant of the estimate from the previous iteration. As the OWL-QN is based on the L-BFGS limited-memory quasi-Newton method, it requires the computation of the gradient and Hessian in each iteration, where the Hessian is approximated using the gradients of the previous iterations instead of computing the full Hessian matrix. The gradients of the log-

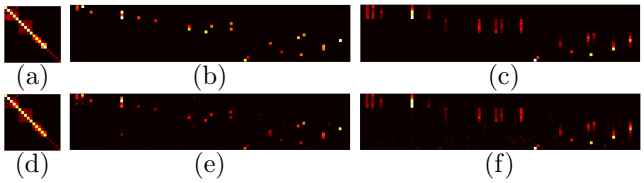

Figure 3: Illustration of our method using a single simulated dataset from Case 1. True parameters are shown in the top row and estimated parameters in the bottom row. (a) $\mathbf{\Theta_{yy}}$, (b) $\mathbf{\Theta_{xy}}$, (c) $\mathbf{B}$, (d) $\hat{\mathbf{\Theta}}_{\mathbf{yy}}$, (e) $\hat{\mathbf{\Theta}}_{\mathbf{xy}}$, (f) $\hat{\mathbf{B}}$.

likelihood of data given a CGGM are as follows:

$$\frac{\partial L(\mathbf{X},\mathbf{Y};\mathbf{\Theta_{xy}},\mathbf{\Theta_{yy}})}{\partial \mathbf{\Theta_{xy}}} = -\mathbf{X}^T\mathbf{Y} + \sum_i E\left[\mathbf{x}_i\mathbf{y}_i^T\right]$$
$$\frac{\partial L(\mathbf{X},\mathbf{Y};\mathbf{\Theta_{xy}},\mathbf{\Theta_{yy}})}{\partial \mathbf{\Theta_{yy}}} = -1/2\mathbf{Y}^T\mathbf{Y} + \sum_i E\left[\mathbf{y}_i\mathbf{y}_i^T\right],$$

where the expectations are taken with respect to $p(\mathbf{y}_i|\mathbf{x}_i,\mathbf{\Theta_{xy}},\mathbf{\Theta_{yy}})$ and can be computed as

$$E\left[\mathbf{x}_i\mathbf{y}_i^T\right] = -\mathbf{x}_i\mathbf{x}_i^T\mathbf{\Theta_{xy}}\mathbf{\Theta_{yy}}^{-1} \quad (11)$$
$$E\left[\mathbf{y}_i\mathbf{y}_i^T\right] = \mathbf{\Theta_{yy}}^{-1} + \mathbf{\Theta_{yy}}^{-1}\mathbf{\Theta_{xy}}^T\mathbf{x}_i\mathbf{x}_i^T\mathbf{\Theta_{xy}}\mathbf{\Theta_{yy}}^{-1}. \quad (12)$$

Each iteration in the OWL-QN involves a line search to obtain the optimal estimate along the line given the gradient and Hessian at the current estimate. In order to ensure the intermediate search point to be positive definite, we make use of the fact that $A - tB$ is positive definite if A is positive-definite, B is symmetric, and t is sufficiently small. That is, we search for sufficiently small $\alpha \in (0,\infty)$ so that the determinant of the next estimate $\mathbf{\Theta}_{\mathbf{yy}}^{k+1}$ along the ray $\mathbf{\Theta}_{\mathbf{yy}}^{k} - \alpha D_{yy}^k$ given the search direction D_{yy}^k is again positive.

4 Experiments

We compare the performance of our method to those of lasso, MRCE, GFlasso, multi-task lasso with an L_1/L_2 regularization for union support recovery, using simulated datasets and two real datasets from genetics and finance applications. For MRCE, we found the alternate optimization that solves a sequence of convex problems given either $\mathbf{B}$ or $\mathbf{\Omega}$ often did not converge, and thus, used the approximate method, as was also suggested by Rothman et al. [16]. In addition, the alternate optimization for MRCE was too slow even for a moderate size of data (taking one full day for a single simulated dataset of size $J = 1000$ and $K = 50$ with cross-validation, compared to a few minutes for other methods) to be applied to a large-scale simulation study. The regularization parameters in each method was determined by five-fold cross validation.

4.1 Simulation Study

We evaluate our proposed method and other methods using simulated data with known parameters. We use two different simulation strategies, based on 1) the standard parameterization with $\mathbf{B}$ and $\mathbf{\Psi}$ in Eq. (1) and 2) the alternative parameterization with $\mathbf{\Theta_{yy}}$ and $\mathbf{\Theta_{xy}}$ in Eq. (6). For each individual i, the input vector $\mathbf{x}_i$ is generated by setting each element of $\mathbf{x}_i$ to a random draw from $N(0,1)$. Given the true parameters $\mathbf{\Theta_{yy}}$ and $\mathbf{\Theta_{xy}}$ (or, $\mathbf{B}$ and $\mathbf{\Psi}$), the output data $\mathbf{y}_i$ are sampled from Eq. (6) (or, from Eq. (1), respectively).

Simulation with $\mathbf{\Theta_{yy}}$ and $\mathbf{\Theta_{xy}}$ We set $\mathbf{\Theta_{yy}}$ and $\mathbf{\Theta_{xy}}$, using different output structures such as independent, tree, graph, and chain, as follows:

- Case 1 ($J = 100$, $K = 20$): Assuming four groups of outputs with each group containing five output variables, we generate the output network structure such that within each of the four groups, the output structure is a complete graph with weak edge connections, a complete graph with strong edge connections, a chain structure, and independent outputs, respectively. Edge weights of this network are drawn from a uniform distribution $[0, 0.5]$ for the first group, and from a uniform distribution $[0.4, 1.0]$ for all the other edges. We set $\mathbf{\Theta_{yy}}$ to the graph Laplacian of this network and add a small positive real value to the diagonal elements to ensure $\mathbf{\Theta_{yy}}$ to be positive definite. We generate $\mathbf{\Theta_{xy}}$ by randomly selecting a single input relevant to each output, and additional zero to three inputs relevant to a subset of randomly selected m outputs within each group, where m is drawn from a uniform distribution $[M/2, M]$.
- Case 2 ($J = 1000$ and $K = 50$): We assume five groups of outputs, each group with 10 outputs. For four of the five groups, we set edge connections/weights using the same strategy as in Case 1. For the additional output group, we assume a binary tree structure with edge weights drawn randomly from a uniform distribution $[0.4, 1.0]$. Then, we set $\mathbf{\Theta_{yy}}$ to the graph Laplacian of this network. We use the same strategy as in Case 1 to set $\mathbf{\Theta_{xy}}$.

Simulation with $\mathbf{B}$ and $\mathbf{\Psi}$ We simulate datasets with known $\mathbf{B}$ and $\mathbf{\Psi}$, using the following scenarios.

- Case 3 ($J = 500$, $K = 50$): We assume a tree structure over outputs, and set $\mathbf{\Psi}$ to the graph Laplacian of a tree with a branching factor of four and edge weights randomly drawn from $[0.4, 1.0]$. We set $\mathbf{B}$ by selecting a common relevant input for each of the internal node and its four child nodes in the tree.
- Case 4 ($J = 500$, $K = 50$): We also consider the traditional scenario that assumes independent noise for different outputs with a diagonal matrix for $\mathbf{\Psi}$. We randomly select two relevant inputs for each output.

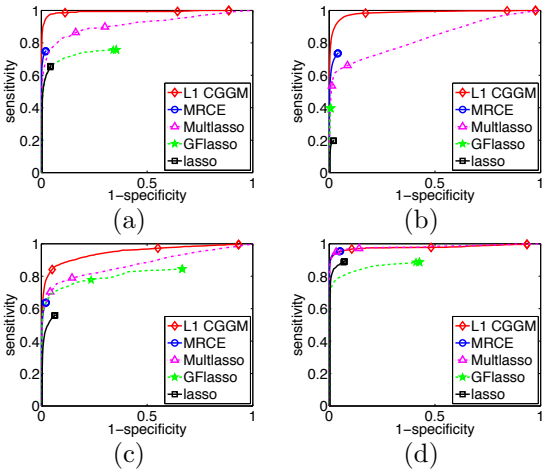

Figure 4: ROC curves for the recovery of the true non-zero regression coefficients in simulation study. The results are shown for Cases 1 – 4 in (a) – (d).

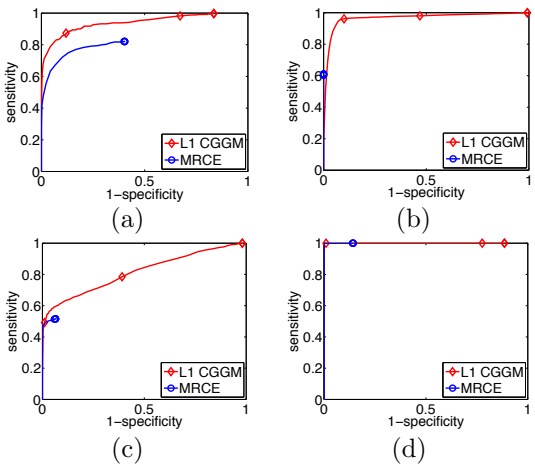

Figure 5: ROC curves for the recovery of the true output structure from simulated data. The results from Cases 1 – 4 are shown in (a) – (d). In (b), the ROC curve for MRCE stops after reaching sensitivity 0.6, implying many false negatives.

Results To illustrate the behavior of our method, in Figure 3, we show the results from a single dataset simulated from Case 1. Figures 3(a)-(c) show the true $\mathbf{\Theta_{yy}}$, $\mathbf{\Theta_{xy}}$, and $\mathbf{B} = -\mathbf{\Theta_{yy}^{-1}\Theta_{xy}^{T}}$, respectively, and Figures 3(d)-(f) present the estimates from a dataset of 800 samples. As shown in Figure 3(b), the non-zero entries in $\mathbf{\Theta_{xy}}$ include inputs relevant to individual outputs as well as inputs affecting multiple outputs. However, as shown in Figure 3(c), the inputs relevant only to individual outputs in $\mathbf{\Theta_{xy}}$ become relevant in $\mathbf{B}$ to multiple related outputs within the corresponding group of outputs in $\mathbf{\Theta_{yy}}$.

In order to quantitatively compare the performance of different methods, we generate 30 simulated datasets of 400 samples from each of Cases 1 – 4. We first evaluate different methods on the accuracy for the recovery of true relevant inputs in $\hat{\mathbf{B}}$ averaged over 30 simulated datasets and show the results as receiver operating characteristic (ROC) curves in Figure 4. For all scenarios, our method significantly outperforms all the other previous methods. The results show that taking into account the output structure while estimating regression coefficients improves the sensitivity and specificity for recovering sparse structure.

Unlike other regression methods, our method and MRCE have the ability to estimate the output structure in addition to the regression coefficients. We compare the performance of the two methods on the recovery of the true output structure as reflected in $\mathbf{\Theta_{yy}}$ for CGGM and $\mathbf{\Omega}$ (which is equivalent to $\mathbf{\Theta_{yy}}$) for MRCE. As can be seen in Figure 5, our method significantly outperforms MRCE.

We also compare the performance of the methods on prediction accuracy by generating additional 200 test samples for each simulated dataset and computing prediction errors on these test sets. As shown in Figure 6, our method achieves lower prediction errors than any other methods. Even when the outputs are independent with no shared sparsity (Case 4, Figure 6(b)), the performance of our algorithm is still comparable to or better than all the other methods because the independent output structure can be properly reflected as a diagonal inverse covariance matrix in $\mathbf{\Theta_{yy}}$.

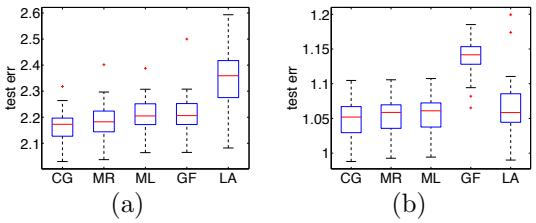

Figure 6: Prediction errors in simulation study. (a) Case 2, (b) Case 4. (CG: sparse CGGM, MR: MRCE, ML: multi-task lasso, GF: GFlasso, LA: lasso)

Table 1: Prediction errors (Human Liver Cohort data)

Sparse CGGM	GFlasso	MultLasso	lasso
2.359	2.369	2.621	4.212

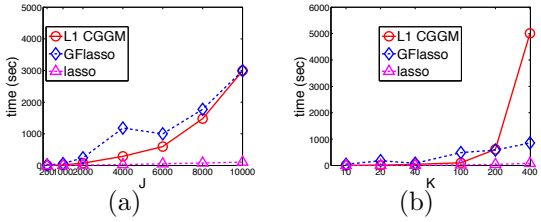

Figure 7: Computation time for different methods. (a) Varying the input size J ($K = 20$) and (b) varying the output size K ($J = 1000$).

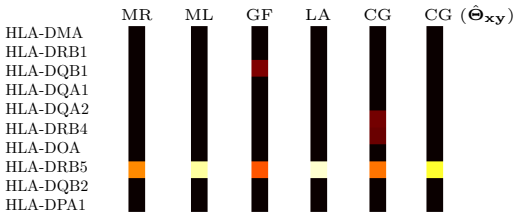

Figure 8: Estimated association strengths of SNP rs9271366 with HLA Class II gene expressions.

Computation time We compare the computation time of lasso, GFlasso, and sparse CGGM for varying input and output sizes in Figures 7(a) and (b), respectively. We used the lasso implementation available in the R package glmnet, and used a Matlab implementation of GFlasso with proximal gradient method for optimization [4]. We do not include the results for the approximate MRCE, since it was slower than all of the other methods by orders of magnitude. Our method scales reasonably well as J and K increase, although for large K the main bottleneck is the inversion of $\mathbf{\Theta_{yy}}$ for computing the expectation in Eq. (12).

4.2 Real Dataset from eQTL Mapping

We apply our and other methods to the gene expression traits (outputs) and single nucleotide polymorphism (SNP) data (inputs) of 178 samples from the Human Liver Cohort study [17] to discover SNPs that influence the expression levels of genes. This type of study is widely known as an expression quantitative trait locus (eQTL) mapping in the genetics community. It is generally believed that when a genetic variation in the genome such as a SNP perturbs the expression of a gene, the effect propagates through the gene network to influence the expressions of genes in the downstream of the pathway. In this case, the casual or relevant SNP affects the expression of the target gene directly, and affects the downstream genes indirectly. Thus, using our new method, we expect not only to learn the gene network and relevant SNPs for gene expressions but also to distinguish between direct and indirect influences of SNPs on genes that are connected in a gene network.

The input dataset consists of $J = 937$ SNPs on chromosome 6 that have minor allele frequency greater than 0.05 and whose pair-wise correlations are less than 0.1. The output dataset includes $K = 100$ gene expression traits that have variance greater than 0.05. We estimate the regression coefficients using 143 samples and then compute the prediction error on the remaining 35 samples. As shown in Table 1, our sparse CGGM produces the smallest prediction error.

We closely examine a small number of genes in HLA class II for associations with SNPs on chromosome 6, motivated by the findings in previous studies [19]. We apply each method to the subset of outputs consisting of only 11 HLA gene expressions that are known to be highly correlated. Overall, all of the methods found the strongest association between SNP rs9270986 and the expression trait for HLA-DRB5 gene. The estimated regression coefficients between this SNP and all genes in HLA class II are shown in Figure 8. While some methods find associations between the SNP rs9270986 and expression levels of genes other than the HLA-DRB5 gene, our method identifies only the HLA-DRB5 gene as being directly perturbed by the SNP and suggests indirect perturbations for other genes in HLA Class II by the same SNP.

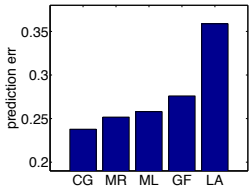

Figure 9: Prediction errors for S&P stock price data.

4.3 Real Dataset from S&P 500 stock prices

We apply each method to the daily stock price data of the S&P 500 companies in 2005 to learn the model that can predict the stock prices in the future by using the stock prices in the past as inputs. We use a first-order auto-regressive model $\mathbf{y}_t = \mathbf{B}\mathbf{y}_{t-1} + \boldsymbol{\epsilon}_t$, where $\mathbf{y}_t$ represents the stock prices of the companies at time t. We select 128 companies with a moderate level of variance over time ($15 < \sigma^2 < 80$). Each model is trained using the data from the first 50 days, and tested on the data from the next 50 days.

We show the prediction accuracies for different methods in Figure 9. Overall, our method outperforms all the other methods. Given the highly structured nature of the stock data with correlated stock prices for many companies (especially within the same sector), our results show that taking into account the output structure and structured sparsity as in our approach can increase the prediction accuracy.

5 Conclusions

In this paper, we proposed a new approach for a spare estimation of multiple-output regression that can estimate both the output structure and regression coefficients with structured sparsity at the same time. The future work includes exploring the use of an alternative optimization method such as a fast iterative shrinkage thresholding algorithm [2].

Acknowledgements

SYK is supported by Okawa Foundation Research Grant.

References

[1] G. Andrew and J. Gao. Scalable training of l_1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.

[2] A. Beck and M. Teboulle. A fast iterative shrinkage thresholding algorithm for linear inverse problems. *SIAM Journal of Image Science*, 2(1):183–202, 2009.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[4] X. Chen, Q. Lin, S. Kim, J. Carbonell, and E.P. Xing. Smoothing proximal gradient method for general structured sparse learning. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 105–114. AUAI Press, 2011.

[5] J. Friedman, T. Hastie, and R. Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

[6] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–41, 2008.

[7] R. Jenatton, J. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. Technical report, INRIA, 2009.

[8] Z. Kang, K. Grauman, and F. Sha. Learning with whom to share in multi-task feature learning. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

[9] S. Kim and E.P. Xing. Statistical estimation of correlated genome associations to a quantitative trait network. *PLoS Genetics*, 5(8):e1000587, 2009.

[10] S. Kim and E.P. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *Proceedings of the 27th International Conference on Machine Learning*, pages 543–550. Omnipress, 2010.

[11] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[12] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, 2001.

[13] K. Mardia, J. Kent, and J Bibby. *Multivariate Analysis*. Academic Press, 1979.

[14] N. Meinshausen and P. Buhlmann. High dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34:1436–1462, 2006.

[15] G. Obozinski, M.J. Wainwright, and M.J. Jordan. High-dimensional union support recovery in multivariate regression. In *Advances in Neural Information Processing Systems 21*, 2008.

[16] A. Rothman, E. Levina, and J. Zhu. Sparse multivariate regression with covariance estimation. *Journal of Computational and Graphical Statistics*, 19(4):947–962, 2010.

[17] E. Schadt, C. Molony, E. Chudin, K. Hao, and X. et al Yang. Mapping the genetic architecture of gene expression in human liver. *PLoS Biology*, 6(5):e107, 2008.

[18] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society, Series B*, 58(1):267–288, 1996.

[19] Wellcome Trust Case Control Consortium. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447:661–78, 2007.

[20] D. Witten and R. Tibshirani. Covariance-regularized regression and classification for high-dimensional problems. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 71(3):615–636, 2009.

[21] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of Royal Statistical Society, Series B*, 68(1):49–67, 2006.

[22] P. Zhao, G. Rocha, and B. Yu. Grouped and hierarchical model selection through composite absolute penalties. Technical Report 703, Department of Statistics, University of California, Berkeley, 2008.