# Structured Sparsity with Group-Graph Regularization

**Xin-Yu Dai, Jian-Bing Zhang, Shu-Jian Huang, Jia-Jun Chen, Zhi-Hua Zhou**
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China
{daixinyu,zjb,huangsj,chenjj,zhouzh}@nju.edu.cn

## Abstract

In many learning tasks with structural properties, structural sparsity methods help induce sparse models, usually leading to better interpretability and higher generalization performance. One popular approach is to use group sparsity regularization that enforces sparsity on the clustered groups of features, while another popular approach is to adopt graph sparsity regularization that considers sparsity on the link structure of graph embedded features. Both the group and graph structural properties co-exist in many applications. However, group sparsity and graph sparsity have not been considered simultaneously yet. In this paper, we propose a $g^2$-regularization that takes group and graph sparsity into joint consideration, and present an effective approach for its optimization. Experiments on both synthetic and real data show that, enforcing group-graph sparsity lead to better performance than using group sparsity or graph sparsity only.

## Introduction

It is well known that sparse models can be better than non-sparse models in many scenarios, and many regularizers have been developed to enforce sparsity constraints (Tibshirani 1994; Yuan and Lin 2006; Huang, Zhang, and Metaxas 2011; Chen et al. 2013). Formally, given the optimization objective:

$$\min_{\mathbf{w}\in\mathbb{R}^p} L(\mathbf{w}) + \lambda\Omega(\mathbf{w}) \tag{1}$$

where $L(\mathbf{w})$ is the loss function, $\Omega(\mathbf{w})$ is the regularizer, $\mathbf{w}$ and $\lambda$ are the model parameter and control parameter, respectively. One can adopt $\ell_1$-norm to realize the regularizer such that only a few features will be used in the learned model. Such an approach, known as *Lasso* (Tibshirani 1994), has been applied to many tasks as a demonstration on how sparse models can achieve better generalization performance than non-sparse models; moreover, with fewer features used, the sparse models are usually with reasonable interpretability. Thus, the exploration of effective sparsity regularization methods has become a hot topic during the past few years.

In many application areas, such as computer vision, bioinformatics and natural language processing, there are usually inherent structural properties in the data. Adequately exploiting such structural information may lead to better performance, and thus, great efforts have been made to developing structured sparsity regularization methods.

One popular approach, known as group sparsity (Yuan and Lin 2006), is to consider the feature clustering structures. By clustering features into groups and then enforcing sparsity at the group level, improved learning performances have been observed in many tasks (Yuan and Lin 2006; Bach 2008; Guo and Xue 2013). Another popular approach, known as graph sparsity (Jacob, Obozinski, and Vert 2009; Huang, Zhang, and Metaxas 2011; Mairal and Yu 2013), is to consider the link structure of graph embedded features. By embedding the features into a graph and enforcing sparsity in the connectivity, better performances may be obtained with a subgraph containing a small number of connected features.

It is noteworthy that in previous studies either group sparsity regularization or graph sparsity regularization was used. However, in many scenarios both group and graph structure properties exist simultaneously in an essential way. Thus considering only group sparsity or graph sparsity may lead to loss of structured information. For example, in the task of splice site prediction, the genes at different positions naturally form many groups, among where there are some dependencies. The importances of the groups vary, depending on their positions and dependences to each other groups. Thus, the graph sparsity can be enforced by the group structure.

In this paper, we propose the $g^2$-regularization method which enforces group-graph sparsity to make use of the advantages of both the group and graph structures. The combination of group sparsity and graph sparsity enforcement is non-trivial because the groups of features are embedded into the graph. The enforcement of group-graph sparsity with the $g^2$-regularization leads to a solution with a subgraph containing a small number of connected groups. We present an effective approach which performs the optimization with the $g^2$-regularizer using minimum cost network flow and proximal gradient. Experiments on both synthetic and real data show that, the $g^2$-regularization leads to better performance than only group sparsity or graph sparsity. In addition, considering both the group and graph structures together is more efficient than considering the graph structure only, because the group structure will help reduce the graph structure to a smaller scale.

In the following we start by presenting some preliminaries. Then we present the $g^2$-regularization method, followed by experiments and concluding remarks.

# Preliminaries

## Group Sparsity

In group sparsity, we denote by $I = \{1, 2, ..., p\}$ as the index set of the model parameter $\mathbf{w} \in \mathbb{R}^p$ in Eq.(1), then partition it into groups. Each of the groups is denoted as $A_i$. So we have a group structure $\pi = \{A_1, ..., A_q\}$, where (i). $I = \bigcup_{A \in \pi} A$, (ii).$A_i \neq \emptyset, \forall A_i \in \pi$, (iii).$A_i \cap A_j = \emptyset, \forall A_i, A_j \in \pi, i \neq j$. Group regularization is defined as follows:

$$\Omega_{Group}(\mathbf{w}) = \sum_{j=1}^{q} d_j \left\| \mathbf{w}_{A_j} \right\|_2 \qquad (2)$$

where $\mathbf{w}_{A_j} = \langle \mathbf{w}_i \rangle_{i \in A_j}$ is a sub-vector of the features in the $j_{th}$ group, and $d_j$ is a nonnegative scalar for the group $A_j$.

The group sparsity is firstly proposed by (Yuan and Lin 2006) as group lasso. The features are clustered into pre-specified groups. Group lasso enforces sparsity at the group level with $\ell_1$-norm regularization, so that features in one group will either all be selected or all be discarded. Many various methods (Roth and Fischer 2008; Meier, van de Geer, and Buhlmann 2008; Liu, Ji, and Ye 2009; Kowalski, Szafranski, and Ralaivola 2009) have been proposed for optimization with the group sparsity regularization.

## Graph Sparsity

In graph sparsity, we have a directed acyclic graph (DAG) $G = (V, E)$ on the index set $I = \{1, 2, ..., p\}$ of the model parameter $\mathbf{w} \in \mathbb{R}^p$, where $V = I$ is the vertex set and $E = \{(i, j) \,|\, i, j \in V\}$ is the edge set. Let $g$ be a path in the graph of $G$, denoted as $g = (v_1, v_2, ..., v_k)$, where $v_i \in V, i = 1, ..., k$ and $(v_i, v_{i+1}) \in E, i = 1, ..., k-1$. Let $\mathcal{G}$ be the set of all paths in the graph $G$. We denote the positive weight of each path $g \in \mathcal{G}$ by $\eta_g$ ($\eta_g > 0$). The graph sparsity regularization is defined as follows (Huang, Zhang, and Metaxas 2011):

$$\Omega_{Graph}(\mathbf{w}) = \min_{\mathcal{J} \subseteq \mathcal{G}} \left\{ \sum_{g \in \mathcal{J}} \eta_g \ \text{ s.t. } \ \mathrm{Supp}(\mathbf{w}) \subseteq \bigcup_{g \in \mathcal{J}} g \right\} \qquad (3)$$

where Supp$(\cdot)$ stands for the nonzero index set of a vector. $\mathcal{J}$ is a subset of $\mathcal{G}$ whose union covers the support of $w$.

The graph sparsity enforces us to select a subgraph $\mathcal{J}$ containing a small number of connected features (That is to cover Supp($\mathbf{w}$)).

Optimization with the graph sparsity regularization is not exactly tractable when the graph scale is too large. Some approximate optimization methods (Jacob, Obozinski, and Vert 2009; Huang, Zhang, and Metaxas 2011; Mairal and Yu 2013) have been proposed for graph sparsity.

# $g^2$-regularization

We now propose the $g^2$-regularization which enforces group-graph sparsity to make use of the group and graph structures simultaneously. The group structure is embeded into a graph structure. $g^2$-regularization leads to a solution with a subgraph containing a small number of connected groups. In this section, we firstly give the formulation of our $g^2$-regularization. The optimization method with $g^2$-regularization is then presented.

## Formulation

We denote by $I = \{1, 2, ..., p\}$ the index set of the model parameter $\mathbf{w} \in \mathbb{R}^p$, and let $\pi = \{A_1, ..., A_q\}$ the group structure on $I$. Given a DAG $G = (V, E)$ on the group index $J = \{1, 2, ..., q\}$, where $V = J$ is the vertex set and $E = \{(i, j) \,|\, i, j \in V\}$ is the edge set. $g = (v_1, v_2, ..., v_k)$ is a path in the graph of $G$, where $v_i \in V, i = 1, ..., k$ and $(v_i, v_{i+1}) \in E, i = 1, ..., k-1$. Let $\mathcal{G}$ denote the set of all paths in graph $G$. $\eta_g > 0$ is non-negative weight of the path $g \in \mathcal{G}$. The formulation of $g^2$-regularization is as follows:

$$\Omega_{g^2}(\mathbf{w}) \triangleq \min_{\mathcal{J} \subseteq \mathcal{G}} \left\{ \sum_{g \in \mathcal{J}} \eta_g \ \text{ s.t. } \ \mathrm{Supp}(\sigma(\mathbf{w})) \subseteq \bigcup_{g \in \mathcal{J}} g \right\} \qquad (4)$$

where Supp$(\cdot)$ stands for the nonzero index set of a vector. $\mathcal{J}$ is a subset of $\mathcal{G}$ whose union covers the support of $\sigma(\mathbf{w})$. $\sigma(\cdot)$ is a group function $\sigma : \mathbb{R}^p \to \mathbb{R}^q$ which can be defined as follows:

$$\sigma(\mathbf{w}) = \left( \left\| \mathbf{w}_{A_1} \right\|, \left\| \mathbf{w}_{A_2} \right\|, \cdots, \left\| \mathbf{w}_{A_q} \right\| \right) \qquad (5)$$

where $\mathbf{w}_{A_j} = \langle \mathbf{w}_i \rangle_{i \in A_j}$ and $\|\cdot\|$ stands for the $\ell_2$-norm.

The above formulation is an extension of the graph sparsity formulation. The features are firstly clustered into groups, then a graph structure is constructed with the connectivity between groups. Our $g^2$-regularization is applied to select a subgraph with connected groups as supports.

Note that if the edge set $E$ of $G$ is an empty set, $g^2$-regularization will degenerate into group sparsity. If the function is defined as $\sigma : \mathbf{w} \to (|\mathbf{w}_1|, ..., |\mathbf{w}_p|)$, $g^2$-regularization degenerates into graph sparsity. From this point of view, our $g^2$-regularization can be viewed as a natural combination of group sparsity and graph sparsity with their two kinds of structures.

In addition, the group lasso can be rewritten as follows:

$$\Omega_{Group}(\mathbf{w}) = \left\| \sigma(\mathbf{w}) \right\|_1 \qquad (6)$$

Group lasso uses $\ell_1$-norm to select groups. However, we use graph sparsity method to induce sparsity over groups which forms the $g^2$-regularization.

## Optimization with $g^2$-regularizer

The $g^2$-regularizer (defined in Eq.(4)) can be rewritten as a boolean linear program as follows:

$$\Omega_{Conv}(\mathbf{w}) \triangleq \min_{\mathbf{x} \in \{0,1\}^{|\mathcal{G}|}} \left\{ \eta^{\mathrm{T}} \mathbf{x} \ \text{ s.t. } \ \mathbf{N}\mathbf{x} \geq \mathrm{Supp}(\sigma(\mathbf{w})) \right\} \qquad (7)$$

**Algorithm 1** The $g^2$-regularization method

```
Input: Loss function $L(\mathbf{w})$, group structure $\pi$, graph structure DAG $G'$, initial $\mathbf{w}_0$, parameter $\lambda$, maximum iterations $M$, threshold $\tau$
Output: $\mathbf{w}_k$
1: Compute $\rho$, a Lipschitz constant of $\nabla L(\mathbf{w})$
2: Take $\mathbf{y}_1 = \mathbf{w}_0$, $t_1 = 1$, $k = 1$
repeat
    3: $\mathbf{a}_k \leftarrow \mathbf{y}_k - \frac{1}{\rho} \cdot \nabla L(\mathbf{y}_k)$
    4: $f^* \leftarrow \arg\min_{f \in \mathcal{F}} \frac{\lambda}{\rho} \sum_{(u,v) \in E'} f_{uv} c_{uv} + \sum_{j=1}^{q} \frac{1}{2} \max\left( (\sigma(\mathbf{a}_k))_j - s_j(f), 0 \right)^2$
    5: $(\mathbf{w}_k)_i \leftarrow \min\left( (\mathbf{a}_k)_i, \frac{s_j(f^*)}{(\sigma(\mathbf{a}_k))_j} (\mathbf{a}_k)_i \right), i \in A_j, j \in \{1, 2, ..., q\}$
    6: $t_{k+1} \leftarrow \frac{1+\sqrt{1+4t_k^2}}{2}$
    7: $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + \left( \frac{t_k - 1}{t_{k+1}} \right) (\mathbf{w}_k - \mathbf{w}_{k-1})$
    8: $k \leftarrow k + 1$
until $|\mathbf{w}_{k-1} - \mathbf{w}_{k-2}| < \tau$ or $M$ iterations finish.
```

where $\eta([\eta_g]_{g \in \mathcal{G}})$ is the vector in $\mathbb{R}_+^{|\mathcal{G}|}$, and $\mathbf{N}$ is a binary matrix in $\{0,1\}^{q \times |\mathcal{G}|}$ which indicates if each vertex is in each path of $\mathcal{G}$ or not. Supp($\sigma(\mathbf{w})$) is a vector in $\{0,1\}^q$ such that its $j_{th}$ entry is one if $j$ is in the support of $\sigma(\mathbf{w})$ or zero if not.

Since the $g^2$-regularizer defined in the Eq.(7) is non-convex, we can now formulate a convex relaxation of $\Omega_{Conv}(\mathbf{w})$:

$$\Omega_{Conv}(\mathbf{w}) \triangleq \min_{\mathbf{x} \in \mathbb{R}_+^{|\mathcal{G}|}} \left\{ \eta^{\mathrm{T}} \mathbf{x} \text{ s.t. } \mathbf{N}\mathbf{x} \geq |\sigma(\mathbf{w})| \right\} \quad (8)$$

Such a relaxation is classical and corresponds to the same mechanism relating the $\ell_0$ to the $\ell_1$-penalty. $|\sigma(\mathbf{w})|$ is the vector in $\mathbb{R}_+^{|\mathcal{G}|}$ obtained by replacing the entries of $\sigma(\mathbf{w})$ by their absolute values.

In Algorithm 1, we present an effective optimization approach with the $g2$-regularizer in Eq.(8) using minimum cost network flow and proximal gradient.

The weight of a path is the sum of the costs of all edges on this path. Denote $s$ and $t$ as two additional nodes which stand for the source and sink node on graph $G$. Define $G' = (V', E')$ as $V' = V \cup \{s,t\}$ and $E' = E \cup \{(s,u) \,|u \in V\} \cup \{(u,t) \,|u \in V\}$. We denote by $c_{uv}$ the cost of the edge $uv \in E'$ where $uv$ stands for $(u,v)$ for short. For a path $g = (v_1, v_2, \cdots, v_k)$, the weight $\eta_g$ is computed as follows:

$$\eta_g = c_{sv_1} + \sum\nolimits_{i=1}^{k-1} c_{v_i v_{i+1}} + c_{v_k t} \quad (9)$$

The loss function $L$ is convex and differentiable with a Lipschitz continuous gradient. Here $\rho > 0$ is a parameter which is an upper bound on the Lipschitz constant of $\nabla L$. $(\cdot)_i$ stands for the $i$th entry of a vector. $f$ stands for a network flow on the graph $G'$. $\mathcal{F}$ stands for the set of network flow $f$ on $G'$. $s_j(f) \triangleq \sum_{u \in V':(u,j) \in E'} f_{uj}$ stands for the amount of flow going through a vertex $j$.

We apply the proximal gradient method for optimization under the FISTA (Beck and Teboulle 2009) algorithm implementation. Computing the proximal operator in step 4 and 5 is equivalent to computing network flow inspired by (Mairal and Yu 2013). Theorem 1 shows the equivalence of proximal operator of $\Omega_{Conv}$ in Eq.(8) and network flow problem computed in step 4 and step 5. (The proof is attached in supplemental material.)

**Theorem 1.** *For $\mathbf{b} \in \mathbb{R}^p$, here, the group function $\sigma$, and graph $G'$ with costs $[c_{uv}]_{uv \in E'}$ are defined above. The proximal operator of $\Omega_{Conv}(\mathbf{w})$:*

$$\mathrm{Prox}_{\Omega_{Conv}}(\mathbf{b}) = arg \min_{\mathbf{w} \in \mathbb{R}^p} \frac{1}{2} \|\mathbf{b} - \mathbf{w}\|^2 + \Omega_{Conv}(\mathbf{w}) \quad (10)$$

*is equivalent to compute:*

$$\mathbf{w}_i^* = \min\left( \mathbf{b}_i, \frac{s_j(f^*)}{(\sigma(\mathbf{b}))_j} \mathbf{b}_i \right), i \in A_j, \forall j \in \{1, ..., q\} \quad (11)$$

*where*

$$f^* \in \arg\min_{f \in \mathcal{F}} \left\{ \sum_{(u,v) \in E'} f_{uv} c_{uv} + \sum_{j=1}^{q} \frac{1}{2} \max\left( (\sigma(\mathbf{b}))_j - s_j(f), 0 \right)^2 \right\} \quad (12)$$

It is easy to see that Eq.(8) is a norm. Hence, according to previous studies (Nesterov 2007; Beck and Teboulle 2009), the solution will converge to the solution of Eq.(1) when we use the proximal gradient method with a convex regularization. The complexity of solving Eq.(12) with an $\varepsilon$-accurate solution is a time polynomial in $|E|$, $|V|$, $\log(\|\sigma(b)\|_\infty / \varepsilon)$. In our method, the groups are viewed as nodes, $|V|$ is consequently reduced to a small scale. As the number of node decreases, the number of edge decreases accordingly. This shows that $|E|$ and $|V|$ become small. So, as for optimization, it is more efficient with the $g^2$-regularization since the graph structure can be reduced to a reasonable scale with the help of group structure.

## Error Bound Analysis

In this subsection, we discuss the estimation error bound of our $g^2$-regularization to illustrate the benefits of constructing a graph structure on groups.

According to previous study (Huang and Zhang 2010) , the parameter estimation error bound of group lasso is defined as follows:

$$\|\mathbf{w}_{est} - \bar{\mathbf{w}}\| \leq \frac{\sqrt{4.5}}{\rho_-(s)\sqrt{n}} \left(1 + 0.25\tau^{-1}\right) \sqrt{\kappa D^2 + \alpha B^2} \quad (13)$$

and

$$\tau^2 \leq \frac{lD^2 + \alpha_l B^2}{72\left(\kappa D^2 + \alpha B^2\right)}$$

where $\bar{\mathbf{w}}$ is a $(\alpha, \kappa)$ strongly group-sparse ground-truth. $\mathbf{w}_{set}$ is the estimated parameter. $D$, $B$ are two constant scalars depending on the data. $s$ and $l$ are two constant scalars depending on $\kappa$. $n$ is the number of samples. $\rho_-\left(s\right)$ is the group sparse eigenvalue, also defined in (Huang and Zhang 2010). $\alpha_l$ stands for the minimum group number where the size of the union of these groups is larger than $l$.

In $g^2$-regularizer, we have a graph structure (DAG) on groups, this can be seen as a kind of overlap group lasso (each overlapped group can be seen as a union of some groups), which has a very small $\alpha$ (Jenatton, Audibert, and Bach 2011). Thus, we have a small $\tau^{-1}$. Besides, after the construction of the graph structure on the groups, $\rho_-\left(s\right)$ will remain the same. These two aspects show that we can get a smaller bound from the graph structure.

Moreover, Huang, Zhang, and Metaxas( 2011) uses the coding complexity to interpret the structured sparsity. The parameter estimation error bound depends on the coding length. The bound grows with the increasing of coding length. From this theory, if the regularizer is defined as the graph sparsity (Eq.(3)), for a subset $K \subseteq I = \{1, 2, ..., p\}$, the suggested coding length $cl\left(K\right)$ is computed as follows

$$cl\left(K\right) = C\left|K\right| + \theta \log_2 p \tag{14}$$

where $C$ is a constant. $\theta$ stands for the number of connected parts of $K$ on a graph, i.e. $K = \bigcup_{i=1}^{\theta} K_i$ and each $K_i$ is connected.

In our $g^2$-regularization method, nodes on the graph are composed of groups. So, for $K \subseteq I = \{1, 2, ..., p\}$, the coding length $cl\left(K\right)$ of $g^2$-regularization can be computed as follows:

$$cl\left(K\right) = \left\{ \begin{array}{l} C'\left|Q\right| + \theta' \log_2 q \\ \infty, K \text{ is not a union of groups} \end{array} \right. \tag{15}$$

where $C'$ is a constant. $Q$ is a subset of $J = \{1, 2, ..., q\}$. Each element in $Q$ stands for one node (i.e. one group) in the graph $G$. $\left|Q\right|$ stands for the number of the groups in $K$. $\theta'$ is the number of connected part of $Q$ on the graph, i.e. $Q = \bigcup_{i=1}^{\theta'} Q_i$ and $Q_i$ is connected. Since $q$ is smaller than $p$, this coding length is smaller than that of graph sparsity defined in Eq. (14).

Above analysis shows that we can obtain smaller estimation error bound with $g^2$-regularization than considering group or graph sparsity only.

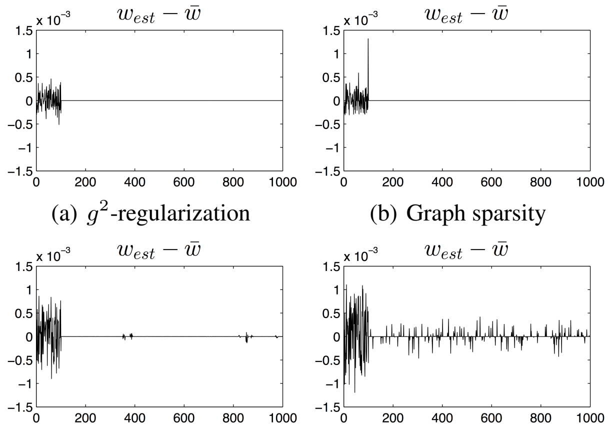
# Experiments

We use synthetic and real data to evaluate the $g^2$-regularizer. A modified open-source software named SPAMS from http://spams-devel.gforge.inria.fr/ is used to implement our algorithm.

## Synthetic Data

Table 1: Relative difference $(r)$ of $g^2$-regularization and other sparsity regularizations.

| | $g^2$-regularization | Graph sparsity | Group Lasso | Lasso |
|---|---|---|---|---|
| r $(\times 10^{-4})$ | **1.8** | 2.2 | 4.0 | 5.9 |



(a) $g^2$-regularization (b) Graph sparsity

(c) Group lasso (d) Lasso

Figure 1: Plot $w_{est} - \bar{w}$ to analysis the parameter estimation of different methods

Firstly, we use the synthetic data to evaluate our method when the graph structure over groups is given as a priori property. We generate 500 data points $(\mathbf{x}_i, \mathbf{y}_i)$ where each point has 1000 features. These features are divided into 100 groups (each group has 10 features) . The features in the first 10 groups are selected as the support of $\mathbf{w}$. Each $\mathbf{y}_i$ is computed by $\mathbf{y}_i = \mathbf{w}^{\mathbf{T}}\mathbf{x}_i + \varepsilon$, where $\varepsilon \sim \mathcal{N}\left(0, \sigma^2\right)$ and $\sigma$ is Gaussian-distributed noise. We use $Least\ Square$ as the loss function $L\left(\mathbf{w}\right) = \left\|\mathbf{Y} - \mathbf{X}\mathbf{w}\right\|^2$.

Following the comparison method presented by (Huang, Zhang, and Metaxas 2011), we use relative difference $r = \left\|\mathbf{w}_{est} - \bar{\mathbf{w}}\right\|_2 / \left\|\bar{\mathbf{w}}\right\|_2$ for comparison, where $\bar{\mathbf{w}}$ and $\mathbf{w}_{est}$ stand for the ground-truth and the estimated parameter, respectively. In this experiment, the graph structure over these 100 groups is generated as follows:

- **The Group and Graph Structures**: Based on the priori properties, we generate a graph where the first 10 groups are connected as a path and the cost of each edge on this path is $-0.05$. Other groups are isolated in the graph. For the edges of the source node $s$ and the sink node $t$, we set $\{c_{su} = 0 \left| u \in V\right.\}$ and $\{c_{ut} = 1 \left| u \in V\right.\}$

We compare the relative difference of our $g^2$-regularization method with Lasso, group lasso and graph (without group) sparsity . The group structure used in the group lasso is exactly the same as that used in the $g^2$-regularizer. As for graph (without groups) sparsity, we construct a graph where the first 100 features are connected and the cost of each connected edge is $-0.005$. Other features are isolated in this graph.

Based on the exact group and graph structural properties, Table 1 shows that our method gets the best result comparing with any of the other methods.

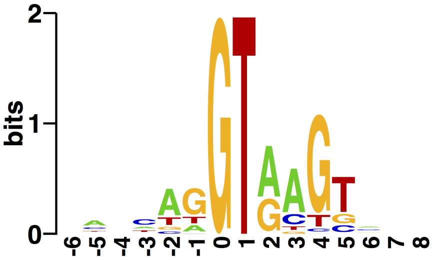


Figure 2: Sequence Logo representation of the 5' splice site case. It is modeled within a window from $-3$ to 5. The consensus 'GT' appears at positions 0, 1.

Moreover, in order to show the robust of the $g^2$-regularization, we generate a random graph structure. The result of relative difference from the random graph is also 4.0, which is still comparable with group lasso.

In Figure 1, we further analyze the parameter estimation of the different regularizations. The $g^2$-regularization, group lasso and graph sparsity (Fig. 1(a)-1(c)) can select features more accurately than Lasso (Fig. 1(d)). Graph-based sparsity regularization methods (Fig. 1(a) and 1(b)) do better than group lasso(Fig. 1(c)). It is noteworthy that the $g^2$-regularization (Fig. 1(a)) selects the features with a smaller deviation than the graph sparsity regularization without the group structure (Fig. 1(b)). This further analysis illustrates that the combination of the group and graph structures is better than using the graph or group structure only.

## Application on Splice Site Prediction

In order to evaluate the performance of our method on real-world applications, we apply our method to two splice site prediction datasets, MEMset and NN269 dataset. Splice site prediction plays an important role in gene finding. The samples of splice site prediction data are sequences of factors with 4 levels $\{A, C, G, T\}$.

**MEMset Dataset** ***(1) Experiment Setup*** This dataset is available at http://genes.mit.edu/burgelab/maxent/ssdata/.

It can be divided into two cases: 5' splice site case and 3' splice site case. In 5' case, the sequences have length 7 with 4 level $\{A, C, G, T\}$, while in 3' case, the sequences have length 21 with the same 4 level. More information about this dataset can be found in (Yeo and Burge 2004). MEMset is a very unbalanced dataset. In our experiments, we randomly choose a balanced subset with 6000 true and 6000 false donor splice sites for evaluation in 5' case and in 3' case, respectively. We use the $logistic\ Regression$ loss function $L(\mathbf{w}) = \sum_{i=1}^{N} \log\left(1 + e^{-\mathbf{y}_i \mathbf{x}^{\mathbf{T}} \mathbf{x}_i}\right)$. In addition, we randomly choose another 600 true and 600 false splice sites as validation data in 5' case and 3' case, respectively. The control parameter of $\lambda$ in Eq.1 is tuned on the validation data.

***(2) The Group and Graph Structures*** Identified in previous study (Roth and Fischer 2008), features are firstly grouped with factor interactions. In 5' case, a DNA sequence is modeled within a window from $-3$ to 5 as in Figure 2. Removing the consensus sequence 'GT' results in sequences of length 7, i.e., sequences of 7 factors with 4 levels {A, C, G, T }. The Sequences Logo in Figure 2 are represented as a collection of all factor interactions up to order 4, such as $\{(-2), (-1,), (2), ...\}$ of $1_{st}$-order, $\{(-2, -1), (-2, 2), ...\}$ of $2_{nd}$-order, and so on. Each interaction is treated as a group, leading to **98** groups that include 7 groups of $1_{st}$-order, 21 groups of $2_{nd}$-order, 35 groups of $3_{rd}$-order, and 35 groups of $4_{th}$-order. The dimension of feature space is 11564. In 3' case, with the same setup, we have **1,561** groups. The dimension of feature space is $88,564$.

Secondly, according to the order, all groups are partitioned into four lists ($1_{st}$-order, $2_{nd}$-order, $3_{rd}$-order and $4_{th}$-order) in 5' case. The groups in each list are with the same size. In each list, the groups are sorted by the dictionary order of factors, such as $((-2, -1), (-2, 2), (-2, 3), ...)$. Two adjacent groups are linked from lower rank to higher rank. The edges between these adjacent groups are set with a cost of 10.

The splice site data have the properties that the groups close to the consensus pair 'GT' or 'AG' are considered to be more dependent (Yeo and Burge 2004). Consensus 'GT' (or 'AG') is a pair which appears in both the true and false samples at a fixed position in 5' case (or in 3' case). According to above data properties, in each list, the concentrated groups around the consensus pair are linked in 5' case (or 3' case). So, we link $1_{st}$-order groups of $(-2, -1, 2, 3)$ together, and $2_{nd}$-order groups $((-2, -1), (-1, 2), (2, 3))$ together, etc. These edges are set with a small cost of 0.1. For the edges connecting the source and sink nodes $s$ and $t$, we set $\{c_{su} = 0\,|u \in V\}$ and $\{c_{ut} = 1\,|u \in V\}$. As for the graph sparsity without groups, we use the similar graph structure except that we link all the inner group features.

The above graph construction processes is far from optimal; however, it is sufficient to be used in our experiments, though a better graph may lead to a better performance.

***(3) Results*** We use the Maximum Correlation Coefficient (MCC) and Receiver Operating Curve(ROC) as two performance measures. The detailed description of MCC and ROC is attached in the supplement material of this paper.

We randomly partition the data into the training and test sets for 10 times, and report the average results as well as standard deviations over the 10 repetitions. In Table 2, we compare the $g^2$-regularization with $\ell_1$-norm and the group lasso regularizations. Our method is significantly better than the other methods based on the $t$-test at 95% significance level in both 5' case and 3' case. Group structure can help enforce sparsity on the group level, but it is still not enough to represent the linked structure in the feature space. When we make additional use of the graph structures on groups, the performance is improved significantly.

In Table 2, please note that results of graph sparsity without groups cannot be obtained within 24 hours or longer, since the scale of the graph is too dense and large to be optimized. However, our $g^2$-regularization can obtain results within about 20.9 seconds in 5' case and 201.2 seconds in 3' case. It works efficiently because the group structure helps to

Table 2: Maximum correlation coefficient(MCC, mean±std.) of the $g^2$-regularization and other methods on the MEMset and NN269 dataset. N/A stands for no results obtained within 24 hours.

| | $\ell_1$ regularizer | Group sparsity | $g^2$-regularization | Graph sparsity |
|---|---|---|---|---|
| MEMset - 5'case | 0.8716±0.0102 | 0.8847±0.0113 | **0.9267±0.0071** | N/A |
| MEMset - 3'case | 0.8215 ±0.0162 | 0.8285±0.0139 | **0.8577±0.0179** | N/A |
| NN269 - 5'case | 0.8711±0.0205 | 0.8920±0.0238 | **0.9436±0.0238** | N/A |
| NN269 - 3'case | 0.8203±0.0213 | 0.8312±0.0226 | **0.8508±0.0185** | N/A |

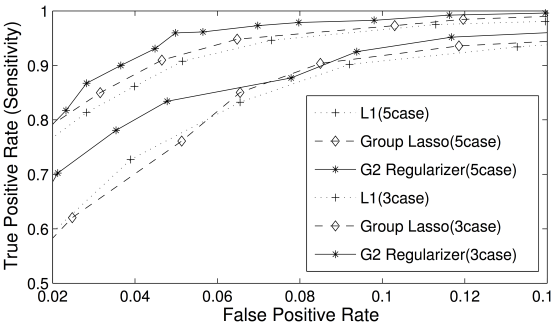


Figure 3: ROC for $\ell_1$-norm, Group Lasso, and $g^2$-regularization on MEMset, 5' case and 3' case, respectively.

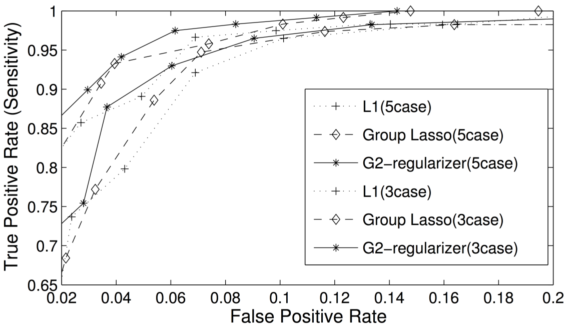


Figure 4: ROC curves for $\ell_1$-norm, Group Lasso, and $g^2$-regularization on NN269, 5' case and 3' case, respectively.

reduce the graph structure to a smaller scale. We have only 98 and 1561 groups (nodes in the graph) in 5' case and 3' case, respectively.

In addition to MCC, we applied another measure of ROC for evaluation with a view to illustrating the performance in a binary classifier hypothesis test. In Figure 3, $g^2$ regularization shows the best performance where the two solid curves (5' case and 3' case, respectively) are closer to the left and top borders.

**NN269 Dataset** ***(1) Experiment Setup*** We use the *NN269* dataset for more real-world data evaluation (Reese et al. 1997), which is available at http://www.fruitfly.org/data/seq_tools/datasets/Human/GENIE_96/splicesets/.

It can also be divided into two cases: the 5' splice site and the 3' splice site. In 5' case, the sequences have length 15 with 4 level $\{A, C, G, T\}$, while the sequences in 3' case have length 90 with 4 level $\{A, C, G, T\}$. It is not so unbalanced of NN269 dataset. Thus, in 5' case, we directly use the dataset with 1116 true and 4140 false donor sites for evaluation (another 208 true and 782 false donor site for validation). In 3' case, we use the dataset with 1116 true and 4672 false acceptor sites for evaluation (another 208 true and 881 false sites for validation). The control parameter of λ in Eq.1 is tuned on the validation data.

***(2) The Group and Graph Structures*** The sequence length in this dataset is much longer than that in the MEMset, especially in 3' case. Using all the factor interactions up to order 4 will lead to more than 2.5 million groups. So, in the NN269 dataset, only succession interactions up to order 4 are grouped in both 5' and 3' case. The graph structure is then used to model the long range interactions. Several small groups in one path could be viewed as a big group with long range interactions. According to above structural properties, we use a connected DAG with cost 0.1 for each edge. We set $\{c_{su} = 0 \,|u \in V\}$ and $\{c_{ut} = 1 \,|u \in V\}$. Finally, we have **54** groups with 4188 features in 5'case and **354** groups with 29688 features in 3' case. For the graph sparsity without groups, we use a connected DAG on the each feature, and the cost of each edge is similar with that in $g^2$-regularization.

***(3) Results*** We randomly partition the data into the training and test sets for 10 times, and report the average results as well as standard deviations over the 10 repetitions. As shown in Table 2, the performance of using the $g^2$ regularization is significantly better compared with the $\ell_1$-norm and group lasso based on the $t$-test at 95% significance level in both 5' case and 3' case. Through ROC measures, $g^2$-regularization also shows the best performance in NN269 dataset, as shown in Figure 4.

## Conclusions

In this paper, we propose a new form of structured sparsity called $g^2$-regularization. Theoretical properties of the proposed regularization are discussed. With the graph structures on groups, we apply the minimum network flow and proximal gradient method for the optimization. Experiments on both synthetic and real data demonstrate its superiority over some other sparse or structured sparse models. In the future, we will explore the graph structure with more inherent structural properties in data. Further research will be conducted to find more efficient optimization method on large-scale graph.

# References

Bach, F. 2008. Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research* 9:1179–1225.

Beck, A., and Teboulle, M. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences* 2(1):183–202.

Chen, S.-B.; Ding, C.; Luo, B.; and Xie, Y. 2013. Uncorrelated lasso. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 166–172.

Guo, Y., and Xue, W. 2013. Probabilistic multi-label classification with sparse feature learning. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, 1373–1380.

Huang, J. Z., and Zhang, T. 2010. The benefit of group sparsity. *Annals of Statistics* 38(4):19782004.

Huang, J. Z.; Zhang, T.; and Metaxas, D. 2011. Learning with structured sparsity. *Journal of Machine Learning Research* 12:3371–3412.

Jacob, L.; Obozinski, G.; and Vert, J.-P. 2009. Group lasso with overlap and graph lasso. In *Proceedings of the 26th International Conference on Machine Learning*, 433–440.

Jenatton, R.; Audibert, J.-Y.; and Bach, F. 2011. Structured variable selection with sparsity-inducing norms. *Journal of Machine Learning Research* 12:2777–2824.

Kowalski, M.; Szafranski, M.; and Ralaivola, L. 2009. Multiple indefinite kernel learning with mixed norm regularization. In *Proceedings of the 26th International Conference on Machine learning*, 545–552.

Liu, J.; Ji, S. W.; and Ye, J. P. 2009. Multi-task feature learning via efficient l2,1-norm minimization. In *Uncertainty in Artificial Intelligence (UAI)*, 339–348.

Mairal, J., and Yu, B. 2013. Supervised feature selection in graphs with path coding penalties and network flows. *Journal of Machine Learning Research* 14:2449–2485.

Meier, L.; van de Geer, S.; and Buhlmann, P. 2008. The group lasso for logistic regression. *Journal of the Royal Statistical Society. Series B* 70(1):53–71.

Nesterov, Y. 2007. Gradient methods for minimizing composite objective function. Technical report, Technical report, CORE Discussion paper.

Reese, M. G.; Eechman, F. H.; Kulp, D.; and Haussler, D. 1997. Improved splice site detection in genie. *Journal of Computational Biology* 4(3):311–324.

Roth, V., and Fischer, B. 2008. The group-lasso for generalized linear models: uniqueness of solutions and efficient. In *Proceedings of the 25th International Conference on Machine learning*, 848–855.

Tibshirani, R. 1994. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B* 58(1):267–288.

Yeo, G., and Burge, C. B. 2004. Maximum entropy modeling of short sequence motifs with applications to rna splicing signals. *Journal of Computational Biology* 11(2/3):377–394.

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