Group and Graph Joint Sparsity for Linked Data Classification

Longwen Gao and **Shuigeng Zhou***
Shanghai Key Lab of Intelligent Information Processing, and School of Computer Science
Fudan University, Shanghai 200433, China
{lwgao, sgzhou}@fudan.edu.cn

Abstract

Various sparse regularizers have been applied to machine learning problems, among which structured sparsity has been proposed for a better adaption to structured data. In this paper, motivated by effectively classifying linked data (e.g. Web pages, tweets, articles with references, and biological network data) where a group structure exists over the whole dataset and links exist between specific samples, we propose a joint sparse representation model that combines group sparsity and graph sparsity, to select a small number of connected components from the graph of linked samples, meanwhile promoting the sparsity of edges that link samples from different groups in each connected component. Consequently, linked samples are selected from a few sparsely-connected groups. Both theoretical analysis and experimental results on four benchmark datasets show that the joint sparsity model outperforms traditional group sparsity model and graph sparsity model, as well as the latest group-graph sparsity model.

Introduction

Sparse representation (SR) has been proposed to do classification due to its simple formulation and natural relationship with the parsimony principle (Wright et al., 2009). It represents each test sample as a sparse linear combination of the training samples and then classifies the sample according to its coefficient vector. The ℓ_1-norm is used to promote the sparsity of the coefficient vector. SR has been applied to classification tasks in various domains, such as images (Majumdar and Ward, 2009; Wright et al., 2009; Gao and Zhou, 2014), texts (Sainath et al., 2010) and biological data (Li and Ngom, 2012; Yuan et al., 2012).

For a better adaption to various classification tasks, regularizers of *structured sparsity* were introduced. These regularizers use the structure of the training data as prior knowledge. For example, *group sparse representation* (GSR) (Jenatton, Audibert, and Bach, 2011) imposes the sparsity among groups of samples. It works better than SR when the underlying samples are strongly group-sparse (Huang and Zhang, 2009). For data with overlapping/hierarchical class structure, *overlapping group sparsity* (Jacob, Obozinski, and Vert, 2009) allows groups to be overlapped, and composite absolute penalties (CAPs) (Zhao, Rocha, and Yu, 2006) put a hierarchical group structure among the samples. When samples are not grouped but linked with each other, *graph structure sparsity* (Huang, Zhang, and Metaxas, 2009) tends to select a subset of connected training samples. Recently, Gao and Zhou (2015) proposed *uncertain group sparse representation* to handle data with uncertain group membership.

Nowadays, *linked data* exist ubiquitously and are amassed rapidly. A common feature of linked data is that a group structure exists over the whole dataset and links exist between specific samples. For instance, Web pages are not only naturally grouped via their semantic topics but also interconnected by their hyperlinks; Tweets have topics and are also connected in terms of the relationships between their authors (or followers and readers). The links between samples provide complementary clues to the class labels of those samples. Exploiting the link information (i.e., the *graph structure* of samples) and the label information (i.e., the *group structure* of samples) simultaneously can definitely facilitate the classification tasks on linked data. As both label information and link information may contain noise, exploring them together can also enhance learning robustness.

Recently, Dai et al. (2015) proposed the so-called *group-graph* (g^2 in short) *sparsity* to exploit the advantages of both group and graph structures simultaneously. g^2 sparsity is actually to impose graph sparsity on groups, in stead of on samples. They assumed that a graph structure among the groups is given explicitly, and samples in one group are linked with all samples in another group if these two corresponding groups are connected in the graph of groups. That is, they assumed dense connections between samples from different groups connected in the graph. However, such assumption is not always applicable. And with the given graph of groups, detailed links between samples are not considered any more. For linked data, explicit graph structure exists among samples, instead of among groups, and there may not be dense connections between samples of different groups even though the involved groups are related. These will unavoidably cause performance degradation of g^2 sparsity in classifying linked data.

In this paper, motivated by the significance of classifying the increasing linked data and the limitations of existing structured sparsity models in linked data classification, we propose a new joint regularizer to nontrivially combine

*Correspondence author.

group sparsity and graph sparsity at sample level for effectively classifying linked data. Our regularizer selects a small number of connected components of the graph of samples, while the edges between different groups in each connected component should be sparse. That is, our regularizer exploits the advantage of graph structure by selecting connected components of samples as in graph sparsity, and the advantage of group structure by restraining the connections across groups to be sparse. Contributions of this paper are: 1) We propose a new joint sparse regularizer that exploits group and graph structures simultaneously for effectively classifying linked data; 2) We design an efficient algorithm to solve the new sparsity model and theoretically show the advantage of the new sparsity model over the existing ones; 3) We carry out extensive experiments on real datasets, which show that the new joint sparsity outperforms traditional group sparsity and graph sparsity, as well as the latest g^2 sparsity.

Major Existing Structured-Sparsity Models

Here we briefly introduce *group sparsity*, *graph sparsity* and *group-graph sparsity* (*g^2 sparsity* in short) in the context of representation. Assume we have a dataset such that: 1) there are M training samples $\boldsymbol{D}^{1\cdots M}$ in $\mathcal{R}^d$ that fall into G different classes, where each training sample i has a label in $\{1..G\}$; and 2) the M training samples are linked by the edges of a given graph $\boldsymbol{W} \in \{0,1\}^{M\times M}$. Given a test sample $\boldsymbol{y}$, we are to represent it using a dictionary $\boldsymbol{D} \in \mathcal{R}^{d\times M}$ constituted by all the training samples.

Group sparsity or *group sparse representation* (*GSR*) (Jenatton, Audibert, and Bach, 2011) relates the test sample with a small number of groups of training samples. GSR requires the coefficients of different groups to be sparse:

$$\min_{\boldsymbol{x}\in\mathcal{R}^M} \left\{ \frac{1}{2} \left\|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\right\|_2^2 + \lambda\Omega_{Group}(\boldsymbol{x}) \right\}$$
$$with\ \Omega_{Group}(\boldsymbol{x}) = \sum_{g=1}^{G} \sqrt{\sum_{i\in\mathcal{G}_g} \boldsymbol{x}_i^2} = \sum_{g=1}^{G} \left\|\boldsymbol{x}_{\mathcal{G}_g}\right\|_2. \tag{1}$$

Above, $\mathcal{G}_g$ is the set of indices of training samples with label $g \in \{1..G\}$, $\lambda > 0$ is a tradeoff parameter, and each class is considered to be a group. The first term is the regression error for linear representation and the second term is the group sparse regularizer. If we define a group function $\sigma(\boldsymbol{x})$ as in (Dai et al., 2015):

$$\sigma(\boldsymbol{x}) = \left(\left\|\boldsymbol{x}_{\mathcal{G}_1}\right\|_2, \left\|\boldsymbol{x}_{\mathcal{G}_2}\right\|_2, \ldots, \left\|\boldsymbol{x}_{\mathcal{G}_G}\right\|_2\right), \tag{2}$$

then the group sparse regularizer is a composition of ℓ_1 sparse norm and the group function σ:

$$\Omega_{Group}(\boldsymbol{x}) = \left\|\sigma(\boldsymbol{x})\right\|_1 = \mathbf{1}^\top \left|\sigma(\boldsymbol{x})\right|. \tag{3}$$

Graph sparsity (Huang, Zhang, and Metaxas, 2009) selects a small number of connected components of the graph to represent the test sample. Let $\mathcal{P}$ be the set of all paths in graph $\boldsymbol{W}$ and $\eta_p > 0$ be the weight of each path $p \in \mathcal{P}$. The representation process with the graph sparse regularizer is defined as follows:

$$\min_{\boldsymbol{x}\in\mathcal{R}^M} \left\{ \frac{1}{2} \left\|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\right\|_2^2 + \lambda\Omega_{Graph}(\boldsymbol{x}) \right\}$$
$$with\ \Omega_{Graph}(\boldsymbol{x}) = \min_{\mathcal{J}\subset\mathcal{P}} \left\{ \sum_{p\in\mathcal{J}} \eta_p\ s.t.\ Supp(\boldsymbol{x}) \subseteq \bigcup_{p\in\mathcal{J}} p \right\}, \tag{4}$$

where $Supp(\cdot)$ finds the index set of non-zero elements for a vector. The path weight η_p is usually set to the sum of edge weights in path p (Dai et al., 2015). If the relationship between any two samples is assumed to be similar importance, the edge weights are constant.

g^2 sparsity (Dai et al., 2015) is a combination of group sparsity and graph sparsity by imposing graph structure over groups, which selects a small number of connected groups of samples to represent the test sample. Here, vertices of graph are groups and a path in the graph consists of groups and edges between them. It is formulated as below:

$$\min_{\boldsymbol{x}\in\mathcal{R}^M} \left\{ \frac{1}{2} \left\|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\right\|_2^2 + \lambda\Omega_{g^2}(\boldsymbol{x}) \right\}$$
$$with\ \Omega_{g^2}(\boldsymbol{x}) = \min_{\mathcal{J}\subset\mathcal{P}} \left\{ \sum_{p\in\mathcal{J}} \eta_p\ s.t.\ Supp(\sigma(\boldsymbol{x})) \subseteq \bigcup_{p\in\mathcal{J}} p \right\}, \tag{5}$$

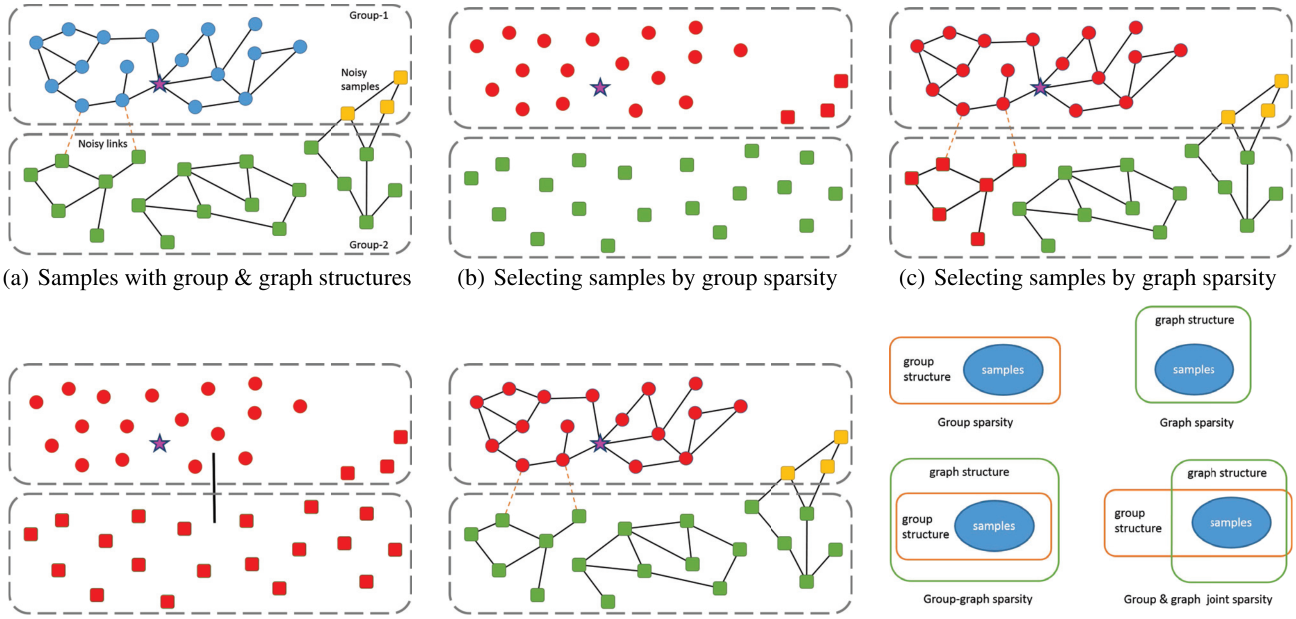
Group and Graph Joint Sparsity

Here, we first give a motivation example to illustrate the limitations of the existing sparsity models in handling linked data, and the differences between our model and the existing models. Then we present the new model and an optimization algorithm to solve the model. Finally, we theoretically show the advantage of our model over the existing models.

A motivation example

We consider a 2-class classification problem. Fig. 1(a) shows the training data that fall into two classes (*group-1/2*) corresponding to the upper and lower dashed rectangles. The training samples are locally interlinked (*graph structure*). There is also noisy label/link information: the yellow squares located in the bottom-right corner of the upper rectangle are noisy samples (belonging to *group-2* but incorrectly labeled to *group-1*) and the red dotted edges across the two groups are noisy links (error links). Given a test sample (the purple star) located in the upper rectangle, we are to find its class label by different models: group sparsity, graph sparsity, g^2 sparsity and our new sparsity, respectively.

Different models tend to select different collections of training samples to represent the test sample. For group sparsity, as shown in Fig. 1(b), it selects exactly the samples in *group-1*, including the noisy samples. Here, all selected samples are red colored. Fig. 1(b) shows no links between samples because group sparsity considers no graph structure. For graph sparsity, it selects the samples in a connected component of the graph, including the samples in *group-2* connected by the noisy links, as shown in Fig. 1(c). The



(a) Samples with group & graph structures (b) Selecting samples by group sparsity (c) Selecting samples by graph sparsity

(d) Selecting samples by g^2 sparsity (e) Selecting samples by our new sparsity (f) Comparing different sparsity models

Figure 1: A motivation example for illustrating the limitations of three existing structured sparsity models and the differences between them and our group & graph joint sparsity model. (a) shows the training samples falling to two classes. (b), (c), (d) and (e) show the samples selected by the four models respectively. All selected samples are in red color. (f) shows how the four models exploit group structure and/or graph structure differently.

noisy samples are not selected because they are not reachable to the test sample. For g^2 sparsity, there are links between the two groups, which makes it believe that the two groups are connected, thus it selects all the samples in the two groups, as shown in Fig. 1(d). Here links between samples are not shown as g^2 does not consider them. Instead, we draw a link between the two groups as they are regarded "connected". In summary, all the three models have selected inappropriate samples to represent the test sample, which is unfavorable to the classification task.

Intuitively, the ideal result is to select only the blue samples in *group-1*, as shown in Fig. 1(e). So we have to leave out the unlinked noisy samples (the yellow squares) in the upper rectangle, and try to cut off the connections across groups. For this end, our new sparsity model is designed to select samples in a small number of connected components (*graph sparsity*) while requiring that each connected component contains as few edges interlinking samples of different groups as possible (*group sparsity*). Consequently, our selected samples are connected, and cover fewer groups than those selected by graph sparsity and g^2 sparsity. In the sequel, we call edges across different groups *border edges*.

Fig. 1(f) further illustrates how the four models exploit group structure and/or graph structure differently. Group/graph sparsity imposes only group/graph structure directly on samples. Though both g^2 sparsity and our sparsity exploit the advantages of group and graph structures simultaneously, g^2 sparsity imposes graph structure on groups, in other words, it embeds groups into the graph structure, while our sparsity nontrivially imposes group and graph structures simultaneously on samples. So we call our sparsity *group & graph joint sparsity*, simply *GGJ sparsity*.

Regularizer formulation and optimization

Let $\mathcal{Q}$ be the set of all border edges. We use a vector $\boldsymbol{\eta} \in \mathcal{R}_+^{|\mathcal{P}|}$ as a weight vector for all paths in $\mathcal{P}$. Let $\boldsymbol{N}$ be a binary matrix in $\{0,1\}^{|\mathcal{P}|\times M}$ where each column indicates if a certain sample is in each path or not. Similarly, let $\boldsymbol{E}$ be a binary matrix in $\{0,1\}^{|\mathcal{Q}|\times|\mathcal{P}|}$ where each column indicates if a certain path contains each border edge or not. We penalize the number of selected border edges as follows:

$$\begin{aligned} &\Omega_{GGJ}(\boldsymbol{x}) = \min_{\boldsymbol{w}\in\{0,1\}^{|\mathcal{P}|}} \left\{\boldsymbol{\eta}^\top \boldsymbol{w} + \gamma \|\boldsymbol{E}\boldsymbol{w}\|_0\right\} \\ &s.t.\ \boldsymbol{w} \geq Supp(\boldsymbol{N}\cdot Supp(\boldsymbol{x})). \end{aligned} \tag{6}$$

Here, $\boldsymbol{w}$ indicates whether each path is selected or not, $\|\cdot\|_0$ counts the number of border edges contained by each selected path, and γ is the trade-off parameter. Minimizing the term $\boldsymbol{\eta}^\top\boldsymbol{w}$ equals to finding a small number of paths. The two terms in the objective function together are for finding the desirable samples: they are in a small number of connected components and those connected components contain as few border edges as possible. This constraint is different from that in (Dai et al., 2015), still, it replaces the original constraint $Supp(\boldsymbol{x}) \subseteq \bigcup_{p\in\mathcal{J}} p$ in graph sparsity by requiring $\boldsymbol{w}$ to cover the paths containing all the vertices corresponding to the non-zeros elements in $\boldsymbol{x}$.

Convex relaxation One drawback of the regularizer in Eq. (6) is that the objective is difficult to optimize because

$\|\cdot\|_0$ is not convex. In the literature, $\|\cdot\|_0$ is usually approximated by its convex relaxation ℓ_1-norm. We follow the same routine and approximate our regularizer as follows:

$$\begin{aligned}\Omega_{GGJ}(\boldsymbol{x}) &\approx \min_{\boldsymbol{w}\in\{0,1\}^{|\mathcal{P}|}} \left\{\boldsymbol{\eta}^\top \boldsymbol{w} + \gamma \left\|\boldsymbol{E}\boldsymbol{w}\right\|_1\right\} \\ &= \min_{\boldsymbol{w}\in\{0,1\}^{|\mathcal{P}|}} \left\{\boldsymbol{\eta}^\top \boldsymbol{w} + \gamma \boldsymbol{1}^\top(\boldsymbol{E}\boldsymbol{w})\right\} \\ &= \min_{\boldsymbol{w}\in\{0,1\}^{|\mathcal{P}|}} \left\{\boldsymbol{\eta}^\top \boldsymbol{w} + \gamma (\boldsymbol{1}^\top\boldsymbol{E})\boldsymbol{w}\right\} \\ &= \min_{\boldsymbol{w}\in\{0,1\}^{|\mathcal{P}|}} \left\{\boldsymbol{\eta}^\top \boldsymbol{w} + \gamma \boldsymbol{e}^\top\boldsymbol{w}\right\} \\ &s.t.\ \boldsymbol{w} \geq Supp(\boldsymbol{N}\cdot Supp(\boldsymbol{x})).\end{aligned} \tag{7}$$

Here $\boldsymbol{e}$ is a vector whose elements are the number of border edges in each path. As we have relaxed the non-convex term in Eq. (6), we do not penalize the number of border edges, instead, we penalize the number of paths containing border edges, by using the number of border edges on each path.

In real applications, we need not count the border edges for each path, since the vector $\boldsymbol{e}$ can be further approximated as $\widetilde{\boldsymbol{e}}$ whose elements are the number of groups in each path. $\widetilde{\boldsymbol{e}}$ is much simpler and more "group-sparse":

$$\begin{aligned}\Omega_{GGJ}(\boldsymbol{x}) &= \min_{\boldsymbol{w}\in\{0,1\}^{|\mathcal{P}|}} \left\{\boldsymbol{\eta}^\top \boldsymbol{w} + \gamma \widetilde{\boldsymbol{e}}^\top\boldsymbol{w}\right\} \\ &s.t.\ \boldsymbol{w} \geq Supp(\boldsymbol{N}\cdot Supp(\boldsymbol{x})).\end{aligned} \tag{8}$$

Therefore, the objective function of our sparsity model is:

$$\min_{\boldsymbol{x}\in\mathcal{R}^M} \left\{\frac{1}{2}\left\|\boldsymbol{y}-\boldsymbol{D}\boldsymbol{x}\right\|_2^2 + \lambda\Omega_{GGJ}(\boldsymbol{x})\right\} \tag{9}$$

Smoothing discrete functions and variables It is easy to see that the Eq. (8) is still non-convex due to its non-convex constraint. We use a convex and derivable function $\delta(\boldsymbol{x})$ to replace $\boldsymbol{N}\cdot Supp(\boldsymbol{x})$ and use a convex function $|\cdot|$ to replace the outer $Supp(\cdot)$, and have

$$\begin{aligned}\Omega_{GGJ}(\boldsymbol{x}) &\approx \min_{\boldsymbol{w}\in\mathcal{R}_+^{|\mathcal{P}|}} \left\{\boldsymbol{\eta}^\top \boldsymbol{w} + \gamma \widetilde{\boldsymbol{e}}^\top\boldsymbol{w}\right\} \\ &s.t.\ \boldsymbol{w} \geq \left|\delta(\boldsymbol{x})\right|,\end{aligned} \tag{10}$$

where the group function $\delta(\boldsymbol{x})$ is defined as follows:

$$\delta(\boldsymbol{x}) = \left(\left\|\boldsymbol{x}_{p_1}\right\|_2, \left\|\boldsymbol{x}_{p_2}\right\|_2, \ldots, \left\|\boldsymbol{x}_{p_{|\mathcal{P}|}}\right\|_2\right),\ p_i\in\mathcal{P}. \tag{11}$$

The reason is:

$$\begin{cases}\left\|\boldsymbol{x}_{p_i}\right\|_2 = 0 \iff \boldsymbol{N}_i\cdot Supp(\boldsymbol{x}) = 0; \\ \left\|\boldsymbol{x}_{p_i}\right\|_2 > 0 \iff \boldsymbol{N}_i\cdot Supp(\boldsymbol{x}) \neq 0.\end{cases} \tag{12}$$

In Eq. (10), $\boldsymbol{\eta} \geq \boldsymbol{0}$ and $\gamma\widetilde{\boldsymbol{e}} \geq \boldsymbol{0}$. Thus according to (Bertsimas and Tsitsiklis, 1997), the optimum of the linear programming is $\boldsymbol{w} = |\delta(\boldsymbol{x})|$. Substituting this into the representation model (9), we have:

$$\min_{\boldsymbol{x}\in\mathcal{R}^M} \left\{\frac{1}{2}\left\|\boldsymbol{y}-\boldsymbol{D}\boldsymbol{x}\right\|_2^2 + \left(\lambda\boldsymbol{\eta} + \lambda\gamma\widetilde{\boldsymbol{e}}^\top\right)\left|\delta(\boldsymbol{x})\right|\right\} \tag{13}$$

Optimization The model (13) can be solved using the proximal method. The regression error term is a smooth differentiable function and can be linearized around the current point $\boldsymbol{x}^t$ at each iteration as in (Bach et al., 2012):

$$\min_{\boldsymbol{x}\in\mathcal{R}^M} \left\{\frac{1}{2}\left\|\boldsymbol{x} - (\boldsymbol{x}^t - \frac{1}{L}\nabla f(\boldsymbol{x}^t))\right\|_2^2 + \frac{\lambda}{L}\Omega_{GGJ}(\boldsymbol{x})\right\}, \tag{14}$$

where f is the regression error function and L is the upper bound of $\left\|\boldsymbol{D}^\top\boldsymbol{D}\right\|_2$ (Baldassarre et al., 2012).

Recall that the proximity operator introduced by Moreau (Moreau, 1962) can be defined as follows:

Definition 1. *φ is a real-valued convex function on $\mathcal{R}^M$, its proximity operator $Prox_\varphi \boldsymbol{x}(\boldsymbol{u})$ is defined as:*

$$\arg\min_{\boldsymbol{x}} \left\{\frac{1}{2}\left\|\boldsymbol{u}-\boldsymbol{x}\right\|_2^2 + \varphi(\boldsymbol{x}) : \boldsymbol{u}\in\mathcal{R}^M\right\}. \tag{15}$$

Employing the notation of proximity operator, the update rule for Eq. (14) is

$$\boldsymbol{x} = Prox_{\frac{\lambda}{L}\Omega_{GGJ}}(\boldsymbol{x}^t - \frac{1}{L}\nabla f(\boldsymbol{x}^t)). \tag{16}$$

Now we try to compute $Prox_{\frac{\lambda}{L}\Omega_{GGJ}}$ by relating the computation of $Prox_{\frac{\lambda}{L}\Omega_{GGJ}}$ with the computation of $Prox_{\Omega_{Group}}$ as in (Micchelli, Shen, and Xu, 2011):

$$Prox_{\frac{\lambda}{L}\Omega_{GGJ}}(\boldsymbol{u}) = \boldsymbol{u} - c\boldsymbol{B}^\top\boldsymbol{v}. \tag{17}$$

Here, $\boldsymbol{B}\in\mathcal{R}^{(|\mathcal{P}|M)\times M}$ is a projection matrix defined as:

$$\begin{aligned}&\boldsymbol{B} = \left[\boldsymbol{B}_1^\top\boldsymbol{B}_2^\top\cdots\boldsymbol{B}_{|\mathcal{P}|}^\top\right]^\top, \\ &\boldsymbol{B}_i\in\mathcal{R}^{M\times M}\ and\ (\boldsymbol{B}_i)_j^j = 1\ if\ j\in p_i,\end{aligned} \tag{18}$$

and $\boldsymbol{v}\in\mathcal{R}^{|\mathcal{P}|M}$ is a fixed-point of H defined as follows:

$$H(\boldsymbol{v}) = \left(\boldsymbol{I} - Prox_{\frac{\lambda}{cL}\Omega_{Group}}\right)\left(\left(\boldsymbol{I} - c\boldsymbol{B}\boldsymbol{B}^\top\right)\boldsymbol{v} + \boldsymbol{B}\boldsymbol{u}\right). \tag{19}$$

Here, each group corresponds to a path in $\mathcal{P}$. As the proximity operator of Ω_{Group} can be computed directly as in (Bach et al., 2012), $Prox_{\frac{\lambda}{L}\Omega_{GGJ}}(\boldsymbol{u})$ is computed by first finding a fixed-point and then using Eq. (17). This algorithm is guaranteed to converge (Micchelli, Shen, and Xu, 2011).

Theoretical analysis

We give the following proposition to indicate the advantage of our regularizer over the two existing graph-based regularizers: graph sparsity regularizer and g^2 sparsity regularizer.

Proposition 1. *Give a dataset $\boldsymbol{D}$ whose samples fall into 2 groups $\mathcal{G}_1, \mathcal{G}_2$ and there are edges connects them. Let $\boldsymbol{x}^{GGJ}, \boldsymbol{x}^G, \boldsymbol{x}^{g^2}$ be the optimums for Eq. (9), Eq. (4) and Eq. (5) respectively, where the test sample is supported by $\mathcal{G}_1$. If $\Omega_{GGJ} = \Omega_G = \Omega_{g^2}$ and $\left\|\boldsymbol{x}^{GGJ}\right\|_2 = \left\|\boldsymbol{x}^G\right\|_2 = \left\|\boldsymbol{x}^{g^2}\right\|_2 = T$, then $\exists\gamma$ such that $\left\|\boldsymbol{x}_{\mathcal{G}_2}^{GGJ}\right\|_2 \leq \left\|\boldsymbol{x}_{\mathcal{G}_2}^G\right\|_2$ and $\left\|\boldsymbol{x}_{\mathcal{G}_2}^{GGJ}\right\|_2 \leq \left\|\boldsymbol{x}_{\mathcal{G}_2}^{g^2}\right\|_2$.*

Proof. We prove the above proposition in a smoothed situation by substituting the smoothed $\boldsymbol{w} = |\delta(\boldsymbol{x})|$ into the objective of our sparse regularizer in Eq. (7):

$$\Omega_{GGJ} = \boldsymbol{\eta}^\top\left|\delta(\boldsymbol{x}^{GGJ})\right| + \gamma\left\|\boldsymbol{E}\left|\delta(\boldsymbol{x}^{GGJ})\right|\right\|_1. \tag{20}$$

Samples in $\mathcal{G}_1$ can be divided into two sets $\mathcal{C}_1$ and $\mathcal{C}_2$, where only samples in $\mathcal{C}_2$ connect samples in $\mathcal{G}_2$. Without loss of generality, we simply assume that there is only one path in each of $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{G}_2$ and $\boldsymbol{\eta} = \mathbf{1}$. Thus $|\mathcal{Q}| = 1$ and the second term becomes $\gamma \left\| \boldsymbol{x}^{GGJ}_{\mathcal{C}_2 \cup \mathcal{G}_2} \right\|_2$. Hence,

$$\begin{aligned}\Omega_{GGJ} &= \left\| \boldsymbol{x}^{GGJ}_{\mathcal{C}_1} \right\|_2 + \left\| \boldsymbol{x}^{GGJ}_{\mathcal{C}_2} \right\|_2 + \left\| \boldsymbol{x}^{GGJ}_{\mathcal{G}_2} \right\|_2 \\ &+ (1+\gamma) \left\| \boldsymbol{x}^{GGJ}_{\mathcal{C}_2 \cup \mathcal{G}_2} \right\|_2 , \\ \Omega_G &= \left\| \boldsymbol{x}^{G}_{\mathcal{C}_1} \right\|_2 + \left\| \boldsymbol{x}^{G}_{\mathcal{C}_2} \right\|_2 + \left\| \boldsymbol{x}^{G}_{\mathcal{G}_2} \right\|_2 + \left\| \boldsymbol{x}^{G}_{\mathcal{C}_2 \cup \mathcal{G}_2} \right\|_2 .\end{aligned} \tag{21}$$

For a vector $\boldsymbol{x}$, we consider two norms defined as

$$\begin{aligned}\|\boldsymbol{x}\|_a &= \|\boldsymbol{x}_{\mathcal{C}_1}\|_2 + \|\boldsymbol{x}_{\mathcal{C}_2}\|_2 + \|\boldsymbol{x}_{\mathcal{G}_2}\|_2 + \|\boldsymbol{x}_{\mathcal{C}_2 \cup \mathcal{G}_2}\|_2 , \\ \|\boldsymbol{x}\|_b^2 &= \|\boldsymbol{x}_{\mathcal{C}_1}\|_2^2 + \|\boldsymbol{x}_{\mathcal{C}_2}\|_2^2 + \|\boldsymbol{x}_{\mathcal{G}_2}\|_2^2 + \|\boldsymbol{x}_{\mathcal{C}_2 \cup \mathcal{G}_2}\|_2^2 .\end{aligned} \tag{22}$$

They can be restricted by each other due to the property of the norm in a finite dimensional Hilbert space:

$$c_{low} \|\boldsymbol{x}\|_b \le \|\boldsymbol{x}\|_a \le c_{high} \|\boldsymbol{x}\|_b . \tag{23}$$

Hence, for a fixed vector $\boldsymbol{x}$, we can find a corresponding $c \in [c_{low}, c_{high}]$ such that $\|\boldsymbol{x}\|_a = c \|\boldsymbol{x}\|_b$. Back to our proof, from the above equation, we have:

$$\begin{aligned}\Omega_{GGJ} &= \left\| \boldsymbol{x}^{GGJ} \right\|_a = c_1 \left\| \boldsymbol{x}^{GGJ} \right\|_b , \\ \Omega_G &= \left\| \boldsymbol{x}^{G} \right\|_a = c_2 \left\| \boldsymbol{x}^{G} \right\|_b .\end{aligned} \tag{24}$$

As $\mathcal{C}_1, \mathcal{C}_2$ and $\mathcal{G}_2$ are non-overlapping sets, they become:

$$\begin{aligned}\Omega^2_{GGJ} &= c_1^2 \left(\|\boldsymbol{x}^{GGJ}\|_2^2 + (1+\gamma) \left(\|\boldsymbol{x}^{GGJ}_{\mathcal{C}_2}\|_2^2 + \|\boldsymbol{x}^{GGJ}_{\mathcal{G}_2}\|_2^2 \right) \right) , \\ \Omega^2_G &= c_2^2 \left(\|\boldsymbol{x}^{G}\|_2^2 + \|\boldsymbol{x}^{G}_{\mathcal{C}_2}\|_2^2 + \|\boldsymbol{x}^{G}_{\mathcal{G}_2}\|_2^2 \right) .\end{aligned} \tag{25}$$

By letting $\Omega_{GGJ} = \Omega_G$ and $\left\| \boldsymbol{x}^{GGJ} \right\|_2 = \left\| \boldsymbol{x}^{G} \right\|_2 = T$ we have:

$$\begin{aligned}(1+\gamma) \left(\left\| \boldsymbol{x}^{GGJ}_{\mathcal{G}_2} \right\|_2^2 - \left\| \boldsymbol{x}^{G}_{\mathcal{G}_2} \right\|_2^2 \right) &= \left(\frac{c_2^2}{c_1^2} - 1 \right) T + \frac{c_2^2}{c_1^2} \left\| \boldsymbol{x}^{G}_{\mathcal{C}_2} \right\|_2^2 \\ - \left(1 + \gamma - \frac{c_2^2}{c_1^2} \right) \left\| \boldsymbol{x}^{G}_{\mathcal{G}_2} \right\|_2^2 &- (1+\gamma) \left\| \boldsymbol{x}^{GGJ}_{\mathcal{C}_2} \right\|_2^2 .\end{aligned} \tag{26}$$

Since the test sample is supported by $\mathcal{G}_1$, $\left\| \boldsymbol{x}^{GGJ}_{\mathcal{C}_2} \right\|_2 > 0$. Thus there exists a γ such that the right hand side of the above equation is less than 0, namely, $\left\| \boldsymbol{x}^{GGJ}_{\mathcal{G}_2} \right\|_2 \le \left\| \boldsymbol{x}^{G}_{\mathcal{G}_2} \right\|_2$. Similarly, we can show $\left\| \boldsymbol{x}^{GGJ}_{G_2} \right\|_2 \le \left\| \boldsymbol{x}^{g^2}_{G_2} \right\|_2$. □

The above proposition indicates that our model works better than the two graph sparsity models when there are edges connecting two different groups, because our regularizer puts smaller coefficients on the samples of the other group. Also, the approximated sparse regularizer in Eq. (13) looks similar to the formulation of group sparsity in Eq. (1), the difference is that our regularizer uses paths instead of groups, thus it finds more connected samples than group sparsity. Comparing with traditional group sparsity, the use of graph structure makes our model more robust.

Table 1: Details of datasets

	Cora	**Twitter**	**Gene**	**Protein**
# of Samples	2708	6000	1243	3185
# of Features	1433	6367	461	8000
# of Links	5278	6943	1326	6378
# of Classes	7	6	2	5

Table 2: Accuracy results on the four datasets (%)

	Cora	**Twitter**	**Gene**	**Protein**
Group	90.8065	84.8921	67.9839	83.1668
Graph	90.9225	85.2146	69.3669	82.9579
g^2	91.3548	85.5314	67.4597	83.4779
GGJ	**91.9135**	**85.8850**	**76.1694**	**84.4239**

Classification rule

After the coefficient vector $\boldsymbol{x}$ is computed, we can decide which is the most suitable label for a test sample $\boldsymbol{y}$. The maximum ℓ_2 support rule (Sainath et al., 2010) classifies a test sample $\boldsymbol{y}$ as follows:

$$label^* = \arg \max_{g \in \{1..G\}} \left\| \boldsymbol{x}_{\mathcal{G}_g} \right\|_2 . \tag{27}$$

Experiments

Four datasets are used: Cora, Twitter, Gene and Protein. The details of these datasets are summarized in Table 1.

Cora is a publication dataset (Mccallum et al., 2000) that contains 2708 machine learning papers falling to 7 classes: “Case Based”, “Genetic Algorithms”, “Neural Networks”, “Probabilistic Methods”, “Reinforcement Learning”, “Rule Learning” and “Theory”. The features consist of 1433 words with document frequency no less than 10 after stemming and stop word removal. The links are the citations of all papers.

Twitter is a collection of 6000 tweets with 6 meaningful hash tags: “redsox”, “jobs”, “IranElection”, “tlot”, “Twitter” and “music”. Hash tags are used as class labels, each of which is related to 1000 tweets. We add links between tweets that contain the same URLs as in (Duan et al.).

Gene is a protein interaction network data from KDD cup 2001. The task is to label the proteins with “nucleus” or “non-nucleus”. Each protein is represented as a 461 dimensional binary vector using categorical features including Essential, Class, Complex, Phenotype, Motif and Chromosome. The links are the interactions between those proteins and 1326 remains after removing negative weighted links.

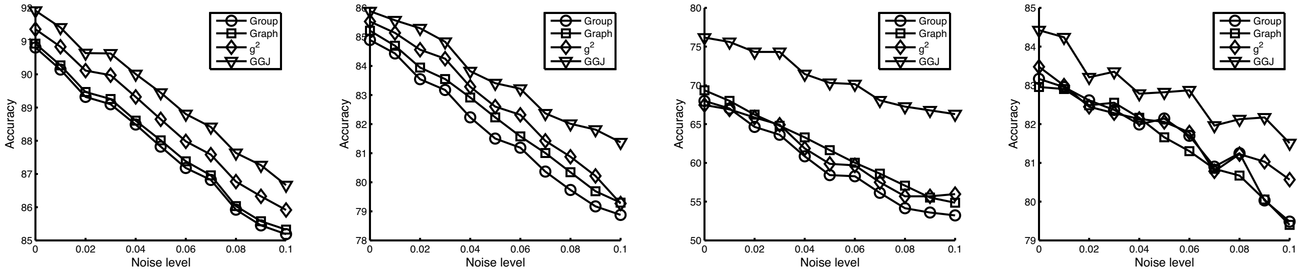
Protein is a dataset extracted from the databases Yeast[1] and STRING[2]. We are to predict the functions of proteins.

[1] http://genomics.stanford.edu

[2] http://string-db.org/

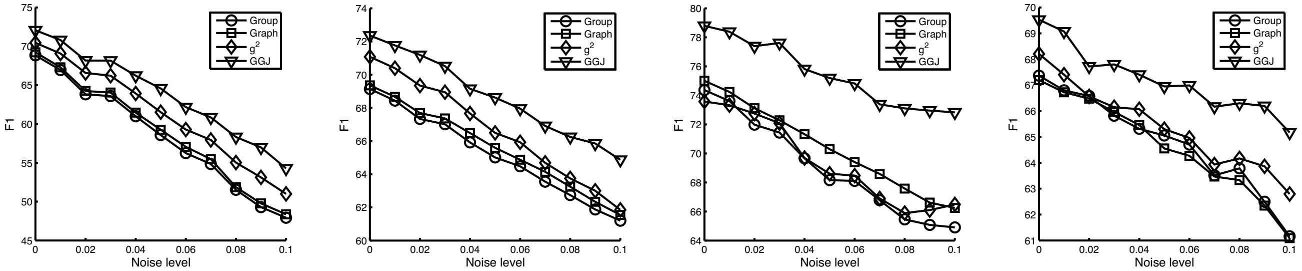
Table 3: *F1* results on the four datasets (%)

	Cora	**Twitter**	**Gene**	**Protein**
Group	68.8353	69.1360	74.3656	67.3633
Graph	69.2303	69.3564	75.0093	67.1941
g^2	70.4479	71.0746	73.5754	68.2048
GGJ	**72.0636**	**72.3564**	**78.7931**	**69.5269**



(a) Cora dataset (b) Twitter dataset (c) Gene dataset (d) Protein dataset

Figure 2: Accuracy results of different structured sparsity models on the four datasets with varying noise level.



(a) Cora dataset (b) Twitter dataset (c) Gene dataset (d) Protein dataset

Figure 3: *F1* results of different structured sparsity models on the four datasets with varying noise level.

Function labels are extracted from Yeast and protein interactions are extracted from STRING. All the combinations of 3 amino acids out of 20 are used to represent the proteins.

We compare our regularizer with three existing regularizers: group sparsity, graph sparsity and the latest group-graph (g^2) sparsity. Performance metrics are *Accuracy* and *F1*. *F1* is a combined measure of *precision* and *recall*, which is defined as $F1$=2*$precision$*$recall$/($precision$+$recall$). For each dataset, we perform 10-fold cross-validation by separating the dataset into 10 subsets and each time we draw 1 subset out as test samples and then average the performance. For group sparsity, graph sparsity and our GGJ sparsity, the structures are directly adopted from classes and links. For g^2 sparsity, the groups are the classes and we connect two groups if the number of links between them is $> 25\%$ times that of the corresponding complete graph.

Classification performance We first evaluate our regularizer on the four datasets and compare it with the other three models. The accuracy and F1 of each method on each dataset are shown in Table 2 and Table 3 respectively. We can see that our model clearly outperforms the other three models. The reason is twofold: 1) group/graph sparsity makes use of only group/graph structure, i.e., label/link information; 2) Though both g^2 sparsity and GGJ sparsity explore group and graph structures simultaneously, the former uses the graph structure of groups based on some assumption that is not applicable to all data, while the latter uses directly the group and graph structures of samples. Note that linked data usually have no explicit graph structure of groups, while inferring graph structure of groups from sample links cannot keep all useful information, thus is not reliable.

Robustness against noisy labels We then test the robustness of our regularizer and the other three models on the four datasets. We add noise to the labels of training samples, because in the real world, labels are often annotated manually and might be inconsistent because different persons participate the work. We uniformly select 1% to 10% training samples and give them random labels, and present the performance in Fig. 2 and Fig. 3. Though the curves of all the four models show down trends, the performance of our model keeps the best and decreases slower than the other three models. Group sparsity and g^2 sparsity do not work well because the noisy labels impair the consistency with the underlying group/graph structure of samples/groups. The performance of graph sparsity is neither good because it selects many samples with unreliable labels and thus the predictions are not trustable. Our regularizer selects much sparser samples where there are fewer samples with unreliable labels, due to the sparse constraint on border edges.

Conclusion

In this paper, we propose a group & graph joint sparse regularizer for effectively classifying linked data where both group structure and graph structure exist. The regularizer finds a small number of connected components while requiring those connected components contain as few border edges as possible. Both theoretical analysis and experimental results on four datasets validate the advantage of our model over three existing structured sparsity models.

Acknowledgements

This work was partially supported by National Natural Science Foundation of China (NSFC) under grant No. 61272380 and the Key Projects of Fundamental Research Program of Shanghai Municipal Commission of Science and Technology under grant No. 14JC1400300 .

References

Bach, F.; Jenatton, R.; Mairal, J.; and Obozinski, G. 2012. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning* 4(1):1–106.

Baldassarre, L.; Morales, J.; Argyriou, A.; and Pontil, M. 2012. A general framework for structured sparsity via proximal optimization. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS'12)*, 82–90.

Bertsimas, D., and Tsitsiklis, J. N. 1997. *Introduction to linear optimization*. Athena Scientific Belmont.

Dai, X.-Y.; Zhang, J.-B.; Huang, S.-J.; Chen, J.-J.; and Zhou, Z.-H. 2015. Structured sparsity with group-graph regularization. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15)*, 1714–1720.

Duan, Y.; Wei, F.; Zhou, M.; and Shum, H.-Y. Graph-based collective classification for tweets. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM'12)*.

Gao, L., and Zhou, S. 2014. Towards Topological-Transformation Robust Shape Comparison: A Sparse Representation Based Manifold Embedding Approach. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI 2014)*, 2753–2759.

Gao, L., and Zhou, S. 2015. Learning Sparse Representations from Datasets with Uncertain Group Structures: Model, Algorithm and Applications. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, 2603–2609.

Huang, J. Z., and Zhang, T. 2009. The benefit of group sparsity. *Annals of Statistics* 38(4):1978–2004.

Huang, J.; Zhang, T.; and Metaxas, D. 2009. Learning with structured sparsity. *Journal of Machine Learning Research* 12(7):3371–3412.

Jacob, L.; Obozinski, G.; and Vert, J.-P. 2009. Group lasso with overlap and graph lasso. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML'09)*, 433–440. ACM.

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

Li, Y., and Ngom, A. 2012. Fast sparse representation approaches for the classification of high-dimensional biological data. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM'12)*, 1–6. IEEE.

Majumdar, A., and Ward, R. 2009. Classification via group sparsity promoting regularization. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'09)*, 861 –864.

Mccallum, A. K.; Nigam, K.; Rennie, J.; and Seymore, K. 2000. Automating the construction of internet portals with machine learning. In *Information Retrieval*, 127–163.

Micchelli, C. A.; Shen, L.; and Xu, Y. 2011. Proximity algorithms for image models: denoising. *Inverse Problems* 27(4):045009.

Moreau, J.-J. 1962. Fonctions convexes duales et points proximaux dans un espace hilbertien. *CR Acad. Sci. Paris Sér. A Math* 255:2897–2899.

Sainath, T. N.; Maskey, S.; Kanevsky, D.; Ramabhadran, B.; Nahamoo, D.; and Hirschberg, J. 2010. Sparse Representations for Text Categorization. In *Proceedings of 11st Annual Conference of the International Speech Communication Association (INTERSPEECH'10)*, 2266–2269.

Wright, J.; Yang, A.; Ganesh, A.; Sastry, S.; and Ma, Y. 2009. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31(2):210 –227.

Yuan, L.; Woodard, A.; Ji, S.; Jiang, Y.; Zhou, Z.-H.; Kumar, S.; and Ye, J. 2012. Learning sparse representations for fruit-fly gene expression pattern image annotation and retrieval. *BMC Bioinformatics* 13(1):107.

Zhao, P.; Rocha, G.; and Yu, B. 2006. Grouped and hierarchical model selection through composite absolute penalties. *Department of Statistics, UC Berkeley, Tech. Rep* 703.