Higher Order Fused Regularization for Supervised Learning with Grouped Parameters

Koh Takeuchi[1(✉)], Yoshinobu Kawahara[2], and Tomoharu Iwata[1]

[1] NTT Communication Science Laboratories, Kyoto, Japan
{takeuchi.koh,iwata.tomoharu}@lab.ntt.co.jp
[2] The Institute of Scientific and Industrial Research (ISIR),
Osaka University, Osaka, Japan
ykawahara@sanken.osaka-u.ac.jp

Abstract. We often encounter situations in supervised learning where there exist possibly groups that consist of more than two parameters. For example, we might work on parameters that correspond to words expressing the same meaning, music pieces in the same genre, and books released in the same year. Based on such auxiliary information, we could suppose that parameters in a group have similar roles in a problem and similar values. In this paper, we propose the Higher Order Fused (HOF) regularization that can incorporate smoothness among parameters with group structures as prior knowledge in supervised learning. We define the HOF penalty as the Lovász extension of a submodular higher-order potential function, which encourages parameters in a group to take similar estimated values when used as a regularizer. Moreover, we develop an efficient network flow algorithm for calculating the proximity operator for the regularized problem. We investigate the empirical performance of the proposed algorithm by using synthetic and real-world data.

1 Introduction

Various regularizers for supervised learning have been proposed, aiming at preventing a model from overfitting and at making estimated parameters more interpretable [1,3,16,30,31]. Least absolute shrinkage and selection operator (Lasso) [30] is one of the most well-known regularizers that employs the ℓ_1 norm over a parameter vector as a penalty. This penalty enables a sparse estimation of parameters that is robust to noise in situations with high-dimensional data. However, Lasso does not explicitly consider relationships among parameters. Recently, *structured* regularizers have been proposed to incorporate auxiliary information about structures in parameters [3]. For example, the Fused Lasso proposed in [31] can incorporate the smoothness encoded with a similarity graph defined over the parameters into its penalty.

While such a graph representation is useful to incorporate information about pairwise interactions of variables (i.e. the second-order information), we often encounter situations where there exist possibly overlapping groups that consist of more than two parameters. For example, we might work on parameters that

A. Appice et al. (Eds.): ECML PKDD 2015, Part I, LNAI 9284, pp. 577–593, 2015.
DOI: 10.1007/978-3-319-23528-8_36

correspond to words expressing the same meaning, music pieces in the same genre, and books released in the same year. Based on such auxiliary information, we naturally suppose that a group of parameters would provide similar functionality in a supervised learning problem and thus take similar values.

In this paper, we propose Higher Order Fused (HOF) regularization that allows us to employ such prior knowledge about the similarity on groups of parameters as a regularizer. We define the HOF penalty as the Lovász extension of a submodular higher-order potential function, which encourages parameters in a group to take similar estimated values when used as a regularizer. Our penalty has effects not only on such variations of estimated values in a group but also on supports over the groups. That is, it could detect whether a group is effective for a problem, and utilize only effective ones by solving the regularized estimation. Moreover, our penalty is robust to noise of the group structure because it encourages an effective part of parameters within the group to have the same value and allows the rest of the parameters to have different estimated values.

The HOF penalty is defined as a non-smooth convex function. Therefore, a forward-backward splitting algorithm [7] can be applied to solve the regularized problem with the HOF penalty, where the calculation of a proximity operator [22] is a key for the efficiency. Although it is not straightforward to develop an efficient way of solving the proximity operator for the HOF penalty due to its inseparable form of the HOF penalty, we develop an efficient network flow algorithm based on [14] for calculating the proximity operator.

Note that Group Lasso (GL) [34] is also known as a class of regularizers to use explicitly a group structure of parameters. However, while our HOF penalty encourages the smoothness over parameters in a group, GL imposes parameters to be sparse in a group-wise manner.

In this paper, we conduct experiments on regression with both synthetic and real-world data. In the experiments with the synthetic data, we investigate the comparative performance of our method on two settings of overlapping and non-overlapping groups. In the experiments with the real-world data, We first test the predictive performances about the average rating of each item (such as movie and book) from a set of users who watched or read items, given user demographic groups. And then, we confirm the predictive performance on a rating value from a review text given semantic and positive-negative word groups.

The rest of this paper is organized as below. In Section 2, we introduce regularized supervised learning and the forward-backward splitting algorithm. In Section 3, we propose Higher Order Fused regularizer. In Section 4, we derive a efficient flow algorithm for solving the proximity operator of HOF. In Section 5, we review related work of our method. In Section 6, we conduct experiments to compare our methods and existing regularizers. We conclude this paper and discuss future work in Section 6.

2 Regularized Supervised Learning

We denote the number of observations as N and the number of variables as M. An observed sample is denoted as $\{y_n, \boldsymbol{x}_n\}$ where $y_n \in \mathcal{Y}$ is a target value

Algorithm 1. Forward-backward splitting algorithm with Nesterov's acceleration

```
Initialize $\boldsymbol{\beta}_0 \in \mathbb{R}^d$, set $\boldsymbol{\zeta}_0 = \boldsymbol{\beta}_0$ and $\eta_0 = 1$.
for $t = 0, 1, \cdots$ do
    $\hat{\boldsymbol{\beta}}_t = \boldsymbol{\zeta}_t - L^{-1}\nabla l(\boldsymbol{\zeta}_t)$.
    $\boldsymbol{\beta}_{t+1} = \mathrm{prox}_{L^{-1}\Omega}\,\hat{\boldsymbol{\beta}}_t$.
    $\eta_{t+1} = (1 + \sqrt{4\eta_t^2 + 1})/2$.
    $\lambda_t = 1 + (\eta_t - 1)/\eta_{t+1}$.
    $\boldsymbol{\zeta}_{t+1} = \boldsymbol{\beta}_t + \lambda_t(\boldsymbol{\beta}_{t+1} - \boldsymbol{\beta}_t)$.
end for
```

and $\boldsymbol{x}_n = (x_1, x_2, \cdots, x_M) \in \mathbb{R}^M$ is an explanatory variable vector. We denote a parameter vector as $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_d) \in \mathbb{R}^d$ where d is the total number of parameters. An object function of regularized supervised learning problem is: $\mathcal{L}(\boldsymbol{\beta}) = \frac{1}{N}\sum_{n=1}^{N} l(\boldsymbol{\beta}; y_n, \boldsymbol{x}_n) + \gamma\Omega(\boldsymbol{\beta})$, where $l(\boldsymbol{\beta}; y_n, \boldsymbol{x}_n) : \mathbb{R}^d \to \mathbb{R}$ is an empirical risk, $\Omega(\boldsymbol{\beta}) : \mathbb{R}^d \to \mathbb{R}$ is a regularizer, and γ is a hyper parameter of the regularizer. A problem of supervised learning attains a solution: $\arg\min_{\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{\beta})$. This formulation includes well-known regularized supervised learning problems such as Lasso, logistic regression [17], elastic net [36], and SVM [28].

When l is a differentiable convex function where its gradient ∇l is L-Lipschitz continuous , i.e.,

$$\left(\forall(\boldsymbol{\beta}, \hat{\boldsymbol{\beta}}) \in \mathbb{R}^d \times \mathbb{R}^d\right) \ \ \|\nabla l(\boldsymbol{\beta}) - \nabla l(\hat{\boldsymbol{\beta}})\|_2^2 \le L\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\|_2^2, \tag{1}$$

where $L \in (0, +\infty)$. And Ω is a lower semicontinuous function whose proximity operator is provided, a minimization problem of $\mathcal{L}$ can be solved by employing the forward-backward splitting algorithm[6,7]. Its solutions are characterized by the fixed point equation.

$$\boldsymbol{\beta} = \mathrm{prox}_{\gamma\Omega}\big(\boldsymbol{\beta} - \gamma\nabla l(\boldsymbol{\beta})\big), \tag{2}$$

where $\mathrm{prox}_{\gamma\Omega} : \mathbb{R}^d \to \mathbb{R}^d$ is a proximity operator [6,22] for Ω and $\gamma \in (0, +\infty)$. The proximity operator utilizes the Moreau envelope [22] of the regularizer ${}^{\gamma}\Omega : \mathbb{R}^d \to \mathbb{R} : \hat{\boldsymbol{\beta}} \to \min_{\boldsymbol{\beta}} \Omega(\boldsymbol{\beta}) + 1/2\gamma\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\|_2^2$, whose gradient is $1/\gamma$-Lipschitz continuous [5]. The forward-backward splitting algorithm is also known to the proximal gradient method. The convergence of the forward-backward splitting algorithm can achieve $O(1/t^2)$ rate by utilizing Nesterov's acceleration [25,26] (the same idea is also proposed in FISTA [4]), where t is the number of iteration counts, see Algorithm 1.

3 Higher Order Fused Regularizer

In this section, we define Higher Order Fused (HOF) regularizer through the Lovász extension of the higher order potential function, called the robust P^n potential function, and discuss the sparsity property in supervised learning with the HOF penalty.

3.1 Review of Submodular Functions and Robust P^n Potential

Let $V = \{1, 2, \dots, d\}$. A set function $f : 2^V \rightarrow \mathbb{R}$ is called *submodular* if it satisfies:

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T), \tag{3}$$

for any $S, T \subseteq V$ [8]. A submodular function is known to be a counterpart of a convex function, which is described through a continuous relaxation of a set function called *the Lovász extension*. The Lovász extension $\hat{f} : \mathbb{R}^V \rightarrow \mathbb{R}$ of a set function f is defined as:

$$\hat{f}(\boldsymbol{\beta}) = \sum_{i=1}^{d} \beta_{j_i} \left(f(\{j_1, \dots, j_i) - f(\{j_1, \dots, j_{i-1})) \right), \tag{4}$$

where $j_1, j_2, \dots, j_d \in V$ are the distinct indices corresponding to a permutation that arranges the entries of $\boldsymbol{\beta}$ in non increasing order, i.e., $\beta_{j_1} \geq \beta_{j_2} \geq \cdots \geq \beta_{j_d}$. It is known that a set function f is submodular if and only if its Lovász extension $\hat{f}$ is convex [21]. For a submodular function f with $f(\emptyset) = 0$, *the base polyhedron* is defined as:

$$B(f) = \{\mathbf{x} \in \mathbb{R}^V \mid \mathbf{x}(S) \leq f(S)\ (\forall S \subseteq V), \mathbf{x}(V) = f(V)\}. \tag{5}$$

Many problems in computer vision are formulated as the energy minimization problem, where a graph-cut function is often used as the energy for incorporating *the smoothness* in an image. A graph-cut function is known to be almost equivalent to a second order submodular function [13] (i.e., it represents a relationship between two nodes). Meanwhile, recently several higher order potentials have been considered for taking into account the smoothness among more than two. For example, Kohli et al.[18] propose the robust P^n model, which can be minimized efficiently with a network flow algorithm. Let us denote a group of indices as $g \subset V$ and a set of groups as $\mathcal{G} = \{g_1,\ g_2, \cdots, g_K\}$, where K is the number of groups. We denote hyper parameters that are weights of parameters in the k-th group as:

$$\boldsymbol{c}_0^k, \boldsymbol{c}_1^k \in \mathbb{R}^d_{\geq 0},\ c_{0,i}^k = \begin{cases} c_{0,i}^k & \text{if } i \in g_k, \\ 0 & \text{otherwise} \end{cases},\ c_{1,i}^k = \begin{cases} c_{1,i}^k & \text{if } i \in g_k, \\ 0 & \text{otherwise} \end{cases},\ (i \in V). \tag{6}$$

The potential can be represented in the form of a set function as:

$$f_{\text{ho}}(S) = \sum_{k=1}^{K} \min\left(\theta_0^k + \boldsymbol{c}_0^k(V \setminus S),\ \theta_1^k + \boldsymbol{c}_1^k(S),\ \theta_{\max}^k\right), \tag{7}$$

where θ_0^k, θ_1^k and $\theta_{\max}^k \in \mathbb{R}_{\geq 0}$ are hyper parameters for controlling consistency of estimated parameters in the k-th group that satisfy $\theta_{\max}^k \geq \theta_0^k$, $\theta_{\max}^k \geq \theta_1^k$ and, for all $S \subset V$, $(\theta_0^k + \boldsymbol{c}_0^k(V \setminus S) \geq \theta_{\max}^k) \vee (\theta_1^k + \boldsymbol{c}_1^k(S) \geq \theta_{\max}^k) = 1$.

3.2 Definition of HOF Penalty

As mentioned in [2,32], Generalized Fused Lasso (GFL) can be obtained as the Lovász extension of a graph-cut function. This penalty, used in supervised learning, prefers parameters that take similar values if a pair of them are adjacent on a given graph, which is a similar structured property to a graph-cut function as an energy function. Now, based on an analogy with this relationship between GFL and a graph-cut function, we define our HOF penalty, which encourages parameters in a groups to take similar values, using the structural property of the higher order potential Eq. (7).

Suppose that a set of groups is given as described in the previous section. Then, we define the HOF penalty as the Lovász extension of the higher order potential Eq. (7), which is described as:

$$\Omega_{\mathrm{ho}}(\boldsymbol{\beta}) = \sum_{k=1}^{K} \Bigg(\sum_{i \in \{j_1,\ldots,j_{s-1}\}} (\beta_i - \beta_{j_s}) c_{1,i}^k + \beta_{j_s}(\theta_{\max}^k - \theta_1^k) \\ + \beta_{j_t}(\theta_0^k - \theta_{\max}^k) + \sum_{i \in \{j_{t+1},\cdots,j_d\}} (\beta_{j_t} - \beta_i) c_{0,i}^k \Bigg), \tag{8}$$

where $\boldsymbol{c}_0^k, \boldsymbol{c}_1^k, \theta_0^k, \theta_1^k, \theta_{\max}^k$ correspond to the ones in Eq. (7) and,

$$j_s^k = \min \left\{ j' \mid \theta_1^k + \textstyle\sum_{i \in \{j_1,\cdots,j'\}} c_{1,i}^k \geq \theta_{\max}^k \right\}, \\ j_t^k = \min \left\{ j' \mid \theta_0^k + \textstyle\sum_{i \in \{j',\cdots,j_d\}} c_{0,i}^k < \theta_{\max}^k \right\}. \tag{9}$$

The first term in Eq. (8) enforces parameters larger than β_{j_s} to have the same value of β_{j_s}. The second and third terms can be rewritten as $\theta_{\max}^k(\beta_{j_s} - \beta_{j_t}) - \beta_{j_s}\theta_1^k + \beta_{j_t}\theta_0^k$. $\theta_{\max}^k(\beta_{j_s} - \beta_{j_t})$ enforces all of parameters between β_{j_s} and β_{j_t} to have the same value because parameters are sorted by the decreasing order and $\beta_{j_s} = \beta_{j_t}$ can be satisfied if and only if all parameters between β_{j_s} and β_{j_t} have the same estimated value (see an example of parameters between s and t in Figure 1(b)). $-\beta_{j_s}\theta_1^k + \beta_{j_t}\theta_0^k$ encourages β_{j_s} and β_{j_t} to have larger and smaller estimated values, respectively. The fourth term enforces parameters smaller than β_{j_t} to have the same value of β_{j_t}. The HOF penalty is robust to noise of the group structure because it allows parameters outside of $(\beta_{j_s}, \cdots, \beta_{j_t})$ to have different estimated values and then it utilizes only an effective part of the group and discard the others.

Proposition 1. *$\Omega_{\mathrm{ho}}(\boldsymbol{\beta})$ is the Lovász extension of the higher order potential Eq. (7).*

Proof. We denote $U_i = \{j_1, \ldots, j_i\}$ and $f_{\mathrm{ho}}^k(U_i) = \min \left(\theta_0^k + \boldsymbol{c}_0^k(V \setminus S),\ \theta_1^k + \boldsymbol{c}_1^k(S),\ \theta_{\max}^k\right)$, then,

$$f_{\mathrm{ho}}^k(U_i) = \begin{cases} \theta_1^k + \boldsymbol{c}_1^k(U_i) & (1 \leq i < s) \\ \theta_{\max}^k & (s \leq i < t) \\ \theta_0^k + \boldsymbol{c}_0^k(V \setminus U_i) & (t \leq i \leq d) \end{cases}, \tag{10}$$

and hence,

$$\beta_{j_i}\left(f_{\text{ho}}^k(U_i) - f_{\text{ho}}^k(U_{i-1})\right) = \begin{cases} \beta_{j_i}\boldsymbol{c}_1^k(\{j_i\}) & (1 \le i < s) \\ \beta_{j_s}\left(\theta_{\max}^k - (\theta_1^k + \boldsymbol{c}_1^k(U_{s-1}))\right) & (i = s) \\ 0 & (s < i < t) \\ \beta_{j_t}\left(\theta_0^k + \boldsymbol{c}_0^k(V \setminus U_t) - \theta_{\max}^k\right) & (i = t) \\ -\beta_{j_i}\boldsymbol{c}_0^k(\{j_i\}) & (t < i \le d) \end{cases}, \quad (11)$$

where $\boldsymbol{c}_1^k(U_i) = \sum_{i\in\{j_1,\cdots,j_i\}} c_{1,i}^k$ and $\boldsymbol{c}_0^k(V \setminus U_i) = \sum_{i\in\{j_{i+1},\cdots,j_d\}} c_{0,i}^k$. As a result, we have $\Omega_{\text{ho}}(\boldsymbol{\beta})$ by summing all of these from the definition of the Lovász extension Eq. (4).

Although the penalty $\Omega_{\text{ho}}(\boldsymbol{\beta})$ includes many hyper parameters (such as $\boldsymbol{c}_0^k, \boldsymbol{c}_1^k$, θ_0^k, θ_1^k and $\theta_{\max}^k$), it would be convenient to use the same value for $\theta_0^k, \theta_0^k, \theta_{\max}^k$ for different $g \in \mathcal{G}$ and constant values for non-zero elements in $\boldsymbol{c}_0^k$ and $\boldsymbol{c}_1^k$, respectively, in practice. We show an example of Eq. (10) in Figure 1(a), and parameters that minimizes the potential in Figure 1(b). As described in [1], the Lovász extension of a submodular function with $f(\emptyset) = f(V) = 0$ has the sparsity effects not only on the support of $\boldsymbol{\beta}$ but also on all sup-level set $\{\boldsymbol{\beta} \ge \alpha\}$ $(\alpha \in \mathbb{R})$.[1] A necessary condition for $S \subseteq V$ to be inseparable for the function $g : A \to f_{\text{ho}}(S \cup A) - f_{\text{ho}}(S)$ is that S is a set included in some unique group g_k. Thus, Ω_{ho} as a regularizer has an effect to encourage the values of parameters in a group to be close.

4 Optimization

4.1 Proximity Operator via Minimum-Norm-Point Problem

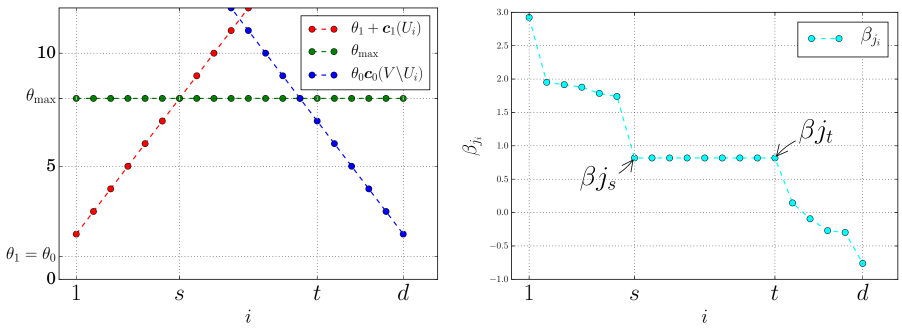
From the definition, the HOF penalty belongs to the class of the lower semicontinuous convex function but is non-smooth. To attain a solution of the penalty, we define the proximity operator as:

$$\text{prox}_{\gamma\Omega_{\text{ho}}} \hat{\boldsymbol{\beta}} = \mathop{\arg\min}_{\boldsymbol{\beta}\in\mathbb{R}^d} \Omega_{\text{ho}}(\boldsymbol{\beta}) + \frac{1}{2\gamma}\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2^2, \quad (12)$$

and we denote a solution of the proximity operator $\text{prox}_{\gamma\Omega_{\text{ho}}} \hat{\boldsymbol{\beta}}$ as $\boldsymbol{\beta}^*$. By plugging $\Omega_{\text{ho}}(\boldsymbol{\beta}) = \max_{\boldsymbol{s}\in B(f_{\text{ho}})} \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{s}$ [11] into Eq. (12), the proximity operator can be shown as the following minimization problem on a base polyhedron [32].

$$\begin{aligned} &\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \Omega_{\text{ho}}(\boldsymbol{\beta}) + \frac{1}{2\gamma}\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2^2 = \min_{\boldsymbol{\beta}} \max_{\boldsymbol{s}\in B(f_{\text{ho}})} \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{s} + \frac{1}{2\gamma}\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2^2 \\ &= \max_{\boldsymbol{s}\in B(f_{\text{ho}})} -\frac{1}{2}\|\boldsymbol{s} - \gamma^{-1}\hat{\boldsymbol{\beta}}\|_2^2 + \frac{1}{2\gamma}\|\hat{\boldsymbol{\beta}}\|_2^2 \ \left(\because \ \arg\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{s} + \tfrac{1}{2\gamma}\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2^2 = \boldsymbol{\beta} - \gamma\boldsymbol{s}\right) \\ &\leftrightarrow \min_{\boldsymbol{s}\in B(f_{\text{ho}})} \|\boldsymbol{s} - \gamma^{-1}\hat{\boldsymbol{\beta}}\|_2^2. \end{aligned} \quad (13)$$

[1] The higher order potential $f_{\text{ho}}(S)$ can be always transformed by excluding the constant terms θ_0 and θ_1 and by accordingly normalizing $\mathbf{c}_0$ and $\mathbf{c}_1$ respectively.



(a) values of the set functions

(b) parameters sorted by the decreasing order

Fig. 1. (**a**) An example of f_{ho} where $K = 1$, $c_{1,i} = c_{0,i} = 1$ $(i \in V)$, $\theta_1 = \theta_0 = 1$, and $\theta_{\max} = 8$. The horizontal and vertical axes correspond to the index of parameters and values of set functions, respectively. Red, Green, and Blue lines correspond to each lines in Eq. (10), respectively. (**b**) Parameters $\boldsymbol{\beta}$ sorted by the decreasing order. The horizontal and vertical axes correspond to the sorted index and values of parameters, respectively.

Let $\boldsymbol{t} = \boldsymbol{s} - \gamma^{-1}\hat{\boldsymbol{\beta}}$ and, with the basic property of the base polyhedron of a submodular function, the proximity operator goes equal to a minimal point problem,

$$\min_{\boldsymbol{s} \in B(g)} \|\boldsymbol{s} - \gamma^{-1}\hat{\boldsymbol{\beta}}\|_2^2 = \min_{\boldsymbol{t} \in B(f_{\text{ho}} - \gamma^{-1}\hat{\boldsymbol{\beta}})} \|\boldsymbol{t}\|_2^2. \tag{14}$$

From the derivation, it follows that $\boldsymbol{\beta}^* = -\gamma \boldsymbol{t}^*$ where $\boldsymbol{t}^*$ is the solution of Eq. (14).

In general, the problem in Eq. (14) can be solved with submodular minimization algorithms including Minimum-Norm-Point (MNP) algorithm proposed by [12]. However, the time complexity of the fastest algorithm among existing submodular minimization algorithms is $O(d^5 EO + d^6)$, where EO is a cost for evaluating the function. Therefore, those algorithm are infeasible when the size of parameters d is large.

4.2 Network Flow Algorithm

We utilize a parametric property of MNP problem to solve the problem in Eq. (14). With this property, we can apply a parametric flow algorithm that attains the exact solution of the problem more efficiently than existing submodular minimization algorithms.

The set function $h(S) = f_{\text{ho}}(S) - \hat{\boldsymbol{\beta}}(S)$ in Eq. (14) is submodular because the sum of a submodular and modular functions are submodular [11]. Therefore, Eq. (14) is a special case of a minimization problem of a separable convex function under submodular constraints [23] that can be solved via parametric optimization. We denote a parameter $\alpha \in \mathbb{R}_{\geq 0}$ and define a set function $h_\alpha(S) = h(S) - \alpha \mathbf{1}(S)$, $(\forall S \subset V)$, where $\mathbf{1}(S) = \sum_{i \in S} 1$. When h is non-decreasing submodular function, there exists a set of $r+1$ $(\leq d)$ subsets: $S^* = \{S_0 \subset S_1 \subset \cdots \subset S_r\}$, where $S_j \subset V$, $S_0 =$, and $S_r = V$, respectively. And there are $r+1$ subintervals Q_r of α: $Q_0 = [0, \alpha_0), Q_1 = [\alpha_1, \alpha_2), \cdots, Q_r = [\alpha_r, \infty)$, such that, for each $j \in \{0, 1, \cdots, r\}$, S_j is the unique maximal minimizer of $h_\alpha(S), \forall \alpha \in Q_j$ [23]. The optimal minimizer of Eq. (14) $\boldsymbol{t}^* = (t_1^*, t_2^*, \cdots, t_d^*)$ is then determined as:

$$t_i^* = \frac{f_{\text{ho}}(S_{j+1}) - f_{\text{ho}}(S_j)}{\mathbf{1}(S_{j+1} \setminus S_j)}, \ \forall i \in (S_{j+1} \setminus S_j), \ j = (1, \cdots, r). \tag{15}$$

We introduce two lemmas from [24] to ensure that h is a non-decreasing submodular function.

Lemma 1. *For any $\eta \in \mathbb{R}$ and a submodular function h, $\boldsymbol{t}^*$ is an optimal solution to $\min_{\boldsymbol{t} \in \mathcal{B}(h)} \|\boldsymbol{t}\|_2^2$ if and only if $\boldsymbol{t}^* - \eta \mathbf{1}$ is an optimal solution to $\min_{\boldsymbol{t} \in \mathcal{B}(h) + \eta \mathbf{1}} \|\boldsymbol{t}\|_2^2$.*

Lemma 2. *Set $\eta = \max_{i=1,\cdots,d}\{0, h(V \setminus \{i\}) - h(V)\}$, then $h + \eta \mathbf{1}$ is a non-decreasing submodular function.*

With Lemma 2, we solve

$$\min_{S \subset V} f_{\text{ho}}(S) - \hat{\boldsymbol{\beta}}(S) + (\eta - \alpha)\mathbf{1}(S), \tag{16}$$

and then apply Lemma 1 to obtain a solution of the original problem. Because Eq. (16) is a specific form of a min cut problem, we can be solved the problem efficiently.

Theorem 1. *Problem in Eq. (16) is equivalent to a minimum s/t-cut problem defined as in Figure. 2.*

Proof. The cost function in Eq. (16) is a sum of a modular and submodular functions, because the higher order potential can be transformed as a second order submodular function. Therefore, this cost function is a $\mathcal{F}^2$ energy function [19] that is known to be "graph-representative". In Figure. 2, the groups of parameters are represented with hyper nodes u_1^k, u_0^k that correspond to each group, and capacities of edges between hyper nodes and ordinal nodes $v_i \in V$. These structures are not employed in [32]. Edges between source and sink nodes correspond to input parameters like [32]. We can attain a solution of s/t min cut problem via graph cut algorithms. We employ an efficient parametric flow algorithm provided by [14] that run in $O(d|E|\log(d^2/|E|))$ as the worst case, where $|E|$ is the number of edges of the graph in Figure 2.

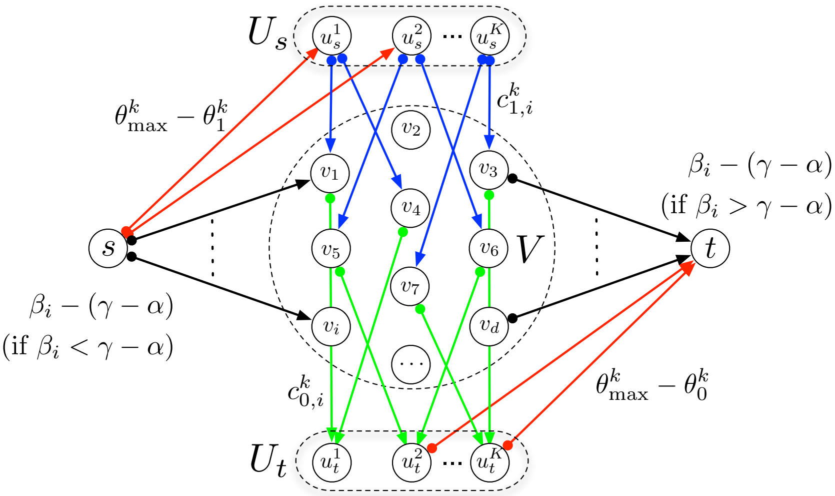

U_s
u_s^1
u_s^2
$\cdots$
u_s^K
$\theta^k_{\max} - \theta^k_1$
$c^k_{1,i}$
v_2
v_1
v_3
v_4
$\beta_i - (\gamma - \alpha)$
(if $\beta_i > \gamma - \alpha$)
s
v_5
v_6
V
t
v_7
$\beta_i - (\gamma - \alpha)$
(if $\beta_i < \gamma - \alpha$)
v_i
v_d
$\cdots$
$c^k_{0,i}$
$\theta^k_{\max} - \theta^k_0$
U_t
u_t^1
u_t^2
$\cdots$
u_t^K

Fig. 2. A minimum s/t-cut problem of Problem 16. Given a graph $G = (V, E)$ for the HOF penalty, capacities of edges are defined as: $c(s, u^k_1) = \theta_{\max} - \theta_1$, $c(u^k_1, v_i) = c^k_{1,i}$, $c(v_i, u^k_0) = c^k_{0,i}$, $c(u^k_0, t) = \theta^k_{\max} - \theta^k_0$, $c(s, v_i) = z_i - (\gamma - \alpha)$ if $z_i > \gamma - \alpha$, and $c(v_i, t) = (\gamma - \alpha) - z_i$ if $z_i < \gamma - \alpha$. Nodes u^k_1 and u^k_0, $k = (1, \cdots, K)$ are hyper nodes that correspond to the groups. And s, t, and v_i are source-node, sink-node, and nodes of parameters, respectively.

5 Related Work

Lasso [30] is one of the most well-known sparsity-inducing reguralizers, which employs a sum of ℓ_1 norm of parameters as a penalty: $\Omega_{\text{Lasso}}(\boldsymbol{\beta}) = \sum_{i=1}^{d} \|\beta_i\|_1$. The penalty is often minimized by the soft-thresholding that is a proximity operator of ℓ_1 norm. Fused Lasso (FL) is a one of the structured regularizers proposed by [31] to utilize similarities of parameters. FL is also known as the total variation [27] in the field of optimization. Generalized Fused Lasso (GFL) is an extension of FL to adopt a graph structure into the structured norm. We denote a similarity between parameters i and j as $w_{i,j} \in \mathbb{R}_{\geq 0}$. Let us denote a set of edges among parameters, whose similarities are not equal to zero as $E = \{(i, j) | w_{i,j} \neq 0\}$. GFL imposes a fused penalty as: $\Omega_{\text{GFL}}(\boldsymbol{\beta}) = \sum_{(i,j)\in E} w_{i,j} \|\beta_i - \beta_j\|_1$. Because the penalty of GFL is not separable, efficient minimization for this penalty is a challenging problem. A flow algorithm for GFL was proposed by [32] that showed significant improvement on a computational time of proximity operator from existing algorithm. The computational time was reduced by transforming the minimization problem into a separable problem under a submodular constraint [23].

Group Lasso was proposed by [34] to impose a group sparsity as a ℓ_1/ℓ_2 norm on grouped parameters. The Group Lasso imposes a group-wise sparsity penalty as: $\Omega_{\text{GL}}(\boldsymbol{\beta}) = \sum_{k=1}^{K} \|\boldsymbol{\beta}(g_k)\|_2^2$. The penalty works as a group-wise Lasso that selects feature groups effective to a problem. Group Lasso has been extended to groups having overlaps, and efficient calculations of the proximity operator were proposed in [15,33,35]. [9] proposed the Sparse Group Lasso (SGL) which combines both the $\ell 1$ norm and $\ell 1/\ell 2$ norm imposed it as a regularizer. Group Lasso was extended to groups having overlap by [35].

6 Experiments

In this section, we compared our proposed method with existing methods on linear regression problems[2] using both synthetic and real-world data. We employed the ordinary least squares (OLS), Lasso[3], Sparse Group Lasso (SGL) [20], and Generalized Fused Lasso (GFL) as comparison methods. We added the ℓ_1 penalty of Lasso to GFL and our proposed method by utilizing a property: $\mathrm{prox}_{\Omega_{\mathrm{Lasso}}+\Omega} = \mathrm{prox}_{\Omega_{\mathrm{Lasso}}} \circ \mathrm{prox}_{\Omega}$ [10]. With GFL, we encoded groups of parameters by constructing cliques that connect edges between whole pairs of parameters in the group.

6.1 Synthetic Data

We conducted regression experiments with arfically synthesized data. We employed two settings in which parameters had different group structures. In the first setting, parameters had five non-overlapping groups. In the second setting, groups were overlapped.

With the first non-overlapping groups setting, we set the true parameters of features within the group to the same value. With the second overlapping groups setting, we set the true parameters of features having no overlap to the same value, and those of features belonging to two groups to a value of either of the two groups. The explanatory variables $x_{n,i}$ were randomly generated with the Gaussian distribution with mean 0 and variance 1. Then, we obtained target values $\boldsymbol{y}$ from the Gaussian distribution where its mean and variance are $\sum_{i=1}^{d} \beta_i x_{n,i}$ and 5, respectively. The size of the feature dimension D was 100 and the number of observed data points N was $30, 50, 70, 100$, and 150. Hyper parameters were selected by 10-fold cross validation. The hyper parameters of regularizers γ were selected from $\{0.0001, 0.001, 0.01, 0.1, 1.0, 10.0\}$. $\theta^k_{\max}$ was selected from $0.01, 0.1$, and 1.0. $c^k_{0,i}$ and $c^k_{1,i}$ were set to have the same value that was selected from 1.0 and 10.0. θ^k_0 and θ^k_1 were set to 0. We employed the following Root Mean Squared Error (RMSE) on the test data to evaluate the performances: $\sqrt{\frac{1}{N}\sum_{n=1}^{N} \|y_n - \hat{y}_n\|_2^2}$.

The results are summarized in Table 1. In the first setting with non-overlapping groups, our proposed method and GFL showed superior performances than SGL, Lasso, and OLS. Errors of our proposed method and GFL were almost similar. The SGL and Lasso fell in low performances since these

[2] Where the number of variables and features are equal ($m = d$).

[3] We used matlab built-in codes of OLS and Lasso.

Table 1. Average RMSE and their standard deviations with synthetic data. Hyper parameters were selected from 10-fold cross validation. Values in bold typeface are statistically better ($p < 0.01$) than those in normal typeface as indicated by a paired t-test.

(a) *non-overlapping* groups

N	Proposed	SGL	GFL	Lasso	OLS
30	$\mathbf{0.58 \pm 0.32}$	174.40 ± 75.60	$\mathbf{0.48 \pm 0.29}$	189.90 ± 74.50	208.80 ± 119.00
50	$\mathbf{0.56 \pm 0.14}$	119.70 ± 40.80	$\mathbf{0.57 \pm 0.14}$	115.30 ± 54.10	260.40 ± 68.80
70	$\mathbf{0.40 \pm 0.19}$	128.10 ± 39.90	$\mathbf{0.40 \pm 0.19}$	125.00 ± 48.10	313.20 ± 42.40
100	$\mathbf{0.47 \pm 0.13}$	120.40 ± 42.00	$\mathbf{0.47 \pm 0.13}$	112.80 ± 45.60	177.10 ± 68.90
150	$\mathbf{0.51 \pm 0.08}$	106.80 ± 22.00	$\mathbf{0.51 \pm 0.08}$	79.40 ± 20.90	1.08 ± 0.13

(b) *overlapping* groups

N	Proposed	SGL	GFL	Lasso	OLS
30	$\mathbf{84.40 \pm 76.40}$	156.20 ± 64.20	173.50 ± 67.30	162.10 ± 97.10	187.70 ± 108.30
50	$\mathbf{40.90 \pm 11.30}$	108.60 ± 43.80	103.20 ± 27.10	122.80 ± 57.70	246.40 ± 70.50
70	$\mathbf{9.95 \pm 9.22}$	119.40 ± 36.20	138.40 ± 54.10	138.80 ± 44.20	317.80 ± 36.60
100	$\mathbf{3.19 \pm 6.15}$	115.70 ± 38.20	149.20 ± 28.90	101.50 ± 37.70	208.50 ± 76.30
150	$\mathbf{0.53 \pm 0.06}$	104.50 ± 15.50	135.30 ± 21.00	12.30 ± 4.93	1.08 ± 0.13

methods had no ability to fuse parameters. In the second setting with overlapping groups, our proposed method showed superior performance than SGL, GFL, Lasso, and OLS. When $N < D$, existing methods suffered from overfitting; however, our proposed method showed small errors even if $N = 30$. GFL showed low performance in this setting because the graph cannot represents groups.

Examples of estimated parameters on an experiment ($N = 30$) are shown in Figures 3 and 4. In this situation ($N < D$), the number of observation was less than the number of features; therefore, the problems of parameter estimation became undetermined system problems. From Figure 3, we confirmed that our proposed method and GFL successfully recovered the true parameters by utilizing the group structure. From Figure 4, we confirmed that our proposed methods were able to recover true parameters with overlapping groups. This is because our proposed method can represent overlapping groups appropriately. GFL fell into an imperfect result because it employed the pairwise representation that cannot describe groups.

6.2 Real-World Data

We conducted two settings of experiments with real-world data sets. With the first setting, we predicted the average rating of each item (movie or book) from a set of users who watched or read items. We used publicly available real-world data provided by MovieLens100k, EachMovie, and Book-Crossing[4]. We utilized a group structure of users, for example; age, gender, occupation and country as auxiliary

[4] http://grouplens.org

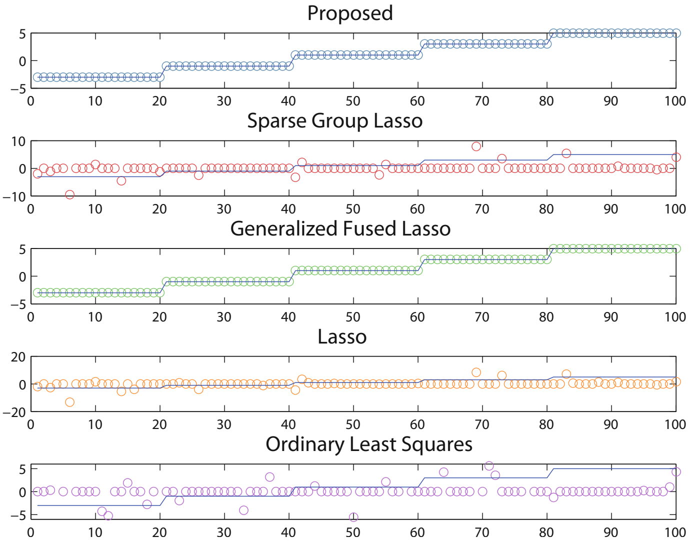


Fig. 3. Estimated parameters from synthetic data with five *non-overlapping* groups. Circles and Blue lines correspond to estimated and true parameter values, respectively.

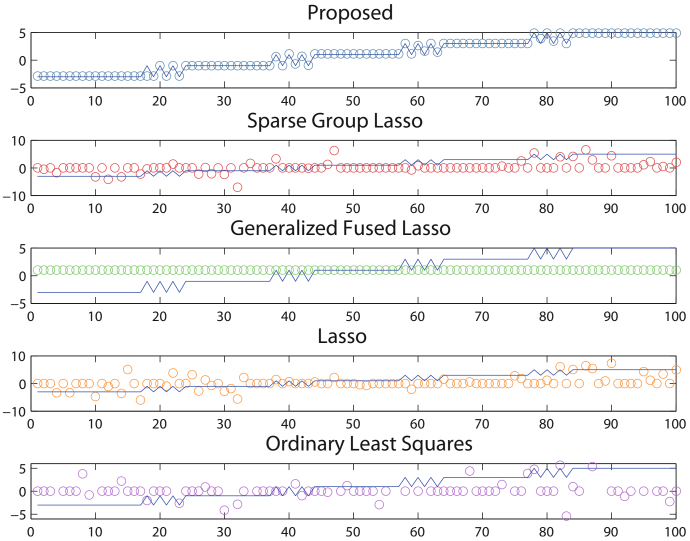


Fig. 4. Estimated parameters from synthetic data with five *overlapping* groups. Circles and Blue lines correspond to estimated and true parameter values, respectively.

information. The MovieLens100k data contained movie rating records with three types groups including ages, genders and occupations. The EachMovie data consisted of movie rating records with two types groups including ages and genders.

Table 2. Summaries of real-world data. $N_{\rm all}, D$ and G correspond to a total number of observations, a dimension of features, and a total number of groups, respectively.

	$N_{\rm all}$	D	K	types of groups
MovieLens100k	$1,620$	942	31	8 age, 2 gender, 21 occupation
EachMovie	$1,623$	$1,000$	21	11 age, 2 gender
Book-Crossing	$1,835$	$1,275$	41	12 age , 29 country

We used the $1,000$ most frequently watching users. The Book-Crossing data was made up of book rating records with two types of groups including ages and countries. We eliminated users and books whose total reading counts were less than 30 from the Book-Crossing data. Summaries of real-world data are shown in Table 2. To check the performance of each method, we changed the number of training data N. $c^k_{0,i}$ and $c^k_{1,i}$ were set to have the same value that was 1.0 if the i-th item belonged to the k-th group or 0.0 otherwise. In each experiment, other hyper parameters were selected by 10-fold cross validation in the same manner as previous experiments.

The results are summarized in Table 3. With the MovieLens100k data, our proposed method showed the best performance on whole settings of N because it was able to utilize groups as auxiliary information for parameter estimations. When $N = 1600$, SGL and GFL also showed competitive performance. With the EachMovie and Book-Crossing data, we confirmed that our proposed model showed the best performance. SGL and Lasso showed competitive performance on some settings of N. With the EachMovie and Book-Crossing data sets, estimated parameters were almost sparse therefore SGL and Lasso showed competitive performance.

Next, we conducted another type of an experiment employing the Yelp data[5]. The task of this experiment was to predict a rating value from a review text. We randomly extracted reviews and used the $1,000$ most frequently occurred words, where stop words were eliminated by using a list[6]. We employed two types of groups of words. We attained 50 semantic groups of words by applying k-means to semantic vectors of words. The semantic vectors were learned form the GoogleNews data by word2vec[7]. We also utilized a positive-negative word dictionary[8] to construct two positive and negative groups of words [29]. Other settings were set to be the same as the MovieLens100k data.

The results are shown in Table 4. We confirmed that our proposed method showed significant improvements over other existing methods with the Yelp data. GFL also showed competitive performance when the number of training data $N = 1,000$. The semantic groups constructed by k-means have no overlap and overlap was only appeared between semantic and positive-negative groups. When

[5] http://www.yelp.com/dataset_challenge
[6] https://code.google.com/p/stop-words
[7] https://code.google.com/p/word2vec/
[8] http://www.lr.pi.titech.ac.jp/~takamura/pndic_en.html

Table 3. Average RMSE and their standard deviations with real-world data sets. Hyper Parameters were selected from 10-fold cross validation. Values in bold typeface are statistically better ($p < 0.01$) than those in normal typeface as indicated by a paired t-test.

(a) MovieLens100k

N	Proposed	SGL	GFL	Lasso	OLS
200	**0.30 ± 0.02**	0.32 ± 0.02	0.33 ± 0.02	1.11 ± 0.71	3.81 ± 1.97
400	**0.28 ± 0.02**	0.32 ± 0.02	0.33 ± 0.02	0.82 ± 0.31	2718.00 ± 6575.00
800	**0.27 ± 0.02**	0.31 ± 0.02	0.33 ± 0.02	0.54 ± 0.21	134144.00 ± 370452.00
1200	**0.27 ± 0.03**	0.32 ± 0.03	0.33 ± 0.03	0.48 ± 0.31	4.19 ± 2.97
1600	**0.27 ± 0.07**	**0.30 ± 0.09**	**0.31 ± 0.09**	0.44 ± 0.45	1.01 ± 0.81

(b) EachMovie

N	Proposed	SGL	GFL	Lasso	OLS
200	**0.86 ± 0.03**	**0.86 ± 0.02**	0.92 ± 0.02	1.24 ± 0.15	2.15 ± 1.17
400	**0.83 ± 0.03**	0.85 ± 0.02	0.90 ± 0.03	1.17 ± 0.09	3.20 ± 1.66
800	**0.81 ± 0.03**	0.84 ± 0.02	0.89 ± 0.03	1.09 ± 0.06	14.30 ± 14.70
1200	**0.80 ± 0.05**	0.84 ± 0.05	0.88 ± 0.05	1.06 ± 0.07	2479.00 ± 9684.00
1500	**0.79 ± 0.09**	**0.83 ± 0.07**	0.87 ± 0.09	1.01 ± 0.12	29.90 ± 29.60

(c) Book-Crossing

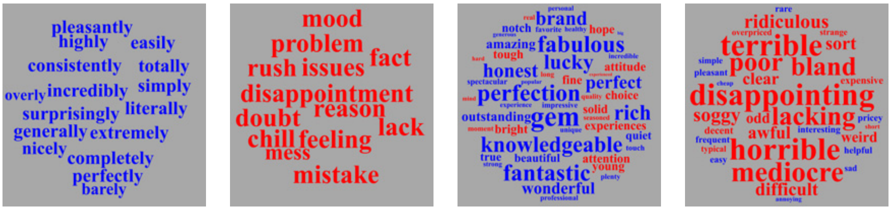
N	Proposed	SGL	GFL	Lasso	OLS
200	**0.71 ± 0.02**	0.73 ± 0.02	0.82 ± 0.02	0.92 ± 0.14	3.98 ± 0.83
400	**0.70 ± 0.02**	**0.72 ± 0.02**	0.82 ± 0.02	0.79 ± 0.03	66.60 ± 109.20
800	**0.68 ± 0.02**	0.70 ± 0.02	0.81 ± 0.02	0.71 ± 0.02	34.00 ± 27.70
1200	**0.67 ± 0.04**	0.71 ± 0.04	0.82 ± 0.04	**0.70 ± 0.03**	551.00 ± 1532.00
1700	**0.64 ± 0.07**	**0.68 ± 0.07**	0.78 ± 0.08	**0.66 ± 0.06**	1.18 ± 0.12

N is small, words having overlap scarcely appeared in review texts. Therefore, GFL showed competitive performance.

We show estimated parameters of four semantic groups in Figure 5. Colors of words corresponded to a sign of an estimated parameter value. Blue corresponds to the plus (positive) value and red corresponds to minus (negative) value. The size of words indicates absolute values of an estimated parameter value. As we have explained in Section 3.2, to make our proposed method robust, our proposed method is designed to allow inconsistency of estimated values within a group. This effect was confirmed by those illustrations. In the top two figures, parameters of words attained almost the same values. On the other hand, in the bottom two figures, parameters of words attained different estimated signs and absolute values. We supposed that the first two semantic groups of words were fitted for this regression problem. Therefore, consistency of estimated values was high. Whereas, the second two semantic groups of words were not fitted, and then resulted in low consistency of estimated values. Those results indicated that our proposed method was able to detect effective groups of words from given overlapping groups with Yelp data.

Table 4. Linear regression problems with Yelp data ($D = 1,000$) and $G = 52$ (50 semantic groups and two positive and negative groups). Means and standard deviations of the loss on the test data are shown. Parameters were selected from 10-fold cross validation. Bold font corresponds to significant difference of t-test ($p < 0.01$).

N	Proposed	SGL	GFL	Lasso	OLS
1000	$\mathbf{1.23 \pm 0.02}$	3.31 ± 0.20	$\mathbf{1.24 \pm 0.01}$	1.62 ± 0.11	135.60 ± 211.00
2000	$\mathbf{1.20 \pm 0.02}$	1.58 ± 0.05	1.23 ± 0.01	1.27 ± 0.02	1.61 ± 0.06
3000	$\mathbf{1.13 \pm 0.02}$	1.34 ± 0.07	1.22 ± 0.01	1.18 ± 0.03	1.35 ± 0.07
5000	$\mathbf{1.10 \pm 0.02}$	1.18 ± 0.03	1.22 ± 0.01	1.12 ± 0.02	1.18 ± 0.03



(a) Positive group. (b) Negative group. (c) Positive dominant group. (d) Negative dominant group.

Fig. 5. Estimated parameters of four semantic groups of words. Blue and Red correspond to plus and minus of estimated parameters, respectively. The size of words correspond to the absolute values of estimated parameters.

7 Conclusion

We proposed a structured regularizer named Higher Order Fused (HOF) regularization in this paper. HOF regularizer exploits groups of parameters as a penalty in regularized supervised learning. We defined the HOF penalty as a Lovaśtz extension of a robust higher order potential named the robust P^n potential. Because the penalty is non-smooth and non-separable convex function, we provided the proximity operator of the HOF penalty. We also derived a flow algorithm to calculate the proximity operator efficiently, by showing that the robust P^n potential is graph-representative. We examined experiments of linear regression problems with both synthetic and real-world data and confirmed that our proposed method showed significantly higher performance than existing structured regularizers. We also showed that our proposed method can incorporate groups properly by utilizing the robust higher-order representation.

We provided the proximity operator of the HOF penalty but only adopted it to linear regression problems in this paper. We can apply the HOF penalty to other supervised or unsupervised learning problems including classification and learning to rank, and also to other applicational fields including signal processing and relational data analysis.

Acknowledgements. This work was partially supported by JSPS KAKENHI Grant Numbers 14435225 and 14500801.

References

1. Bach, F.R.: Structured sparsity-inducing norms through submodular functions. In: Proc. of NIPS, pp. 118–126 (2010)
2. Bach, F.R.: Shaping level sets with submodular functions. In: Proc. of NIPS, pp. 10–18 (2011)
3. Bach, F.R., Jenatton, R., Mairal, J., Obozinski, G.: Structured sparsity through convex optimization. Statistical Science **27**(4), 450–468 (2012)
4. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences **2**(1), 183–202 (2009)
5. Chaux, C., Combettes, P.L., Pesquet, J.C., Wajs, V.R.: A variational formulation for frame-based inverse problems. Inverse Problems **23**(4), 1495 (2007)
6. Combettes, P.L., Pesquet, J.C.: Proximal splitting methods in signal processing. In: Fixed-Point Algorithms for Inverse Problems in Science and Engineering, pp. 185–212. Springer (2011)
7. Combettes, P.L., Wajs, V.R.: Signal recovery by proximal forward-backward splitting. Multiscale Modeling & Simulation **4**(4), 1168–1200 (2005)
8. Edmonds, J.: Submodular functions, matroids, and certain polyhedra. In: Combinatorial Structures and their Applications, pp. 69–87 (1970)
9. Friedman, J., Hastie, T., Tibshirani, R.: A note on the group lasso and a sparse group lasso. arXiv preprint arXiv:1001.0736 (2010)
10. Friedman, J., Hastie, T., Höfling, H., Tibshirani, R., et al.: Pathwise coordinate optimization. The Annals of Applied Statistics **1**(2), 302–332 (2007)
11. Fujishige, S.: Submodular functions and optimization, vol. 58. Elsevier (2005)
12. Fujishige, S., Hayashi, T., Isotani, S.: The minimum-norm-point algorithm applied to submodular function minimization and linear programming. Technical report, Research Institute for Mathematical Sciences Preprint RIMS-1571, Kyoto University, Kyoto, Japan (2006)
13. Fujishige, S., Patkar, S.B.: Realization of set functions as cut functions of graphs and hypergraphs. Discrete Mathematics **226**(1), 199–210 (2001)
14. Gallo, G., Grigoriadis, M.D., Tarjan, R.E.: A fast parametric maximum flow algorithm and applications. SIAM Journal on Computing **18**(1), 30–55 (1989)
15. Jacob, L., Obozinski, G., Vert, J.P.: Group lasso with overlap and graph lasso. In: Proc. of ICML, pp. 433–440 (2009)
16. Jenatton, R., Audibert, J.Y., Bach, F.: Structured variable selection with sparsity-inducing norms. The Journal of Machine Learning Research **12**, 2777–2824 (2011)
17. Koh, K., Kim, S.J., Boyd, S.P.: An interior-point method for large-scale l1-regularized logistic regression. Journal of Machine Learning Research **8**(8), 1519–1555 (2007)
18. Kohli, P., Ladicky, L., Torr, P.H.S.: Robust higher order potentials for enforcing label consistency. International Journal of Computer Vision **82**(3), 302–324 (2009)
19. Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(2), 147–159 (2004)
20. Liu, J., Ji, S., Ye, J.: SLEP: Sparse Learning with Efficient Projections. Arizona State University (2009)

21. Lovász, L.: Submodular functions and convexity. In: Mathematical Programming the State of the Art, pp. 235–257. Springer (1983)
22. Moreau, J.J.: Fonctions convexes duales et points proximaux dans un espace hilbertien. CR Acad. Sci. Paris Sér. A Math. **255**, 2897–2899 (1962)
23. Nagano, K., Kawahara, Y.: Structured convex optimization under submodular constraints. In: Proc. of UAI, pp. 459–468 (2013)
24. Nagano, K., Kawahara, Y., Aihara, K.: Size-constrained submodular minimization through minimum norm base. In: Proc. of ICML, pp. 977–984 (2011)
25. Nesterov, Y.E.: A method of solving a convex programming problem with convergence rate $O(1/k^2)$. Soviet Mathematics Doklady **27**, 372–376 (1983)
26. Nesterov, Y.E.: Smooth minimization of non-smooth functions. Mathematical Programming **103**(1), 127–152 (2005)
27. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena **60**(1), 259–268 (1992)
28. Suykens, J.A., Vandewalle, J.: Least squares support vector machine classifiers. Neural Processing Letters **9**(3), 293–300 (1999)
29. Takamura, H., Inui, T., Okumura, M.: Extracting semantic orientations of words using spin model. In: Proc. of ACL, pp. 133–140. Association for Computational Linguistics (2005)
30. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological), 267–288 (1996)
31. Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., Knight, K.: Sparsity and smoothness via the fused lasso. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **67**(1), 91–108 (2005)
32. Xin, B., Kawahara, Y., Wang, Y., Gao, W.: Efficient generalized fused lasso with its application to the diagnosis of alzheimers disease. In: Proc. of AAAI, pp. 2163–2169 (2014)
33. Yuan, L., Liu, J., Ye, J.: Efficient methods for overlapping group lasso. In: Proc. of NIPS, pp. 352–360 (2011)
34. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **68**(1), 49–67 (2006)
35. Zhang, X., Burger, M., Osher, S.: A unified primal-dual algorithm framework based on bregman iteration. Journal of Scientific Computing **46**(1), 20–46 (2011)
36. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **67**(2), 301–320 (2005)