Groups-Keeping Solution Path Algorithm for Sparse Regression with Automatic Feature Grouping

Bin Gu
Computer Science and Engineering
University of Texas at Arlington
Texas, 76019, USA
jsgubin@gmail.com

Guodong Liu
Computer Science and Engineering
University of Texas at Arlington
Texas, 76019, USA
mealsd@gmail.com

Heng Huang*
Computer Science and Engineering
University of Texas at Arlington
Texas, 76019, USA
heng@uta.edu

ABSTRACT

Feature selection is one of the most important data mining research topics with many applications. In practical problems, features often have group structure to effect the outcomes. Thus, it is crucial to automatically identify homogenous groups of features for high-dimensional data analysis. Octagonal shrinkage and clustering algorithm for regression (OSCAR) is an important sparse regression approach with automatic feature grouping and selection by ℓ_1 norm and pairwise ℓ_∞ norm. However, due to over-complex representation of the penalty (especially the pairwise ℓ_∞ norm), so far OSCAR has no solution path algorithm which is mostly useful for tuning the model. To address this challenge, in this paper, we propose a groups-keeping solution path algorithm to solve the OSCAR model (OscarGKPath). Given a set of homogenous groups of features and an accuracy bound ε, OscarGKPath can fit the solutions in an interval of regularization parameters while keeping the feature groups. The entire solution path can be obtained by combining multiple such intervals. We prove that all solutions in the solution path produced by OscarGKPath can strictly satisfy the given accuracy bound ε. The experimental results on benchmark datasets not only confirm the effectiveness of our OscarGKPath algorithm, but also show the superiority of our OscarGKPath in cross validation compared with the existing batch algorithm.

CCS CONCEPTS

•Information systems →Data mining; •Computing methodologies →Machine learning;

KEYWORDS

Solution path, OSCAR, automatic feature grouping, feature selection, sparse regression

*To whom all correspondence should be addressed. This work was partially supported by U.S. NIH R01 AG049371, NSF IIS 1302675, IIS 1344152, DBI 1356628, IIS 1619308 and IIS 1633753.

KDD'17, August 13–17, 2017, Halifax, NS, Canada.

DOI: http://dx.doi.org/10.1145/3097983.3098010

1 INTRODUCTION

The high-dimensional data are increasingly available in many data mining applications as the data collection technologies evolve. For example, DNA microarray technology can produce a large number of measurements [14, 29]. Medical imaging technologies (*e.g.*, MRI, CT, and Ultrasound) can produce high resolution 3-dimensional and 4-dimensional images [15]. The high resolution images are essentially high-dimensional data defined by the large number of voxels. To efficiently and effectively analyze the high-dimensional data, feature selection techniques have been introduced to identify the significant features associated to response variables [2] and to enhance the prediction tasks. Feature selection methods are particularly important and useful in bioinformatics and computational medicine (also called as biomarker selection).

To conduct the feature selection in high-dimensional data analysis, many sparse learning methods [8, 9, 19, 26, 30, 31] have been proposed. These sparse learning methods use the sparsity-inducing norms (*e.g.* to force the coefficients of non-important features to be zero. As a result, the features which have non-zero coefficients can be easily 'selected'. In the high-dimensional data, the highly correlated features widely exist [13, 15]. However, the sparse learning methods tend to arbitrarily select only one of them as mentioned in [6]. Thus, the estimation can be unstable, and the resulted model is difficult to interpret [33]. Especially, in bioinformatics research, some genes from the same family always work together to show the biological function, thus it is incorrect to only select one of them as biomarker. However, most existing feature selection methods ignore the feature group structure. Although the group LASSO model and its variants have been proposed, these methods require the feature group information to be known in advance [33]. Thus, it is crucial to design new feature selection with automatically identifying homogenous groups of features.

To tackle this challenging problem, several sparse learning methods have been proposed. For example, the elastic net [35] encourages β_i to be close to β_j for highly correlated features i, j by a ℓ_2-norm, where β_i's are the feature coefficients of the regression model. The fused LASSO [27] directly enforces the successive feature coefficients to be similar by the regularizer $|\beta_i - \beta_{i-1}|$, if the features are ordered in some meaningful way. The method proposed by Wu *et al.* [32] uses the ℓ_∞-norm to encourage the equality of coefficients for the features with maximum absolute value. The clustered LASSO [25] constraints all feature coefficients to be similar by the regularizer $\sum_{i<j} |\beta_i - \beta_j|$ (also called pairwise penalty). Different to the above feature grouping which cannot clearly and adaptively reveal the feature group structure, the OSCAR [6] (octagonal shrinkage and clustering algorithm for regression) method

uses the pairwise ℓ_∞-norm to encourage the equality of coefficients for highly correlated features. Among these methods, OSCAR can adaptively capture the feature groups, and clearly reveal the feature group structure by the equality of coefficients. In this paper, we focus on OSCAR model due to the ability of automatic feature grouping. The sparse learning based feature selection models usually have parameters and tuning parameters is time-consuming and could lead to sub-optimal results.

To address the parameter tuning issue and generate stable and optimal results, the solution path algorithm can provide a compact representation of all exact (or approximate) optimal solutions, which is extremely useful for model selection [10]. Several solution path algorithms have been proposed for sparse learning. For example, Rosset and Zhu [23] proposed a solution path algorithm for LASSO. Zhu *et al.* [34] proposed a solution path algorithm for ℓ_1-norm support vector machine. Park and Hastie [21] introduced a solution path algorithm for ℓ_1-norm regularized generalized linear models. Tibshirani and Taylor [28] presented the solution path algorithm for the generalized LASSO, where LASSO and fused LASSO are two special cases of the generalized LASSO. These solution path algorithms were designed for the learning problems with ℓ_1-norm. However, due to the difficulty in treating over-complex representation of the penalty (especially the pairwise ℓ_∞-norm) in OSCAR, there is still no solution path algorithm for OSCAR model. More importantly, we hope that the designed solution path algorithm can efficiently handle the pairwise ℓ_∞-norm. Note that Zhong and Kwok [33] proposed a fast batch algorithm of OSCAR (FastOSCAR) based on the accelerated proximal gradient method, which only gives one solution for one execution, but cannot give a continuous solution path for OSCAR.

In this paper, we propose a novel groups-keeping solution path algorithm for OSCAR (OscarGKPath), which can significantly improve the regularization parameters tuning of OSCAR model. Specifically, given a set of homogenous groups of features produced by a batch algorithm (*e.g.* FastOSCAR) and an accuracy bound ε, our new OscarGKPath algorithm can fit the solutions in an interval of regularization parameters while keeping the feature groups. Theoretically, we prove that any solution in this interval can strictly satisfy the given accuracy bound ε. The entire solution path can be obtained by combining multiple such intervals. We conduct the experiments on seven benchmark datasets. The experimental results confirm the effectiveness and efficiency of our OscarGKPath method.

Notations. β_j denote the j-th element of vector β. Δ denotes the amount of the change of each variable. sign(x) is a sign function which returns 1 if $x > 0$, otherwise returns -1.

2 REVIEW OF OSCAR MODEL

In this section, we first introduce the formulation of OSCAR, and then provide the optimality conditions of OSCAR correspondingly.

2.1 OSCAR

Given a training set $S = \{(x_i, y_i)\}_{i=1}^l$ with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. We assume that y_i is centered, *i.e.*, $\sum_{i=1}^l y_i = 0$, and each feature of the training set S is standardized, *i.e.*, $\sum_{i=1}^l x_{ij} = 0$ and $\sum_{i=1}^l x_{ij}^2 = 1$. Because the response is centered, OSCAR considers a linear regression model without the intercept. Thus, the formulation of OSCAR is considered as follows:

$$\begin{aligned} \min_{\beta} \quad & \frac{1}{2}\sum_{i=1}^{l}\left(y_i - x_i^T\beta\right)^2 \\ s.t. \quad & \|\beta\|_1 + c\sum_{j<k}\max\{|\beta_j|, |\beta_k|\} \le t\,, \end{aligned} \tag{1}$$

where the ℓ_1-norm enforces the sparsity, and the pairwise ℓ_1-norm encourages every coefficient pair $|\beta_j|$ and $|\beta_k|$ to be equal which can automatically group highly correlated features. $c \ge 0$ an $t > 0$ are tuning constants. c is controlling the relative weighting of the norms and t is controlling the magnitude. Specially, if $c = 0$, OSCAR degenerates to LASSO. If $c = \infty$, OSCAR clusters all features as a group but without variable selection. Thus, selecting appropriate values of c and t plays an essential role for OSCAR.

The formulation (1) is a constrained optimization problem which can be written in the penalized form (2) according to the subdifferential version of Karush-Kuhn-Tucker (KKT) conditions [24]:

$$\begin{aligned} F(\beta, \lambda_1, \lambda_2) = \min_{\beta} \quad & \frac{1}{2}\sum_{i=1}^{l}\left(y_i - x_i^T\beta\right)^2 \\ & +\lambda_1\|\beta\|_1 + \lambda_2\sum_{i<j}\max\{|\beta_i|, |\beta_j|\}\,, \end{aligned} \tag{2}$$

where $\lambda_1 \ge 0$ and $\lambda_2 \ge 0$ are two regularization parameters. For a pair of λ_1 and λ_2, there exists a pair of t and c such that (2) and (1) share the same solution, and vice versa. As mentioned above, selecting the values of λ_1 and λ_2 also plays an essential role for OSCAR. This paper will propose a novel group-keeping solution path algorithm for (2) which greatly benefits the regularization parameters tuning in solving OSCAR model.

2.2 Optimality Conditions of OSCAR Model

The formulation (2) has the pair ℓ_∞-norm which make it nontrivial to derive the optimality conditions of OSCAR. In this section, we first derive an equivalent formulation of (2) which is based on the feature groups and orders of the optimal solution, and then present the optimality conditions of OSCAR.

We denote β as an optimal solution of OSCAR. Let $o(j) \in \{1, \cdots, d\}$ denote the order of $|\beta_j|$ among $\{|\beta_1|, |\beta_2|, \cdots, |\beta_d|\}$ such that if $o(j_1) < o(j_2)$, we have: $|\beta_{j_1}| \le |\beta_{j_2}|$. Based on the orders $o(j)$, we can define the feature group $\mathcal{G}_g$ as following:

Definition 2.1. Given the orders $o(j)$ of $|\beta_j|$. The set $\mathcal{G}_g \subseteq \{1, \cdots, d\}$ is called a group of features if the following conditions are satisfied.

(1) $\forall j_1, j_2 \in \mathcal{G}_g$, and $j_1 \ne j_2$, we have $|\beta_{j_1}| = |\beta_{j_2}| \stackrel{\text{def}}{=} \theta_g$.
(2) If $j \in \{1, 2, \cdots, d\}$ and $j \notin \mathcal{G}_g$, we have that $|\beta_j| \ne \theta_g$.

In Definition 2.1, we denote θ_g as the common value of $|\beta_j|$ for the group $\mathcal{G}_g$. Thus, we have a set of $\mathcal{G}_g$, $g = 1, \cdots, G$, such that $\mathcal{G}_1 \cup \mathcal{G}_2 \cup \cdots \cup \mathcal{G}_G = \{1, 2, \cdots, d\}$, and $0 \le \theta_1 < \theta_2 < \cdots < \theta_G$. Based on the groups $\mathcal{G}_g$, $g = 1, \cdots, G$, the formulation (2) can be rewritten as (3) which is free of the ℓ_1-norm and the pair ℓ_∞-norm:

$$\begin{aligned} \min_{\theta} \quad & \frac{1}{2}\sum_{i=1}^{l}\left(y_i - \widetilde{x}_i^T\theta\right)^2 + \sum_{g=1}^{G} w_g\theta_g \\ s.t. \quad & 0 \le \theta_1 < \theta_2 < \cdots < \theta_G\,, \end{aligned} \tag{3}$$

where $\widetilde{x}_i = [\widetilde{x}_{i1}\ \widetilde{x}_{i2} \cdots \widetilde{x}_{iG}]$ and $\widetilde{x}_{ig} = \sum_{j \in \mathcal{G}_g} \text{sign}(\beta_j) x_{ij}$. $w_g = \sum_{j \in \mathcal{G}_g} (\lambda_1 + (o(j)-1)\lambda_2)$.

According to the value of θ_g, we can define an active set: $\mathcal{A} = \{g \in \{1, \cdots, G\} \mid \theta_g > 0\}$. Correspondingly, we define $\overline{\mathcal{A}} = \{1, \cdots, G\} - \mathcal{A}$. Thus, the optimality conditions of (3) can be presented as following:

$$\sum_{i=1}^{l} -\widetilde{x}_{ig} \left(y_i - \widetilde{x}_i^T \theta \right) + w_g = 0, \quad \forall g \in \mathcal{A} \tag{4}$$

$$0 < \theta_{2-|\overline{\mathcal{A}}|} < \cdots < \theta_{G-1} < \theta_G \tag{5}$$

$$\theta_{\overline{\mathcal{A}}} = 0\,. \tag{6}$$

3 OUR NEW SOLUTION PATH ALGORITHM

In this section, we propose a groups-keeping solution path algorithm of OSCAR (*i.e.*, OscarGKPath) with respect to two regularization parameters λ_1 and λ_2. Let $\Delta\lambda = \begin{bmatrix} \Delta\lambda_1 \\ \Delta\lambda_2 \end{bmatrix}$ denote the changes of λ_1 and λ_2, and given a direction $d = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$ to $\Delta\lambda$, we have $\Delta\lambda = d\Delta\eta$, where $\Delta\eta$ is a parameter to control the adjustment qualities of λ_1 and λ_2. Thus, OscarGKPath actually tries to produce a solution path of OSCAR with respect to the direction d on λ_1 and λ_2.

Normally, solution path algorithms include two steps, *i.e.*, initializing the solution for a fixed values of λ_1 and λ_2, and computing the solution path based on the initial solution. As mentioned previously, FastOSCAR [33] is a fast batch algorithm of OSCAR based on the accelerated proximal gradient method. As far as we know, so far FastOSCAR is the fastest batch learning algorithm of OSCAR. Thus, we use FastOSCAR to produce the initial solution and the initial groups $\mathcal{G}_g$ to the optimization (3). In the following, we provide the detailed descriptions for computing the solution path based on the initial solution.

Let $\Delta\theta$ represent the changes of the coefficients θ. To compute the solution path, the first issue is which direction of $\Delta\theta$ (denoted by ξ) is with respect to the direction d on λ_1 and λ_2. The key of computing the direction ξ is trying to hold all the solutions satisfying the optimality conditions (4)-(6) during the adjustments (see Fig. 1), given the feature groups. Note that the unchanged feature groups may not be true for new values of λ_1 and λ_2. Thus, the solutions on the direction ξ could be approximate solutions. We use an accuracy bound ε to control the quality of solutions. After computing the direction ξ, the next issue is what maximum adjustment of $\Delta\eta$ (denoted by $\Delta\eta^{max}$) is such that the optimality conditions (4)-(6) will not be satisfied if $\Delta\eta$ exceeds $\Delta\eta^{max}$. After finding the maximum adjustment quantity $\Delta\eta^{max}$, we can update θ, $\lambda_1, \lambda_2, \mathcal{A}, \overline{\mathcal{A}}$ and the orders of θ_g. Repeating this procedure until the duality gap $G(\theta, \lambda_1, \lambda_2) > \varepsilon$ (see Fig. 1). Finally, we should backtrack the last piece of solution path to make sure the end solution to satisfy the accuracy ε, which can be easily implemented by binary search. The above procedure is our OscarGKPath algorithm which is summarized in Algorithm 1.

As mentioned above, three main steps of OscarGKPath are:

(1) computing the directions of $\Delta\theta$;
(2) computing the maximum adjustment of $\Delta\eta$;
(3) and computing the duality gap $G(\theta, \lambda_1, \lambda_2)$.

In the following, we provide the detailed descriptions for these three steps respectively.

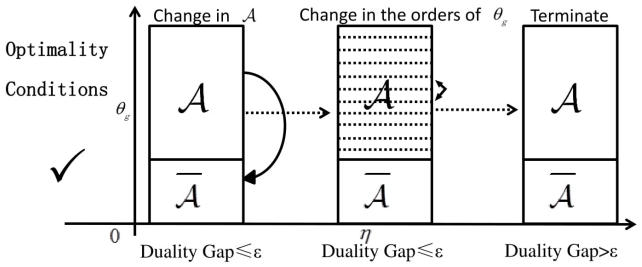


Figure 1: The fundamental principle of OscarGKPath is holding all the solutions satisfying the accuracy bound ε when given a set of feature groups, and adjusting the solutions based on the optimality conditions.

Algorithm 1 OscarGKPath

Input: The direction d, and the accuracy ε, an interval $[\underline{\eta}, \overline{\eta}]$ of η.
Output: Approximate solution path of OSCAR w.r.t. d on λ_1 and λ_2 in an interval of $[\underline{\eta}, \overline{\eta}]$.

1: Compute the solution θ and the groups $\mathcal{G}_g$ for $\eta = \underline{\eta}$ based on FastOSCAR.
2: **repeat**
3: Compute the directions of $\Delta\theta$.
4: Compute the maximum adjustment of $\Delta\eta$.
5: Update η, θ, λ_1, λ_2, $\mathcal{A}$ and the orders of θ_g.
6: Compute the duality gap $G(\theta, \lambda_1, \lambda_2)$ according to Algorithm 1.
7: **until** $\eta > \overline{\eta}$ or $G(\theta, \lambda_1, \lambda_2) > \varepsilon$
8: Backtrack the last piece of solution path to make sure the end solution to satisfy the accuracy ε.

3.1 Compute the Directions of $\Delta\theta$

As mentioned above, the principle during the adjustment of λ_1 and λ_2 is to keep the optimality conditions of OSCAR. According to the optimality condition (4), it is easy to find $\theta_{\overline{\mathcal{A}}}$ is fixed on the bound 0. Only $\theta_{\mathcal{A}}$ have the possibility to be adjusted when changing λ_1 and λ_2. Thus, we have the following linear system:

$$\sum_{i=1}^{l} \widetilde{x}_{ig} \widetilde{x}_{i\mathcal{A}}^T \Delta\theta_{\mathcal{A}} + \widetilde{w}_g \Delta\eta = 0, \quad \forall g \in \mathcal{A}\,, \tag{7}$$

where $\widetilde{w}_g = \sum_{j \in \mathcal{G}_g} (d_1 + (o(j)-1)d_2)$. Let $\widetilde{W}$ be the $G \times G$ diagonal matrix with $\widetilde{W}_{gg} = \widetilde{w}_g$, and $\widetilde{X}$ be a $l \times G$ matrix whose i-th row is equal to $\widetilde{x}_i^T$. We can represent the above linear system (7) as the following in the matrix form:

$$\underbrace{\widetilde{X}_{*\mathcal{A}}^T \widetilde{X}_{*\mathcal{A}}}_{H_{\mathcal{A}\mathcal{A}}} \Delta\theta_{\mathcal{A}} + \widetilde{W}_{\mathcal{A}\mathcal{A}} \Delta\eta = \mathbf{0}\,. \tag{8}$$

Let $\xi_{\mathcal{A}}$ denote $\frac{\Delta\theta_{\mathcal{A}}}{\Delta\eta}$ (*i.e.*, the direction of $\Delta\theta_{\mathcal{A}}$ *w.r.t.* $\Delta\eta$), the linear system (8) can be rewritten as:

$$H_{\mathcal{A}\mathcal{A}}\xi_{\mathcal{A}} = -\widetilde{W}_{\mathcal{A}\mathcal{A}}\,. \tag{9}$$

Thus, we can get $\xi_{\mathcal{A}}$ by solving the linear system (9). Traditional way for solving the linear system (9) is by the direct matrix inverse of $H_{\mathcal{A}\mathcal{A}}$. As mentioned in [12, 20], the key matrix $H_{\mathcal{A}\mathcal{A}}$ will encounter singularities. For the robustness of OscarGKPath, we can compute $\xi_{\mathcal{A}}$ based on the QR decomposition with column pivoting [22] without directly computing the inverse of $H_{\mathcal{A}\mathcal{A}}$. Because $\Delta\theta_{\overline{\mathcal{A}}} = \mathbf{0}$, actually we know the direction of $\Delta\theta$.

3.2 Compute the Maximum Adjustment of $\Delta\eta$

After obtaining the linear relationships $\xi_{\mathcal{A}}$, we need to compute the maximum adjustment $\Delta\eta^{max}$ as mentioned previously. As shown in Fig. 1, there are three main types of cases which should be considered for the computation of $\Delta\eta^{max}$.

(1) A certain coefficient θ_g in $\mathcal{A}$ reaches 0. Thus we can compute the maximal possible $\Delta\eta^{\mathcal{A}}$ before a certain θ_g in $\mathcal{A}$ moves to $\overline{\mathcal{A}}$, by the constraints $\theta_g + \xi_g\Delta\eta > 0, \forall g \in \mathcal{A}$ in the optimality conditions of OSCAR.
(2) A pair of feature groups swap their orders of θ_g. As mentioned in (5), the optimality conditions of OSCAR are based on a given orders of θ_g. Thus, we can compute the maximal possible $\Delta\eta^{o}$ before a pair of adjacent groups $\mathcal{G}_g$ and $\mathcal{G}_{g+1}$ swap their orders, by the constraints $\theta_g + \xi_g\Delta\eta < \theta_{g+1} + \xi_{g+1}\Delta\eta$.
(3) η reaches $\overline{\eta}$, i.e., the termination condition is met. Then the maximal adjustment quantity before the solution path algorithm meets the termination condition is $\overline{\eta} - \eta$.

Thus, the smallest of three values $\{\Delta\eta^{\mathcal{A}}, \Delta\eta^{o}, \overline{\eta} - \eta\}$ constitutes the maximal adjustment quantity $\Delta\eta^{max}$.

3.3 Check the Duality Gap

The optimization problem $F(\beta)$ is a convex problem. Thus, we can guarantee the solution β is a ε-approximation solution with $F(\beta, \lambda_1, \lambda_2) - F(\beta^*, \lambda_1, \lambda_2) \leq \varepsilon$ by the duality gap $G(\beta, \lambda_1, \lambda_2) = F(\beta, \lambda_1, \lambda_2) - \widetilde{F}(\alpha, \lambda_1, \lambda_2) \leq \varepsilon$, where β^* is an optimal solution of $F(\beta, \lambda_1, \lambda_2)$, α is the dual variable, and $\widetilde{F}(\alpha, \lambda_1, \lambda_2)$ is the dual of $F(\beta, \lambda_1, \lambda_2)$. This conclusion holds because $F(\beta, \lambda_1, \lambda_2) - F(\beta^*, \lambda_1, \lambda_2) \leq G(\beta, \lambda_1, \lambda_2)$ [7].

As discussed in [4, 33], the dual function $\widetilde{F}(\alpha, \lambda_1, \lambda_2)$ can be computed as:

$$\begin{aligned} \widetilde{F}(\alpha, \lambda_1, \lambda_2) = \max_{\alpha} \quad & \frac{-1}{2}\alpha^T\alpha - \alpha^T y \\ s.t. \quad & \max_{\sum_{j=1}^{d}(\lambda_1 + \lambda_2(o(j)-1))|\beta_j| = 1} \alpha^T X\beta \leq 1\,, \end{aligned} \tag{10}$$

where X is an $l \times d$ matrix whose i-th row is equal to x_i^T. Further, [33] proved that the optimal α of $\widetilde{F}(\alpha)$ can be analytically computed as:

$$\alpha = \min\{1, \frac{1}{r^*(X^T\nabla f(\beta))}\}\nabla f(\beta)\,, \tag{11}$$

where $\nabla f(\beta) = X\beta - y$. Assuming the indices of γ are sorted by $|\gamma_1| \leq |\gamma_2| \leq \cdots \leq |\gamma_d|$, we have:

$$r^*(\gamma) = \max_{j \in \{1, 2, \cdots, d\}} \frac{\sum_{i=1}^{j} |\gamma_j|}{\sum_{i=1}^{j} \lambda_1 + (i-1)\lambda_2}\,. \tag{12}$$

The algorithm for computing the duality gap was originally proposed by [33]. In a word, it can be computed according to (10) based on the optimal α of $\widetilde{F}(\alpha, \lambda_1, \lambda_2)$ (11) which is analytically represented. Thus, the duality gap can be computed efficiently. We also present them in Algorithm 2 to be consistent with the formulation of (2).

Algorithm 2 Duality Gap

Input: β or θ, λ_1 and λ_2.
Output: The duality gap $G(\theta, \lambda_1, \lambda_2)$.
1: Compute $\gamma = X^T\nabla f(\beta)$ and sort γ_i in ascend order.
2: Compute $r^*(\gamma)$.
3: Compute the optimal α of $\widetilde{F}(\alpha)$ according to (11).
4: Compute the duality gap $G(\beta, \lambda_1, \lambda_2) = F(\beta, \lambda_1, \lambda_2) - \widetilde{F}(\alpha, \lambda_1, \lambda_2)$ according to (10).

4 ε-APPROXIMATION ANALYSIS OF OSCARGKPATH ALGORITHM

As shown in Algorithm 1, we check the duality gap only for several single points of η. How to guarantee that the whole solution path produced by OscarGKPath is ε-approximation is the focus of this section. In this section, we will prove that any solution in the solution path produced by OscarGKPath can strictly satisfy that $G(\theta, \lambda_1, \lambda_2) \leq \varepsilon$ (Corollary 4.3). In addition, we provide a guideline for choosing $\underline{\eta}$ and $\overline{\eta}$ (Theorem 4.5 and 4.6), which guide the choices for the start and ending points of the interval $[\underline{\eta}, \overline{\eta}]$.

Before answering the question, we first give a definition of piecewise linearity [10] of the solution path as following:

Definition 4.1. Suppose $\theta(\eta)$ is returned by a solution path. The solution $\theta(\eta)$ is called piecewise linear as a function of η, if existing $\underline{\eta} = \eta_0 < \eta_1 < \eta_2 < \cdots < \eta_m = \overline{\eta}$, and the corresponding vectors $\xi^{[1]}, \xi^{[2]}, \cdots, \xi^{[m]}$, such that the solution $\theta(\eta)$ is given exactly or approximately, by $\theta(\eta_k) + \xi^{[k]}(\eta - \eta_k), \forall\eta \in [\eta_k, \eta_{k+1}]$.

Based on Definition 4.1, it is easy to verify that $\theta(\eta)$ produced by OscarGKPath is piecewise linear, where each interval $[\eta_k, \eta_{k+1}]$ corresponds the interval produced by one iteration of OscarGKPath. Based on the piecewise linearity of OscarGKPath, we can prove that all the solutions $\beta(\eta)$, $\forall\eta \in [\eta_k, \eta_{k+1}]$, strictly satisfy that $G(\theta(\eta), \lambda_1, \lambda_2) \leq \varepsilon$ (Theorem 4.5), which means that any solution in the solution path produced by OscarGKPath can strictly satisfy that $G(\theta(\eta), d_1\eta, d_2\eta) \leq \varepsilon$ (Corollary 4.3).

Theorem 4.2. *For the interval $[\eta_k, \eta_{k+1}]$ produced by one iteration of OscarGKPath, we have that all the solutions $\theta(\eta)$, $\forall\eta \in [\eta_k, \eta_{k+1}]$, strictly satisfy that $G(\theta(\eta), d_1\eta, d_2\eta) \leq \varepsilon$.*

Proof. According to (11), we have that $\alpha = \nabla f(\beta)$ if $r^*(X^T\nabla f(\beta)) < 1$, otherwise $\alpha = \frac{1}{r^*(X^T\nabla f(\beta))}\nabla f(\beta)$. We first prove that the solutions $\theta(\eta)$, $\forall\eta \in [\eta_{k-1}, \eta_k]$, strictly satisfy that $G(\theta(\eta), d_1\eta, d_2\eta) \leq \varepsilon$

if $\alpha(\eta) = \nabla f(\beta(\eta))$.

$$
\begin{aligned}
& F(\theta(\eta), d_1\eta, d_2\eta) \qquad (13)\\
= \;& \frac{1}{2}\|\widetilde{X}\theta(\eta) - y\|^2 + \sum_{g=1}^{G} w_g\theta_g(\eta)\\
= \;& \frac{1}{2}\|\widetilde{X}(\theta(\eta_k) + \xi\Delta\eta) - y\|^2\\
& + \sum_{g=1}^{G} \widetilde{w}_g(\eta_k + \Delta\eta)(\theta_g(\eta_k) + \xi_g\Delta\eta)\\
= \;& \underbrace{F(\theta(\eta_t))}_{c_1} + \underbrace{\left(\xi^T\xi + \sum_{g=1}^{G}\widetilde{w}_g\xi_g\right)}_{a_1}(\Delta\eta)^2 +\\
& \underbrace{\left(\sum_{g=1}^{G}\widetilde{w}_g(\eta_k\xi_g + \theta_g(\eta_k)) - 2\xi^T(\widetilde{X}\theta(\eta_k) - y)\right)}_{b_1}\Delta\eta
\end{aligned}
$$

$$
\begin{aligned}
& -\widetilde{F}(\alpha(\eta), d_1\eta, d_2\eta) \qquad (14)\\
= \;& \frac{1}{2}\alpha(\beta(\eta))^T\alpha(\beta(\eta)) + \alpha(\eta)^T y\\
= \;& \frac{1}{2}\left(X(\beta(\eta_t) + \widetilde{\xi}\Delta\eta) - y\right)^T\left(X(\beta(\eta_t) + \widetilde{\xi}\Delta\eta) - y\right)\\
& + \left(X(\beta(\eta_t) + \widetilde{\xi}\Delta\eta) - y\right)^T y\\
= \;& \underbrace{\widetilde{F}(\alpha(\eta_t))}_{c_2} + \underbrace{\frac{1}{2}\left(X\widetilde{\xi}\right)^T\left(X\widetilde{\xi}\right)}_{a_2}(\Delta\eta)^2\\
& + \underbrace{\left((X\beta(\eta_t) - y^T)X\widetilde{\xi} + X\widetilde{\xi}^T y\right)}_{b_2}\Delta\eta
\end{aligned}
$$

where β can be converted from θ, $\widetilde{\xi}$ is the directions of $\Delta\beta$ which also can be converted from ξ. Based on (13)-(14), we can denote $G(\theta(\eta), d_1\eta, d_2\eta)$ as $G(\theta(\eta), d_1\eta, d_2\eta) = a(\Delta\eta)^2 + b\Delta\eta + c$, where

$$
\begin{aligned}
a &= a_1 + a_2 \qquad (15)\\
b &= b_1 + b_2 \qquad (16)\\
c &= c_1 + c_2 \qquad (17)\\
\Delta\eta &= \eta - \eta_t \qquad (18)
\end{aligned}
$$

Thus, it is easy to verify that $a > 0$ or $a = 0$ and $b \geq 0$. Otherwise, we can get $G(\beta(\eta), d_1\eta, d_2\eta) < 0$ for some $\eta > \eta_t$, which contradicts with the fact $G(\beta(\eta), d_1\eta, d_2\eta) \geq 0$ for all $\eta \geq 0$. Thus, the maximum of $G(\theta(\eta), d_1\eta, d_2\eta)$ for $\eta \in [\eta_k, \eta_{k+1}]$ is either $G(\theta(\eta_t), d_1\eta_t, d_2\eta_t)$ or $G(\theta(\eta_{t+1}), d_1\eta_{t+1}, d_2\eta_{t+1})$. This completes the proof for $\alpha(\eta) = \nabla f(\beta(\eta))$.

If $\alpha = \frac{1}{r^*(X^T\nabla f(\beta))}\nabla f(\beta)$, we have:

$$
\begin{aligned}
& X^T\nabla f(\beta(\eta)) = X^T(X\beta(\eta) - y) \qquad (19)\\
= \;& X^T X(\beta(\eta_t) + \widetilde{\xi}\Delta\eta) - X^T y\\
= \;& X^T\nabla f(\beta(\eta_t)) + X^T X\widetilde{\xi}\Delta\eta\,.
\end{aligned}
$$

Assuming $r^*(\gamma(\eta))$ achieves the maximum at the index j_0, we have:

$$
\begin{aligned}
r^*(\gamma(\eta)) &= \frac{\sum_{i=1}^{j_0}|\gamma_j(\eta)|}{\sum_{i=1}^{j_0}\lambda_1 + (i-1)\lambda_2} \qquad (20)\\
&= \frac{\sum_{i=1}^{j_0}|X_{*i}^T\nabla f(\beta(\eta_t)) + X_{*i}^T X\widetilde{\xi}\Delta\eta|}{\sum_{i=1}^{j_0}\lambda_1 + (i-1)\lambda_2}\\
&\stackrel{\text{def}}{=} r^*(\gamma(\eta_t)) + \widetilde{a}\Delta\eta\,.
\end{aligned}
$$

Thus, we have $\alpha(\eta) = \frac{X(\beta(\eta_t)) - y + \widetilde{\xi}\Delta\eta}{r^*(\gamma(\eta_t)) + \widetilde{a}\Delta\eta}$. Let

$$
\begin{aligned}
& (X(\beta(\eta_t)) - y + \widetilde{\xi}\Delta\eta)^T y(r^*(\gamma(\eta_t)) + \widetilde{a}\Delta\eta)\\
\stackrel{\text{def}}{=} \;& a_3(\Delta\eta)^2 + b_3\Delta\eta + c_3\,, \qquad (21)
\end{aligned}
$$

we have:

$$
\begin{aligned}
& -\widetilde{F}(\alpha(\eta), d_1\eta, d_2\eta) \qquad (22)\\
= \;& \frac{(a_2 + a_3)(\Delta\eta)^2 + (b_2 + b_3)\Delta\eta + c_2 + c_3}{(r^*(\gamma(\eta_t)) + \widetilde{a}\Delta\eta)^2}
\end{aligned}
$$

We can conclude that $\frac{-r^*(\gamma(\eta_t))}{\widetilde{a}} < 0$. Otherwise, it is easy to verify that the duality gap is negative or infinite for some $\Delta\eta > 0$ because the singular point $\Delta\eta = \frac{-r^*(\gamma(\eta_t))}{\widetilde{a}}$. Further, we have that $r^*(\gamma(\eta_t)) > 0$ and $\widetilde{a} > 0$. We have that $a_2 + a_3 > 0$ because $\lim_{\eta\to\infty} -\widetilde{F}(\alpha(\eta), d_1\eta, d_2\eta) = \frac{a_2+a_3}{\widetilde{a}^2}$. Further, $a_2 + a_3$ should be much larger than $-\widetilde{a}^2\widetilde{F}(\alpha(\eta_k), d_1\eta_k, d_2\eta_k)$. Thus, $-\widetilde{F}(\alpha(\eta), d_1\eta, d_2\eta)$ should not be monotonically increasing then decreasing for $\eta > 0$.

Similarly, we can conclude that $a_1 > 0$ or or $a_1 = 0$ and $b_1 \geq 0$. Otherwise, we can get $G(\beta(\eta), d_1\eta, d_2\eta) < 0$ for some $\eta > \eta_t$, which contradicts with the fact $G(\beta(\eta), d_1\eta, d_2\eta) \geq 0$ for all $\eta \geq 0$. Thus, $F(\theta(\eta), d_1\eta, d_2\eta)$ should not be monotonically increasing then decreasing for $\eta > 0$.

Thus, combining the analysis for $-\widetilde{F}(\alpha(\eta), d_1\eta, d_2\eta)$ and $F(\theta(\eta), d_1\eta, d_2\eta)$ as mentioned above, we can conclude that $G(\theta(\eta), d_1\eta, d_2\eta)$ should not be monotonically increasing then decreasing for $\eta > 0$. Thus, the maximum of $G(\theta(\eta), d_1\eta, d_2\eta)$ for $\eta \in [\eta_k, \eta_{k+1}]$ would be at the start point or endpoint of the interval $[\eta_k, \eta_{k+1}]$. This completes the proof. □

Corollary 4.3. *For all solutions $\theta(\eta)$ produced by OscarGKPath, we have that $\theta(\eta)$ satisfies that $G(\theta(\eta), d_1\eta, d_2\eta) \leq \varepsilon$.*

Corollary 4.3 can be easily obtained based on Theorem 4.5. In the following, we give the definition of the maximal of η (denoted by $\eta_{\min}$) such that $\beta(\eta_{\min})$ is a solution of the ordinary least squares. We also give the the definition of the minimum of η (denoted by $\eta_{\max}$) such that $\beta(\eta_{\max}) = \mathbf{0}$. It is intuitive that $[\eta_{\min}, \eta_{\max}]$ is a good choice for the interval $[\underline{\eta}, \overline{\eta}]$ in Algorithm 1.

Definition 4.4. Let $\beta(0) = (\widetilde{X}^T\widetilde{X})^{-1}\widetilde{X}^T y$ is the solution of (2) with $\eta = 0$. $\eta_{\min}$ is the maximal of η in (2) such that $F(\beta(\eta_{\min}), d_1\eta_{\min}, d_2\eta_{\min}) = F(\beta(0), 0, 0)$. Let $\beta(\infty) = \mathbf{0}$ is the solution of (2) with $\eta = \infty$. $\eta_{\max}$ is the minimum of η in (2) such that $F(\beta(\eta_{\max}), d_1\eta_{\max}, d_2\eta_{\max}) = F(\mathbf{0}, \infty, \infty)$.

Theorem 4.5. *We define $t_{\max}$ as*

$$
t_{\max} = \min_{v} \qquad \|\beta(0) + v\|_1 + \qquad (23)
$$

$$\frac{d_2}{d_1}\sum_{j<k}\max\{|\beta(0)_j+v_j|,|\beta(0)_k+v_k|\}$$
$$s.t. \quad X^T v = \mathbf{0}$$

Let $\beta = \beta(0) + v$. We have $\eta_{\min} = \|D^{-1}X^T(X\beta - y)\|_\infty$, where D is a $d \times d$ diagonal matrix with $D_{jj} = d_1 + d_2(o(j) - 1)$.

Proof. According to the subdifferential version of KKT conditions [24], we have that $X^T(X\beta - y) + Dv\eta = \mathbf{0}$, where $v \in \partial\|\beta\|_1$. Thus, we have $\eta = \|D^{-1}X^T(X\beta - y)\|_\infty$ if $\beta \neq \mathbf{0}$. As mentioned in [6], there exists a direct correspondence between t and η. Specifically, the larger t is, and the smaller η. Thus, $\eta_{\min} = \|D^{-1}X^T(X(\beta(0) + v) - y)\|_\infty$. This completes the proof. □

The problem (23) can be solved by the augmented Lagrangian algorithm efficiently with the iteration complexity $O(\log(\epsilon^{-1}))$ [3], where ϵ is the accuracy of the solution.

Theorem 4.6. *Let Ω denote the set of all possible orders for $|\beta = \mathbf{0}|$, and let D^Ω denote the set of all possible D based on a given Ω. We have*

$$\eta_{\max} = \min \quad \|D^{-1}X^T y\|_\infty \qquad (24)$$
$$s.t. \quad D \in D^\Omega$$

Proof. According to the subdifferential version of KKT conditions [24], we have $Dv\eta - X^T y = \mathbf{0}$, where $v \in \partial\|\beta\|_1$. Because $v_j \in [-1, 1]$, we have that $\eta = \|D^{-1}X^T y\|_\infty$. This completes the proof. □

Essentially, (24) is a combinatorial optimization, which can not be solved exactly in polynomial time. In practice, we can set the orders of $|\beta_j|$ as the ones of the $|X^T y|$ in ascending order, which gives a good approximation of $\eta_{\max}$.

5 EXPERIMENTAL RESULTS

In this section, we first describe the experimental setup, and then provide the experimental results and discussions.

5.1 Experimental Setup

5.1.1 Design of Experiments. The experiments include two parts. The first part is to verify the effectiveness of OscarGKPath, and the second part is to show the advantage of OscarGKPath for model selection.

To verify the effectiveness of OscarGKPath, we count the numbers of the iterations (denoted as *#Iterations*) of OscarGKPath and the numbers of calling FastOSCAR (denoted as *#FastOSCAR*) for producing the entire solution path. By counting the *#FastOSCAR*, we want to check how many the calling of FastOSCAR are needed for producing the entire solution path. By counting the *#Iterations*, we empirically show that finite convergence of OscarGKPath. To show the advantage of OscarGKPath for model selection, we compare the running time, cross validation errors and testing errors of our OscarGKPath and the batch algorithm (*i.e.*, FastOSCAR) for 5-fold cross validation.

5.1.2 Implementation Details. We implement our proposed OscarGKPath in MATLAB. As mentioned previously, to the best of our knowledge, the FastOSCAR algorithm [33] is the fastest batch learning algorithm for OSCAR. To compare the run-time at the same platform, we implement the FastOSCAR algorithm in MATLAB to compare the running time of cross validation with our OscarGKPath.

The 5-fold cross validation is done on a two-step grid search strategy [11]. The initial search is done on a 20 coarse grid linearly spaced in the region $\{\log_2\eta| - 4 \leq \log_2\eta \leq 15\}$, followed by a fine search on a 20 uniform grid linearly spaced by 0.1 in the $(\log_2\eta)$ space. To compare the running time of FastOSCAR and OscarGKPath and verify the effectiveness of OscarGKPath, we do the experiments on three representative directions of d, *i.e.*, $\begin{bmatrix}1\\0.5\end{bmatrix}$, $\begin{bmatrix}1\\1\end{bmatrix}$ and $\begin{bmatrix}1\\2\end{bmatrix}$. Thus, we fix $d_1 = 1$, and set $d_2 = 0.5$, 1 and 2 respectively. In the experiments, we set the duality gap $G(\theta(\eta), d_1\eta, d_2\eta) \leq \varepsilon = 0.1 \times F(\beta^*, d_1, d_2)$.

5.1.3 Datasets. Table 1 summarizes the details of seven benchmark datasets used in our experiments, where the first and last two datasets (*i.e.*, YearPredictionMSD, USPS, SensIT Vehicle (combined) and Protein datasets) are from the LIBSVM Data[1], and the IndoorLoc Longitude, IndoorLoc Latitude and Slice Localization datasets are from the UCI benchmark repository [5]. The Left Ventricle and Right Ventricle dataset were collected from 3360 MRI images by hospital and each image has 400 pixels. Both of the datasets are encouraged to find the homogenous groups of features. The Left and right Ventricle datasets are to predict the areas for left ventricle and right ventricle respectively [16, 17]. Note that, the alphabets in the $(\cdot)$ are the abbreviation of the name of the corresponding dataset.

Table 1: The summary of datasets used in our experiments.

Dataset	Sample size	Attributes
YearPredictionMSD(YP)	51,630	90
USPS	7,291	256
Left Ventricle(LV)	3,360	400
Right Ventricle(RV)	3,360	400
IndoorLoc Longitude(InLo)	21,048	529
IndoorLoc Latitude(InLa)	21,048	529
Slice Localization(SL)	53,500	386
SensIT Vehicle Combined	78,823	100
Protein	17,766	357

5.2 Experimental Results and Discussions

Effectiveness of OscarGKPath: Table 2 presents the average numbers of *#Iterations* for OscarGKPath over 10 trials with different values of d_2 on the YearPredictionMSD, USPS, Left Ventricle, Right Ventricle, IndoorLoc Longitude, IndoorLoc Latitude, Slice Localization, SensIT Vehicle Combined and Protein datasets. The results show that OscarGKPath can fit the entire approximate solution path of OSCAR within a finite number of iterations. Table 2 also presents

[1] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

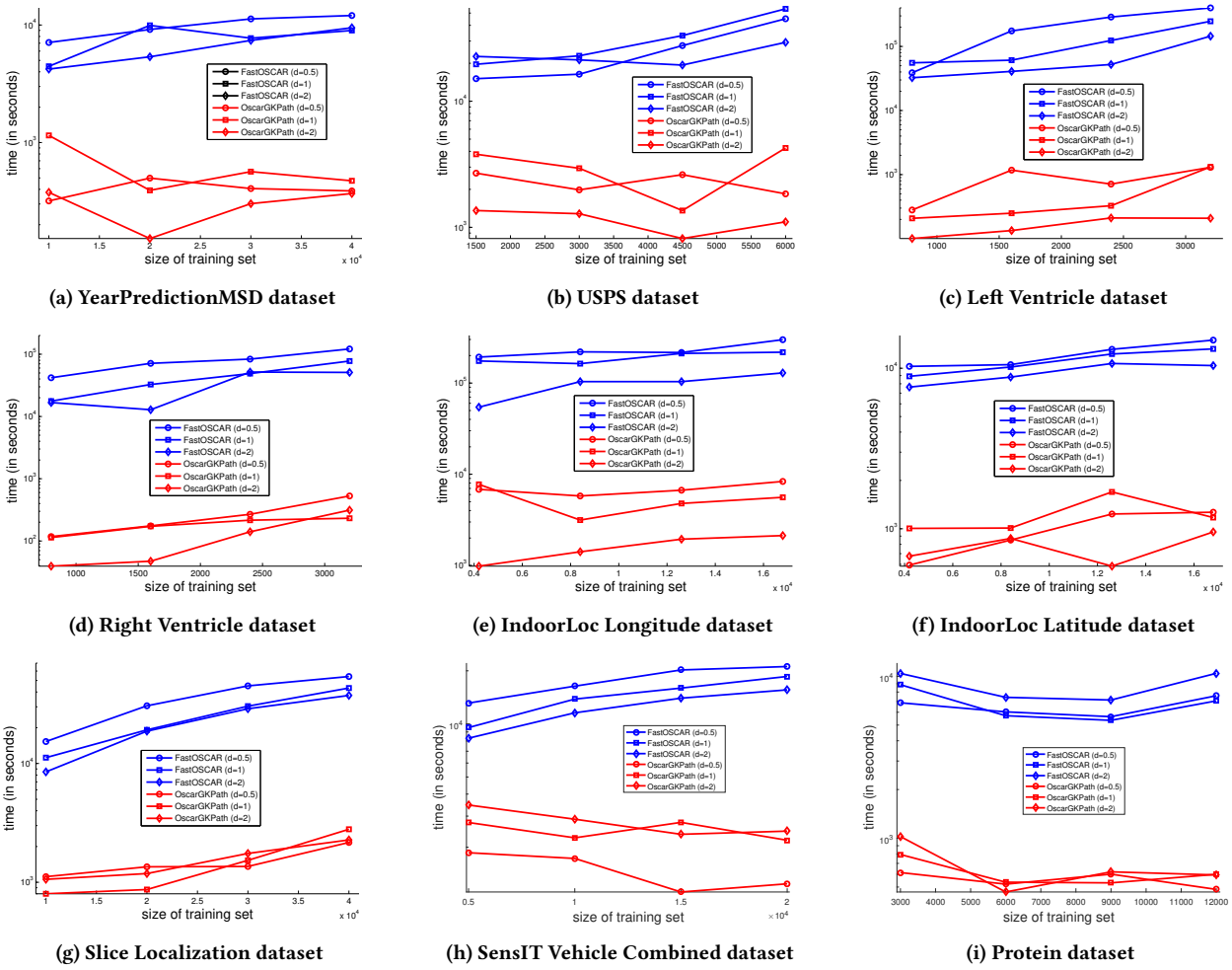
Table 2: Average results of OscarGKPath over 10 trials.

Dataset	Size	*#Iterations*			*#FastOSCAR*		
		$d_2 = 0.5$	$d_2 = 1$	$d_2 = 2$	$d_2 = 0.5$	$d_2 = 1$	$d_2 = 2$
YearPredictionMSD	10,000	2,628	2,742	1,484	8	7	6
	20,000	3,063	2,497	1,694	8	8	7
	30,000	4,012	4,285	2,266	8	8	8
	40,000	3,789	5,128	1,891	8	8	9
USPS	1,500	2,032	2,536	1,874	6	7	7
	3,000	1,865	3,142	2,893	9	9	8
	4,500	3,512	2,892	2,571	10	10	10
	6,000	3,357	4,932	2,974	12	11	11
Left Ventricle	800	7,324	8,731	6,319	10	8	7
	1,600	8,629	10,896	6,232	13	10	9
	2,400	7,149	9,643	6,882	13	12	9
	3,200	9,798	9,921	7,237	14	13	11
Right Ventricle	800	8,163	8,932	6,654	9	8	7
	1,600	8,676	11,324	6,134	11	10	8
	2,400	9,564	9,251	7,234	12	12	10
	3,200	11,941	10,453	8,951	13	12	11
IndoorLoc Longitude	4,200	903	954	502	23	19	12
	8,400	615	891	499	21	18	12
	12,600	469	604	439	24	18	15
	16,800	558	753	695	26	21	11
IndoorLoc Latitude	4,200	7,124	6,212	7,290	6	5	5
	8,400	6,161	7,404	6,260	6	6	5
	12,600	7,908	6,348	4,767	5	5	5
	16,800	5,837	5,210	4,980	5	5	5
Slice Localization	10,000	9,073	8,738	9,346	8	8	7
	20,000	7,330	5,328	7,207	9	8	8
	30,000	4,442	7,620	6,804	8	9	9
	40,000	6,220	9,483	7,896	10	10	9
SensIT Vehicle Combined	5,000	1,844	2,321	2,435	5	6	6
	10,000	1,667	2,073	2,656	5	5	6
	15,000	1,430	2,001	2,092	5	5	5
	20,000	910	1,844	2,012	4	5	5
Protein	3,000	793	937	587	5	4	3
	6,000	765	898	485	6	5	4
	9,000	988	715	689	6	5	4
	12,000	567	815	743	6	6	5

#FastOSCAR for OSCAR over 10 trials with different values of d_2 on different datasets. The results empirically show that only the limited calling of FastOSCAR can produce the entire solution path. Based on the results of *#Iterations* and *#FastOSCAR*, we verify that OscarGKPath is an effective algorithm to fit the entire approximate solution path of OSCAR.

Advantages of OscarGKPath: Fig. 2 plots the running time of OscarGKPath and FastOSCAR in 5-fold cross validation on the YearPredictionMSD, USPS, Left Ventricle, IndoorLoc Longitude, IndoorLoc Latitude, Slice Localization, SensIT Vehicle Combined and Protein datasets with different values of d_2. The results demonstrate that the cross validation based on our OscarGKPath is generally much faster than the one based on FastOSCAR. Because our OscarGKPath only runs five times for the 5-fold cross validation, however, FastOSCAR has to run 400×5 times for the 5-fold cross validation.

Fig. 3 illustrates the cross validation errors and testing errors of OscarGKPath and FastOSCAR for 5-fold cross validation over 10 trials with notched box plot, where the mean squared error (MSE) is used as the performance criterion. From the results, we can find that OscarGKPath performs better than or equally to FastOSCAR on the cross validation errors. This is because that our OscarGKPath fits the entire approximate solution path with a given accuracy bound ε which guarantees that the cross validation error of our OscarGKPath is not worse than the one of FastOSCAR with the same accuracy bound ε. The results of Fig. 3 also confirm the generalization of our OscarGKPath on the testing sets.



(a) YearPredictionMSD dataset (b) USPS dataset (c) Left Ventricle dataset

(d) Right Ventricle dataset (e) IndoorLoc Longitude dataset (f) IndoorLoc Latitude dataset

(g) Slice Localization dataset (h) SensIT Vehicle Combined dataset (i) Protein dataset

Figure 2: Average running time (in seconds) of OscarGKPath and FastOSCAR with 5-fold cross validation over 10 trials.

In addition to showing the advantage of OscarGKPath for model selection, we also validate the advantage of OSCAR for automatic feature grouping. Facebook Comment Volume dataset from the UCI benchmark repository [5] is originally with 54 features. We select 6 most important features by LASSO and duplicate these features 20 times with noise from $N(0, 0.16)$. Fig. 3 presents the absolute coefficients of OSCAR and LASSO on the revised Facebook Comment Volume dataset. The results show that LASSO arbitrarily selects several (maybe zero) of them from a group. However, OSCAR can group the features automatically, and its solution is closer to the ground truth.

6 CONCLUSION

OSCAR is an effective feature selection approach which can automatically group homogenous features in feature selection process. In this paper, we proposed a novel groups-keeping solution path algorithm of OSCAR (OscarGKPath), which can effectively handle the pairwise ℓ_∞-norm in OSCAR, and produce an approximate solution path. More importantly, we theoretically prove that all solutions from OscarGKPath can strictly satisfy a given accuracy bound ε. OscarGKPath can greatly benefit the regularization parameters tuning of OSCAR and generate stable and optimal results. The experimental results on a variety of datasets not only confirm the effectiveness of our OscarGKPath, but also show the superiority of our OscarGKPath in cross validation compared with the existing batch algorithm. In the further, we plan to extend the OscarGKPath algorithm to the general structured sparse learning method with the ℓ_∞-norm [1, 18].

REFERENCES

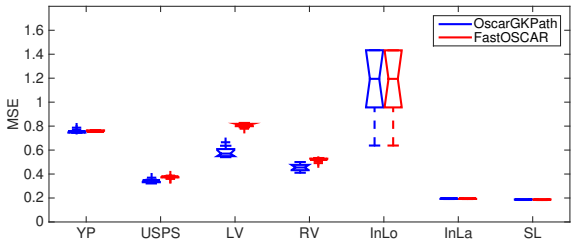
[1] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. 2017. Persistence Images: A Stable Vector Representation of Persistent Homology. *Journal of Machine Learning Research* 18, 8 (2017), 1–35.

[2] Charlotte Møller Andersen and Rasmus Bro. 2010. Variable selection in regressiona tutorial. *Journal of Chemometrics* 24, 11-12 (2010), 728–737.

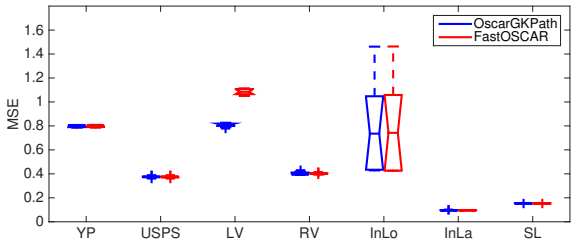
[3] Necdet Aybat, Zi Wang, and Garud Iyengar. 2015. An Asynchronous Distributed Proximal Gradient Method for Composite Convex Optimization. In *Proceedings of The 32nd International Conference on Machine Learning*. 2454–2462.

[4] Francis Bach, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, and others. 2011. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning* 5 (2011).

[5] K. Bache and M. Lichman. 2013. UCI Machine Learning Repository. (2013). http://archive.ics.uci.edu/ml

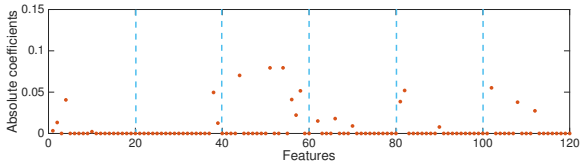


(a) Cross validation errors

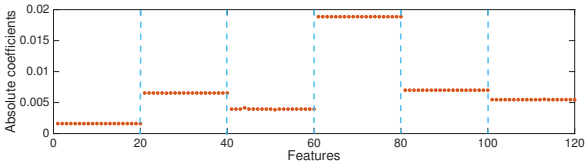


(b) Testing errors

Figure 3: Cross validation errors and testing errors of OscarGKPath and FastOSCAR with 5-fold cross validation over 10 trials.



(a) LASSO



(b) OSCAR

Figure 4: Absolute coefficients of LASSO and OSCAR on the Facebook Comment Volume dataset.

[6] Howard D Bondell and Brian J Reich. 2008. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics* 64, 1 (2008), 115–123.

[7] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

[8] Xiao Cai, Feiping Nie, and Heng Huang. 2013. Exact Top-k Feature Selection via l2,0-Norm Constraint. *23rd International Joint Conference on Artificial Intelligence (IJCAI)* (2013), 1240–1246.

[9] Xiao Cai, Feiping Nie, Heng Huang, and Chris Ding. 2011. Feature Selection via l2,1-Norm Support Vector Machine. In *IEEE International Conference on Data Mining*. 91–100.

[10] Bin Gu and Charles X Ling. 2015. A New Generalized Error Path Algorithm for Model Selection. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. 2549–2558.

[11] Bin Gu, Victor Sheng, Keng Tay, Walter Romano, and Shuo Li. 2016. Cross Validation Through Two-dimensional Solution Surface for Cost-Sensitive SVM. (2016).

[12] B. Gu and V. S. Sheng. 2016. A Robust Regularization Path Algorithm for ν-Support Vector Classification. *IEEE Transactions on Neural Networks and Learning Systems* PP, 99 (2016), 1–8. https://doi.org/10.1109/TNNLS.2016.2527796

[13] Jiang Gui and Hongzhe Li. 2005. Penalized Cox regression analysis in the high-dimensional and low-sample size settings, with applications to microarray gene expression data. *Bioinformatics* 21, 13 (2005), 3001–3008.

[14] Michael J Heller. 2002. DNA microarray technology: devices, systems, and applications. *Annual review of biomedical engineering* 4, 1 (2002), 129–153.

[15] Gabor T Herman. 1990. A survey of 3D medical imaging technologies. *IEEE engineering in medicine and biology magazine: the quarterly magazine of the Engineering in Medicine & Biology Society* 9, 4 (1990), 15.

[16] Rainer Hoffmann, Giuseppe Barletta, Stephan von Bardeleben, Jean Louis Vanoverschelde, Jaroslaw Kasprzak, Christian Greis, and Harald Becher. 2014. Analysis of left ventricular volumes and function: a multicenter comparison of cardiac magnetic resonance imaging, cine ventriculography, and unenhanced and contrast-enhanced two-dimensional and three-dimensional echocardiography. *Journal of the American Society of Echocardiography* 27, 3 (2014), 292–301.

[17] Roberto M Lang, Luigi P Badano, Victor Mor-Avi, Jonathan Afilalo, Anderson Armstrong, Laura Ernande, Frank A Flachskampf, Elyse Foster, Steven A Goldstein, Tatiana Kuznetsova, and others. 2015. Recommendations for cardiac chamber quantification by echocardiography in adults: an update from the American Society of Echocardiography and the European Association of Cardiovascular Imaging. *Journal of the American Society of Echocardiography* 28, 1 (2015), 1–39.

[18] Julien Mairal, Rodolphe Jenatton, Francis R Bach, and Guillaume R Obozinski. 2010. Network flow algorithms for structured sparsity. In *Advances in Neural Information Processing Systems*. 1558–1566.

[19] Feiping Nie, Heng Huang, Xiao Cai, and Chris Ding. 2010. Efficient and Robust Feature Selection via Joint l2,1-Norms Minimization. *NIPS* (2010).

[20] Chong-Jin Ong, Shiyun Shao, and Jianbo Yang. 2010. An improved algorithm for the solution of the regularization path of support vector machine. *IEEE Transactions on Neural Networks* 21, 3 (2010), 451–462.

[21] Mee Young Park and Trevor Hastie. 2007. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69, 4 (2007), 659–677.

[22] David Poole. 2014. *Linear algebra: A modern introduction*. Cengage Learning.

[23] Saharon Rosset and Ji Zhu. 2007. Piecewise linear regularized solution paths. *The Annals of Statistics* (2007), 1012–1030.

[24] Andrzej P Ruszczyński. 2006. *Nonlinear Optimization*. Number 13 in Nonlinear optimization. Princeton University Press.

[25] Yiyuan She. 2010. Sparse regression with exact clustering. *Electronic Journal of Statistics* 4 (2010), 1055–1096.

[26] Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* (1996), 267–288.

[27] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. 2005. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67, 1 (2005), 91–108.

[28] Ryan Joseph Tibshirani, Jonathan E Taylor, Emmanuel Jean Candes, and Trevor Hastie. 2011. *The solution path of the generalized lasso*. Stanford University.

[29] H. Wang, F. Nie, H. Huang, S. Kim, K. Nho, S. L. Risacher, A. J. Saykin, and L. Shen. 2012. Identifying quantitative trait loci via group-sparse multitask regression and feature selection: an imaging genetics study of the ADNI cohort. *Bioinformatics* 28, 2 (2012), 229–37.

[30] Hua Wang, Feiping Nie, Heng Huang, Shannon Risacher, Chris Ding, Andrew J Saykin, Li Shen, and ADNI. 2011. Sparse Multi-Task Regression and Feature Selection to Identify Brain Imaging Predictors for Memory Performance. In *IEEE Conference on Computer Vision*. 557–562.

[31] Larry Wasserman and Kathryn Roeder. 2009. High dimensional variable selection. *Annals of statistics* 37, 5A (2009), 2178.

[32] Seongho Wu, Xiaotong Shen, and Charles J Geyer. 2009. Adaptive regularization using the entire solution surface. *Biometrika* 96, 3 (2009), 513–527.

[33] Leon Wenliang Zhong and James T Kwok. 2012. Efficient sparse modeling with automatic feature grouping. *IEEE transactions on neural networks and learning systems* 23, 9 (2012), 1436–1447.

[34] Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani. 2004. 1-norm support vector machines. *Advances in neural information processing systems* 16, 1 (2004), 49–56.

[35] Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67, 2 (2005), 301–320.