Efficient Approximate Solution Path Algorithm for Order Weight L_1-Norm with Accuracy Guarantee

Runxue Bao[1], Bin Gu[2], Heng Huang[1,2]
[1]Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA, United States
[2]JD Finance America Corporation, Mountain View, CA, United States
Email: runxue.bao@pitt.edu, jsgubin@gmail.com, heng.huang@pitt.edu

***Abstract*—Variable selection is a challenging problem in high-dimensional linear regression problems with a large number of predictors. Thus, sparsity-inducing and clustering-inducing regularization methods are widely used to identify highly correlated covariates. Ordered Weight L_1 (OWL) family of regularizers for linear regression perform well to identify precise clusters of correlated covariates and interpret the effect of each variable. Solution path algorithms are helpful to select hyperparameters to tune the OWL model. Due to over-complex representation of the penalty, so far the OWL model has no solution path algorithms for hyperparameter selection. To address this challenge, in this paper, we propose an efficient approximate solution path algorithm (OWLAGPath) to solve the OWL model with accuracy guarantee. For a given accuracy bound ϵ, OWLAGPath can find the corresponding solutions for the OWL model with numerous hyperparameters while keeping the sparsity and precise features grouping properties. Theoretically, we prove that all the solutions produced by OWLAGPath can strictly satisfy the given accuracy bound ϵ. The experimental results on three benchmark datasets not only confirm the effectiveness and efficiency of our OWLAGPath algorithm, but also show the advantages of OWLAGPath for model selection than the existing algorithms.**

***Index Terms*—Sparse regression, variable selection, hyperparameter selection, solution path algorithm.**

I. Introduction

With the rapid development of data mining and data collection technologies, high-dimensional data widely exists in the real world. High-dimensional data processing and mining are becoming more and more essential in many application scenarios, such as bioinformatics [15], computer vision [5] and financial portfolio [7]. In order to process high-dimensional data with correlated and superfluous features more efficiently and effectively, sparse learning methods which select correlated features [11] are becoming very important.

Recently, a variety of sparsity-inducing feature selection methods [8], [12], [16] are introduced to analyze high-dimensional data. These sparse-inducing regularizers put coefficients of non-relevant features to be zero and thus select the remaining features. However, these methods tend to arbitrarily select only one of the highly correlated features in high-dimensional data. Therefore, the model learned can be unstable and difficult to interpret. To solve the deficiency above, several clustering-inducing methods [13], [17], [18], [21] were proposed. However, the prior information of feature cluster structures is required by the group Lasso model [18] and its variants. The elastic net [21] cannot find the specific clusters. The fused Lasso [13] only can be applicable when features are ordered naturally. Wu et al. [17] only encourages the coefficients of the features with maximum absolute value to be equal. The clustered Lasso is computationally intensive and thus not scalable for large dimension d.

To simultaneously promote sparsity and clustering, a new family of regularizers, termed Ordered Weight L_1-Norms, were proposed in [6], [19], which can acquire the clustering structure of highly correlated features and deselect irrelevant features. Specifically, the coefficients of the features and the corresponding weights (*i.e.*, hyperparameters) in the OWL model are sorted in a non-increasing order. Unlike group Lasso and its variants, the OWL model can find the clustering structures automatically without any prior information of the feature clusters. However, the OWL model involves a large number of hyperparameters and tuning these hyperparameters plays a pivotal role to the performance of the model.

To solve the hyperparameter selection problems, many solution path algorithms [9], [10], [14] were proposed to generate an path of the exact or approximate solutions with the possible values of hyperparameters in the search space. Specifically, [10] proposed a piecewise linear path algorithm for Lasso. [9] proposed a path algorithm for generally l_1-norm regularized linear models. [14] proposed a path algorithm for the generalized Lasso models. However, for the OWL model, the data with dimension d projects a subspace of $\mathbb{R}^d$ for hyperparameter selection. Due to the complexity of the OWL model, so far there is still no path algorithm of the OWL model for model selection. Generally, grid search methods can produce a coarse solution path. However, hyperparameters selection with multi-dimensional grid search could be extremely time-consuming for high-dimensional data with large d and cannot provide any accuracy guarantee for model selection.

To address this challenge, in this paper, we propose an efficient approximate solution path algorithm for the OWL model (OWLAGPath) with accuracy guarantee, which is significantly helpful for the model selection of the family of the OWL model. For a given accuracy bound ϵ, OWLAGPath can find the solutions for the OWL model with numerous hyperparameters while keeping the sparsity and precise features clustering properties during the learning process. Specifically, OWLAGPath can find a series of solutions of the OWL model with the corresponding hyperparameters first and then find a piecewise solution path based on the previous solutions with accuracy guarantee. Theoretically, we rigorously prove that all the solutions in the path can strictly satisfy the given accuracy bound ϵ. The experiments on three benchmark datasets not only confirm the effectiveness and efficiency of our OWLAGPath algorithm, but also show the advantages of OWLAGPath for model selection than the existing algorithms.

II. OWL Regularized Regression

In this section, we will first introduce the OWL norm and then derive the formulation of linear regression problems with the OWL norm considered in this paper.

A. The OWL Norm

The OWL norm is defined as

$$\Omega_{\boldsymbol{\lambda}}(\boldsymbol{\beta}) = \sum_{i=1}^{d} \lambda_i |\beta|_{[i]} = \boldsymbol{\lambda}^T |\boldsymbol{\beta}|_{\downarrow}, \tag{1}$$

where $\boldsymbol{\lambda}$ is a non-negative vector of d non-increasing weights, $\beta_{[i]}$ denotes the i-th largest element of vector $|\boldsymbol{\beta}|$ and $\boldsymbol{\beta}_{\downarrow}$ is the vector that sorts the components of $\boldsymbol{\beta}$ in non-increasing order. The hyperparameter space of the OWL model is a monotone non-negative cone [4] that can be defined as:

$$\mathcal{K}_{m+} = \{\boldsymbol{\lambda} \in \mathbb{R}^d : \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0\} \subset \mathbb{R}^d_+. \tag{2}$$

The OWL regularizer penalizes the regression coefficients according to their magnitude: the larger the magnitude, the larger the penalty. The OWL norm can automatically group highly correlated covariates to make the coefficients associated with these correlated covariates equal. As an extension of l_1-norm, it can also enforce the sparsity of the model.

B. Linear Regression with the OWL Norm

This paper studies classical linear regression problems with the OWL norm under the squared error loss. We consider a training set $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$ where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. The unconstrained formulation of the linear regression with the OWL norm is as follows:

$$F(\boldsymbol{\beta}, \boldsymbol{\lambda}) = \frac{1}{2}\sum_{i=1}^{l}(y_i - \boldsymbol{x}_i^T\boldsymbol{\beta})^2 + \sum_{j=1}^{d}\lambda_j |\beta|_{[j]}, \tag{3}$$

where λ_j is the j-largest non-negative parameter. Each feature has a corresponding hyperparameter.

The formulation (3) is a general formulation of many learning problems. For example, Lasso [12] is a special case of (3) if $\lambda_1 = \lambda_2 = \cdots = \lambda_d$, where $\lambda_j > 0, j = 1, 2, \cdots, d$. Linear regression with L_∞-norm is a special case of the OWL norm if $\lambda_1 > 0$ and $\lambda_2 = \cdots = \lambda_d = 0$. OSCAR [3] is a special case of (3) if $\lambda_j = \alpha_1 + \alpha_2(d-j), j = 1, 2, \cdots, d$, where α_1 and α_2 are non-negative parameters. Each set of parameters correspond to a specific model. Thus, hyperparameter selection plays an key role for OWL regularized regression.

III. An Approximate Solution Path Algorithm for the OWL Model

In this section, we propose an efficient approximate and accuracy-guaranteed solution path algorithm (OWLAGPath) for the OWL regularized regression (3). Supposing the training data $\boldsymbol{x}_i \in \mathbb{R}^d$ has d features, the hyperparameters $\boldsymbol{\lambda} \in \mathcal{K}_{m+}$ of the OWL model have d non-negative parameters with a d dimensional search space. Let $\boldsymbol{d} \in \mathbb{R}^d$ denote the search direction of $\boldsymbol{\lambda}$ in the hyperparameter space, we have $\Delta\boldsymbol{\lambda} = \boldsymbol{d}\Delta\eta$, where $\Delta\eta$ is the adjustment of hyperparameters. Our OWLAGPath algorithm is presented in Algorithm 1.

Algorithm 1 OWLAGPath

Input: Search direction d, accuracy bound ϵ, an search interval $[\underline{\lambda}, \bar{\lambda}]$

```
procedure
    Compute the solution $\boldsymbol{\theta}$ for $\eta = \underline{\lambda}$ based on SLOPE.
    while $d_1\eta \leq \bar{\lambda}$ and $G \leq \epsilon$ do
        Compute the search direction of $\Delta\boldsymbol{\theta}$.
        Compute the maximum adjustment $\Delta\eta^{max}$.
        Update $\eta, \boldsymbol{\theta}, \boldsymbol{\lambda}$ and $\theta_g$.
        Compute the duality gap based on Algorithm 2.
        if $G > \epsilon$ then
            Backtrack to the last piece of solutions that satisfies the error $\epsilon$.
    for $i = 1 : m$ do
        Compute the solution path based on (17).
```

Output: Solution path for the OWL model in $[\underline{\lambda}, \bar{\lambda}]$

Specifically, to produce the solution path, we compute the initial solution for an initial hyperparameter $\boldsymbol{\lambda}$ by SLOPE [2] which is a fast batch algorithm for solving the OWL model. Then, we compute the entire solution path based on the initial solution. To update the hyperparameters, the search direction of the d hyperparameters needs to be computed. To guarantee the accuracy bound of the solutions, the entire solution path is required to satisfy the optimality conditions and thus we compute the maximum adjustment $\Delta\eta^{max}$ of the hyperparameters based on the accuracy bound ϵ. Next, the solution and cluster structures can be updated for the new model. We can produce the a series of solutions by repeating the above procedures. If the last piece of solution path cannot satisfy the error ϵ, we should backtrack the last piece to make sure that the end solution satisfies the error ϵ. By applying (17), we can get the solution path for the whole hyperparameter space with accuracy guarantee.

A. Optimality Conditions of the OWL Model

The ordered term $\sum_{j=1}^{d}\lambda_j|\beta|_{[j]}$ makes it difficult to derive the optimality conditions of the OWL model directly. Based on the sparsity and clustering properties of the OWL model, we give Definition III.1 and derive the equivalent formulation of (3) and the equivalent optimality conditions of the OWL model.

Definition III.1. *Supposing $\boldsymbol{\beta}$ denotes an optimal solution of the OWL model and $o(j) \in \{1, ..., d\}$ denotes the order of $|\beta_j|$ among $\{|\beta_1|, |\beta_2|, ..., |\beta_d|\}$, we have $|\beta_{j_1}| \le |\beta_{j_2}|$ if $o(j_1) < o(j_2)$. Based on the order $o(j)$, the feature cluster is defined as the set $\mathcal{G}_g \subseteq \{1, ..., d\}$ with the same absolute value of the coefficient β_j where $\forall j_1$, $j_2 \in \mathcal{G}_g$, we have $|\beta_{j_1}| = |\beta_{j_2}| \overset{def}{=} \theta_g$.*

According to Definition III.1, we have a series of $\mathcal{G}_g$ with the ordered weights, $g = 1, \cdots, G$, such that $\mathcal{G}_1 \cup \mathcal{G}_2 \cup \cdots \cup \mathcal{G}_G = \{1, ..., d\}$and $\theta_1 > \theta_2 > \cdots > \theta_G \ge 0$. Thus, (3) can be rewritten as (4) in a clustering way:

$$\begin{aligned} &\min_{\theta} \frac{1}{2}\sum_{i=1}^{l}(y_i - \tilde{\boldsymbol{x}}_i^T\theta)^2 + \sum_{g=1}^{G}\omega_g\theta_g \\ &\quad s.t. \quad \theta_1 > \theta_2 > \cdots \ge \theta_G \ge 0, \end{aligned} \tag{4}$$

where $\tilde{\boldsymbol{x}}_i = [\tilde{x}_{i1}\, \tilde{x}_{i2} \cdots \tilde{x}_{iG}]$ and $\tilde{x}_{ig} = \sum_{j\in\mathcal{G}_g} \text{sign}(\beta_j)x_{ij}$, $\omega_g = \sum_{j\in\mathcal{G}_g} \lambda_j$.

According to the sparsity, the coefficients for some features could be 0. For convenience, we can suppose that we have k non-zero clusters. The number of the clusters and the features in each cluster could change as the parameters change. For $\theta_g > 0$ in each cluster $\mathcal{G}_g$, the optimality conditions of (4) can be presented as follows:

$$\sum_{i=1}^{l} -\tilde{x}_{ig}(y_i - \tilde{\boldsymbol{x}}_i^T\theta) + \omega_g = 0, \quad \forall\theta_g > 0 \tag{5}$$

$$\theta_1 > \theta_2 > \cdots > \theta_k > \theta_{k+1} = 0 \tag{6}$$

$$\theta_{k+1} = \theta_{k+2} = \cdots = \theta_G = 0 \tag{7}$$

B. Compute the Search Direction of $\Delta\boldsymbol{\theta}$

To update the parameters, the search direction of d parameters needs to be computed while keeping the optimality conditions of the OWL model. For $\theta_g = 0$ in cluster $\mathcal{G}_g$, according to (5), the value θ_g is fixed to 0 during the whole process. Only for $\theta_g > 0$ in each cluster $\mathcal{G}_g$, they have the possibility to be adjusted when hyperparameter $\boldsymbol{\lambda}$ is changed. Let $\Delta\theta_g$ denote the changes of the solutions $\boldsymbol{\theta}$ in cluster $\mathcal{G}_g$, we can get the following equations:

$$\sum_{i=1}^{l} \tilde{x}_{ig}\tilde{x}_i^T\Delta\theta + \tilde{\omega}_g\Delta\eta = 0, \quad \forall\theta_g > 0, \tag{8}$$

where $\tilde{\omega}_g = \sum_{j\in\mathcal{G}_g} d_j$. Let ξ_g denote $\frac{\Delta\theta_g}{\Delta\eta}$ as the direction of $\Delta\theta_g$ w.r.t. $\Delta\eta$, we can get search directions $\boldsymbol{\xi}$ by solving (8).

C. Compute the Maximum Adjustment of $\Delta\boldsymbol{\theta}$

After obtaining the search direction, we want to search as far as possible within the accuracy bound and thus compute the maximum adjustment $\Delta\eta^{max}$. For each cluster with parameter $\theta_g > 0$, we compute the maximum adjustment $\Delta\eta_g$ when the coefficient θ_g reaches 0 by the constraint $\theta_g + \xi_g\Delta\eta > 0$ according to (6). Thus, $\Delta\eta_1^{max}$ is the smallest one of a set of values that can be solved as follows:

$$\theta_g + \xi_g\Delta\eta_g^{max} = 0, \quad \forall g = 1, 2, \cdots, k \tag{9}$$

Similarly, we compute the maximum adjustment $\Delta\eta_g$ when the coefficient θ_g reaches θ_{g+1} by the constraint $\theta_g + \xi_g\Delta\eta > \theta_{g+1} + \xi_{g+1}\Delta\eta$. Iteratively, $\Delta\eta_2^{max}$ can be solved as follows:

$$\theta_g + \xi_g\Delta\eta_g^{max} = \theta_{g+1} + \xi_{g+1}\Delta\eta_g^{max}, \quad \forall g = 1, 2, \cdots, k \tag{10}$$

Considering the termination condition, $d_1\eta$ with the new η should be smaller than $\bar{\lambda}$. $\Delta\eta_3^{max}$ can be solved as follows:

$$\Delta\eta^{max} = \bar{\lambda} - d_1\eta \tag{11}$$

Finally, we compute the smallest one of values $\Delta\eta_i^{max}$, $i = 1, 2, 3$, and get the maximum adjustment $\Delta\eta^{max}$.

D. Check the Duality Gap

The problem (3) is a convex optimization problem. Let $\boldsymbol{\beta}^*$ be the optimal solution of $F(\boldsymbol{\beta}^*, \boldsymbol{\lambda})$ and $\boldsymbol{\beta}$ be an ϵ-approximation solution with $F(\boldsymbol{\beta}, \boldsymbol{\lambda}) - F(\boldsymbol{\beta}^*, \boldsymbol{\lambda}) \le \epsilon$. The duality gap $G(\boldsymbol{\beta}, \boldsymbol{\lambda})$ is defined as follows:

$$G(\boldsymbol{\beta}, \boldsymbol{\lambda}) = F(\boldsymbol{\beta}, \boldsymbol{\lambda}) - \tilde{F}(\boldsymbol{\alpha}, \boldsymbol{\lambda}) \tag{12}$$

where $\boldsymbol{\alpha}$ is the dual variable, and $\tilde{F}(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ is the dual of $F(\boldsymbol{\beta}^*, \boldsymbol{\lambda})$. We have:

$$F(\boldsymbol{\beta}, \boldsymbol{\lambda}) - F(\boldsymbol{\beta}^*, \boldsymbol{\lambda}) \le F(\boldsymbol{\beta}, \boldsymbol{\lambda}) - \tilde{F}(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = G(\boldsymbol{\beta}, \boldsymbol{\lambda}) \tag{13}$$

Therefore, we can guarantee that the solution $\boldsymbol{\beta}$ produced by our OWLAGPath is a ϵ-approximation solution by $G(\boldsymbol{\beta}, \boldsymbol{\lambda}) \le \epsilon$.

Inspired by [1], we can extend the algorithm in [20] to compute the duality gap of the OWL model. First, $F(\boldsymbol{\beta}, \boldsymbol{\lambda})$ can be computed by (3). According to [1], the dual function in our problem can be computed as follows:

$$\begin{aligned} F(\boldsymbol{\beta}^*, \boldsymbol{\lambda}) &= \max_{\boldsymbol{\alpha}} -\frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{\alpha} - \boldsymbol{\alpha}^T\boldsymbol{y} \\ s.t. &\max_{\sum_{j=1}^{d}\lambda_j|\beta_j|\le 1} \boldsymbol{\alpha}^T X\boldsymbol{\beta} \le 1, \end{aligned} \tag{14}$$

From [20], we know $\boldsymbol{\alpha}$ can be computed as follows:

$$\boldsymbol{\alpha} = \min\{1, \frac{1}{r^*(\boldsymbol{X}^T\nabla f(\boldsymbol{X\beta}))}\}\nabla f(\boldsymbol{X\beta}), \tag{15}$$

where $\nabla f(\boldsymbol{X\beta}) = 2(\boldsymbol{X\beta} - \boldsymbol{y})$. Assuming $\boldsymbol{\gamma}$ are sorted in an decreasing way as $|\gamma_1| \ge |\gamma_2| \ge \cdots \ge |\gamma_d|$, $r^*(\boldsymbol{\gamma})$ can be computed as:

$$r^*(\boldsymbol{\gamma}) = \max_{j\in\{1,2,\cdots,d\}} \frac{\sum_{i=1}^{j}|\gamma_i|}{\sum_{i=1}^{j}\lambda_i}. \tag{16}$$

Here, we complete the process of solving the duality gap. The procedures are summarized in Algorithm 2.

IV. ϵ-APPROXIMATION PROOF OF OWLAGPATH ALGORITHM

In this part, we will prove that any solution generated by OWLAGPath can meet the ϵ accuracy bound. First, I will give the piecewise solution path of OWLAGPath in (17). Supposing $\boldsymbol{\theta}(\eta)$ denotes a solution of the regression task with the OWL model, $\exists\underline{\lambda} = \eta_0 \le \eta_1 \le \eta_2 \le \cdots \le \eta_m = \frac{\bar{\lambda}}{d_1}$, there

Algorithm 2 Duality Gap

```
Input: $\boldsymbol{\beta}, \boldsymbol{\lambda}$
procedure
    Compute $\boldsymbol{\gamma} = \boldsymbol{X}^T \nabla f(\boldsymbol{X\beta})$ and sort $\gamma_i$ in descending order.
    Compute $r^*(\boldsymbol{\gamma})$ based on (16).
    Compute the optimal $\alpha$ of $\tilde{F}(\boldsymbol{\alpha}, \boldsymbol{\lambda})$ based on (15).
    Compute $G(\boldsymbol{\beta}, \boldsymbol{\lambda})$ based on (3), (14) and (12).
Output: The duality gap $G(\boldsymbol{\beta}, \boldsymbol{\lambda})$.
```

are corresponding solutions $\boldsymbol{\theta}(\eta_0), \boldsymbol{\theta}(\eta_1), \boldsymbol{\theta}(\eta_2), \cdots, \boldsymbol{\theta}(\eta_m)$ produced by our OWLAGPath algorithm. We can compute the solution path between η_k and η_{k+1} as follows:

$$\boldsymbol{\theta}(\eta) = \boldsymbol{\theta}(\eta_k) + \boldsymbol{\xi}^k(\eta - \eta_k), \quad \forall \eta \in [\eta_k, \eta_{k+1}] \tag{17}$$

where $k = 0, 1, \cdots, m-1$.

The solutions $\boldsymbol{\theta}(\eta)$ produced by our OWLAGPath is piecewise linear because the solution in each interval $[\eta_k, \eta_{k+1}]$ is linear respectively. Checked in Algorithm 1, we know that the duality gap $G(\boldsymbol{\beta}(\eta_k), \boldsymbol{d}\eta_k)$ and $G(\boldsymbol{\beta}(\eta_{k+1}), \boldsymbol{d}\eta_{k+1})$ of the end points $\boldsymbol{\theta}(\eta_k)$ and $\boldsymbol{\theta}(\eta_{k+1})$ satisfy the accuracy bound. We give Theorem IV.1 as follows:

Theorem IV.1. *For $\forall \eta \in [\eta_k, \eta_{k+1}]$ in the search interval produced by OWLAGPath, we have that the solution $\boldsymbol{\beta}(\eta)$ strictly satisfy $G(\boldsymbol{\beta}(\eta), \boldsymbol{d}\eta) \leq \epsilon$.*

According to Theorem IV.1, we can further conclude that all the solutions produced by our OWLAGPath strictly satisfy $G(\boldsymbol{\beta}(\eta), \boldsymbol{d}\eta) \leq \epsilon$ easily. Here we give the proof of Theorem IV.1.

Proof. We will prove Theorem IV.1 in two cases. First, if $r^*(X^T \nabla f(\boldsymbol{\beta})) < 1$, $\alpha(\eta) = \nabla f(\boldsymbol{X\beta}(\eta))$. We have:

$$\begin{aligned} F(\boldsymbol{\theta}(\eta), \boldsymbol{d}\eta) &= \frac{1}{2}\|\tilde{\boldsymbol{X}}\boldsymbol{\theta}(\eta) - \boldsymbol{y}\|^2 + \sum_{g=1}^{G} \omega_g \theta_g(\eta) \\ &= \frac{1}{2}\|\tilde{\boldsymbol{X}}(\boldsymbol{\theta}(\eta_k) + \boldsymbol{\xi}^k \Delta\eta) - \boldsymbol{y}\|^2 \\ &\quad + \sum_{g=1}^{G} \tilde{\omega}_g(\eta_k + \Delta\eta)(\theta_g(\eta_k) + \xi_g^k \Delta\eta) \\ &= a_1(\Delta\eta)^2 + b_1(\Delta\eta) + c_1 \end{aligned} \tag{18}$$

Meanwhile we have:

$$\begin{aligned} & -\tilde{F}(\boldsymbol{\alpha}(\eta), \boldsymbol{d}\eta) = \frac{1}{2}\boldsymbol{\alpha}(\boldsymbol{\beta}(\eta))^T \boldsymbol{\alpha}(\boldsymbol{\beta}(\eta)) + \boldsymbol{\alpha}(\eta)^T \boldsymbol{y} \\ = \; & 2(\boldsymbol{X}(\boldsymbol{\beta}(\eta_k) + \tilde{\boldsymbol{\xi}}^k \Delta\eta) - \boldsymbol{y})^T (\boldsymbol{X}(\boldsymbol{\beta}(\eta_k) + \tilde{\boldsymbol{\xi}}^k \Delta\eta) - \boldsymbol{y}) \\ & + 2(\boldsymbol{X}(\boldsymbol{\beta}(\eta_k) + \tilde{\boldsymbol{\xi}}^k \Delta\eta) - \boldsymbol{y})^T \boldsymbol{y} \\ = \; & a_2(\Delta\eta)^2 + b_2(\Delta\eta) + c_2 \end{aligned} \tag{19}$$

where $\boldsymbol{\beta}$ can be converted from $\boldsymbol{\theta}$ and $\tilde{\boldsymbol{\xi}}^k$ is the direction of $\Delta\boldsymbol{\beta}$ which can be converted from $\boldsymbol{\xi}^k$. Based on (18) and (19), we can denote $G(\boldsymbol{\beta}(\eta), \boldsymbol{d}\eta)$ as

$$G(\boldsymbol{\beta}(\eta), \boldsymbol{d}\eta) = a(\Delta\eta)^2 + b(\Delta\eta) + c \tag{20}$$

We can easily get $a > 0$ or $a = 0$ because the duality gap $G(\boldsymbol{\beta}(\eta), \boldsymbol{d}\eta) \geq 0$ for all $\eta \geq 0$. Otherwise, we can get $G(\boldsymbol{\beta}(\eta), \boldsymbol{d}\eta) < 0$ for some $\eta > \eta_k$. Thus, the maximum of $G(\boldsymbol{\beta}(\eta), \boldsymbol{d}\eta)$ for $\eta \in [\eta_k, \eta_{k+1}]$ is either $G(\boldsymbol{\beta}(\eta_k), \boldsymbol{d}\eta_k)$ or $G(\boldsymbol{\beta}(\eta_{k+1}), \boldsymbol{d}\eta_{k+1})$. We know the duality gap of the end points strictly satisfy the accuracy bound. Therefore, we complete the proof in the case of $\alpha(\eta) = \nabla f(\boldsymbol{X\beta}(\eta))$.

Similarly, we can proof $G(\boldsymbol{\beta}(\eta), \boldsymbol{d}\eta) \leq \epsilon$ in the case that $r^*(X^T \nabla f(\boldsymbol{\beta})) \geq 1$ and $\alpha = \frac{\nabla f(\boldsymbol{X\beta})}{r^*(\boldsymbol{X}^T \nabla f(\boldsymbol{X\beta}))}$.

Thus, we complete the proof for Theorem IV.1 and prove that all the solutions produced by our OWLAGPath strictly satisfy $G(\boldsymbol{\beta}, \boldsymbol{d}\eta) \leq \epsilon$. □

Table I: The real-world datasets used in the experiments.

Dataset	*Sample size*	*Attributes*
YearPredictionMSD (YP)	51630	90
SensIT Vehicle Combined (SV)	78823	100
Protein	17766	357

V. Experimental Results

In this section, we first give the experimental setup and then present our experimental results with discussions.

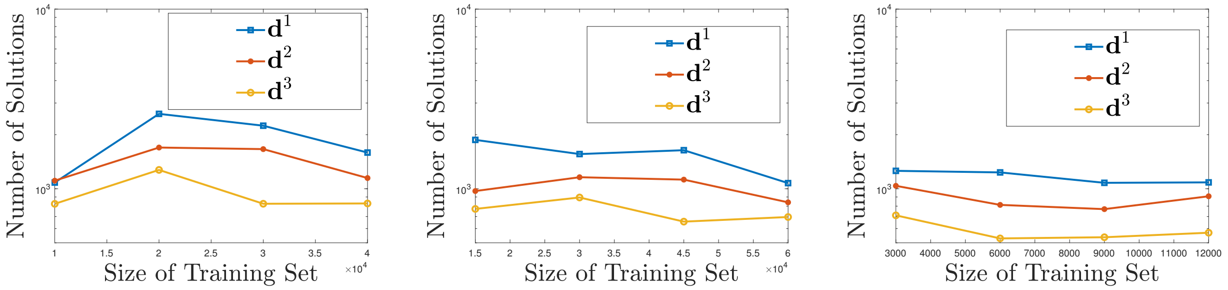
A. Experimental Setup

1) Design of Experiments: We conduct experiments to verify the effectiveness, efficiency and advantages on the generalization of our OWLAGPath algorithm for model selection.

To validate the effectiveness of OWLAGPath, we count the number of solutions produced by our OWLAGPath algorithm to show finite convergence of OWLAGPath. To the best of our knowledge, SLOPE is a fast batch algorithm for solving the OWL model. Grid search methods with SLOPE (denoted as GridSearchSLOPE) can help produce a coarse solution path with different parameters. To verify the efficiency of our OWLAGPath algorithm, we compare the running time of OWLAGPath on different search directions with GridSearchSLOPE. To show the advantage of OWLAGPath on generalization, we compare the cross validation error and testing error of our OWLAGPath and GridSearchSLOPE with 5-fold cross validation.

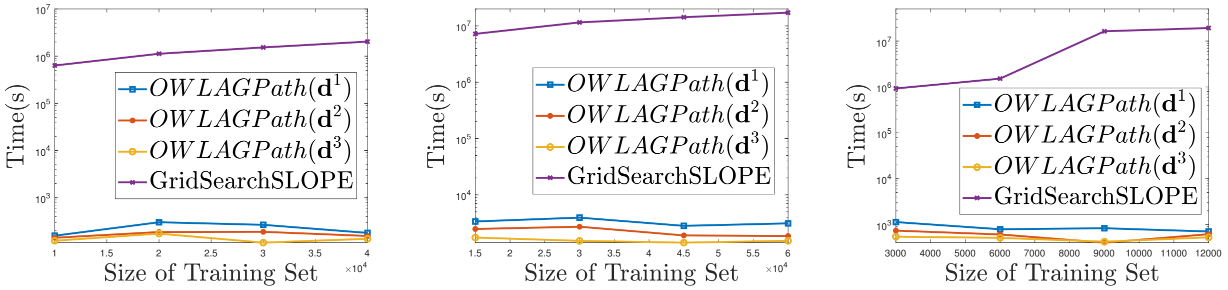
2) Implementation Details: Our experiments were performed on an 4-core Intel i7-6820 machine. We implement our OWLAGPath algorithm in MATLAB. We compare the running time of OWLAGPath and GridSearchSLOPE at the same platform. The duality gap condition in our experiments is set as $G(\boldsymbol{\beta}, \boldsymbol{\lambda}) \leq \epsilon = \alpha * F(\boldsymbol{\beta}, \boldsymbol{\lambda})$ where $\alpha = 0.1$.

GridSearchSLOPE can be done by a multi-dimensional grid search strategy. Empirically, we can choose 0.001 as the lower bound so that the penalty has little influence and choose 1000 as the upper bound so that the penalty can enforce all the coefficients close to 0. We compare our OWLAGPath algorithm with GridSearchSLOPE in the search space bounded as above. To make GridSearchSLOPE more efficient, we can do a coarse search at first and do a fine search for the final



(a) YP (b) SV (c) Protein

Figure 1: Number of solutions produced by our OWLAGPath algorithm *w.r.t.* the size of training set for different search directions.



(a) YP (b) SV (c) Protein

Figure 2: Running time of our OWLAGPath algorithm on different search directions and GridSearchSLOPE algorithms *w.r.t.* the size of training set.

hyperparameters. The coarse search for the parameters can be done on a 6 uniform coarse grid linearly spaced by 1 in the region $\{\log_{10}\lambda_i| - 3 \leq \log_{10}\lambda_i \leq 3\}$ where $i = 1, 2, \dots, k$ and then the fine search can be done on a 10 uniform fine grid linearly space by 0.1 in the $log_{10}\lambda_i$ search space. We do the experiments with different search direction $\boldsymbol{d}$. The representative search direction can be formulated as follows:

$$d_i = q * (k - i) + 1; \quad i = 1, 2, \cdots, k, \tag{21}$$

where q is a parameter to control the search direction, we set $q = 1, 2$ and 3 in our experiments. The directions are denoted as $\boldsymbol{d}^1$, $\boldsymbol{d}^2$ and $\boldsymbol{d}^3$. Please note we can choose any direction that satisfies the setting of the OWL model as search direction for our OWLAGPath algorithm.

To compare the performance of the generalization of OWLAGPath and GridSearchSLOPE, we randomly divide the dataset into training set and testing set in proportion to $4:1$ to test the testing error and divide the training set in proportion to $4:1$ to test the cross validation error similarly.

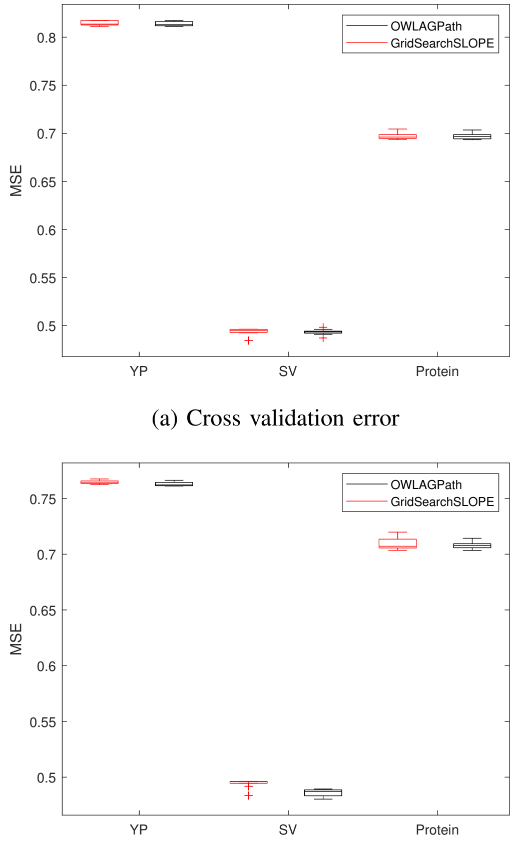
3) Datasets: Table 1 summarizes three benchmark datasets used in our experiments. YearPredictionMSD, SensIT Vehicle Combined and Protein datasets are from the LIBSVM repository which is available at https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

B. Experimental Results and Discussions

1) Effectiveness of OWLAGPath on the Model Selection: Figures 1(a)-(c) present the results of the number of solutions produced by our OWLAGPath over 5 trails. The results show the algorithm can converge with fewer iterations with larger q. This is because larger q can make larger hyperparameter like λ_1 search in the hyperparameter space with a larger step. The results empirically show that OWLAGPath can produce the solution path of the OWL model in finite iterations. The results support the conclusion that OWLAGPath is an effective algorithm to produce the approximate solution path of the OWL model.

2) Efficiency of OWLAGPath on the Model Selection: Figures 2(a)-(c) provide the results of the running time of our OWLAGPath algorithm with three different directions and GridSearchSLOPE over 5 trails on the datasets with different sizes of training set. The results confirm that our OWLAGPath is always much faster than GridSearchSLOPE for model selection. The reasons are as follows. First, our OWLAGPath only need to solve the OWL model by calling SLOPE once in each interval while the GridSearchSLOPE need to solve the OWL model by SLOPE for each set of parameters. Second, our OWLAGPath explores maximum adjustment of the hyperparameters of the OWL model and thus it only need make a small number of adjustments for the whole process.

3) Better Generalization of OWLAGPath on the Model Selection: Figures 3(a)-(b) provide the results of cross validation error and testing error of OWLAGPath and GridSearchSLOPE for 5-fold cross validation over 10 trails. According to the experimental results, our OWLAGPath performs better than or equally to GridSearchSLOPE both on the cross validation error



(a) Cross validation error

(b) Testing error

Figure 3: Cross validation error and testing error of our OWLAGPath algorithm and GridSearchSLOPE.

and testing error. The reason is that grid search methods only do a spaced search and thus cannot provide any accuracy guarantee for model selection. Contrary to that, all the solutions produced by our OWLAGPath can strictly satisfy the given accuracy bound ϵ. To sum up, our proposed OWLAGPath algorithm performs much better than the existing algorithm for model selection with better generalization and much less computational time.

VI. CONCLUSION

Ordered weight L_1 family of regularizers for linear regression perform well in feature selection to generate sparse solutions and identify precise clusters of correlated covariates. It involves a large number of hyperparameters and tuning these hyperparameters plays a pivotal role to the performance of the OWL model. In this paper, we propose an efficient approximate solution path algorithm (OWLAGPath) to solve the OWL model with accuracy guarantee, which can be extremely helpful for tuning the OWL model. For a given accuracy bound ϵ, OWLAGPath can find the solution path for the OWL model with numerous hyperparameters while keeping the precise features grouping and sparsity properties. More importantly, we prove that all the solutions in the solution path produced by OWLAGPath can strictly satisfy the given accuracy bound ϵ by rigorous theoretical analysis. The experimental results on three benchmark datasets not only confirm the effectiveness and efficiency of our OWLAGPath algorithm on model selection, but also show our OWLAGPath has better generalization than the existing algorithms.

ACKNOWLEDGEMENT

This work was partially supported by U.S. NSF IIS 1836945, IIS 1836938, DBI 1836866, IIS 1845666, IIS 1852606, IIS 1838627, IIS 1837956.

REFERENCES

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

[2] M. Bogdan, E. Van Den Berg, C. Sabatti, W. Su, and E. J. Candès. Slopeadaptive variable selection via convex optimization. *The annals of applied statistics*, 9(3):1103, 2015.

[3] H. D. Bondell and B. J. Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with oscar. *Biometrics*, 64(1):115–123, 2008.

[4] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[5] J. Guo, Y. Zhang, Z. Huang, and W. Qiu. Person re-identification by weighted integration of sparse and collaborative representation. *IEEE Access*, 5:21632–21639, 2017.

[6] M. lgorzata Bogdana, E. van den Bergb, W. Suc, and E. J. Candesc. Statistical estimation and testing via the ordered l1 norm. 2013.

[7] K. Morik, S. Turek, and V. Kliewer. Lasso vs. slope: Vergleich und deren praktische umsetzung anhand von camda-und tcga-daten.

[8] F. Nie, H. Huang, X. Cai, and C. H. Ding. Efficient and robust feature selection via joint 2, 1-norms minimization. In *Advances in neural information processing systems*, pages 1813–1821, 2010.

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

[10] S. Rosset and J. Zhu. Piecewise linear regularized solution paths. *The Annals of Statistics*, pages 1012–1030, 2007.

[11] A. Smolinska, L. Blanchet, L. M. Buydens, and S. S. Wijmenga. Nmr and pattern recognition methods in metabolomics: from data acquisition to biomarker discovery: a review. *Analytica chimica acta*, 750:82–97, 2012.

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

[13] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[14] R. J. Tibshirani. *The solution path of the generalized lasso*. Stanford University, 2011.

[15] H. Wang, F. Nie, H. Huang, S. Kim, K. Nho, S. L. Risacher, A. J. Saykin, L. Shen, and A. D. N. Initiative. Identifying quantitative trait loci via group-sparse multitask regression and feature selection: an imaging genetics study of the adni cohort. *Bioinformatics*, 28(2):229–237, 2011.

[16] L. Wasserman and K. Roeder. High dimensional variable selection. *Annals of statistics*, 37(5A):2178, 2009.

[17] S. Wu, X. Shen, and C. J. Geyer. Adaptive regularization using the entire solution surface. *Biometrika*, 96(3):513–527, 2009.

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

[19] X. Zeng and M. A. Figueiredo. Decreasing weighted sorted l_1 regularization. *IEEE Signal Processing Letters*, 21(10):1240–1244, 2014.

[20] L. W. Zhong and J. T. Kwok. Efficient sparse modeling with automatic feature grouping. *IEEE transactions on neural networks and learning systems*, 23(9):1436–1447, 2012.

[21] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, 67(2):301–320, 2005.