Enumerate Lasso Solutions for Feature Selection

Satoshi Hara,[1,3] **Takanori Maehara**[2,3]
1. National Institute of Informatics, Tokyo, Japan
2. Shizuoka University, Shizuoka, Japan
3. JST, ERATO, Kawarabayashi Large Graph Project
satohara@nii.ac.jp, maehara.takanori@shizuoka.ac.jp

Abstract

We propose an algorithm for enumerating solutions to the Lasso regression problem. In ordinary Lasso regression, one global optimum is obtained and the resulting features are interpreted as task-relevant features. However, this can overlook possibly relevant features not selected by the Lasso. With the proposed method, we can enumerate many possible feature sets for human inspection, thus recording all the important features. We prove that by enumerating solutions, we can recover a true feature set exactly under less restrictive conditions compared with the ordinary Lasso. We confirm our theoretical results also in numerical simulations. Finally, in the gene expression and the text data, we demonstrate that the proposed method can enumerate a wide variety of meaningful feature sets, which are overlooked by the global optima.

1 Introduction

Background and Motivation Feature selection is a procedure that selects a subset of relevant features (i.e., variables) for model construction. It plays a central role in many tasks in artificial intelligence and data mining.

One of the most common feature selection methods is Lasso regression (Tibshirani 1996; Chen, Donoho, and Saunders 2001). We consider a prediction problem with n observations and p predictors. Here, we have a response vector $y \in \mathbb{R}^n$ and a predictor matrix $X \in \mathbb{R}^{n \times p}$. The Lasso regression seeks $\beta \in \mathbb{R}^p$ that minimizes ℓ_1-regularized residual sum of squares:

$$L(\beta) := \frac{1}{2}\|X\beta - y\|_2^2 + \rho\|\beta\|_1 \tag{1}$$

where $\rho \in \mathbb{R}_{\geq 0}$ is a regularization parameter. The optimal solution $\beta^* \in \mathbb{R}^p$ to (1) is usually sparse; therefore, we can extract a set of features as the support of the optimal solution, $\mathrm{supp}(\beta^*) = \{i : |\beta_i^*| > 0\}$.

In this study, instead of finding a single optimal solution to the Lasso regression problem, we try to *enumerate good solutions* with different supports in ascending order based on their objective values. "Solutions enumeration" is often used in real applications in various areas such as networks (Brander and Sinclair 1996), databases (Chang et al. 2015), power engineering (Voll et al. 2015), and computational biology (Naor and Brutlag 1994). It has many advantages to enumerate multiple solutions in both theory and applications, as follows.

1. In many real-world tasks, mathematical models include some inaccuracy/approximations. Therefore, the optimal solution to the mathematical model is a good approximation but not necessarily the best solution. By enumerating many solutions, we have a chance to obtain more and better solutions for the real tasks.
2. In many real-world tasks, some constraints are too vague and complex to formalize. Thus, it is hard to incorporate such constraints in the mathematical model. In such a case, enumerating solutions and then selecting the one that satisfies the non-formalized constraints would be a more practical and efficient approach.
3. In theory, the Lasso method can recover the true feature, if some conditions regarding incoherence are satisfied. By enumerating more than one solution, we have a chance to recover the true feature even if these conditions are not satisfied.

Contributions In this study, we make the following contributions:

- We formulate an enumeration version of the Lasso regression problem (Section 3) and propose an algorithm for this problem, which enumerates solutions with different supports in ascending order of objective values (Section 4).
- We prove an exact support recovery theorem: If there exists a sparse linear model, by setting the regularization parameter ρ appropriately, we can obtain a solution that recovers the exact solution by enumerating solutions up to some threshold (Section 6).
- We conduct experiments to evaluate the proposed algorithm using a synthetic dataset and a real-world dataset from computational biology and text categorization (Section 7). The results show that using the proposed method, we can obtain better solutions than the Lasso optimal solution in terms of test error or the solutions tend to involve important features that are absent in the optimal solution.

The proposed algorithm can be easily extended to more general sparsity-inducing convex models such as Lasso Cox regression (Tibshirani 1996), Lasso logistic regression (Lee et al. 2006), elastic-net (Zou and Hastie 2005), fused-Lasso (Tibshirani et al. 2005), and group-Lasso (Yuan and Lin 2006). For simplicity, here, we only consider the standard Lasso problem.

2 Preliminaries

Notation For positive integer p, we denote by $[p] = \{1, 2, \dots, p\}$. For p vector u, $n \times p$ matrix X, and $S \subseteq [p]$, u_S and X_S are the $|S|$ vector indexed by S and the $n \times |S|$ matrix whose columns are indexed by S, respectively. I is the identity matrix. $X^\top$ is the transposed matrix of X. $\|\cdot\|_1$, $\|\cdot\|_2$, and $\|\cdot\|_\infty$ denote ℓ_1, ℓ_2, and ℓ_∞ norm, respectively. The vector $\beta^* \in \mathbb{R}^p$ denotes the minimizer of $L(\beta)$ in (1). We also use $\beta^{(k)}$ to denote the k-th enumerated solution, where $\beta^{(1)} = \beta^*$.

Lasso As described in Section 1, the (regularized) Lasso problem seeks the vector $\beta^* \in \mathbb{R}^p$ that minimizes the ℓ_1-regularized residual sum of squares, (1). This problem is a convex quadratic programming problem. Therefore, we can solve this efficiently using various methods such as proximal gradient method (Boyd and Vandenberghe 2004).

The most important property of Lasso is that it yields a sparse solution. Therefore, the obtained features can be easily interpreted by a human. In theory, under some conditions, Lasso can recover a sparse model (Knight and Fu 2000; Wainwright 2009). See (Hastie, Tibshirani, and Wainwright 2015) for more details of Lasso regression.

3 Problem Formulation

Here, we formulate our enumeration problem.

For a subset $S \subseteq [p]$, we consider the Lasso regression problem $\texttt{Lasso}_=(S)$, where the support is specified by S:

$$\texttt{Lasso}_=(S): \quad \min L(\beta) \text{ s.t. } \operatorname{supp}(\beta) = S. \tag{2}$$

Basically, we want to enumerate solutions to $\texttt{Lasso}_=(S)$ for all $S \subseteq [p]$. However, $\texttt{Lasso}_=(S)$ may not have a solution as the constraint $\operatorname{supp}(\beta) = S$ is discontinuous. Therefore, we relax the equality constraint to the subset constraint and consider the following problem:

$$\texttt{Lasso}(S): \quad \min L(\beta) \text{ s.t. } \operatorname{supp}(\beta) \subseteq S. \tag{3}$$

$\texttt{Lasso}(S)$ can be solved efficiently as it is a Lasso regression problem wherein the variables are restricted to S. In addition, the infimum value of $\texttt{Lasso}_=(S)$ is attained from the optimal value of $\texttt{Lasso}(S)$; see Proposition 2 below. Thus, these problems are essentially equivalent. Therefore, our problem is formulated as follows.

Problem 1 (Lasso Enumeration Problem)**.** Enumerate the top-k different support solutions to $\texttt{Lasso}(S)$ for all $S \subseteq [p]$ in ascending order of their objective function values.

Note that, for different S and S', $\texttt{Lasso}(S)$ and $\texttt{Lasso}(S')$ may produce the same solution. Since we are interested in feature selection, we require our problem to output only different support solutions.

Algorithm 1 Enumeration algorithm

```
Compute $\beta^* \in \texttt{Lasso}([p])$ and insert $(\beta^*, [p], \emptyset)$.
for $k = 1, 2, \dots$ do
    Extract $(\beta, S, F)$ from the heap and output $\beta$ as the $k$-th solution $\beta^{(k)}$ if it is not already output.
    for $i \in \operatorname{supp}(\beta)$ and $i \notin F$ do
        Compute $\beta' \in \texttt{Lasso}(S \setminus \{i\})$ and insert $(\beta', S \setminus \{i\}, F)$ to the heap.
        $F \leftarrow F \cup \{i\}$
    end for
end for
```

Proposition 2.

$$\min\{L(\beta) : \operatorname{supp}(\beta) \subseteq S\} = \inf\{L(\beta) : \operatorname{supp}(\beta) = S\}. \tag{4}$$

Proof. Clearly, we have LHS $\leq$ RHS. Thus, we prove converse inequality. Let $\beta^* \in \mathbb{R}^p$ be an arbitrary solution to LHS. For any $\epsilon > 0$, consider $\beta^* + \delta\vec{1}_S$ for a sufficiently small $\delta > 0$. Then $\beta^* + \delta\vec{1}_S$ is a feasible solution to RHS and $L(\beta^* + \delta\vec{1}_S) \leq L(\beta^*) + \epsilon$. This shows converse inequality. □

4 Algorithm

In this section, we propose an algorithm to solve Problem 1. Efficient implementation is given in a later section. For notational convenience, we denote by $\beta \in \texttt{Lasso}(S)$, if β is the optimal solution to problem $\texttt{Lasso}(S)$.

Our approach follows Lawler's framework (Lawler 1972), which successively computes the optimal solution and then constructs subproblems that exclude the obtained optimal solution.

We maintain a heap (or sorted list) data structure to store tuples of one vector and two subsets, $(\beta, S, F) \in \mathbb{R}^p \times 2^{[p]} \times 2^{[p]}$, where $\beta \in \texttt{Lasso}(S)$. The heap is ordered by the non-decreasing order of the objective function value, $L(\beta)$. F is used to avoid inserting the same set twice to the data structure.

At the beginning of the algorithm, we insert $(\beta^*, [p], \emptyset)$ to the heap, where $\beta^* \in \texttt{Lasso}([p])$. Then, the algorithm repeats the following procedure: For the k-th iteration, the algorithm extracts tuple (β, S, F) and outputs β as the k-th solution if it is not already output. We then "branch" this node to create subproblems that exclude β. The key observation is that, if subset S' satisfies $\operatorname{supp}(\beta) \subseteq S' \subseteq S$, β is also the optimal solution to $\texttt{Lasso}(S')$. Thus, we consider the subsets $S \setminus \{i\}$ for each $i \in \operatorname{supp}(\beta)$. Here, to avoid enumerating the same set multiple times, we avoid branching by index $i \in F$, which was skipped before. This procedure is summarized in Algorithm 1

Correctness To prove the correctness of Algorithm 1, we need to prove the following two claims:

- The algorithm outputs solutions in the non-decreasing order of their objective function values. (Lemma 4)

- For any subset $S \subseteq [p]$, there exists $\beta^{(k)} \in \texttt{Lasso}(S)$. (Lemma 6)

These immediately imply the following result.

Theorem 3. Algorithm 1 solves Problem 1.

In the following we prove the claims.

Lemma 4. Algorithm 1 enumerates solutions $\beta^{(1)}, \beta^{(2)}, \dots$ in the non-decreasing order of objective function values.

Proof. Let $\beta^{(k)}$ and $\beta^{(\ell)}$ ($k < \ell$) be two enumerated solutions. Consider the step when $\beta^{(k)}$ is extracted from the heap. If $\beta^{(\ell)}$ is in the heap, then $L(\beta^{(k)}) \leq L(\beta^{(\ell)})$ by the definition of the heap. Otherwise, there exists $(\beta^{(m)}, S^{(m)}, F^{(m)})$ in the heap such that $\beta^{(\ell)}$ is obtained from branches of this tuple. Since $\beta^{(\ell)}$ is a feasible solution to $\texttt{Lasso}(S^{(m)})$, we have $L(\beta^{(k)}) \leq L(\beta^{(m)}) \leq L(\beta^{(\ell)})$. □

Lemma 5. For any $\beta \in \mathbb{R}^p$, there exists $(\beta^{(k)}, S^{(k)}, F^{(k)})$ such that $\operatorname{supp}(\beta^{(k)}) \subseteq \operatorname{supp}(\beta) \subseteq S^{(k)}$ and $L(\beta^{(k)}) \leq L(\beta)$.

Proof. Let $k = 1$ and consider the k-th extracted element $(\beta^{(k)}, S^{(k)}, F^{(k)})$. We keep invariant $\operatorname{supp}(\beta) \subseteq S^{(k)}$ in the following discussion.

If $\operatorname{supp}(\beta^{(k)}) \subseteq \operatorname{supp}(\beta)$, since β is a feasible solution to $\texttt{Lasso}(S^{(k)})$, we obtain the conclusion. Otherwise, there exists $i \in \operatorname{supp}(\beta^{(k)})$ with $i \notin \operatorname{supp}(\beta)$. Following the algorithm, $(\beta', S^{(k)} \setminus \{i\}, F')$ is inserted to the heap. We increase k to the index where this tuple is extracted and continue this discussion.

Since $S^{(k)}$ decreases monotonically during the discussion, this must terminate after finite iterations, i.e., it must fall into the first situation. This concludes the proof. □

Lemma 6. For any subset $S \subseteq [p]$, there exists $\beta^{(k)} \in \texttt{Lasso}(S)$.

Proof. Let $\beta' \in \texttt{Lasso}(S)$. By Lemma 5, there exists $\beta^{(k)}$ such that $\operatorname{supp}(\beta^{(k)}) \subseteq \operatorname{supp}(\beta')$ and $L(\beta^{(k)}) \leq L(\beta')$. Since $\beta^{(k)}$ is feasible to $\texttt{Lasso}(S)$, it is optimal to $\texttt{Lasso}(S)$. □

Complexity Estimating the computational complexity of Algorithm 1 is difficult because it depends on how many subsets give the same solution (we refer to such a situation as *collision*). Our preliminary experiment shows that collision occurs a small fraction in a practical setting. If the collision occurs at most constant fraction during enumerating the top-k solutions, the complexity is $O(ks\mathcal{A})$ time, where s is the average sparsity of the top-k solutions and $\mathcal{A}$ is the complexity of solving the Lasso regression problem.

5 Efficient Implementation

Here, we describe some implementation technique for Algorithm 1.

Avoiding redundant Lasso computations Consider when we want to insert a new subset S' and its optimal solution $\beta' \in \texttt{Lasso}(S')$ to the heap. If we can identify that the optimal solution β' is already in the heap, *without computing* β', we can avoid redundant Lasso computations. Since solving Lasso regression is expensive, avoiding redundant computations improves practical performance.

Let (β, S) be some element in the heap, where $\operatorname{supp}(\beta) \subseteq S'$. Then, β is a solution to $\texttt{Lasso}(S')$, if and only if,

$$X_{S'}^\top (X\beta - y) \in -\rho \partial \|\beta_{S'}\|_1. \tag{5}$$

This condition can be easily verified by evaluating both sides, which is more efficient than solving the Lasso regression problem. In particular, if we compute $\theta^{(k)} = X\beta^{(k)} - y$ when we insert $\beta^{(k)}$ to the heap, the condition can be evaluated in $O(|X_{S'}| + |S'|)$ time, where $|X_{S'}|$ is the number of nonzero elements in $X_{S'}$.

Warm start on branching Consider when (β, S) is extracted and $\texttt{Lasso}(S \setminus \{i\})$ is evaluated to insert a new solution. Then, we observe that the optimal solution to $\texttt{Lasso}(S \setminus \{i\})$ is often close to β. To exploit this property, we can reuse β as an initial solution to $\texttt{Lasso}(S \setminus \{i\})$, where β_i is replaced by zero.

Parallel evaluation We can perform Lasso evaluations for all $i \in \operatorname{supp}(\beta) \setminus F$ in Line 5 in parallel.

6 Support Recovery Theorem

In this section, we assume that there exists a sparse model

$$y = X\beta^\circ + w \tag{6}$$

where β° is an s-sparse vector and $w \in \mathbb{R}^n$ is a Gaussian noise with mean zero and variance σ^2, $N(0, \sigma^2)$. We show that Algorithm 1 can find a solution $\beta^{(k)}$ where $\operatorname{supp}(\beta^{(k)}) = \operatorname{supp}(\beta^\circ)$, under a suitable choice of the regularization parameter ρ (Theorem 9, Theorem 10).

First, we show that by enumerating solutions up to $L(\beta^\circ) \leq L(\beta^{(\ell)})$, we find $\beta^{(k)}$ as follows. This is a direct consequence of Lemma 5.

Lemma 7. By enumerating solutions up to $L(\beta^{(\ell)}) \geq L(\beta^\circ)$, we can find $\beta^{(k)}$ $(1 \leq k \leq \ell)$ such that $\operatorname{supp}(\beta^{(k)}) \subseteq \operatorname{supp}(\beta^\circ) \subseteq S^{(k)}$ and $L(\beta^{(k)}) \leq L(\beta^\circ)$.

Proof. By Lemma 5 applied to β°, we can find $\beta^{(k)}$ with $L(\beta^{(k)}) \leq L(\beta^\circ)$ and $\operatorname{supp}(\beta^{(k)}) \subseteq \operatorname{supp}(\beta^\circ) \subseteq S^{(k)}$. Since $L(\beta^{(k)}) \leq L(\beta^\circ) \leq L(\beta^{(\ell)})$ with Lemma 4, we have $k \leq \ell$. □

Next, we bound $L(\beta)$ in terms of problem parameters.

Lemma 8. Let β° be an s-sparse vector supported by S°, and β^* be the Lasso optimal solution. Suppose the following.

C1. $\|X\beta^\circ - y\|_2 \leq \delta \|X\beta^* - y\|_2$ for some $\delta \geq 0$.

C2. $\|X\beta^* - y\|_2 \leq \epsilon$ for some $\epsilon \geq 0$.

C3. For every $u \neq 0$ with $\|Xu\|_2 \le (1+\delta)\epsilon$, $\|u_{S^\circ}\|_1 < \gamma\|u_{S^{\circ c}}\|_1$ for some $\gamma \ge \max\{1, \delta^2\}$.

Then, we have $L(\beta^\circ) \le \gamma L(\beta^*)$.

Proof. Let $u = \beta^\circ - \beta^*$. Then

$$\|Xu\|_2 \le \|X\beta^\circ - y\| + \|X\beta^* - y\| \le (1+\delta)\epsilon. \tag{7}$$

Therefore, u satisfies the condition C3. Since

$$\|u_{S^\circ}\|_1 = \|\beta^\circ_{S^\circ} - \beta^*_{S^\circ}\|_1 \ge \|\beta^\circ\|_1 - \|\beta^*_{S^\circ}\|_1, \tag{8}$$

$$\|u_{S^{\circ c}}\|_1 = \|\beta^*_{S^{\circ c}}\|_1, \tag{9}$$

we have

$$\|\beta^\circ\|_1 \le \|\beta^*_{S^\circ}\|_1 + \gamma\|\beta^*_{S^{\circ c}}\|_1 \le \gamma\|\beta^*\|_1. \tag{10}$$

Therefore,

$$\begin{aligned} L(\beta^\circ) &= \frac{1}{2}\|X\beta^\circ - y\|_2^2 + \rho\|\beta^\circ\|_1 \\ &\le \frac{\delta^2}{2}\|X\beta^* - y\|_2^2 + \rho\gamma\|\beta^*\|_1 \\ &\le \gamma\left(\frac{1}{2}\|X\beta^* - y\|_2^2 + \rho\|\beta^*\|_1\right) \\ &= \gamma L(\beta^*) \end{aligned} \tag{11}$$

□

Combining Lemmas 7 and 8, we obtain the following theorem.

Theorem 9 (No false inclusion)**.** Assume the same condition as in Lemma 8. Then, by enumerating solutions up to $L(\beta^{(\ell)}) \ge \gamma L(\beta^*)$, we can find $(\beta^{(k)}, S^{(k)})$ $(1 \le k \le \ell)$ such that $\mathrm{supp}(\beta^{(k)}) \subseteq \mathrm{supp}(\beta^\circ) \subseteq S^{(k)}$ and $L(\beta^{(k)}) \le L(\beta^\circ)$.

This theorem is useful to identify the difficulty of a given instance, i.e., the required number of solutions for support recovery. See discussion at the end of this section.

Theorem 10 (No false exclusion)**.** Let $(\beta^{(k)}, S^{(k)})$ be enumerated solution where $\mathrm{supp}(\beta^{(k)}) \subseteq \mathrm{supp}(\beta^\circ) \subseteq S^{(k)}$. If $X_{S^\circ}^\top X_{S^\circ}$ is invertible, then we have

$$\mathrm{supp}(\beta^{(k)}) \supseteq \{i : \|\beta_i^\circ\| > 2\rho\|(X_{S^\circ}^\top X_{S^\circ})^{-1}\|_\infty\} \tag{12}$$

with probability $1 - |S^\circ|\exp(-\rho^2/2\sigma\sqrt{\lambda_{\max}(X_{S^\circ}^\top X_{S^\circ})})$.

Proof. The following proof is the same as that for the support consistency theorem for the standard Lasso regression. Let $u = \beta^{(k)} - \beta^\circ$ and $\overline{S} := S^{(k)} \setminus S^\circ$. Since $\beta^{(k)}$ is the optimal solution to $\mathtt{Lasso}(S^{(k)})$, we have the following subgradient characterization

$$X_{S^{(k)}}^\top(Xu - w) = \rho z \tag{13}$$

where $z \in \partial\|\beta^{(k)}_{S^{(k)}}\|_1 \subseteq [0,1]^{S^{(k)}}$. This is written as

$$\begin{bmatrix} X_{S^\circ}^\top X_{S^\circ} & X_{S^\circ}^\top X_{\overline{S}} \\ X_{\overline{S}}^\top X_{S^\circ} & X_{\overline{S}}^\top X_{\overline{S}} \end{bmatrix} \begin{bmatrix} u_{S^\circ} \\ 0 \end{bmatrix} - \begin{bmatrix} X_{S^\circ}^\top w \\ X_{\overline{S}}^\top w \end{bmatrix} = \rho \begin{bmatrix} z_{S^\circ} \\ z_{\overline{S}} \end{bmatrix}. \tag{14}$$

Therefore,

$$u_{S^\circ} = (X_{S^\circ}^\top X_{S^\circ})^{-1}X_{S^\circ}^\top w + \rho(X_{S^\circ}^\top X_{S^\circ})^{-1}z_{S^\circ}. \tag{15}$$

Here, we evaluate ℓ_∞-norm of u_{S°. By the triangle inequality and the definition of matrix norm, we have

$$\|u_{S^\circ}\|_\infty \le \|(X_{S^\circ}^\top X_{S^\circ})^{-1}\|_\infty \left(\|X_{S^\circ}^\top w\|_\infty + \rho\right). \tag{16}$$

Since $X_{S^\circ}^\top w$ follows a Gaussian distribution of mean zero and covariance $\sigma^2(X_{S^\circ}^\top X_{S^\circ})$, using the union bound, we have $\|X_{S^\circ}^\top w\|_\infty \ge \rho$ with probability $|S^\circ|\exp(-\rho^2/2\sigma\sqrt{\lambda_{\max}(X_{S^\circ}^\top X_{S^\circ})})$. Therefore,

$$\|u\|_\infty = \|u_{S^\circ}\|_\infty \le 2\rho\|(X_{S^\circ}^\top X_{S^\circ})^{-1}\|_\infty \tag{17}$$

with probability $1 - |S^\circ|\exp(-\rho^2/2\sigma\sqrt{\lambda_{\max}(X_{S^\circ}^\top X_{S^\circ})})$. This implies the proposition. □

The condition C3 in Lemma 8 is a relaxation of the nullspace property (Cohen, Dahmen, and DeVore 2009): For all $u \neq 0$ with $Xu = 0$,

$$\|u_{S\circ}\| < \|u_{S^{\circ c}}\|. \tag{18}$$

The nullspace property is a necessary and sufficient condition for the unique recovery theorem in compressed sensing. In our theorem, the nullspace condition is almost equivalent to $\gamma = 1$, which implies that we can recover the solution by enumerating only a single solution. This is the standard exact support recovery theorem.

When $\gamma > 1$, we need to enumerate multiple solutions to recover the support. In particular, if γ is very large, we need to enumerate many solutions. Here, we describe two typical situations that make γ large, depending on the regularization parameter ρ and the data collinearity of the matrix X.

A1. If the regularization parameter ρ is very small, we have a chance to obtain β^* with very small $\|X\beta^* - y\|_2$. Thus, δ in the condition C1 becomes large; hence, γ becomes large.

A2. If some columns of X are nearly collinear, γ becomes large to satisfy the inequality of the condition C3.

It should be emphasized that in the standard support recovery theorem, we cannot obtain anything, if the conditions are not satisfied. Conversely, our conditions C1–C3 are usually satisfied for a sufficiently large γ. Therefore, our algorithm can recover the support (but requires many enumerations in a bad situation).

Note that we cannot identify which $\beta^{(k)}$ recovers the support of β° only by this procedure, which is an issue of our formulation. To select an adequate solution, we need additional criteria (e.g., test error, interpretability) to the problem.

7 Experiments

We conducted experiments to evaluate the proposed algorithm. All codes were implemented in Python 3.5 with scikit-learn[1]. All experiments were conducted on 64-bit CentOS 6.7 with an Intel Xeon E5-2670 2.6GHz CPU and 512GB RAM.

[1] The experiment codes are available at https://github.com/sato9hara/LassoVariants

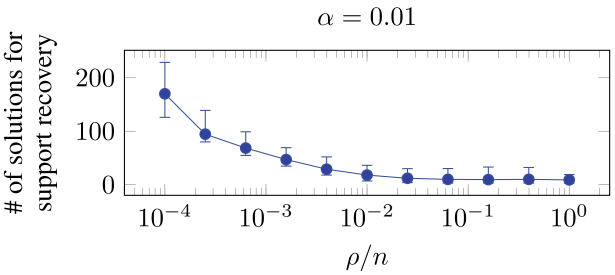


Figure 1: ρ versus the number of solutions required to exact support recovery. The points indicate the median, and the error bars show the 25% and 75% percentiles after 100 trials.

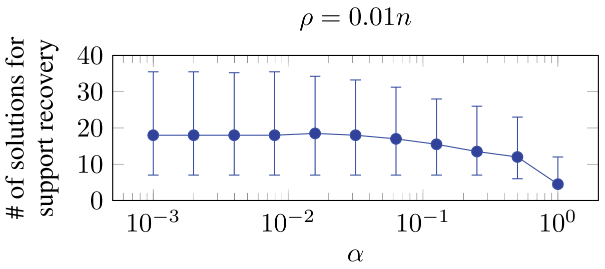


Figure 2: α versus the number of solutions required to exact support recovery. The points indicate the median, and the error bars the show 25% and 75% percentiles after 100 trials.

Synthetic Dataset

We first evaluate the proposed algorithm using a synthetic dataset. The input X is constructed by $X = UA$, where each element of $U \in \mathbb{R}^{n \times p}$ is drawn from a standard normal distribution $N(0, 1)$, and the matrix $A \in \mathbb{R}^{p \times p}$ is given by

$$A = (1 - \alpha)VV^\top + \alpha I, \qquad (19)$$

where $V \in \mathbb{R}^{p \times q}$ is randomly drawn from $N(0, 1)$. The parameter $\alpha \in [0, 1]$ controls the degree of collinearity among the features; the features are highly collinear when α is small, and they are independent when $\alpha = 1$. In the experiment, we set the number of features $p = 10$, the number of samples $n = 100$, and the dimension of V to be $q = 5$. We also set the true parameter β° to be $\beta_1^\circ = \beta_2^\circ = 1$ and $\beta_i^\circ = 0$ otherwise.

Required number of enumerations for exact support recovery We observed the number of enumerations required to exact support recovery. We varied the regularization parameter ρ and the collinearity parameter α. The results are shown in Figures 1 and 2, respectively, which coincide with our theoretical analysis.

Figure 1 shows that if ρ is larger than $0.01n$, we can recover the true support by enumerating only a few solutions. By contrast, the number of required solutions increases as ρ decreases. This matches the results of our analysis A1.

Similarly, Figure 2 also shows that we can recover the true support by enumerating only a few solutions, if α is close to one. By contrast, the number of required solutions increases when α is small, i.e., when the features are collinear. This agrees with our analysis A2.

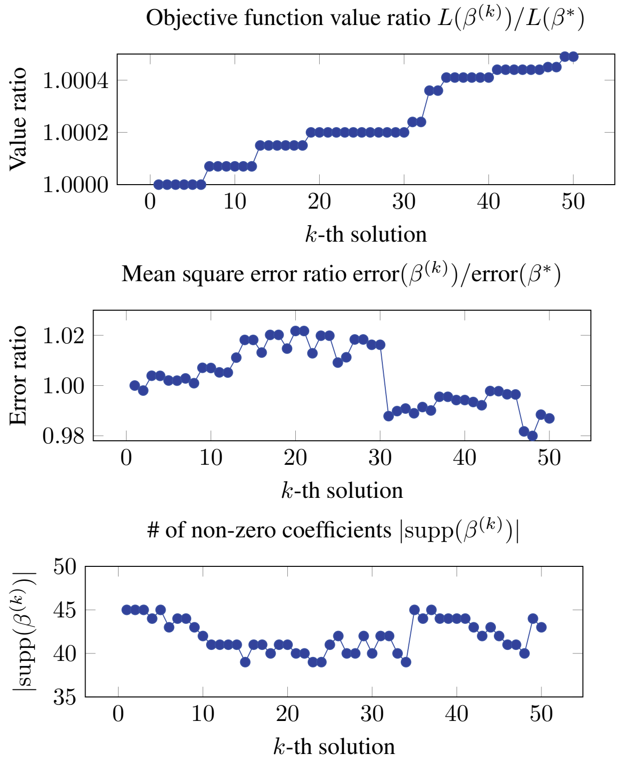
Real Dataset

We applied the proposed method to two real-world datasets. In the experiments, we demonstrate two advantages of the solution enumeration. First, in the gene expression data, we show that through enumeration, we can obtain solutions that predict better than the Lasso optimal solution can. Next, in the 20 newsgroups text data, we show that the proposed algorithm enumerates many interesting features that are missed in the Lasso optimal solution.

When applying the proposed method, we adopted one practical modification to Algorithm 1; we replaced $\mathrm{supp}(\beta)$ in line 4 of Algorithm 1 with $\mathrm{supp}_{>\eta}(\beta) = \{i : |\beta_i| > \eta\}$ with some non-negative η. This modification enabled us to avoid enumerating uninteresting solutions. In the original Algorithm 1, when the k-th solution $\beta^{(k)}$ had a small i-th entry $\beta_i^{(k)} \approx 0$, it was likely that the $k+1$-th solution $\beta^{(k+1)}$ was almost equal to $\beta^{(k)}$, except that the i-th entry was set to exactly zero. Although the support of these two solutions are different, they are almost identical as a solution. With the modified algorithm, we can skip enumerating such almost identical solutions and can enumerate solutions that differ more significantly.

Gene Expression Data We used the thaliana gene expression data used in (Atwell et al. 2010). The data comprises 216,130 genes over 199 different samples. In this experiment, we focused on the FLC gene expression as the response y, which is the one related to flowering. As the feature vector x, we used the majority-based expression. For each gene, we computed the majority out of four genes (A, T, G, and C), and set the i-th feature x_i to be zero if the gene was same as the majority, and x_i to be 1 otherwise. After removing data with missing values, we obtained 167 samples with 216,130 dimensional feature vectors. For the evaluation purpose, we randomly split samples into 134 training samples and 33 test samples.

For the training set, we applied Algorithm 1 with the regularization parameter $\rho = 0.1n$ and the support parameter $\eta = 0.05$. Figure 3 shows that the enumerated top-50 solutions $\beta^{(k)}$ had competitive qualities with the Lasso optimal solution β^*. The increase in the objective function values was limited to up to 0.05%, and the change of the test error was limited to up to ± 2%. It is noteworthy that the solutions after 30 enumerations had smaller test mean square errors compared with the Lasso optimal solution. That is, by enumerating solutions, we obtained a better solution with a smaller test error. It is also interesting to find that such solutions had smaller number of non-zero coefficients compared with the Lasso optimal solution. For instance, $\beta^{(34)}$ had only 39 non-zero coefficients, whereas the Lasso optimal solution β^* had 45 non-zeros: $\beta^{(34)}$ had two additional genes from β^* with eight genes removed. This result implies that, by focusing only on the Lasso optimal solution, we may overlook important features relevant to prediction.


Objective function value ratio $L(\beta^{(k)})/L(\beta^*)$

Mean square error ratio error$(\beta^{(k)})$/error(β^*)

# of non-zero coefficients $|\mathrm{supp}(\beta^{(k)})|$

Figure 3: [Gene Expression] Changes in the objective function values, the test mean square errors, and the number of non-zero coefficients over the enumerated solutions.

20 Newsgroups Data The 20 Newsgroups [2] is a dataset for text categorization. In this experiment, we tried to find discriminative words between the two categories `ibm.pc.hardware` and `mac.hardware`. As a feature vector x, we used tf-idf weighted bag-of-words expression, with stop words and some common verbs removed. The training set comprised $n = 1,168$ samples with $p = 11,648$ words, whereas the test set consisted of $n' = 777$ samples. The task was to find discriminative words that were relevant to classification from these 11,648 words.

Because the task was binary classification between the two categories, we used Lasso logistic regression instead of the ordinary Lasso regression (1). The Lasso logistic regression objective function is defined by

$$L(\beta) := \sum_{i=1}^{n} \log(\exp(-y_i x_i^\top \beta) + 1) + \rho\|\beta\|_1, \qquad (20)$$

where $y_i \in \{-1, 1\}$ is a category indicator. We note that even if we replace the objective function, Algorithm 1 is still valid, and we can enumerate solutions with different supports.

Using Algorithm 1, we enumerated top-50 solutions with the regularization parameter $\rho = 0.001n$ and the support parameter $\eta = 4$. In the first solution β^*, 39 words were selected as relevant for classification. We compared the Lasso optimal solution β^* and the latter 49 solutions $\beta^{(2)}, \beta^{(3)}, \ldots, \beta^{(50)}$. For each solution $(\beta^{(k)}, S^{(k)})$ outputted from the heap, we picked up the following two sets:

Table 1: Words replacements in enumerated solutions.

Original words		Replaced	Subject
bios	→	drive	ibm
ide	→	drive	ibm
dos	→	os, drive	ibm
controller	→	drive	ibm
quadra, centris	→	040, clock	mac
windows, bios, controller	→	disk, drive	ibm
bios, help, controller	→	disk, drive	ibm
centris, pc	→	610	mac

- $B^{(k)} = \mathrm{supp}(\beta^*) \setminus S^{(k)}$: the words removed from β^*,
- $C^{(k)} = \mathrm{supp}(\beta^{(k)}) \setminus \mathrm{supp}(\beta^*)$: the words added to $\beta^{(k)}$.

We then extracted the minimal sets from $\{B^{(k)}, C^{(k)}\}_{k=2}^{50}$, and summarized them into Table 1. Here, the minimal set means that we removed redundant solutions such as $B^{(k)} = \{\text{bios}, \text{ide}\}$, $C^{(k)} = \{\text{drive}\}$ because this word replacement could be explained by $B^{(\ell)} = \{\text{bios}\}$, $C^{(\ell)} = \{\text{drive}\}$ and $B^{(\ell')} = \{\text{ide}\}$, $C^{(\ell')} = \{\text{drive}\}$. Table 1 shows that the words replacements could be categorized into two subjects: one for the words relevant to `ibm.pc.hardware`, and the other for the words relevant to `mac.hardware`. The table indicates that the enumerated solutions were meaningful and diverse, in a sense that the words were replaced with some other relevant words.

We also note that the enumerated top-50 solutions $\beta^{(k)}$ had competitive qualities with the Lasso optimal solution β^* similar to the gene expression data. The increase in the objective function values was limited to up to 2%, and the increase in the test error was limited to up to 4%.

8 Conclusion

We proposed an algorithm to enumerate solutions to the Lasso regression problem. With the algorithm, we could enumerate solutions with different supports in ascending order of their objective function values. We also proved that we could recover the true feature set exactly under less restrictive conditions compared with the ordinary Lasso. The experimental results on the synthetic and real-world datasets demonstrated several advantages of the solution enumeration; the enumerated solutions exhibited a smaller test error, or the solutions tended to involve important yet missing features in the optimal solution.

Acknowledgment

The authors would like to thank Dr. Jun Sese and Dr. Aika Terada for their supports and helpful comments on the thaliana gene expression data experiment.

References

Atwell, S.; Huang, Y. S.; Vilhjálmsson, B. J.; Willems, G.; Horton, M.; Li, Y.; Meng, D.; Platt, A.; Tarone, A. M.;

[2] http://qwone.com/~jason/20Newsgroups/

Hu, T. T.; et al. 2010. Genome-wide association study of 107 phenotypes in arabidopsis thaliana inbred lines. *Nature* 465(7298):627–631.

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

Brander, A. W., and Sinclair, M. C. 1996. A comparative study of k-shortest path algorithms. In *Performance Engineering of Computer and Telecommunications Systems*. Springer. 370–379.

Chang, L.; Lin, X.; Zhang, W.; Yu, J. X.; Zhang, Y.; and Qin, L. 2015. Optimal enumeration: Efficient top-k tree matching. *Proceedings of the VLDB Endowment* 8(5):533–544.

Chen, S. S.; Donoho, D. L.; and Saunders, M. A. 2001. Atomic decomposition by basis pursuit. *SIAM review* 43(1):129–159.

Cohen, A.; Dahmen, W.; and DeVore, R. 2009. Compressed sensing and best -term approximation. *Journal of the American mathematical society* 22(1):211–231.

Hastie, T.; Tibshirani, R.; and Wainwright, M. 2015. *Statistical learning with sparsity: the lasso and generalizations*. CRC Press.

Knight, K., and Fu, W. 2000. Asymptotics for lasso-type estimators. *Annals of statistics* 1356–1378.

Lawler, E. L. 1972. A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem. *Management science* 18(7):401–405.

Lee, S.-I.; Lee, H.; Abbeel, P.; and Ng, A. Y. 2006. Efficient l1 regularized logistic regression. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, 401. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Naor, D., and Brutlag, D. L. 1994. On near-optimal alignments of biological sequences. *Journal of Computational Biology* 1(4):349–366.

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

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

Voll, P.; Jennings, M.; Hennen, M.; Shah, N.; and Bardow, A. 2015. The optimum is not enough: A near-optimal solution paradigm for energy systems synthesis. *Energy* 82:446–456.

Wainwright, M. J. 2009. Sharp thresholds for high-dimensional and noisy sparsity recovery using-constrained quadratic programming (lasso). *IEEE transactions on information theory* 55(5):2183–2202.

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

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