SOR: Scalable Orthogonal Regression for Non-Redundant Feature Selection and its Healthcare Applications

Dijun Luo∗ Fei Wang† Jimeng Sun† Marianthi Markatou† Jianying Hu† Shahram Ebadollahi†

Abstract
As more clinical information with increasing diversity become available for analysis, a large number of features can be constructed and leveraged for predictive modeling. Feature selection is a classic analytic component that faces new challenges due to the new applications: How to handle a diverse set of high dimensional features? How to select features with high predictive power, but low redundant information? How to design methods that can select globally optimal features with theoretical guarantee? How to incorporate and extend existing knowledge driven approach? In this paper, we present Scalable Orthogonal Regression (SOR), an optimization-based feature selection method with the following novelties: 1) Scalability: SOR achieves nearly linear scale-up with respect to the number of input features and the number of samples; 2) Optimality: SOR is formulated as an alternative convex optimization problem with theoretical convergence and global optimality guarantee; 3) Low-redundancy: thanks to the orthogonality objective, SOR is designed specifically to select less redundant features without sacrificing quality; 4) Extendability: SOR can enhance an existing set of preselected features by adding additional features that complement the existing feature set but still with strong predictive power. We present evaluation results showing that SOR consistently outperforms state of the art feature selection methods in a range of quality metrics on several real world data sets. We demonstrate a case study of a large-scale clinical application for predicting early onset of Heart Failure (HF) using real Electronic Health Records (EHRs) data of over 10K patients for over 7 years. Leveraging SOR, we are able to construct accurate and robust predictive models and derive potential clinical insights.

1 Introduction
Recent trends in healthcare and medicine enhance traditional knowledge driven approaches with data extracted information, considered together with knowledge for making treatment and other decisions. As more and more comprehensive EHR data become available, a diverse set of clinical features can be constructed and potentially leveraged for clinical decision support applications. From both theoretical and application perspectives, feature selection is a key component with a lot of challenges.

From statistics and machine learning research, feature selection provides many benefits: 1) speed up the subsequent learning process, 2) improve the model generalizability and alleviate the effect of the curse of dimensionality [12] and overfitting [19]. A large number of feature selection methods have been proposed in the literature [11, 10, 23, 24, 22] and there are many recent reviews and workshops devoted to this topic, e.g., NIPS Conference [7]. Despite the vast literature on feature selection, the problem is by no algorithms solved. Many practical feature selection are developed in the context of concrete applications, such as Bioinformatics applications[3, 18]. A survey on various feature selection methods and applications are presented in Section 2.

Our motivating healthcare application and its associated new challenges for feature selection are presented next.

Motivating example: EHR data provide a longitudinal view of patients. This typically includes diagnosis info such as ICD9 codes, medication info such as drug names, lab results and symptoms. EMR data have been growing rapidly in quantity over the past few years, and are increasingly considered to be a valuable asset by leading medical institutions. Predictive modeling using EHRs for targeted high cost diseases has become highly valuable in modern healthcare. One high cost disease is Heart Failure (HF). The clinical and societal implications of HF are truly staggering. One in 5 US citizens over age 40 is expected to develop HF in their lifetime and HF is the leading cause of hospitalization
among Medicare beneficiaries. With the aging population, HF will continue to be a leading cause of healthcare use. The hope is that through mining the longitudinal EHR data, predictive features can be identified from a large number of input features that will aid us predict HF with high accuracy. Furthermore, the selected features should be parsimonious (i.e., non-redundant). Often there is a known set of features (risk factors) that leads to HF. Any additional features should not only have high predictive value to HF but also complement to the known risk factors in order to minimize redundancy.

Motivated by this clinical application, we propose Scalable Orthogonal Regression (SOR) to address the aforementioned requirements. In particular, SOR has the following properties:

- **Scalable:** SOR achieves nearly linear scale-up with respect to the number of input features and the number of samples;
- **Optimal:** SOR is formulated as a sparse learning problem that can be solved efficiently using alternative convex optimization with theoretical convergence and global optimality guarantee;
- **Non-redundant:** SOR is designed specifically to select less redundant features without sacrificing the quality, where redundancy is measured by an orthogonality measure added as a penalty term in the objective function;
- **Extensible:** SOR can enhance an existing set of preselected features by adding additional features that complement the existing set but still with strong predictive power.

In order to evaluate our algorithm, we compare other state-of-the-art feature selection algorithms in 9 real data sets from various domains, including gene expression, general UCI benchmark data, and multimedia data. Extensive experimental results confirmed that SOR significantly outperforms several state of the art feature selection methods with respect to various quality metrics. In particular, SOR achieves orders of magnitude improvement of speed compared to several other methods. Besides overall competitive AUC measure, SOR can also achieve less redundancy and better stability in terms of selected features.

As a case study, we apply SOR to a clinical application on predictive modeling of HF. The study is done on over 20 million real EHR records on 30K patients over 7 years from a large healthcare provider network. The data contain diagnosis, medication, lab results and HF diagnostic symptoms. The goal is to predict the onset of HF x months before the actual diagnosis. In our cross validation evaluation, we achieve increased AUC measure in comparison to knowledge driven baseline which is provided by clinical experts.

The rest of the paper is organized as the follows. A brief survey on various feature selection methods and applications are presented in Section 2. We then introduce our method and the related optimization algorithms in Section 3. Theoretical analysis for our method is given in Section 4. We demonstrate the quality and scalability of our algorithm in Section 5. Finally we highlight a case study on EHR data in the experimental section.

## 2 Related Work

In feature selection, our purpose is to select a subset of $K$ informative features where $K$ is the number of required features. There are two major sub-problems in feature selection. One is the measurement of how informative a given subset of features is, and the other one is how to obtain the subset of features. Given a measurement of the quality of features, the feature selection problem is essentially a combinatorial optimization problem, and is usually solved by an approximation or greedy search. In general, there are two types of feature selection methods in the literature: (1) filter methods [11] where the selection is independent of classifiers and (2) wrapper methods [10] where the selection is tightly coupled with a specific classifier.

The filter methods evaluate features one by one, then select the top $K$ features according to their scores. This type of scheme can be interpreted as a greedy approach by iteratively selecting one feature from the remaining unselected feature set. Within this category, one can implement it using two approaches. Univariate filtering, e.g. Information Gain, or multivariate filtering, e.g. Minimum Redundancy-Maximum Relevance (mRMR) [18].

Feature selection using wrapper methods provides an alternative way to obtain multivariate subset selection by incorporating the classifiers, e.g. directly approximating the area under the ROC curve [15] or optimization of the LASSO (Least Absolute Shrinkage and Selection Operator) model [4, 21].

The learning of non-redundant features has also been discussed in literature. For example, mRMR explicitly prefers low redundant features [18], and non-redundant codebook feature learning method was also proposed [25].

## 3 Sparse Orthogonal Regression

This section presents the Sparse Orthogonal Regression (SOR) algorithm in detail. First we will introduce some
notation and symbols that will be used throughout the paper.

3.1 Notations We use $X$ to denote the data matrix containing $n$ observations on the $p$ covariates: $X = [x_1, x_2, \cdots, x_p] \in \mathbb{R}^{n \times p}$. Without the loss of generality, we assume all covariate vectors are normalized, i.e., $\|x_i\|_2 = 1$ ($i = 1, \cdots, p$). As we only care about the supervised setting in this paper, we are further given the corresponding response vector $y \in \mathbb{R}^n$, then the feature selection problem is a linear regression under square loss, which takes the following form.

$$\min_{\alpha} J_o(\alpha), \quad J_o(\alpha) = \frac{1}{2}\|y - X\alpha\|^2 = \frac{1}{2}\|y - \sum_j \alpha_j x_j\|^2, $$

where $\alpha = [\alpha_1, \alpha_2, \cdots, \alpha_p]^T \in \mathbb{R}^p$ is the regression coefficient vector. The absolute value of $|\alpha|_i$ can be regarded as the importance of covariate $j$, $j = 1, 2, \cdots, p$. If $\alpha_i = 0$, then that means covariate $i$ is not selected.

3.2 Orthogonality of Features As nonredundancy is one of the major claims of the method we proposed in this paper, we first give the definition of the redundancy between two covariates.

Definition 1 (Redundancy). Given two covariates $x_i$ and $x_j$, as well as their corresponding regression coefficients $\alpha_i$ and $\alpha_j$ (which are fixed) as in Eq. (3.1), we define the redundancy between them as follows,

$$R_{ij} = (\alpha_i \alpha_j x_i^T x_j)^2. $$

Obviously, if $x_i$ and $x_j$ are orthogonal to each other, then $x_i^T x_j = 0$ and $R_{ij} = 0$, indicating that they are totally non-redundant. If $x_i$ and $x_j$ are identical, then $x_i^T x_j$ is maximized. In this case, $x_i$ and $x_j$ are redundant.

Based on definition 1, in order to obtain a set of non-redundant covariates, we can minimize the following objective

$$J_o(\alpha) = \frac{1}{2}\|y - X\alpha\|^2 + \frac{\beta}{4} \sum_{ij} (\alpha_i x_i^T x_j \alpha_j)^2, $$

where the term $\sum_{ij} R_{ij} = \sum_{ij} (\alpha_i \alpha_j x_i^T x_j \alpha_j)^2$ is the summation of the redundancies over all pairwise features, and $\beta$ is a tradeoff parameter which controls the importance of the redundancy.

In feature selection, we also want the number of selected features to be as small as possible, thus we further impose the sparsity penalty term of $\|\alpha\|_1$ on the objective function. Then our goal becomes to minimize the following objective.

$$J(\alpha) = \frac{1}{2}\|y - X\alpha\|^2 + \lambda \|\alpha\|_1 + \frac{\beta}{4} \sum_{ij} (\alpha_i x_i^T x_j \alpha_j)^2,$$

where $\|\alpha\|_1$ is the $\ell_1$ norm of $\alpha : \|\alpha\|_1 = \sum_j |\alpha_j|$. We will show later that $J(\alpha)$ is convex and develop an efficient algorithm to minimize $J(\alpha)$ with respect to $\alpha$.

Here $\lambda$ is a model parameter which controls the sparsity. One can easily show that if $\lambda_i \geq \max_i \|X^Ty\|_i$, $\alpha = 0$ gives the optimal solution of Eq. (3.4). Thus the parameter $\lambda$ has a natural range of $0 \sim \lambda_{\max} = \max_i \|X^Ty\|_i$. In the rest of the paper, without loss of generalization, we use a normalized $\lambda$ (ranging from $0 \sim 1$, where $\lambda = 1$ indicate we use $\lambda_{\max}$). Once the optimal solution of $\alpha^*$ is obtained, we use the absolute values of $|\alpha_i^*|$ as the importance of features.

Our method performs particularly well in cases where the problem includes identifying a set of relevant predictors from a really large collection of variables that are not necessarily independent. We will provide detailed evidence in the experimental section.

3.3 Preliminaries In this section we will present some preliminaries on how to minimize Eq. (3.4). For notational convenience, we will use

$$f(\alpha) = J_o(\alpha) = \frac{1}{2}\|y - X\alpha\|^2 + \frac{\beta}{4} \sum_{ij} (\alpha_i x_i^T x_j \alpha_j)^2,$$

through the rest of this paper. Before diving into the details, first we need to prove that $f(\alpha)$ is locally Lipschitz continuous, which is defined as follows.

Definition 2 (Lipschitz continuous) [17]. A function $f : \mathbb{R}^d \rightarrow \mathbb{R}^m$ is Lipschitz continuous if for $\forall a, b \in \mathbb{R}^d$, we can find a constant $L$ satisfying the following inequality

$$\|a - b\| \leq L \|f(a) - f(b)\|$$

The function $f$ is called locally Lipschitz continuous, if for each $c \in \mathbb{R}^m$, there exists an $L > 0$ such that $f$ is Lipschitz continuous on the open ball of center $c$ and radius $L$.

$$B_L(c) = \{x \in \mathbb{R}^m : \|x - c\| < L\}.$$ 

In our case, as $f(\alpha)$ is continuously smooth, the gradient is locally Lipschitz continuous [6]. Then we have the following inequality [17].

$$f(\alpha) \leq f(\bar{\alpha}) + (\alpha - \bar{\alpha})^T \nabla f(\bar{\alpha}) + \frac{L}{2} \|\alpha - \bar{\alpha}\|^2,$$
which immediately leads to

(3.9) \( \frac{f(\alpha) + \lambda \| \alpha \|_1}{\| f(\alpha) \|_2} \leq f(\hat{\alpha}) + (\alpha - \hat{\alpha})^T \nabla f(\hat{\alpha}) + \frac{L}{2} \| \alpha - \hat{\alpha} \|_2^2 + \lambda \| \alpha \|_1 \).

In this section, we will employ Eq. (3.10) and derive an efficient iterative algorithm which is guaranteed to converge to the global solution of minimizing Eq. (3.4). Denote the right hand side of Eq. (3.10) by \( Z(\alpha, \hat{\alpha}) \), i.e.,

(3.10) \( Z(\alpha, \hat{\alpha}) = f(\hat{\alpha}) + (\alpha - \hat{\alpha})^T \nabla f(\hat{\alpha}) + \frac{L}{2} \| \alpha - \hat{\alpha} \|_2^2 + \lambda \| \alpha \|_1 \),

where \( \nabla f \) is the gradient of \( f \).Bringing \( J(\alpha) \) in Eq.(3.4) into Eq.(3.10), we can easily find that

(3.11) \( J(\alpha) = Z(\alpha, \alpha) \leq Z(\alpha, \hat{\alpha}). \)

Then let \( \hat{\alpha} = \alpha^t \) and

(3.12) \( \alpha^{t+1} = \arg \min_{\alpha} Z(\alpha, \alpha^t) \),

thus we have

(3.13) \( J(\alpha^{t+1}) = Z(\alpha^{t+1}, \alpha^{t+1}) \leq Z(\alpha^{t+1}, \alpha^t) \leq Z(\alpha^t, \alpha^t) = J(\alpha^t) \).

This suggests that we can iteratively update \( \alpha \) by solving problem (3.12) (i.e., minimizing \( Z(\alpha, \hat{\alpha}) \) with \( \hat{\alpha} = \alpha^t \)) to decrease the objective function monotonically.

3.4 Algorithm Details Based on the contents in last subsection, in order to minimize Eq.(3.4), we need to solve the following sub-problem iteratively

(3.14) \( \min_{\alpha} Z(\alpha, \alpha^t). \)

As \( f(\alpha^t) \) is constant with respect to \( \alpha \), we can minimize the following objective instead with respect to \( \alpha \)

(3.15) \( J_m(\alpha) = (\alpha - \alpha^t)^T \nabla f(\alpha^t) + \frac{L}{2} \| \alpha - \alpha^t \|_2^2 + \lambda \| \alpha \|_1 \),

where the gradient of \( f(\alpha) \) is

(3.16) \( \nabla f(\alpha) = (G + \beta A \odot G \odot G) \alpha, \)

which can be written in its matrix form as

(3.17) \( \nabla f(\alpha) = (G + \beta A \odot G \odot G) \alpha, \)

where \( A = \alpha \alpha^T, G = X^T X, \) and \( \odot \) is the matrix Hadamard (elementwise) product.

Next we will show that the minimization of Eq. (3.15) has closed form solution. First, as \( \| \nabla f(\alpha^t) \| \)

is a constant with respect to \( \alpha \), then minimize \( J_m(\alpha) \) in Eq. (3.15) is equivalent to minimize

\( J_m(\alpha) + \frac{1}{2L^2} \| \nabla f(\alpha^t) \|^2 + \frac{L}{2} \| \alpha - \alpha^t \|_2^2 + \frac{1}{2L} \| \nabla f(\alpha^t) \|^2 + \lambda \| \alpha \|_1 \)

\( \frac{L}{2} \| \alpha^t - \left( \alpha^t - \frac{1}{L} \nabla f(\alpha^t) \right) \|_2^2 + \lambda \| \alpha \|_1 \).

Furthermore, we can easily prove the following Lemma.

**Lemma 1.** The global minimum solution of minimizing the following objective over \( u \)

(3.18) \( J(u) = \frac{1}{2} \| u - a \|_2^2 + \mu \| u \|_1, \)

where \( u = [u_1, u_2, \cdots, u_p]^T \) and \( a = [a_1, a_2, \cdots, a_p]^T \) are \( p \times 1 \) vectors, is given by

(3.19) \( u_i = \begin{cases} 0 & \text{if } \mu \geq |a_i|, \\ \frac{a_i - \mu}{|a_i|} & \text{if } \mu < |a_i|, \end{cases} \quad i = 1, 2, \cdots, p, \)

or equivalently,

(3.20) \( u_i = (|a_i| - \mu) \text{sign}(a_i), \)

where \( \text{sign}(x) = x \) if \( x \) is positive, \( \text{sign}(x) = 0 \) if \( x \) is zero and \( \text{sign}(x) \) is the sign function (\( \text{sign}(0) \) is defined as 0 here).

By applying the above lemma, and letting \( \mu = \lambda / L, u = \alpha, a = \alpha^t - \frac{1}{L} \nabla f(\alpha^t) \), one can easily obtain the following close form optimal solution for minimizing Eq. (3.15),

(3.20) \( \alpha_i = \left( \left[ \alpha_i - \frac{1}{L} \nabla f(\alpha^t) \right] \cdot \frac{\lambda}{L} \right) \text{sign} \left( \left[ \alpha_i - \frac{1}{L} \nabla f(\alpha^t) \right] \right), \)

where \( i = 1, 2, \cdots, p. \)

Algorithm 1 summarizes the whole procedure of our Scalable Orthogonal Regression (SOR) algorithm. In the algorithm \( \gamma \) is a optimization parameter to increase \( L \) when the Lipschitz condition is not satisfied and is set to be 1.2 in all experiments. Next section presents some analysis of the algorithm and its extensions.

**Algorithm 3.1.** **Require:** \( \lambda, L_0, \alpha_0, \gamma \)

1: \textbf{while} Not converged \textbf{do}
2: \quad \textbf{ Compute } \nabla f(\alpha) \text{ using Eq. (3.17)}
3: \quad \text{ a } \leftarrow \alpha - \frac{\nabla f(\alpha)}{L}
4: \quad \text{ Solve } \hat{\alpha} \leftarrow \arg \min_{\alpha} \| \alpha - a \|^2 + \lambda \| \alpha \|_1
5: \quad \text{ (Eq. (3.20))}
6: \quad \text{ if } J(\hat{\alpha}) < J(\alpha) \text{ then}
7: \quad \quad \alpha \leftarrow \hat{\alpha}
8: \quad \text{ else}
9: \quad \quad L \leftarrow \gamma L
10: \quad \text{ end if}
11: \textbf{end while}
12: \textbf{return } \alpha
4 Analysis and Extension

In this section, we will provide some analysis and extensions of the SOR algorithm. First we show that the objective Eq. (3.4) is convex with respect to $\alpha$.

4.1 Convexity

We have the following theorem.

Theorem 1 (Convexity). Eq. (3.4) is convex w.r.t. $\alpha$.

Proof: See Appendix A.

Based on the convexity, we can prove the following theorem, which serves as the foundation of the follow up analysis on convergence rate.

Theorem 2 (Lipschitz Continuity). $f$ in Eq. (3.5) is locally Lipschitz continuous. Furthermore, there exists a local $L$ such that Eq. (3.5) is Lipschitz continuous at $\alpha_t$, where $\alpha_t$ is the solution of Algorithm 3.1 at the $t$-th iteration.

Proof: $f(\alpha)$ is continuously smooth, thus it is locally Lipschitz [6]. On the other hand, $f(\alpha)$ is convex and lower bounded, then the set $S = \{\alpha : f(\alpha) \leq f(\alpha^0)\}$ is a closed convex set. Obviously, $\alpha_t \in S$. As $f(\alpha)$ is locally Lipschitz with constant $L_\alpha$, $L = \max_{\alpha \in S} L_\alpha$ is obviously the global Lipschitz constant for the solutions of Algorithm 3.1.

4.2 Convergence

As discussed in section 3.3, SOR can monotonically decrease the value of $J(\alpha)$, and it is obvious that $J(\alpha)$ is lower bounded by zero, thus SOR will converge. Based on Theorem 1 and 2, we can prove the following theorem analyzing the convergence rate of Algorithm 1.

Theorem 3 (Convergence Rate of SOR). Algorithm 1 converges to the global solution of Problem in Eq. (3.4). Furthermore,

$$J(\alpha_T) - J(\alpha^*) \leq \frac{L_T\|x_0 - \alpha^*\|^2}{2T^2},$$

$T$ is number of iterations in Algorithm 3.1, $L_T$ is the value of $L$ in the last iteration, $\alpha^*$ is the global optimal of Eq. (3.4), and $\alpha_T$ is the output of Algorithm 3.1.

Proof: See the Appendix B.

Theorem 3 also guarantees that Algorithm 3.1 converges to the global solution, since $J(\alpha_T) - J(\alpha^*) \rightarrow 0$ as $T \rightarrow \infty$ (notice that $L_T \leq L$ because of the locally Lipschitz continuity of $f(\alpha)$ guaranteed by Theorem 2.

4.3 Accelerated Algorithm

As it is obvious that the $J_m(\alpha)$ in Eq. (3.15) is convex, we can also derive an accelerated algorithm shown in Algorithm 4.1, with much higher convergence rate. For the accelerated SOR (aSOR), we have the following theorem.

Theorem 4 (Convergence Rate of aSOR). Algorithm 1 converges to global solution of Problem in Eq. (3.4). Furthermore,

$$J(\alpha_T) - J(\alpha^*) \leq \frac{L_T\|x_0 - \alpha^*\|^2}{2T^2},$$

$T$ is number of iterations in Algorithm 4.1, $L_T$ is the value of $L$ in the last iteration, $\alpha^*$ is the global optimal of Eq. (3.4), and $\alpha_T$ is the output of Algorithm 3.1.

The theorem can be proved using similar tricks as in [13], and we omit the details here due to limited space.

By comparing the convergence rate of SOR and aSOR, one should notice that the gap to the optimal solution in aSOR decreases as $\frac{1}{T}$, which is much faster than in SOR with $\frac{1}{T^2}$, where $T$ is the number of iterations. We will demonstrate the convergence speed comparison of these two algorithms in the experimental section.

Algorithm 4.1. Require: $\lambda, p_0, x_0, \gamma$

1: $p \leftarrow p_0, x_0, \eta \leftarrow x_0, \alpha \leftarrow \alpha_0, \zeta \leftarrow 1, \alpha_0 \leftarrow 0$
2: while Not converged do
3: $\eta \leftarrow \eta - \nabla J(\eta)/p$
4: $\alpha \leftarrow \arg\min_{\alpha} \|a - \alpha\|^2 + \lambda\|\alpha\|_1$
5: if $J(\alpha) < J(\eta)$ then
6: $\eta \leftarrow \eta + 2(\zeta - 1)(\alpha - \hat{\alpha})/(1 + \sqrt{1 + 4\zeta^2})$
7: $\alpha \leftarrow \alpha + \zeta/(1 + \sqrt{1 + 4\zeta^2})$
8: $\zeta \leftarrow (1 + \sqrt{1 + 4\zeta^2})/2$
9: else
10: $p \leftarrow \eta p$
11: end if
12: end while
13: return $\alpha$

4.4 Computational Complexity

We will analyze the computational complexity of SOR in this section. Specifically, solving $\alpha$ at Step 5 in Algorithm 3.1 needs $O(p)$ time, where $p$ is the dimension of $\alpha$. The computational bottleneck of the Algorithm 1 is the evaluation of the gradient of $f(\alpha)$ in Eq. (3.17), which needs $O(np^2)$ time at the first glance. However, we can develop a more efficient way to obtain the gradient in $O(np^2)$ time. Specifically, we can first compute $B = X \odot (ae^T)$, where $e = [1, 1, \ldots, 1]^T$ with proper size. Then $B_{ij} = a_j x_i^T$, where $x_i^T$ is the $i$-th element of $x_i$ or $b_j = a_j x_j$, where $b_j$ is the $j$-th column of $B$. Obviously, the computation of $B$ only needs $O(np)$ time. Then

$$\sum_{i} (a_i a_j x_i^T x_j) x_j x_i a_j = a_i (x_i^T b_j)^2,$$

the summation of $v = \sum_j b_j$ takes $O(np)$ time, which does not depend on the index $i$. Notice that computing
x^T v only requires O(n) time. One the other hand X^T X_y = X^T (X_y) also requires O(np), thus the whole complexity of computing the gradient is O(np).

We also compare the computational and storage complexity of SOR with some other state-of-the-art approaches (Information Gain, LARS, and mRMR), which are summarized in Table 1.

### 4.5 SOR with Preselected Features

In some real world scenarios, we may already have a set of features preselected with prior knowledge. For example, physicians in hospitals have years of experience on some specific diseases, they have their own knowledge on which features (factors) are more important. In this case, we may want to select a set of features (with data driven approaches) complementary to those preselected features.

Fortunately our SOR algorithm can easily adapted to incorporate this prior knowledge. Assume the preselected feature set is \(P\) and the remaining feature set is \(Q\), then we can partition the whole data matrix as \(X = [X_P, X_Q]\), where \(X_P, X_Q\) only contains the observations on the features in \(P\) and \(Q\) and our goal is to select features from \(Q\). For the feature set \(P\), we first compute their regression coefficients with simple least squares:

\[
\alpha_P = \arg \min_{\alpha} \| y - X_P \alpha \|^2 = (X_P^T X_P)^{-1} X_P^T y.
\]

Then we define

\[
f_p(\alpha) = \frac{1}{2} \| y - X_Q \alpha \|^2
+ \frac{\beta}{4} \left[ \sum_{i,j \in Q} (\alpha_i x_i^T x_j \alpha_j)^2 + \sum_{i \in Q, j \in P} (\alpha_i x_i^T x_j \alpha_j)^2 \right]
\]

where \(\alpha = [\alpha_P, \alpha_Q]^T\) is the concatenated regression coefficient vector with \(\alpha_P\) computed using Eq.(4.21).

Note that there are two terms to punish the feature redundancy. One measures the feature redundancy selected from \(Q\), the other measures the redundancy between the feature selected from \(Q\) and the preselected feature set \(P\). Then we can minimize the following objective with respect to \(\alpha_Q\).

\[
J_p(\alpha) = f_p(\alpha) + \lambda \| \alpha \|_1.
\]

Comparing Eq. (3.4) and Eq. (4.22), one can immediately see that Algorithm still applies for the minimization of Eq. (4.22). The only step we need to change is the computation of gradient. Notice that in this optimization, \(\alpha_j\) is a constant for \(j \in P\). The corresponding gradient is

\[
\nabla f_p(\alpha) = (G + \beta A \odot G_Q \odot G_Q) \alpha + \beta (X_Q^T X_P \alpha_P) \odot \alpha.
\]

### 5 Experimental Results

In this section, will first demonstrate the convergence of SOR and aSOR and the scalability of the algorithm, then evaluate the quality (measured by AUC and stability) and orthogonality of the features selected by our algorithm.

**Datasets:** We evaluate our algorithm on various kinds of data. The first kind is the general datasets from UCI data mining and machine learning repository [5], which include heart and vehicle data sets. The second kind of data are image data, including Columbia object image library (coil) [16] and the Japanese Female Facial Expression (jaffe) Database\(^1\). The third type is gene expression data including MLL [1], and SRBCT [9]. We summarize the data description in Table 2.

**Convergence:** We now present the experiment on the convergence speed in Figure 2. For our algorithms 3.1 and 4.1, we set \(\lambda = 0.1\) and \(\beta = 0.1\). Figure 2 shows the objective function vs. number of iterations. It confirms that aSOR converges much faster than SOR.

---

\(^1\)Available at http://www.kasrl.org/jaffe.html
Scalability: We focus more in feature selection methods designed for specific classifiers (such as SVM-RFE \cite{8,20}). To test scalability, we generate different datasets by subsampling from a large EHR dataset and we observe aSOR is orders of magnitudes more efficient than LARS and mMRMR. Among them, only aSOR and InfoGain can apply to large datasets with over 10K features and samples. In particular, despite its sophisticated optimization mechanism, aSOR achieves similar computational performance to InfoGain, which is a very simple and greedy method.

Classification Accuracy: In all the comparison evaluation, we conduct a standard 80-to-20 split of the data at random at T times (in our case, T=20).

Classification accuracy is captured in terms of Area Under Curve (AUC) measure. To compute AUC, we use a SVM classifier with Gaussian kernel:

\[
K_{ij} = e^{-\frac{||x_i-x_j||^2}{2(\alpha^2)}},
\]

where \(x_i\) and \(x_j\) are data samples and \(\bar{r}\) is the average of pairwise distances among all the data samples and \(a\) is chosen from \([2^{-3},2^{-2},2^{-1},1,2^1,2^2,2^3]\). The SVM trade off parameter \(C\) is chosen from \([0.01,0.1,1,10,100]\). For all data and feature selection methods, we report the best results among all the combinations of \(a\) and \(C\). We directly use the LIBSVM \cite{2} software in our experiments. For SOR, we further choose \(\lambda\) from \([0.001,0.01,0.1,0.5]\) and \(\beta\) from \([0.001,0.01,0.1,0.5]\).

We compare the average of AUC in Figure 3 while varying the number of features selected. We observe the AUC of SOR is clearly above most of the other methods. More specifically, among all 119 comparisons, SOR outperforms the best of the other methods in 88, tie in 17. Our method is only worse than the best of the other methods in 4 cases.

To compare the variability of the AUC, we present the average and standard deviation of the AUC when 5 features are selected in Table 3. For all the 6 data sets, SOR outperforms the other methods in 4 cases.

In terms of parameter stability, SOR requires only two parameters \(\lambda\) and \(\beta\). We show that our method is stable to those parameters in Table 4, where the maximum, minimum, average, and the range of the AUC are reported. One can observe that though the parameters change dramatically in wide ranges, the AUC measure only changes about 1% - 5% for most of the data except for the heart, PROSTATE, and yaleB data sets. In our experiments, we looked into the value of \(\lambda\) which gives the best AUC, and we found that the typical value has a relative narrow range (around 0.1) after the normalization, indicating that \(\lambda\) is not a sensitive parameter.

Redundancy: Next we compare the redundancy of the features selected by different methods. Redundancy is measured by orthogonality between sets of selected features \(S\):

\[
\text{Redundancy} = \frac{1}{T(T-1)} \sum_{i,j \in S, i \neq j} \frac{x_i^T x_j}{\|x_i\| \|x_j\|}.
\]
Figure 1: CPU time comparison of Information Gain (InfoGain), LARS, aSOR, and mRMR. Left: fix the number of samples to 5000, and vary the number of features. Right: fix the number of features to 400, and vary the number of samples.

Figure 2: Convergent rate comparison between algorithm SOR and aSOR on three data sets, vehicle ($p = 18, n = 846$), coil, ($p = 1024, n = 1440$) and MLL, ($p = 12582, n = 72$), where $p$ is the number of dimensions and $n$ is the number of samples.

Figure 3: AUC comparison on 6 data sets (heart, vehicle, jaffe, coil, MLL and SRBCT).
Table 3: AUC and feature stability comparison with SOR, LARS, mRMR, and Information Gain. The best results on each data are highlighted in bold.

<table>
<thead>
<tr>
<th></th>
<th>Our AUC</th>
<th>Stable</th>
<th>LARS AUC</th>
<th>Stable</th>
<th>mRMR AUC</th>
<th>Stable</th>
<th>Information Gain AUC</th>
<th>Stable</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLL</td>
<td>0.9905±0.0245</td>
<td>0.5799</td>
<td>0.9773±0.0447</td>
<td>0.4504</td>
<td>0.9658±0.0541</td>
<td>0.2460</td>
<td>0.9661±0.0472</td>
<td>0.5890</td>
</tr>
<tr>
<td>PROSTATE</td>
<td>0.9676±0.0410</td>
<td>0.8428</td>
<td>0.9563±0.0441</td>
<td>0.7940</td>
<td>0.9447±0.0565</td>
<td>0.4220</td>
<td>0.9593±0.0460</td>
<td>0.7556</td>
</tr>
<tr>
<td>SRBCT</td>
<td>0.9907±0.0257</td>
<td>0.7746</td>
<td>0.9783±0.0391</td>
<td>0.6995</td>
<td>0.9602±0.0592</td>
<td>0.3522</td>
<td>0.9465±0.0662</td>
<td>0.4865</td>
</tr>
<tr>
<td>coil</td>
<td>0.9311±0.0514</td>
<td>0.6714</td>
<td>0.9112±0.0530</td>
<td>0.5097</td>
<td>0.9155±0.0417</td>
<td>0.6452</td>
<td>0.8900±0.0464</td>
<td>0.4865</td>
</tr>
<tr>
<td>heart</td>
<td>0.8464±0.0585</td>
<td>0.9357</td>
<td>0.7754±0.0859</td>
<td>0.9382</td>
<td>0.8277±0.0575</td>
<td>0.7371</td>
<td>0.7851±0.0844</td>
<td>0.8581</td>
</tr>
<tr>
<td>isolet</td>
<td>0.8297±0.0434</td>
<td>0.8537</td>
<td>0.7981±0.0539</td>
<td>0.7168</td>
<td>0.8033±0.0595</td>
<td>0.4360</td>
<td>0.7114±0.0778</td>
<td>0.8849</td>
</tr>
<tr>
<td>jaffe</td>
<td>0.9817±0.0243</td>
<td>0.5121</td>
<td>0.9546±0.0573</td>
<td>0.3467</td>
<td>0.9768±0.0278</td>
<td>0.3502</td>
<td>0.9458±0.0528</td>
<td>0.3198</td>
</tr>
<tr>
<td>vehicle</td>
<td>0.8918±0.0471</td>
<td>0.9910</td>
<td>0.8466±0.0556</td>
<td>0.9183</td>
<td>0.7760±0.0827</td>
<td>0.9647</td>
<td>0.7736±0.0453</td>
<td>0.8937</td>
</tr>
<tr>
<td>yaleB</td>
<td>0.7788±0.1031</td>
<td>0.2882</td>
<td>0.7093±0.1059</td>
<td>0.2509</td>
<td>0.7302±0.0825</td>
<td>0.1471</td>
<td>0.7061±0.0978</td>
<td>0.1545</td>
</tr>
</tbody>
</table>

Table 4: Stability to parameters of SOR. Reported are the AUCs of SOR while $\lambda$ and $\beta$ vary from $[0.001, 0.01, 0.1, 0.5]$.

<table>
<thead>
<tr>
<th>Data</th>
<th>min</th>
<th>max</th>
<th>average</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLL</td>
<td>0.9480</td>
<td>1.0000</td>
<td>0.9900</td>
<td>0.0520</td>
</tr>
<tr>
<td>PROSTATE</td>
<td>0.8571</td>
<td>0.9939</td>
<td>0.9514</td>
<td>0.1369</td>
</tr>
<tr>
<td>SRBCT</td>
<td>0.9789</td>
<td>1.0000</td>
<td>0.9969</td>
<td>0.0211</td>
</tr>
<tr>
<td>coil</td>
<td>0.9441</td>
<td>0.9836</td>
<td>0.9641</td>
<td>0.0395</td>
</tr>
<tr>
<td>heart</td>
<td>0.7572</td>
<td>0.9532</td>
<td>0.8514</td>
<td>0.1959</td>
</tr>
<tr>
<td>isolet</td>
<td>0.8542</td>
<td>0.9073</td>
<td>0.8734</td>
<td>0.0530</td>
</tr>
<tr>
<td>jaffe</td>
<td>0.9849</td>
<td>0.9993</td>
<td>0.9939</td>
<td>0.0144</td>
</tr>
<tr>
<td>vehicle</td>
<td>0.8912</td>
<td>0.9353</td>
<td>0.9149</td>
<td>0.0441</td>
</tr>
<tr>
<td>yaleB</td>
<td>0.6656</td>
<td>0.9088</td>
<td>0.8003</td>
<td>0.2432</td>
</tr>
</tbody>
</table>

Table 5: Number of features.

| Source Diagnosis Medication Lab Vital Framingham |
|----------|----------|---------|--------|---------|
| Count    | 8980     | 976     | 449    | 4       | 34      |

It measures the average cosine similarity between all pairs of features. As shown in Figure 4, SOR clearly has the lowest redundancy in selected features across all settings. In particular, the only scalable method InfoGain performs badly with respect to this measure.

5.1 Case Study in Predictive Modeling on the onset of Heart Failure

As described in the motivating example in Section 1, HF is a complex and costly disease that needs better understanding and management. Leveraging the abundance of EHR data, we aim at developing accurate predictive models that can help predict the onset of HF as early as possible. Feature selection is a crucial step in such a process.

**EHR data:** We have access to real EHR data from a leading healthcare provider, which has over 7 years longitudinal records from most of patients. In this dataset, there are a diverse set of clinical information: 1) diagnosis info consists of ICD9 code; 2) medication info consists generic drug names, pharmacy class and subclass info; 3) lab measures consist of test results of different clinical chemistry measures; 4) vital info includes weight, height, systolic and diastolic blood pressure, and 5) Framingham criteria include positive and negative confirmation of diagnostic signs and symptoms for HF patients. For this study, we include 5K cases (i.e., patients who have diagnosed with HF and also have over 2 years pre-diagnosis EHR info) and 5K controls (i.e., patients who have similar demographic info as cases but have not yet diagnosed with HF). This dataset is constructed carefully by our clinical partners over several years of effort including manually viewing individual EHR records of this patient cohort. We expect the results generated from this study will be of important clinical value.

**Feature construction:** For each patient, we anchor at an index date, and construct a feature vector from the observation window, which is defined as the fixed size time window right before index date. In our study, we fix the observation window to be 1 year. The duration between index date and diagnostic date is called the prediction window. As the prediction window $W$ increases, the model will try to predict the disease onset earlier. Ideally, we want the predictive model to be accurate with large $W$. The feature vector consists of
statistic measures derived from the longitudinal clinical events during the observation window. Each distinct type of clinical event becomes a feature. The number of features from different sources are given in Table 5.1. Feature values are derived from the corresponding EHR records from the observation window for this patient. For discrete events like diagnosis and medication, we use the number of occurrences at the feature value. For continue events such as blood pressure and lab measures, we compute the average of those measures in observation windows after removing invalid and noisy outliers.

**Evaluation:** In this study, we demonstrate the effectiveness of SOR in building models for predicting onset of HF. The evaluation metric is AUC measure which is well accepted in the clinical community. Once the features are constructed using the data in the observation window, we partition the patient feature vectors into 80% training and 20% testing randomly for 10 times. We use SVM classifier with Gaussian kernel with the same parameters as in Section 5.

Figure 5 compares the AUC measures of classifiers using different data sources. The combined features perform best as expected. Diagnosis, Medication and Framingham are also generating competitive results. Another clear pattern is that AUC decreases slowly as the prediction window increases from 0 to 12 months. In particular, even at 12 months earlier, the predictive model can achieve 0.7 AUC, which is clinically significant.

Figure 5: AUC with different prediction window sizes on ICD9, Framingham, Medication, Laboratory test, Vital data and the combination of the 5 feature sets.

Based on the input from clinical experts, we also construct sets of user selected features, one for each data source. In Figure 6, we start from the diagnosis features that are selected by a physician (a knowledge based approach) and gradually add in more features selected by SOR. The AUC consistently increases as more features are included. Note that there are 108
features in the original preselected set. By adding only 5 features, the AUC measure increases close to 10%, which confirms the power of SOR in terms of detecting predictive features. Currently, we are in the process of further validating the selected features with our clinical partners, which upon completion will likely become useful additional risk factors of HF.

6 Conclusions
In this paper, we propose Scalable Orthogonal Regression (SOR) to select low redundancy features. We propose an efficient iterative algorithm to resolve the problem and analyze its convergence rate. Furthermore, we also propose an extension of SOR to incorporate preselected features according to prior expertise knowledge. The effectiveness and efficiency of SOR is demonstrated on several benchmark data sets. Finally we also validate the usefulness of SOR on a real world clinical data set.

References
[22] E. P. Xing, M. I. Jordan, and R. M. Karp. Feature se-


Appendix A: Proof of Theorem 1
We first introduce the following Lemma.

Lemma 2. If A and B are semi-positive definite, then $A \odot B$ is also semi-positive definite.

Proof. We first notice that if $u$ and $v$ are vectors, then

$$[uu^T \odot vv^T]_{ij} = u_i u_j v_i v_j = (u_i v_i)(u_j v_j),$$

Let $w_i = u_i v_i$, or $w = u \odot v$ then

$$[(uu^T) \odot (vv^T)] = ww^T.$$  

Since $A$ is semi-positive definite, there exist $U$ such that $A = UU^T$. For the same reason, let $B = VV^T$. Let $U = [u_1, u_2, \ldots, u_r], V = [v_1, v_2, \ldots, v_s]$ where $r$ and $s$ are the ranks of $A$ and $B$, respectively, and $w_{ij} = u_i \odot v_j$, then

$$A \odot B = \sum_{ij} u_i u_j \odot v_i v_j = \sum_{ij} w_i w_j = WW^T,$$

where $W = [w_{11}, \ldots, w_{1s}, w_{21}, \ldots, w_{rs}]$. Thus $A \odot B$ is semi-positive definite.

Then we can prove the convexity of $f(\alpha)$ by showing the Hessian of $f(\alpha)$ is positive semi-definite. From the gradient of Eq. (3.17), we can compute the Hessian of $f(\alpha)$ as

$$(6.26) \quad H_{pq} = \frac{\partial (\nabla f(\alpha))_p}{\partial \alpha_q} = G_{pq} + \beta \sum_{ij} \alpha_i \alpha_j (x_i^T x_j)^2 a_{ij},$$

Let $\tilde{H}_{pq} = \frac{\partial \sum_{ij} \alpha_i \alpha_j (x_i^T x_j)^2 a_{ij}}{\partial \alpha_q}$, then

$$\tilde{H}_{pq} = \begin{cases} 2\beta a_p a_q (x_p^T x_q)^2 + \sum_{j \neq p} \beta a_j (x_p^T x_j)^2 & \text{if } p = q \\ 2\beta a_p a_q (x_p^T x_q)^2 & \text{if } p \neq q \end{cases}$$

Thus the Hessian matrix is

$$H = G + 2\beta A \odot G + 2\beta \text{diag}(a_1, a_2, \ldots, a_n).$$

where $A = \alpha^T \alpha$ and $a_i = \sum_j a_j (x_i^T x_j)^2, i = 1, 2, \ldots, n$. Since $a_i \geq 0$, $\text{diag}(a_1, a_2, \ldots, a_n)$ is positive semi-definite. And according to Lemma 2, $A \odot G \odot A$ is also positive semi-definite. Thus $H$ is positive semi-definite. And according to Lemma 2, $A \odot G \odot A$ is also positive semi-definite. Thus $H$ is positive semi-definite, and $f(\alpha)$ is convex. Obviously, $\|\alpha\|_1$ is convex, hence $J(\alpha)$ is convex.

Appendix B: Proof of Theorem 3
The following proof is similar to paper [14] for two consecutive solutions $\alpha^T, \alpha^{T+1}$, since $f(\alpha)$ is convex and $\|\alpha\|_1$ is convex,

$$(6.27) \quad f(\alpha^T) \geq f(\alpha^T) + (\alpha^T - \alpha^{T+1})^T \nabla f(\alpha^T),$$

$$\lambda \|\alpha^*\|_1 \geq \lambda \|\alpha^{T+1}\|_1 + \lambda \alpha^* - g^T \alpha^{T+1},$$

where $g \in \partial\|\alpha\|_1$ is any element in the sub-gradient of $\|\alpha\|_1$ at $\alpha^{T+1}$. Since $\alpha^{T+1}$ is the optimal solution of

$$Z(\alpha, \alpha^*) = \frac{L}{2} \|\alpha - \alpha\|^2 + \lambda \|\alpha\|_1 + C,$$

$$0 \in \partial Z(\alpha^{T+1}, \alpha^*),$$

$$0 \in L(\alpha^{T+1} - a) \odot \partial \phi(\alpha^{T+1}).$$

Obviously, $G \equiv L(a - \alpha^{T+1})/\lambda$ must be in $\partial \phi(\alpha^{T+1})$. Thus we have

$$(6.28) \quad \lambda \|\alpha\|_1 \geq \lambda \|\alpha^{T+1}\|_1 + L(\alpha^{T+1} - a).$$

By combining (6.27) and (6.28), we have

$$f(\alpha^T) + \lambda \|\alpha\|_1 \geq f(\alpha^{T+1}) + \lambda \|\alpha^{T+1}\|_1$$

$$+ \frac{L}{2} \|\alpha^{T+1} - \alpha\|^2 + L(\alpha^- \alpha^{T+1})$$

$$+ \frac{L}{2} \left(\|\alpha^* - \alpha\|^2 - \|\alpha^{T+1} - \alpha^*\|^2\right).$$

According to Eq. (3.13), we have $J(\alpha^T) \leq J(\alpha^{T-1}) \leq \cdots \leq J(\alpha^0)$. Thus

$$\sum_{t=T_0}^{T-1} \frac{L}{2} \left(\|\alpha^* - \alpha^T\|^2 - \|\alpha^{T+1} - \alpha^T\|^2\right) \leq \|\alpha^0 - \alpha^*\|^2 - \|\alpha^0 - \alpha^T\|^2 \leq \frac{L}{2} \|\alpha^0 - \alpha^*\|^2,$$

or

$$(6.30) \quad J(\alpha^{T*}) - J(\alpha^*) \leq \frac{L}{2T} \|\alpha^0 - \alpha^*\|^2.$$