LETTER Communicated by Michael Mahoney

Feature Selection for Ridge Regression with Provable Guarantees

Saurabh Paul
saupaul@paypal.com
Global Risk Sciences, Paypal, San Jose, CA 95112, U.S.A.

Petros Drineas
drinep@cs.rpi.edu
Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY 12180, U.S.A.

We introduce single-set spectral sparsification as a deterministic sampling–based feature selection technique for regularized least-squares classification, which is the classification analog to ridge regression. The method is unsupervised and gives worst-case guarantees of the generalization power of the classification function after feature selection with respect to the classification function obtained using all features. We also introduce leverage-score sampling as an unsupervised randomized feature selection method for ridge regression. We provide risk bounds for both single-set spectral sparsification and leverage-score sampling on ridge regression in the fixed design setting and show that the risk in the sampled space is comparable to the risk in the full-feature space. We perform experiments on synthetic and real-world data sets; a subset of TechTC-300 data sets, to support our theory. Experimental results indicate that the proposed methods perform better than the existing feature selection methods.

1 Introduction

Ridge regression, a popular technique in machine learning and statistics, is a commonly used penalized regression method. Regularized least-squares classifier (RLSC) is a simple classifier based on least squares and has a long history in machine learning (Zhang & Peng, 2004; Poggio & Smale, 2003; Rifkin, Yeo, & Poggio, 2003; Fung & Mangasarian, 2001; Suykens & Vandewalle, 1999; Zhang & Oles, 2001; Agarwal, 2002). RLSC is also the classification analogue to ridge regression. It has been known to perform comparably to the popular support vector machines (SVM) (Rifkin et al., 2003; Fung & Mangasarian, 2001; Suykens & Vandewalle, 1999; Zhang & Oles, 2001). RLSC can be solved by simple vector space operations and does not require quadratic optimization techniques like SVM.

Neural Computation **28, 716–742** (2016)
doi:10.1162/NECO_a_00816

We propose a deterministic feature selection technique for RLSC with provable guarantees. There exist numerous feature selection techniques, which work well empirically. There also exist randomized feature selection methods like leverage-score sampling, (Dasgupta, Drineas, Harb, Josifovski, & Mahoney, 2007) with provable guarantees, which work well empirically. But the randomized methods have a failure probability and have to be rerun multiple times to get accurate results. Also, a randomized algorithm may not select the same regardless in different runs. A deterministic algorithm will select the same features regardless of how many times it is run. This becomes important in many applications. Unsupervised feature selection involves selecting features oblivious to the class or labels.

In this work, we present a new, provably accurate unsupervised feature selection technique for RLSC. We study a deterministic sampling-based feature selection strategy for RLSC with provable nontrivial worst-case performance bounds. We also use single-set spectral sparsification and leverage-score sampling as unsupervised feature selection algorithms for ridge regression in the fixed design setting. Since the methods are unsupervised, it will ensure that the methods work well in the fixed design setting, where the target variables have an additive homoskedastic noise. The algorithms sample a subset of the features from the original data matrix and then perform regression tasks on the reduced dimension matrix. We provide risk bounds for the feature selection algorithms on ridge regression in the fixed design setting.

The number of features selected by both algorithms is proportional to the rank of the training set. The deterministic sampling-based feature selection algorithm performs better in practice when compared to existing methods of feature selection.

2 Our Contributions

We introduce single-set spectral sparsification as a provably accurate deterministic feature selection technique for RLSC in an unsupervised setting. The number of features selected by the algorithm is independent of the number of features but depends on the number of data points. The algorithm selects a small number of features and solves the classification problem using those features. Dasgupta et al. (2007) used a leverage-score-based randomized feature selection technique for RLSC and provided worst-case guarantees of the approximate classifier function to that using all features. We use a deterministic algorithm to provide worst-case generalization error guarantees. The deterministic algorithm does not come with a failure probability, and the number of features required by the deterministic algorithm is less than that required by the randomized algorithm. The leverage-score-based algorithm has a sampling complexity of $O(\frac{n}{\epsilon^2}\log(\frac{n}{\epsilon^2\sqrt{\delta}}))$, whereas single-set spectral sparsification requires $O(n/\epsilon^2)$ to be picked, where n is the number of training points, $\delta \in (0,1)$ is a failure probability, and $\epsilon \in (0,1/2]$

is an accuracy parameter. As in Dasgupta et al. (2007), we also provide additive-error approximation guarantees for any test point and relative-error approximation guarantees for test points that satisfy some conditions with respect to the training set.

We introduce single-set spectral sparsification and leverage-score sampling as unsupervised feature selection algorithms for ridge regression and provide risk bounds for the subsampled problems in the fixed design setting. The risk in the sampled space is comparable to the risk in the full-feature space. We give relative-error guarantees of the risk for both feature selection methods in the fixed design setting.

From an empirical perspective, we evaluate single-set spectral sparsification on synthetic data and 48 document-term matrices, which are a subset of the TechTC-300 (Davidov, Gabrilovich, & Markovitch, 2004) data set. We compare the single-set spectral sparsification algorithm with leverage-score sampling, information gain, rank-revealing QR factorization (RRQR), and random feature selection. We do not report running times because feature selection is an offline task. The experimental results indicate that single-set spectral sparsification outperforms all the methods in terms of out-of-sample error for all 48 TechTC-300 data sets. We observe that a much smaller number of features is required by the deterministic algorithm to achieve good performance when compared to leverage-score sampling.

3 Background and Related Work

3.1 Notation. $\mathbf{A}, \mathbf{B}, \ldots$ denote matrices and $\boldsymbol{\alpha}, \mathbf{b}, \ldots$ denote column vectors; $\mathbf{e}_i$ (for all $i = 1 \ldots n$) is the standard basis, whose dimensionality will be clear from context; and $\mathbf{I}_n$ is the $n \times n$ identity matrix. The singular value decomposition (SVD) of a matrix $\mathbf{A} \in \mathbb{R}^{n\times d}$ is equal to $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{n\times d}$ is an orthogonal matrix containing the left singular vectors, $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$ is a diagonal matrix containing the singular values $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_d > 0$, and $\mathbf{V} \in \mathbb{R}^{d\times d}$ is a matrix containing the right singular vectors. The spectral norm of $\mathbf{A}$ is $\|\mathbf{A}\|_2 = \sigma_1$. $\sigma_{\max}$ and $\sigma_{\min}$ are the largest and smallest singular values of $\mathbf{A}$. $\kappa_{\mathbf{A}} = \sigma_{\max}/\sigma_{\min}$ is the condition number of $\mathbf{A}$. $\mathbf{U}^{\perp}$ denotes any $n \times (n-d)$ orthogonal matrix whose columns span the subspace orthogonal to $\mathbf{U}$. A vector $\mathbf{q} \in \mathbb{R}^n$ can be expressed as $\mathbf{q} = \mathbf{A}\boldsymbol{\alpha} + \mathbf{U}^{\perp}\boldsymbol{\beta}$, for some vectors $\boldsymbol{\alpha} \in \mathbb{R}^d$ and $\boldsymbol{\beta} \in \mathbb{R}^{n-d}$, that is, $\mathbf{q}$ has one component along $\mathbf{A}$ and another component orthogonal to $\mathbf{A}$.

3.2 Matrix Sampling Formalism. We now present the tools of feature selection. Let $\mathbf{A} \in \mathbb{R}^{d\times n}$ be the data matrix consisting of n points and d dimensions and $\mathbf{S} \in \mathbb{R}^{r\times d}$ be a matrix such that $\mathbf{SA} \in \mathbb{R}^{r\times n}$ contains r rows of $\mathbf{A}$. Matrix $\mathbf{S}$ is a binary (0/1) indicator matrix, which has exactly one nonzero element in each row. The non-zero element of $\mathbf{S}$ indicates which row of $\mathbf{A}$ will be selected. Let $\mathbf{D} \in \mathbb{R}^{r\times r}$ be the diagonal matrix such that $\mathbf{DSA} \in \mathbb{R}^{r\times n}$ rescales the rows of $\mathbf{A}$ that are in $\mathbf{SA}$. The matrices $\mathbf{S}$ and $\mathbf{D}$ are called the sampling and rescaling matrices, respectively. We will replace the sampling and rescaling matrices by a single matrix $\mathbf{R} \in \mathbb{R}^{r \times d}$, where $\mathbf{R} = \mathbf{DS}$ denotes the matrix specifying which of the r rows of $\mathbf{A}$ are to be sampled and how they are to be rescaled.

3.3 RLSC Basics. Consider a training data of n points in d dimensions with respective labels $y_i \in \{-1, +1\}$ for $i = 1, \ldots, n$. The solution of binary classification problems via Tikhonov regularization in a reproducing kernel Hilbert space (RKHS) using the squared loss function results in the regularized least squares classification (RLSC) problem (Rifkin et al., 2003), which can be stated as

$$\min_{\mathbf{x} \in \mathbb{R}^n} \left\| \mathbf{Kx} - \mathbf{y} \right\|_2^2 + \lambda \mathbf{x}^T \mathbf{Kx}, \tag{3.1}$$

where $\mathbf{K}$ is the $n \times n$ kernel matrix defined over the training data set, λ is a regularization parameter, and $\mathbf{y}$ is the n-dimensional $\{\pm 1\}$ class label vector. In matrix notation, the training data set $\mathbf{X}$ is a $d \times n$ matrix, consisting of n data points and d features $(d \gg n)$. Throughout this study, we assume that $\mathbf{X}$ is a full-rank matrix. We shall consider the linear kernel, which can be written as $\mathbf{K} = \mathbf{X}^T \mathbf{X}$. Using the SVD of $\mathbf{X}$, the optimal solution of equation 3.1 in the full-dimensional space is

$$\mathbf{x}_{opt} = \mathbf{V} \left(\boldsymbol{\Sigma}^2 + \lambda \mathbf{I} \right)^{-1} \mathbf{V}^T \mathbf{y}. \tag{3.2}$$

The vector $\mathbf{x}_{opt}$ can be used as a classification function that generalizes to test data. If $\mathbf{q} \in \mathbb{R}^d$ is the new test point, then the binary classification function is

$$f(\mathbf{q}) = \mathbf{x}_{opt}^T \mathbf{X}^T \mathbf{q}. \tag{3.3}$$

Then, $sign(f(\mathbf{q}))$ gives the predicted label (-1 or $+1$) to be assigned to the new test point $\mathbf{q}$.

Our goal is to study how RLSC performs when the deterministic sampling-based feature selection algorithm is used to select features in an unsupervised setting. Let $\mathbf{R} \in \mathbb{R}^{r \times d}$ be the matrix that samples and rescales r rows of $\mathbf{X}$, thus reducing the dimensionality of the training set from d to $r \ll d$ and r is proportional to the rank of the input matrix. The transformed data set into r dimensions is given by $\tilde{\mathbf{X}} = \mathbf{RX}$, and the RLSC problem becomes

$$\min_{\mathbf{x} \in \mathbb{R}^n} \left\| \tilde{\mathbf{K}}\mathbf{x} - \mathbf{y} \right\|_2^2 + \lambda \mathbf{x}^T \tilde{\mathbf{K}}\mathbf{x}, \tag{3.4}$$

giving an optimal vector $\tilde{\mathbf{x}}_{opt}$. The new test point $\mathbf{q}$ is first dimensionally reduced to $\tilde{\mathbf{q}} = \mathbf{Rq}$, where $\tilde{\mathbf{q}} \in \mathbb{R}^r$, and then classified by the function,

$$\tilde{f} = f(\tilde{\mathbf{q}}) = \tilde{\mathbf{x}}_{opt}^T \tilde{\mathbf{X}}^T \tilde{\mathbf{q}}. \tag{3.5}$$

In subsequent sections, we will assume that the test point $\mathbf{q}$ is of the form $\mathbf{q} = \mathbf{X}\boldsymbol{\alpha} + \mathbf{U}^{\perp}\boldsymbol{\beta}$. The first part of the expression shows the portion of the test point that is similar to the training set, and the second part shows how much the test point is novel compared to the training set, that is, $\|\boldsymbol{\beta}\|_2$ measures how much of $\mathbf{q}$ lies outside the subspace spanned by the training set.

3.4 Ridge Regression Basics. Consider a data set $\mathbf{X}$ of n points in d dimensions with $d \gg n$. Here $\mathbf{X}$ contains n independent and identically distributed (i.i.d.). samples from the d-dimensional independent variable. $\mathbf{y} \in \mathbb{R}^n$ is the real-valued response vector. Ridge regression(RR) or Tikhonov regularization penalizes the ℓ_2 norm of a parameter vector $\boldsymbol{\beta}$ and shrinks the estimated coefficients toward zero. In the fixed design setting, we have $\mathbf{y} = \mathbf{X}^T\boldsymbol{\beta} + \boldsymbol{\omega}$, where $\boldsymbol{\omega} \in \mathbb{R}^n$ is the homoskedastic noise vector with mean 0 and variance σ^2. Let $\boldsymbol{\beta}_\lambda$ be the solution to the ridge regression problem. The RR problem is stated as

$$\hat{\boldsymbol{\beta}}_\lambda = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^d} \frac{1}{n} \left\|\mathbf{y} - \mathbf{X}^T\boldsymbol{\beta}\right\|_2^2 + \lambda \left\|\boldsymbol{\beta}\right\|_2^2. \tag{3.6}$$

The solution to equation 3.6 is $\hat{\boldsymbol{\beta}}_\lambda = (\mathbf{X}\mathbf{X}^T + n\lambda\mathbf{I}_d)^{-1}\mathbf{X}\mathbf{y}$. One can also solve the same problem in the dual space. Using change of variables $\boldsymbol{\beta} = \mathbf{X}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \mathbb{R}^n$, and let $\mathbf{K} = \mathbf{X}^T\mathbf{X}$ be the $n \times n$ linear kernel defined over the training data set. The optimization problem becomes

$$\hat{\boldsymbol{\alpha}}_\lambda = \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \frac{1}{n} \left\|\mathbf{y} - \mathbf{K}\boldsymbol{\alpha}\right\|_2^2 + \lambda\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}. \tag{3.7}$$

Throughout this study, we assume that $\mathbf{X}$ is a full-rank matrix. Using the SVD of $\mathbf{X}$, the optimal solution in the dual space (see equation 3.7) for the full-dimensional data is given by $\hat{\boldsymbol{\alpha}}_\lambda = (\mathbf{K} + n\lambda\mathbf{I}_n)^{-1}\mathbf{y}$. The primal solution is $\hat{\boldsymbol{\beta}}_\lambda = \mathbf{X}\hat{\boldsymbol{\alpha}}_\lambda$.

In the sampled space, we have $\tilde{\mathbf{K}} = \tilde{\mathbf{X}}^T\tilde{\mathbf{X}}$. The dual problem in the sampled space can be posed as

$$\tilde{\boldsymbol{\alpha}}_\lambda = \arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \frac{1}{n} \left\|\mathbf{y} - \tilde{\mathbf{K}}\boldsymbol{\alpha}\right\|_2^2 + \lambda\boldsymbol{\alpha}^T\tilde{\mathbf{K}}\boldsymbol{\alpha}. \tag{3.8}$$

The optimal dual solution in the sampled space is $\tilde{\boldsymbol{\alpha}}_\lambda = (\tilde{\mathbf{K}} + n\lambda\mathbf{I}_n)^{-1}\mathbf{y}$. The primal solution is $\tilde{\boldsymbol{\beta}}_\lambda = \tilde{\mathbf{X}}\tilde{\boldsymbol{\alpha}}_\lambda$.

3.5 Related Work. The work most closely related to ours is that of Dasgupta, Drineas, Harb, Josifovski, and Mahoney (2007), who used a leverage-score-based randomized feature selection technique for RLSC and provided worst-case bounds of the approximate classifier with that of the classifier for all features. The proof of their main quality-of-approximation results provided an intuition of the circumstances when their feature selection method will work well. The running time of leverage-score-based sampling is dominated by the time to compute the SVD of the training set (i.e., $O(n^2d)$), whereas for single-set spectral sparsification, it is $O(rdn^2)$. Single-set spectral sparsification is a slower and more accurate method than leverage-score sampling. Another work on dimensionality reduction of RLSC is that of Avron, Sindhwani, and Woodruff (2013), who used efficient randomized algorithms for solving RLSC in settings where the design matrix has a Vandermonde structure. However, this technique is different from ours, since their work is focused on dimensionality reduction using linear combinations of features but not on actual feature selection.

Lu, Dhillon, Foster, and Ungar (2013) used randomized Walsh-Hadamard transform to lower the dimension of data matrix and subsequently solve the ridge regression problem in the lower-dimensional space. They provided risk bounds of their algorithm in the fixed design setting. However, this is different from our work, since they use linear combinations of features, while we select actual features from the data.

4 Our Main Tools

4.1 Single-Set Spectral Sparsification. We describe the single-set spectral sparsification algorithm (BSS[1], for short) of Batson, Spielman, and Srivastava (2009) as algorithm 1. Algorithm 1 is a greedy technique that selects columns one at a time. Consider the input matrix as a set of d column vectors $\mathbf{U}^T = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d]$, with $\mathbf{u}_i \in \mathbb{R}^\ell (i = 1, .., d)$. Given ℓ and $r > \ell$, we iterate over $\tau = 0, 1, 2, ..r - 1$. Define the parameters $L_\tau = \tau - \sqrt{r\ell}$, $\delta_L = 1$, $U_\tau = \delta_U(\tau + \sqrt{\ell r})$ and $\delta_U = (1 + \sqrt{\ell/r})/(1 - \sqrt{\ell/r})$. For $U, L \in \mathbb{R}$ and $\mathbf{A} \in \mathbb{R}^{\ell\times\ell}$, a symmetric positive-definite matrix with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_\ell$, defines

$$\Phi\left(L, \mathbf{A}\right) = \sum_{i=1}^{\ell} \frac{1}{\lambda_i - L}; \quad \hat{\Phi}\left(U, \mathbf{A}\right) = \sum_{i=1}^{\ell} \frac{1}{U - \lambda_i}$$

as the lower and upper potentials, respectively. These potential functions measure how far the eigenvalues of $\mathbf{A}$ are from the upper and lower barriers

[1] The name BSS comes from the authors: Batson, Spielman, and Srivastava.

U and L, respectively. We define $\mathcal{L}(\mathbf{u}, \delta_L, \mathbf{A}, L)$ and $\mathcal{U}(\mathbf{u}, \delta_U, \mathbf{A}, U)$ as follows:

$$\mathcal{L}\left(\mathbf{u}, \delta_L, \mathbf{A}, L\right) = \frac{\mathbf{u}^T\left(\mathbf{A} - \left(L + \delta_L\right)\mathbf{I}_\ell\right)^{-2}\mathbf{u}}{\Phi\left(L + \delta_L, \mathbf{A}\right) - \Phi\left(L, \mathbf{A}\right)} - \mathbf{u}^T\left(\mathbf{A} - \left(L + \delta_L\right)\mathbf{I}_\ell\right)^{-1}\mathbf{u},$$

$$\mathcal{U}\left(\mathbf{u}, \delta_U, \mathbf{A}, U\right) = \frac{\mathbf{u}^T\left(\left(U + \delta_U\right)\mathbf{I}_\ell - \mathbf{A}\right)^{-2}\mathbf{u}}{\hat{\Phi}\left(U, \mathbf{A}\right) - \hat{\Phi}\left(U + \delta_U, \mathbf{A}\right)} + \mathbf{u}^T\left(\left(U + \delta_U\right)\mathbf{I}_\ell - \mathbf{A}\right)^{-1}\mathbf{u}.$$

At every iteration, there exists an index i_τ and a weight $t_\tau > 0$ such that ${t_\tau}^{-1} \leq \mathcal{L}(\mathbf{u}_{i_\tau}, \delta_L, \mathbf{A}, L)$ and ${t_\tau}^{-1} \geq \mathcal{U}(\mathbf{u}_{i_\tau}, \delta_U, \mathbf{A}, U)$. Thus, there will be at most r columns selected after τ iterations. The running time of the algorithm is dominated by the search for an index i_τ satisfying

$$\mathcal{U}\left(\mathbf{u}_{i_\tau}, \delta_U, \mathbf{A}_\tau, U_\tau\right) \leq \mathcal{L}\left(\mathbf{u}_{i_\tau}, \delta_L, \mathbf{A}_\tau, L_\tau\right)$$

and computing the weight t_τ. One needs to compute the upper and lower potentials $\hat{\Phi}\left(U, \mathbf{A}\right)$ and $\Phi\left(L, \mathbf{A}\right)$ and hence the eigenvalues of $\mathbf{A}$. Cost per iteration is $O\left(\ell^3\right)$, and the total cost is $O\left(r\ell^3\right)$. For $i = 1, \ldots, d$, we need to compute $\mathcal{L}$ and $\mathcal{U}$ for every $\mathbf{u}_i$ which can be done in $O\left(d\ell^2\right)$ for every iteration, for a total of $O\left(rd\ell^2\right)$. Thus, the total running time of the algorithm is $O\left(rd\ell^2\right)$. We present the following lemma for the single-set spectral sparsification algorithm (algorithm 1).

Lemma 1. *BSS (Batson et al., 2009). Given* $\boldsymbol{U} \in \mathbb{R}^{d\times\ell}$ *satisfying* $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_\ell$ *and* $r > \ell$*, we can deterministically construct sampling and rescaling matrices* $\boldsymbol{S} \in \mathbb{R}^{r\times d}$ *and* $\boldsymbol{D} \in \mathbb{R}^{r\times r}$ *with* $\boldsymbol{R} = \boldsymbol{D}\boldsymbol{S}$*, such that, for all* $\boldsymbol{y} \in \mathbb{R}^\ell$*,*

$$\left(1 - \sqrt{\ell/r}\right)^2 \|\boldsymbol{U}\boldsymbol{y}\|_2^2 \leq \|\boldsymbol{R}\boldsymbol{U}\boldsymbol{y}\|_2^2 \leq \left(1 + \sqrt{\ell/r}\right)^2 \|\boldsymbol{U}\boldsymbol{y}\|_2^2.$$

We now present a slightly modified version of lemma 1 for our theorems:

Lemma 2. *Given* $\boldsymbol{U} \in \mathbb{R}^{d\times\ell}$ *satisfying* $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_\ell$ *and* $r > \ell$*, we can deterministically construct sampling and rescaling matrices* $\boldsymbol{S} \in \mathbb{R}^{r\times d}$ *and* $\boldsymbol{D} \in \mathbb{R}^{r\times r}$ *such that for* $\boldsymbol{R} = \boldsymbol{D}\boldsymbol{S}$*,*

$$\|\boldsymbol{U}^T\boldsymbol{U} - \boldsymbol{U}^T\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{U}\|_2 \leq 3\sqrt{\ell/r}.$$

Proof. From lemma 1, it follows that

$$\sigma_\ell\left(\mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U}\right) \geq \left(1 - \sqrt{\ell/r}\right)^2 \text{ and } \sigma_1\left(\mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U}\right) \leq \left(1 + \sqrt{\ell/r}\right)^2.$$

Algorithm 1: Single-Set Spectral Sparsification.

```
Input: $\mathbf{V}^T = [\mathbf{v}_1, \mathbf{v}_2, ...\mathbf{v}_d] \in \mathbb{R}^{\ell \times d}$ with $\mathbf{v}_i \in \mathbb{R}^{\ell}$ and $r > \ell$.

Output: Matrices $\mathbf{S} \in \mathbb{R}^{d \times r}, \mathbf{D} \in \mathbb{R}^{r \times r}$.

1. Initialize $\mathbf{A}_0 = \mathbf{0}_{\ell \times \ell}$, $\mathbf{S} = \mathbf{0}_{d \times r}, \mathbf{D} = \mathbf{0}_{r \times r}$.
2. Set constants $\delta_L = 1$ and $\delta_U = \left(1 + \sqrt{\ell/r}\right) / \left(1 - \sqrt{\ell/r}\right)$.
3. for $\tau = 0$ to $r - 1$ do
    • Let $L_\tau = \tau - \sqrt{r\ell}; U_\tau = \delta_U \left(\tau + \sqrt{\ell r}\right)$.
    • Pick index $i \in \{1, 2, ..d\}$ and number $t_\tau > 0$, such that
          $\mathcal{U}\left(\mathbf{v}_i, \delta_U, \mathbf{A}_\tau, U_\tau\right) \leq \mathcal{L}\left(\mathbf{v}_i, \delta_L, \mathbf{A}_\tau, L_\tau\right)$.
    • Let $t_\tau^{-1} = \frac{1}{2}\left(\mathcal{U}\left(\mathbf{v}_i, \delta_U, \mathbf{A}_\tau, U_\tau\right) + \mathcal{L}\left(\mathbf{v}_i, \delta_L, \mathbf{A}_\tau, L_\tau\right)\right)$
    • Update $\mathbf{A}_{\tau+1} = \mathbf{A}_\tau + t_\tau \mathbf{v}_i \mathbf{v}_i^T$ ; set $\mathbf{S}_{i_\tau, \tau+1} = 1$ and $\mathbf{D}_{\tau+1,\tau+1} = 1/\sqrt{t_\tau}$.
4. end for
5. Multiply all the weights in $\mathbf{D}$ by $\sqrt{r^{-1}\left(1 - \sqrt{(\ell/r)}\right)}$.
6. Return $\mathbf{S}$ and $\mathbf{D}$.
```

Thus,

$$\lambda_{\max}(\mathbf{U}^T\mathbf{U} - \mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U}) \leq (1 - (1 - \sqrt{\ell/r})^2) \leq 2\sqrt{\ell/r}.$$

Similarly,

$$\lambda_{\min}(\mathbf{U}^T\mathbf{U} - \mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U}) \geq (1 - (1 + \sqrt{\ell/r})^2) \geq 3\sqrt{\ell/r}.$$

Combining these, we have $\left\|\mathbf{U}^T\mathbf{U} - \mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U}\right\|_2 \leq 3\sqrt{\ell/r}$.

Note: Let $\epsilon = 3\sqrt{\ell/r}$. It is possible to set an upper bound on ϵ by setting the value of r. We will assume $\epsilon \in (0, 1/2]$.

4.2 Leverage Score Sampling. Our randomized feature selection method is based on importance sampling or the so-called leverage-score sampling of Rudelson and Vershynin (2007). Let **U** be the top-ρ left singular vectors of the training set **X**. We select rows of **U** in i.i.d. trials and rescale the rows with $1/\sqrt{p_i}$ where p_i is a carefully chosen probability distribution of the form

$$p_i = \frac{\|\mathbf{U}_i\|_2^2}{n}, \text{ for } i = 1, 2, \ldots, d, \tag{4.1}$$

that is, proportional to the squared Euclidean norms of the rows of the left-singular vectors. The time complexity is dominated by the time to compute the SVD of **X**.

Lemma 3 *(Rudelson and Vershynin, 2007). Let $\epsilon \in (0, 1/2]$ be an accuracy parameter and $\delta \in (0, 1)$ be the failure probability. Given $\boldsymbol{U} \in \mathbb{R}^{d\times\ell}$ satisfying $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_\ell$, let $\tilde{p} = min\{1, rp_i\}$, p_i be as equation 4.1, and $r = O(\frac{n}{\epsilon^2}\log(\frac{n}{\epsilon^2\sqrt{\delta}}))$. Construct the sampling and rescaling matrix $\boldsymbol{R}$. Then with probability at least $(1-\delta)$, $\|\boldsymbol{U}^T\boldsymbol{U} - \boldsymbol{U}^T\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{U}\|_2 \le \epsilon$.*

5 Theory

In this section, we describe the theoretical guarantees of RLSC using BSS and also the risk bounds of ridge regression using BSS and leverage-score sampling. Before we begin, we state the following lemmas from numerical linear algebra, which will be required for our proofs.

Lemma 4 *(Stewart & Sun, 1990). For any matrix $\boldsymbol{E}$, such that $\boldsymbol{I} + \boldsymbol{E}$ is invertible, $(\boldsymbol{I}+\boldsymbol{E})^{-1} = \boldsymbol{I} + \sum_{i=1}^{\infty}(-\boldsymbol{E})^i$.*

Lemma 5 *(Stewart & Sun, 1990). Let $\boldsymbol{A}$ and $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{E}$ be invertible matrices. Then*

$$\tilde{\boldsymbol{A}}^{-1} - \boldsymbol{A}^{-1} = -\boldsymbol{A}^{-1}\boldsymbol{E}\tilde{\boldsymbol{A}}^{-1}.$$

Lemma 6 *(Demmel & Veselic, 1992). Let $\boldsymbol{D}$ and $\boldsymbol{X}$ be matrices such that the product $\boldsymbol{DXD}$ is a symmetric positive-definite matrix with matrix $\boldsymbol{X}_{ii} = 1$. Let the product $\boldsymbol{DED}$ be a perturbation such that $\|E\|_2 = \eta < \lambda_{min}(\boldsymbol{X})$. Here λ_{min} corresponds to the smallest eigenvalue of $\boldsymbol{X}$. Let λ_i be the* i*th eigenvalue of $\boldsymbol{DXD}$, and let $\tilde{\lambda}_i$ be the* i*th eigenvalue of $\boldsymbol{D}(\boldsymbol{X}+\boldsymbol{E})\boldsymbol{D}$. Then, $\left|\frac{\lambda_i - \tilde{\lambda}_i}{\lambda_i}\right| \le \frac{\eta}{\lambda_{min}(\boldsymbol{X})}$.*

Lemma 7. *Let $\epsilon \in (0, 1/2]$. Then $\left\|\boldsymbol{q}^T\boldsymbol{U}^\perp\boldsymbol{U}^{\perp T}\boldsymbol{R}^T\boldsymbol{R}\boldsymbol{U}\right\|_2 \le \epsilon \left\|\boldsymbol{U}^\perp\boldsymbol{U}^{\perp T}\boldsymbol{q}\right\|_2$.*

The proof of this lemma is similar to lemma 4.3 of Drineas, Mahoney, and Muthukrishnan (2006).

5.1 Our Main Theorems on RLSC. Theorem 1 shows the additive error guarantees of the generalization bounds of the approximate classifer with that of the classifier with no feature selection. The classification error bound of BSS on RLSC depends on the condition number of the training set and on how much of the test set lies in the subspace of the training set.

Theorem 1. *Let $\epsilon \in (0, 1/2]$ be an accuracy parameter and $r = O\left(n/\epsilon^2\right)$ be the number of features selected by BSS. Let $\boldsymbol{R} \in \mathbb{R}^{r\times d}$ be the matrix, as defined in lemma 2. Let $\boldsymbol{X} \in \mathbb{R}^{d\times n}$ with $d >> n$, be the training set, $\tilde{\boldsymbol{X}} = \boldsymbol{RX}$ the reduced dimensional matrix, and $\boldsymbol{q} \in \mathbb{R}^d$ the test point of the form $\boldsymbol{q} = \boldsymbol{X\alpha} + \boldsymbol{U}^{\perp}\boldsymbol{\beta}$. Then the following hold:*

- *If $\lambda = 0$, then $\left|\tilde{\boldsymbol{q}}^T\tilde{\boldsymbol{X}}\tilde{x}_{opt} - \boldsymbol{q}^T\boldsymbol{X}\boldsymbol{x}_{opt}\right| \leq \frac{\epsilon\kappa_X}{\sigma_{max}} \|\boldsymbol{\beta}\|_2 \left\|\boldsymbol{y}\right\|_2.$*
- *If $\lambda > 0$, then $\left|\tilde{\boldsymbol{q}}^T\tilde{\boldsymbol{X}}\tilde{x}_{opt} - \boldsymbol{q}^T\boldsymbol{X}\boldsymbol{x}_{opt}\right| \leq 2\epsilon\kappa_X \|\boldsymbol{\alpha}\|_2 \left\|\boldsymbol{y}\right\|_2 + \frac{2\epsilon\kappa_X}{\sigma_{max}} \|\boldsymbol{\beta}\|_2 \left\|\boldsymbol{y}\right\|_2.$*

Proof. We assume that $\mathbf{X}$ is a full-rank matrix. Let $\mathbf{E} = \mathbf{U}^T\mathbf{U} - \mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U}$ and $\|\mathbf{E}\|_2 = \left\|\mathbf{I} - \mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U}\right\|_2 = \epsilon \leq 1/2$. Using the SVD of $\mathbf{X}$, we define

$$\boldsymbol{\Delta} = \boldsymbol{\Sigma}\mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U}\boldsymbol{\Sigma} = \boldsymbol{\Sigma}\left(\mathbf{I} + \mathbf{E}\right)\boldsymbol{\Sigma}. \tag{5.1}$$

The optimal solution in the sampled space is given by

$$\tilde{\mathbf{x}}_{opt} = \mathbf{V}\left(\boldsymbol{\Delta} + \lambda\mathbf{I}\right)^{-1}\mathbf{V}^T\mathbf{y}. \tag{5.2}$$

It can be proven easily that $\boldsymbol{\Delta}$ and $\boldsymbol{\Delta} + \lambda\mathbf{I}$ are invertible matrices. We focus on the term $\mathbf{q}^T\mathbf{X}\mathbf{x}_{opt}$. Using the SVD of $\mathbf{X}$, we get

$$\begin{aligned}\mathbf{q}^T\mathbf{X}\mathbf{x}_{opt} &= \boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{X}\mathbf{x}_{opt} + \boldsymbol{\beta}\mathbf{U}^{\perp T}\left(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\right)\mathbf{x}_{opt}\\ &= \boldsymbol{\alpha}^T\mathbf{V}\boldsymbol{\Sigma}^2\left(\boldsymbol{\Sigma}^2 + \lambda\mathbf{I}\right)^{-1}\mathbf{V}^T\mathbf{y} \qquad (5.3)\\ &= \boldsymbol{\alpha}^T\mathbf{V}\left(\mathbf{I} + \lambda\boldsymbol{\Sigma}^{-2}\right)^{-1}\mathbf{V}^T\mathbf{y}. \qquad (5.4)\end{aligned}$$

Equation 5.3 follows because of the fact that $\mathbf{U}^{\perp T}\mathbf{U} = 0$ and by substituting $\mathbf{x}_{opt}$ from equation 3.2. Equation 5.4 follows from the fact that the matrices $\boldsymbol{\Sigma}^2$ and $\boldsymbol{\Sigma}^2 + \lambda\mathbf{I}$ are invertible. Now,

$$\begin{aligned}\left|\mathbf{q}^T\mathbf{X}\mathbf{x}_{opt} - \tilde{\mathbf{q}}^T\tilde{\mathbf{X}}\tilde{\mathbf{x}}_{opt}\right| &= \left|\mathbf{q}^T\mathbf{X}\mathbf{x}_{opt} - \mathbf{q}^T\mathbf{R}^T\mathbf{R}\mathbf{X}\tilde{\mathbf{x}}_{opt}\right|\\ &\leq \left|\mathbf{q}^T\mathbf{X}\mathbf{x}_{opt} - \boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{R}^T\mathbf{R}\mathbf{X}\tilde{\mathbf{x}}_{opt}\right| \qquad (5.5)\\ &\quad + \left|\boldsymbol{\beta}^T\mathbf{U}^{\perp T}\mathbf{R}^T\mathbf{R}\mathbf{X}\tilde{\mathbf{x}}_{opt}\right|. \qquad (5.6)\end{aligned}$$

We bound equations 5.5 and 5.6 separately. Substituting the values of $\tilde{\mathbf{x}}_{opt}$ and $\boldsymbol{\Delta}$:

$$\begin{aligned}\boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{R}^T\mathbf{R}\mathbf{X}\tilde{\mathbf{x}}_{opt} &= \boldsymbol{\alpha}^T\mathbf{V}\boldsymbol{\Delta}\mathbf{V}^T\tilde{\mathbf{x}}_{opt}\\ &= \boldsymbol{\alpha}^T\mathbf{V}\boldsymbol{\Delta}\left(\boldsymbol{\Delta}+\lambda\mathbf{I}\right)^{-1}\mathbf{V}^T\mathbf{y}\\ &= \boldsymbol{\alpha}^T\mathbf{V}\left(\mathbf{I}+\lambda\boldsymbol{\Delta}^{-1}\right)^{-1}\mathbf{V}^T\mathbf{y}\\ &= \boldsymbol{\alpha}^T\mathbf{V}\left(\mathbf{I}+\lambda\boldsymbol{\Sigma}^{-1}\left(\mathbf{I}+\mathbf{E}\right)^{-1}\boldsymbol{\Sigma}^{-1}\right)^{-1}\mathbf{V}^T\mathbf{y}\\ &= \boldsymbol{\alpha}^T\mathbf{V}\left(\mathbf{I}+\lambda\boldsymbol{\Sigma}^{-2}+\lambda\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi}\boldsymbol{\Sigma}^{-1}\right)^{-1}\mathbf{V}^T\mathbf{y}. \qquad (5.7)\end{aligned}$$

The last line follows from lemma 4, which states that $(\mathbf{I}+\mathbf{E})^{-1}=\mathbf{I}+\boldsymbol{\Phi}$, where $\boldsymbol{\Phi}=\sum\limits_{i=1}^{\infty}(-\mathbf{E})^i$. The spectral norm of $\boldsymbol{\Phi}$ is bounded by

$$\|\boldsymbol{\Phi}\|_2=\left\|\sum_{i=1}^{\infty}(-\mathbf{E})^i\right\|_2\le\sum_{i=1}^{\infty}\|\mathbf{E}\|_2^i\le\sum_{i=1}^{\infty}\epsilon^i=\epsilon/(1-\epsilon). \qquad (5.8)$$

We now bound equation 5.5. Substituting equations 5.4 and 5.7 in 5.5,

$$\begin{aligned}&\left|\mathbf{q}^T\mathbf{X}\mathbf{x}_{opt}-\boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{R}^T\mathbf{R}\mathbf{X}\tilde{\mathbf{x}}_{opt}\right|\\ &=\left|\boldsymbol{\alpha}^T\mathbf{V}\{\left(\mathbf{I}+\lambda\boldsymbol{\Sigma}^{-2}+\lambda\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi}\boldsymbol{\Sigma}^{-1}\right)^{-1}-\left(\mathbf{I}+\lambda\boldsymbol{\Sigma}^{-2}\right)^{-1}\}\mathbf{V}^T\mathbf{y}\right|\\ &\le\left\|\boldsymbol{\alpha}^T\mathbf{V}\left(\mathbf{I}+\lambda\boldsymbol{\Sigma}^{-2}\right)\right\|_2\left\|\mathbf{V}^T\mathbf{y}\right\|_2\|\boldsymbol{\Psi}\|_2.\end{aligned}$$

The last line follows because of lemma 5 and the fact that all matrices involved are invertible. Here,

$$\begin{aligned}\boldsymbol{\Psi}&=\lambda\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi}\boldsymbol{\Sigma}^{-1}\left(\mathbf{I}+\lambda\boldsymbol{\Sigma}^{-2}+\lambda\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi}\boldsymbol{\Sigma}^{-1}\right)^{-1}\\ &=\lambda\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi}\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\Sigma}^2+\lambda\mathbf{I}+\lambda\boldsymbol{\Phi}\right)\boldsymbol{\Sigma}^{-1}\right)^{-1}\\ &=\lambda\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi}\left(\boldsymbol{\Sigma}^2+\lambda\mathbf{I}+\lambda\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Sigma}.\end{aligned}$$

Since the spectral norms of $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma}^{-1}$ and $\boldsymbol{\Phi}$ are bounded, we need only to bound the spectral norm of $(\boldsymbol{\Sigma}^2+\lambda\mathbf{I}+\lambda\boldsymbol{\Phi})^{-1}$ to bound the spectral norm of $\boldsymbol{\Psi}$. The spectral norm of the matrix $(\boldsymbol{\Sigma}^2+\lambda\mathbf{I}+\lambda\boldsymbol{\Phi})^{-1}$ is the inverse of the smallest singular value of $(\boldsymbol{\Sigma}^2+\lambda\mathbf{I}+\lambda\boldsymbol{\Phi})$. From perturbation theory of matrices (Stewart & Sun, 1990) and equation 5.8, we get

$$\left|\sigma_i\left(\boldsymbol{\Sigma}^2+\lambda\mathbf{I}+\lambda\boldsymbol{\Phi}\right)-\sigma_i\left(\boldsymbol{\Sigma}^2+\lambda\mathbf{I}\right)\right|\le\|\lambda\boldsymbol{\Phi}\|_2\le\epsilon\lambda.$$

Here, $\sigma_i(\mathbf{Q})$ represents the ith singular value of the matrix $\mathbf{Q}$. Also, $\sigma_i^2\left(\boldsymbol{\Sigma}^2+\lambda\mathbf{I}\right)=\sigma_i^{\,2}+\lambda$, where σ_i are the singular values of $\mathbf{X}$:

$$\sigma_i^{\,2}+(1-\epsilon)\lambda \le \sigma_i\left(\boldsymbol{\Sigma}^2+\lambda\mathbf{I}+\lambda\boldsymbol{\Phi}\right) \le \sigma_i^{\,2}+(1+\epsilon)\lambda.$$

Thus,

$$\left\|\left(\boldsymbol{\Sigma}^2+\lambda\mathbf{I}+\lambda\boldsymbol{\Phi}\right)^{-1}\right\|_2 = 1/\sigma_{\min}\left(\boldsymbol{\Sigma}^2+\lambda\mathbf{I}+\lambda\boldsymbol{\Phi}\right) \le 1/\left({\sigma^2}_{\min}+(1-\epsilon)\lambda\right).$$

Here, $\sigma_{\max}$ and $\sigma_{\min}$ denote the largest and smallest singular value of $\mathbf{X}$. Since $\|\boldsymbol{\Sigma}\|_2\left\|\boldsymbol{\Sigma}^{-1}\right\|_2=\sigma_{\max}/\sigma_{\min}\le\kappa_{\mathbf{X}}$, (condition number of $\mathbf{X}$), we bound equation 5.5:

$$\begin{aligned}&\left|\mathbf{q}^T\mathbf{X}\mathbf{x}_{opt}-\boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{R}^T\mathbf{R}\mathbf{X}\tilde{\mathbf{x}}_{opt}\right|\\ &\quad\le\frac{\epsilon\lambda\kappa_{\mathbf{X}}}{{\sigma^2}_{\min}+(1-\epsilon)\lambda}\left\|\boldsymbol{\alpha}^T\mathbf{V}\left(\mathbf{I}+\lambda\boldsymbol{\Sigma}^{-2}\right)^{-1}\right\|_2\left\|\mathbf{V}^T\mathbf{y}\right\|_2.\end{aligned}\tag{5.9}$$

For $\lambda>0$, the term ${\sigma^2}_{\min}+(1-\epsilon)\lambda$ in equation 5.9 is always larger than $(1-\epsilon)\,\lambda$, so it can be upper-bounded by $2\epsilon\kappa_{\mathbf{X}}$ (assuming $\epsilon\le 1/2$). Also,

$$\left\|\boldsymbol{\alpha}^T\mathbf{V}\left(\mathbf{I}+\lambda\boldsymbol{\Sigma}^{-2}\right)^{-1}\right\|_2\le\left\|\boldsymbol{\alpha}^T\mathbf{V}\right\|_2\left\|\left(\mathbf{I}+\lambda\boldsymbol{\Sigma}^{-2}\right)^{-1}\right\|_2\le\|\boldsymbol{\alpha}\|_2.$$

This follows from the fact that $\left\|\boldsymbol{\alpha}^T\mathbf{V}\right\|_2=\|\boldsymbol{\alpha}\|_2$ and $\left\|\mathbf{V}\mathbf{y}\right\|_2=\left\|\mathbf{y}\right\|_2$ as $\mathbf{V}$ is a full-rank orthonormal matrix and the singular values of $\mathbf{I}+\lambda\boldsymbol{\Sigma}^{-2}$ are equal to $1+\lambda/\sigma_i^{\,2}$, making the spectral norm of its inverse at most one. Thus we get,

$$\left|\mathbf{q}^T\mathbf{X}\mathbf{x}_{opt}-\boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{R}^T\mathbf{R}\mathbf{X}\tilde{\mathbf{x}}_{opt}\right|\le 2\epsilon\kappa_{\mathbf{X}}\|\boldsymbol{\alpha}\|_2\left\|\mathbf{y}\right\|_2.\tag{5.10}$$

We now bound equation 5.6. Expanding equation 5.6 using SVD and $\tilde{\mathbf{x}}_{opt}$,

$$\begin{aligned}\left|\boldsymbol{\beta}^T\mathbf{U}^{\perp T}\mathbf{R}^T\mathbf{R}\mathbf{X}\tilde{\mathbf{x}}_{opt}\right|&=\left|\boldsymbol{\beta}^T\mathbf{U}^{\perp T}\mathbf{R}^T\mathbf{R}\mathbf{U}\boldsymbol{\Sigma}\left(\boldsymbol{\Delta}+\lambda\mathbf{I}\right)\mathbf{V}^T\mathbf{y}\right|\\ &\le\left\|\mathbf{q}^T\mathbf{U}^{\perp}\mathbf{U}^{\perp T}\mathbf{R}^T\mathbf{R}\mathbf{U}\right\|_2\left\|\boldsymbol{\Sigma}\left(\boldsymbol{\Delta}+\lambda\mathbf{I}\right)^{-1}\right\|_2\left\|\mathbf{V}^T\mathbf{y}\right\|_2\\ &\le\epsilon\left\|\mathbf{U}^{\perp}\mathbf{U}^{\perp T}\mathbf{q}\right\|_2\left\|\mathbf{V}^T\mathbf{y}\right\|_2\left\|\boldsymbol{\Sigma}\left(\boldsymbol{\Delta}+\lambda\mathbf{I}\right)^{-1}\right\|_2\\ &\le\epsilon\|\boldsymbol{\beta}\|_2\left\|\mathbf{y}\right\|_2\left\|\boldsymbol{\Sigma}\left(\boldsymbol{\Delta}+\lambda\mathbf{I}\right)^{-1}\right\|_2.\end{aligned}$$

The first inequality follows from $\boldsymbol{\beta} = \mathbf{U}^{\perp T}\mathbf{q}$, and the second inequality follows from Lemma 7. To conclude the proof, we bound the spectral norm of $\boldsymbol{\Sigma}\left(\boldsymbol{\Delta} + \lambda\mathbf{I}\right)^{-1}$. Note that from equation 5.1, $\boldsymbol{\Sigma}^{-1}\boldsymbol{\Delta}\boldsymbol{\Sigma}^{-1} = \mathbf{I} + \mathbf{E}$ and $\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1} = \mathbf{I}$,

$$\boldsymbol{\Sigma}\left(\boldsymbol{\Delta} + \lambda\mathbf{I}\right)^{-1} = \left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Delta}\boldsymbol{\Sigma}^{-1} + \lambda\boldsymbol{\Sigma}^{-2}\right)^{-1}\boldsymbol{\Sigma}^{-1} = \left(\mathbf{I} + \lambda\boldsymbol{\Sigma}^{-2} + \mathbf{E}\right)^{-1}\boldsymbol{\Sigma}^{-1}.$$

One can get a lower bound for the smallest singular value of $\left(\mathbf{I} + \lambda\boldsymbol{\Sigma}^{-2} + \mathbf{E}\right)^{-1}$ using matrix perturbation theory and by comparing the singular values of this matrix to the singular values of $\mathbf{I} + \lambda\boldsymbol{\Sigma}^{-2}$. We get

$$(1-\epsilon) + \frac{\lambda}{\sigma_i^2} \le \sigma_i\left(\mathbf{I} + \mathbf{E} + \lambda\boldsymbol{\Sigma}^{-2}\right) \le (1+\epsilon) + \frac{\lambda}{\sigma_i^2}.$$

$$\begin{aligned}\left\|\left(\mathbf{I} + \lambda\boldsymbol{\Sigma}^{-2} + \mathbf{E}\right)^{-1}\boldsymbol{\Sigma}^{-1}\right\|_2 &\le \frac{\sigma^2{}_{\max}}{\left((1-\epsilon)\,\sigma^2{}_{\max} + \lambda\right)\sigma_{\min}} \\ &= \frac{\kappa_{\mathbf{X}}\sigma_{\max}}{(1-\epsilon)\,\sigma^2{}_{\max} + \lambda} \\ &\le \frac{2\kappa_{\mathbf{X}}}{\sigma_{\max}}. \qquad (5.11)\end{aligned}$$

We assumed that $\epsilon \le 1/2$, which implies $(1-\epsilon) + \lambda/\sigma^2{}_{\max} \ge 1/2$. Combining these, we get

$$\left|\boldsymbol{\beta}^T\mathbf{U}^{\perp T}\mathbf{R}^T\mathbf{R}\mathbf{X}\tilde{\mathbf{x}}_{opt}\right| \le \frac{2\epsilon\kappa_{\mathbf{X}}}{\sigma_{max}}\left\|\boldsymbol{\beta}\right\|_2\left\|\mathbf{y}\right\|_2. \qquad (5.12)$$

Combining equations 5.10 and 5.12, we complete the proof for the case $\lambda > 0$. For $\lambda = 0$, equation 5.9 becomes zero, and the result follows.

Our next theorem provides relative-error guarantees to the bound on the classification error when the test point has no new components, that is, $\boldsymbol{\beta} = 0$.

Theorem 2. *Let $\epsilon \in (0, 1/2]$ be an accuracy parameter, $r = O\left(n/\epsilon^2\right)$ be the number of features selected by BSS, and $\lambda > 0$. Let $\boldsymbol{q} \in \mathbb{R}^d$ be the test point of the form $\boldsymbol{q} = \boldsymbol{X}\boldsymbol{\alpha}$, that is, it lies entirely in the subspace spanned by the training set, and the two vectors $\boldsymbol{V}^T\boldsymbol{y}$ and $\left(\boldsymbol{I} + \lambda\boldsymbol{\Sigma}^{-2}\right)^{-1}\boldsymbol{V}^T\boldsymbol{\alpha}$ satisfy the property*

$$\begin{aligned}\left\|\left(\boldsymbol{I} + \lambda\boldsymbol{\Sigma}^{-2}\right)^{-1}\boldsymbol{V}^T\boldsymbol{\alpha}\right\|_2\left\|\boldsymbol{V}^T\boldsymbol{y}\right\|_2 &\le \omega\left\|\left(\left(\boldsymbol{I} + \lambda\boldsymbol{\Sigma}^{-2}\right)^{-1}\boldsymbol{V}^T\boldsymbol{\alpha}\right)^T\boldsymbol{V}^T\boldsymbol{y}\right\|_2 \\ &= \omega\left|\boldsymbol{q}^T\boldsymbol{X}\boldsymbol{x}_{opt}\right|\end{aligned}$$

for some constant ω. If we run RLSC after BSS, then

$$\left|\tilde{\boldsymbol{q}}^T \tilde{\boldsymbol{X}} \tilde{x}_{opt} - \boldsymbol{q}^T \boldsymbol{X} \boldsymbol{x}_{opt}\right| \leq 2\epsilon\omega\kappa_{\boldsymbol{X}} \left|\boldsymbol{q}^T \boldsymbol{X} \boldsymbol{x}_{opt}\right| .$$

The proof follows directly from the proof of theorem 1 if we consider $\boldsymbol{\beta} = 0$.

5.2 Our Main Theorems on Ridge Regression. We compare the risk of subsampled ridge regression with the risk of true dual ridge regression in the fixed design setting. Recall that the response vector $\mathbf{y} = \mathbf{X}^T \boldsymbol{\beta} + \boldsymbol{\omega}$ where $\boldsymbol{\omega} \in \mathbb{R}^n$ is the homoskedastic noise vector with mean 0 and variance σ^2. Also, we assume that the data matrix is of full rank.

Lemma 8. *Let ρ be the rank of $\boldsymbol{X}$. Form $\tilde{\boldsymbol{K}}$ using BSS. Then,*

$$(1 - \Delta)\boldsymbol{K} \preceq \tilde{\boldsymbol{K}} \preceq (1 + \Delta)\boldsymbol{K},$$

where $\Delta = C\sqrt{\rho/r}$. For p.s.d matrices $\boldsymbol{A} \succeq \boldsymbol{B}$ means $\boldsymbol{B} - \boldsymbol{A}$ is p.s.d.

Proof. Using the SVD of $\mathbf{X}$, $\tilde{\mathbf{K}} = \mathbf{V\Sigma}\left(\mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U}\right)\mathbf{\Sigma V}^T$. Lemma 2 implies

$$\mathbf{I}_\rho\,(1 - \Delta) \preceq \left(\mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U}\right) \preceq \mathbf{I}_\rho\,(1 + \Delta) .$$

Multiplying the left- and right-hand sides of the inequality by $\mathbf{V\Sigma}$ and $\mathbf{\Sigma V}^T$, respectively, to the above inequality completes the proof.

Lemma 9. *Let ρ be the rank of $\boldsymbol{X}$. Form $\tilde{\boldsymbol{K}}$ using leverage-score sampling. Then, with probability at least $(1 - \delta)$, where $\delta \in (0, 1)$,*

$$(1 - \Delta)\boldsymbol{K} \preceq \tilde{\boldsymbol{K}} \preceq (1 + \Delta)\boldsymbol{K},$$

where $\Delta = C\frac{\rho}{\epsilon^2}\log\left(\frac{\rho}{\epsilon^2\sqrt{\delta}}\right)$.

5.3 Risk Function for Ridge Regression. Let $\mathbf{z} = \mathbf{E}_\omega[\mathbf{y}] = \mathbf{X}^T\boldsymbol{\beta}$. The risk for a prediction function $\hat{\mathbf{y}} \in \mathbb{R}^n$ is $\frac{1}{n}\mathbf{E}_\omega\|\hat{\mathbf{y}} - \mathbf{z}\|_2^2$. For any $n \times n$ positive symmetric matrix $\mathbf{K}$, we define the following risk function:

$$R(\mathbf{K}) = \frac{\sigma^2}{n}\mathbf{Tr}\left(\mathbf{K}^2\left(\mathbf{K} + n\lambda\mathbf{I}_n\right)^{-2}\right) + n\lambda^2\mathbf{z}^T\left(\mathbf{K} + n\lambda\mathbf{I}_n\right)^{-2}\mathbf{z}.$$

Theorem 3. *Under the fixed design setting, the risk for the ridge regression solution in the full-feature space is $R(\boldsymbol{K})$, and the risk for the ridge regression in the reduced dimensional space is $R(\tilde{\boldsymbol{K}})$.*

Proof. The risk of the ridge regression estimator in the reduced dimensional space is

$$\frac{1}{n}\mathbf{E}_{\omega}\left\|\tilde{\mathbf{K}}\tilde{\boldsymbol{\alpha}}_{\lambda}-\mathbf{z}\right\|_2^2=\frac{1}{n}\mathbf{E}_{\omega}\left\|\tilde{\mathbf{K}}\left(\tilde{\mathbf{K}}+n\lambda\mathbf{I}_n\right)^{-1}\mathbf{y}-\mathbf{z}\right\|_2^2. \tag{5.13}$$

Taking $\tilde{\mathbf{K}}(\tilde{\mathbf{K}}+n\lambda\mathbf{I}_n)^{-1}$ as $\mathbf{Q}$, we can write equation 5.13 as

$$\begin{aligned}
&\frac{1}{n}\mathbf{E}_{\omega}\left\|\mathbf{Q}\mathbf{y}-\mathbf{E}_{\omega}\left[\mathbf{Q}\mathbf{y}\right]\right\|_2^2+\frac{1}{n}\left\|\mathbf{E}_{\omega}\left[\mathbf{Q}\mathbf{y}\right]-\mathbf{z}\right\|_2^2\\
&=\frac{1}{n}\mathbf{E}_{\omega}\left[\left\|\tilde{\mathbf{K}}\left(\tilde{\mathbf{K}}+n\lambda\mathbf{I}_n\right)^{-1}\boldsymbol{\omega}\right\|_2^2\right]+\frac{1}{n}\left\|\tilde{\mathbf{K}}\left(\tilde{\mathbf{K}}+n\lambda\mathbf{I}_n\right)^{-1}\mathbf{z}-\mathbf{z}\right\|_2^2\\
&=\frac{1}{n}\mathbf{Tr}\left(\tilde{\mathbf{K}}^2\left(\tilde{\mathbf{K}}+n\lambda\mathbf{I}_n\right)^{-2}\boldsymbol{\omega}\boldsymbol{\omega}^T\right)+\frac{1}{n}\mathbf{z}^T\left(\mathbf{I}_n-\tilde{\mathbf{K}}\left(\tilde{\mathbf{K}}+n\lambda\mathbf{I}_n\right)^{-1}\right)^2\mathbf{z}\\
&=\frac{\sigma^2}{n}\mathbf{Tr}\left(\tilde{\mathbf{K}}^2\left(\tilde{\mathbf{K}}+n\lambda\mathbf{I}_n\right)^{-2}\right)+n\lambda^2\mathbf{z}^T\left(\tilde{\mathbf{K}}+n\lambda\mathbf{I}_n\right)^{-2}\mathbf{z}.
\end{aligned}$$

The expectation is only over the random noise $\boldsymbol{\omega}$ and is conditional on the feature selection method used.

Our next theorem bounds the risk inflation of ridge regression in the reduced dimensional space compared with the ridge regression solution in the full-feature space.

Theorem 4. *Let ρ be the rank of the matrix $\mathbf{X}$. When using leverage-score sampling as a feature selection technique, with probability at least $1-\delta$, where $\delta\in(0,1)$,*

$$R(\tilde{\boldsymbol{K}})\leq(1-\Delta)^{-2}R(\boldsymbol{K}),$$

where $\Delta=C\frac{\rho}{\epsilon^2}\log\left(\frac{\rho}{\epsilon^2\sqrt{\delta}}\right)$.

Proof. For any positive semidefinite matrix, $\mathbf{K}\in\mathbb{R}^{n\times n}$, we define the bias $B(\mathbf{K})$ and variance $V(\mathbf{K})$ of the risk function as follows:

$$\begin{aligned}
B(\mathbf{K})&=n\lambda^2\mathbf{z}^T\left(\mathbf{K}+n\lambda\mathbf{I}_n\right)^{-2}\mathbf{z},\\
V(\mathbf{K})&=\frac{\sigma^2}{n}\mathbf{Tr}\left(\tilde{\mathbf{K}}^2\left(\tilde{\mathbf{K}}+n\lambda\mathbf{I}_n\right)^{-2}\right).
\end{aligned}$$

Therefore, $R(\mathbf{K})=B(\mathbf{K})+V(\mathbf{K})$. Now, due to, Bach (2013), we know that $B(\mathbf{K})$ is nonincreasing in $\mathbf{K}$ and $V(\mathbf{K})$ is nondecreasing in $\mathbf{K}$. When lemma 9 holds,

$$\begin{aligned}R(\tilde{\mathbf{K}}) &= V(\tilde{\mathbf{K}}) + B(\tilde{\mathbf{K}})\\ &\leq V\left((1+\Delta)\,\mathbf{K}\right) + B\left((1-\Delta)\,\mathbf{K}\right)\\ &\leq (1+\Delta)^2\,V(\mathbf{K}) + (1-\Delta)^{-2}\,B(\mathbf{K})\\ &\leq (1-\Delta)^{-2}\left(V(\mathbf{K}) + B(\mathbf{K})\right)\\ &= (1-\Delta)^{-2}\,R(\mathbf{K}).\end{aligned}$$

We can prove a similar theorem for BSS.

Theorem 5. *Let ρ be the rank of the matrix $\mathbf{X}$. When using BSS as a feature selection technique, with $\Delta = C\rho/\epsilon^2$,*

$$R(\tilde{\mathbf{K}}) \leq (1-\Delta)^{-2} R(\mathbf{K}).$$

6 Experiments

All experiments were performed in Matlab R2013b on an Intel i-7 processor with 16GB RAM.

6.1 BSS Implementation Issues. Batson et al. (2009) do not provide any implementation details of the BSS algorithm. Here we discuss several issues arising during the implementation.

6.1.1 Choice of Column Selection. At every iteration, there are multiple columns that satisfy the condition $\mathcal{U}(\mathbf{u}_i, \delta_U, \mathbf{A}_\tau, U_\tau) \leq \mathcal{L}(\mathbf{u}_i, \delta_L, \mathbf{A}_\tau, L_\tau)$. Batson et al. (2009) suggest picking any column that satisfies this constraint. Instead of breaking ties arbitrarily, we choose the column $\mathbf{u}_i$, which has not been selected in previous iterations and whose Euclidean norm is highest among the candidate set. Columns with zero Euclidean norm never get selected by the algorithm. In the inner loop of algorithm 1, $\mathcal{U}$ and $\mathcal{L}$ has to be computed for all the d columns in order to pick a good column. This step can be done efficiently using a single line of Matlab code by making use of matrix and vector operations.

6.2 Other Feature Selection Methods. In this section, we describe other feature-selection methods with which we compare BSS.

6.2.1 Rank-Revealing QR Factorization. Within the numerical linear algebra community, subset selection algorithms use the so-called rank revealing QR (RRQR) factorization. Here we slightly abuse notation and state $\mathbf{A}$ as a short and fat matrix as opposed to the tall and thin matrix. Let $\mathbf{A}$ be an

$n \times d$ matrix with $(n < d)$ and an integer $k(k < d)$ and assume partial QR factorizations of the form

$$\mathbf{AP} = \mathbf{Q}\begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{R}_{22} \end{pmatrix},$$

where $\mathbf{Q} \in \mathbb{R}^{n\times n}$ is an orthogonal matrix, $\mathbf{P} \in \mathbb{R}^{d\times d}$ is a permutation matrix, $\mathbf{R}_{11} \in \mathbb{R}^{k\times k}$, $\mathbf{R}_{12} \in \mathbb{R}^{k\times(d-k)}$, and $\mathbf{R}_{22} \in \mathbb{R}^{(d-k)\times(d-k)}$. The above factorization is called an RRQR factorization if $\sigma_{\min}(\mathbf{R}_{11}) \geq \sigma_k(\mathbf{A})/p(k, d)$, $\sigma_{\max}(\mathbf{R}_{22}) \leq \sigma_{\min}(\mathbf{A})p(k, d)$, where $p(k,d)$ is a function bounded by a low-degree polynomial in k and d. The important columns are given by $\mathbf{A}_1 = \mathbf{Q}(\begin{smallmatrix}\mathbf{R}_{11}\\ \mathbf{0}\end{smallmatrix})$ and $\sigma_i(\mathbf{A}_1) = \sigma_i(\mathbf{R}_{11})$ with $1 \leq i \leq k$. We perform feature selection using RRQR by picking the important columns that preserve the rank of the matrix.

6.2.2 Random Feature Selection. We select features uniformly at random without replacement, which serves as a baseline method. To get around the randomness, we repeat the sampling process five times.

6.2.3 Leverage-Score Sampling. For leverage-score sampling, we repeat the experiments five times to get around the randomness. We pick the top-ρ left singular vectors of $\mathbf{X}$, where ρ is the rank of the matrix $\mathbf{X}$.

6.2.4 Information Gain. The information gain (IG) feature selection method (Yang & Pedersen, 1997) measures the amount of information obtained for binary class prediction by knowing the presence or absence of a feature in a data set. The method is a supervised strategy, whereas the other methods used here are unsupervised.

6.3 Experiments on RLSC. The goal of this section is to compare BSS with existing feature selection methods for RLSC and show that BSS is better than the other methods.

6.3.1 Synthetic Data. We run our experiments on synthetic data where we control the number of relevant features in the data set and demonstrate the working of algorithm 1 on RLSC. We generate synthetic data in the same manner as given in Bhattacharyya (2004). The data set has n data points and d features. The class label y_i of each data point was randomly chosen to be 1 or -1 with equal probability. The first k features of each data point $\mathbf{x}_i$ are drawn from $y_i\mathcal{N}\left(-j, 1\right)$ distribution, where $\mathcal{N}\left(\mu, \sigma^2\right)$ is a random normal distribution with mean μ and variance σ^2 and j varies from 1 to k. The remaining $d-k$ features are chosen from a $\mathcal{N}(0, 1)$ distribution. Thus, the data set has k relevant features and $(d - k)$ noisy features. By construction, among the first k features, the kth feature has the most discriminatory power,

Table 1: Most Frequently Selected Features Using the Synthetic Data Set.

$r = 80$	$k = 90$	$k = 100$
BSS	89, 88, 87, 86, 85	100, 99, 98, 97, 95
RRQR	90, 80, 79, 78, 77	100, 80, 79, 78, 77
Lvg-Score	73, 85, 84, 81, 87	93, 87, 95, 97, 96
IG	80, 79, 78, 77, 76	80, 79, 78, 77, 76
$r = 90$	$k = 90$	$k = 100$
BSS	90, 88, 87, 86, 85	100, 99, 98, 97, 96
RRQR	90, 89, 88, 87, 86	100, 90, 89, 88, 87
Lvg-Score	67, 88, 83, 87, 85	100, 97, 92, 48, 58
IG	90, 89, 88, 87, 86	90, 89, 88, 87, 86

followed by $(k - 1)$th feature and so on. We set n to 30 and d to 1000. We set k to 90 and 100 and ran two sets of experiments.

We set the value of r (i.e., the number of features selected by BSS) to 80 and 90 for all experiments. We performed ten-fold cross-validation and repeated it 10 times. The value of λ was set to 0, 0.1, 0.3, 0.5, 0.7, and 0.9. We compared BSS with RRQR, IG, and leverage-score sampling. The mean out-of-sample error was 0 for all methods for both $k = 90$ and $k = 100$. Table 1 shows the set of five most frequently selected features by the different methods for one such synthetic data set across 100 training sets. The top features picked up by the different methods are the relevant features by construction and also have good discriminatory power. This shows that BSS is as good as any other method in terms of feature selection and often picks more discriminatory features than the other methods. We repeated our experiments on 10 different synthetic data sets, and each time, the 5 most frequently selected features were from the set of relevant features. Thus, by selecting only 8% to 9% of all features, we show that we are able to obtain the most discriminatory features along with good out-of-sample error using BSS.

Though running time is not the main subject of this study, we point out that we computed the running time of the different feature selection methods averaged over 10 ten-fold cross-validation experiments. The time to perform feature selection for each of the methods averaged over 10 ten-fold cross-validation experiments was less than 1 second (see Table 2), which shows that the methods can be implemented in practice.

6.3.2 TechTC-300. We use the TechTC-300 data (Davidov et al., 2004) consisting of a family of 295 document-term data matrices. The TechTC-300 data set comes from the Open Directory Project (ODP), a large, comprehensive directory of the web, maintained by volunteer editors. Each matrix in

Table 2: Running Time of Various Feature Selection Methods (in Seconds).

	BSS	IG	LVG	RRQR
Synthetic data	0.1025	0.0003	0.0031	0.0016
TechTC-300	75.7624	0.0242	0.4054	0.2631

Notes: For synthetic data, the running time corresponds to the experiment when $r = 80$ and $k = 90$ and is averaged over 10 ten-fold cross-validation experiments. For TechTC-300, the running time corresponds to the experiment when $r = 400$ and is averaged over 10 ten-fold cross-validation experiments and over 48 TehTC-300 data sets.

the TechTC-300 data set contains a pair of categories from the ODP. Each category corresponds to a label, and thus the resulting classification task is binary. The documents that are collected from the union of all the subcategories within each category are represented in the bag-of-words model, with the words constituting the features of the data (Davidov et al., 2004). Each data matrix consists of 150 to 280 documents, and each document is described with respect to 10,000 to 50,000 words. Thus, TechTC-300 provides a diverse collection of data sets for a systematic study of the performance of the RLSC using BSS. We removed all words of length at most four from the data sets. Next, we grouped the data sets based on the categories and selected data sets whose categories appeared at least thrice. There were 147 data sets, and we performed ten-fold cross-validation and repeated it 10 times on 48 such data sets. We set the values of the regularization parameter of RLSC to 0.1, 0.3, 0.5 and 0.7.

We set r to 300, 400, and 500. We report the out-of-sample error for all 48 data sets. BSS consistently outperforms leverage-score sampling, IG, RRQR and random feature selection on all 48 data sets for all values of the regularization parameter. Table 3 and Figure 1 show the results. The out-of-sample error decreases with an increase in the number of features for all methods. In terms of out-of-sample error, BSS is the best, followed by leverage-score sampling, IG, RRQR, and random feature selection. BSS is at least 3% to 7% better than the other methods when averaged over 48 document matrices. From Figures 1 and 2, it is evident that BSS is comparable to the other methods and often better on all 48 data sets. Leverage-score sampling requires a greater number of samples to achieve the same out-of-sample error as BSS (see table 3, $r = 500$ for Lvg-Score and $r = 300$ for BSS). Therefore, for the same number of samples, BSS outperforms leverage-score sampling in terms of out-of-sample error. The out-of-sample error of supervised IG is worse than that of unsupervised BSS, which could be due to the worse generalization of the supervised IG metric. We also observe that the out-of-sample error decreases with increase in λ for the different feature selection methods.

Table 3: Out-of-Sample Error of TechTC-300 Data Sets Averaged over 10 Ten-Fold Cross-Validation and over 48 Data Sets for Three Values of r.

$r = 300$	$\lambda = 0.1$	$\lambda = 0.3$	$\lambda = 0.5$	$\lambda = 0.7$
BSS	**31.76 ± 0.68**	**31.46 ± 0.67**	**31.24 ± 0.65**	**31.03 ± 0.66**
Lvg-Score	38.22 ± 1.26	37.63 ± 1.25	37.23 ± 1.24	36.94 ± 1.24
RRQR	37.84 ± 1.20	37.07 ± 1.19	36.57 ± 1.18	36.10 ± 1.18
Randomfs	50.01 ± 1.2	49.43 ± 1.2	49.18 ± 1.19	49.04 ± 1.19
IG	38.35 ± 1.21	36.64 ± 1.18	35.81 ± 1.18	35.15 ± 1.17
$r = 400$	$\lambda = 0.1$	$\lambda = 0.3$	$\lambda = 0.5$	$\lambda = 0.7$
BSS	**30.59 ± 0.66**	**30.33 ± 0.65**	**30.11 ± 0.65**	**29.96 ± 0.65**
Lvg-Score	35.06 ± 1.21	34.63 ± 1.20	34.32 ± 1.2	34.11 ± 1.19
RRQR	36.61 ± 1.19	36.04 ± 1.19	35.46 ± 1.18	35.05 ± 1.17
Randomfs	47.82 ± 1.2	47.02 ± 1.21	46.59 ± 1.21	46.27 ± 1.2
IG	37.37 ± 1.21	35.73 ± 1.19	34.88 ± 1.18	34.19 ± 1.18
$r = 500$	$\lambda = 0.1$	$\lambda = 0.3$	$\lambda = 0.5$	$\lambda = 0.7$
BSS	**29.80 ± 0.77**	**29.53 ± 0.77**	**29.34 ± 0.76**	**29.18 ± 0.75**
Lvg-Score	33.33 ± 1.19	32.98 ± 1.18	32.73 ± 1.18	32.52 ± 1.17
RRQR	35.77 ± 1.18	35.18 ± 1.16	34.67 ± 1.16	34.25 ± 1.14
Randomfs	46.26 ± 1.21	45.39 ± 1.19	44.96 ± 1.19	44.65 ± 1.18
IG	36.24 ± 1.20	34.80 ± 1.19	33.94 ± 1.18	33.39 ± 1.17

Note: The first and second entries of each cell represent the mean and standard deviation. Items in bold indicate the best results.

We list the most frequently occurring words selected by BSS and leverage-score sampling for the $r = 300$ case for seven TechTC-300 data sets over 100 training sets used in the cross-validation experiments. Table 4 shows the names of the seven TechTC-300 document term matrices. The words shown in Tables 5 and 6 were selected in all cross-validation experiments for these seven data sets. The words are closely related to the categories to which the documents belong, which shows that BSS and leverage-score sampling select important features from the training set. For example, for the document pair (1092_789236), where 1092 belongs to the category of "Arts:Music:Styles:Opera" and 789236 belongs to the category of "US:Navy: Decommisioned Submarines," the BSS algorithm selects *submarine, shipyard, triton, opera, libretto, theatre*, which are closely related to the two classes. The top words selected by leverage-score sampling for the same document pair are *seawolf, sturgeon, opera, triton finback*, which are closely related to the class. Another example is the document pair 10539_300332, where 10539 belongs to "US:Indiana:Localities:S" and 300332 belongs to the category of "Canada: Ontario: Localities:E." The top words selected for this document pair are *ontario, elliot, shelbyville, indiana, schererville*, which are

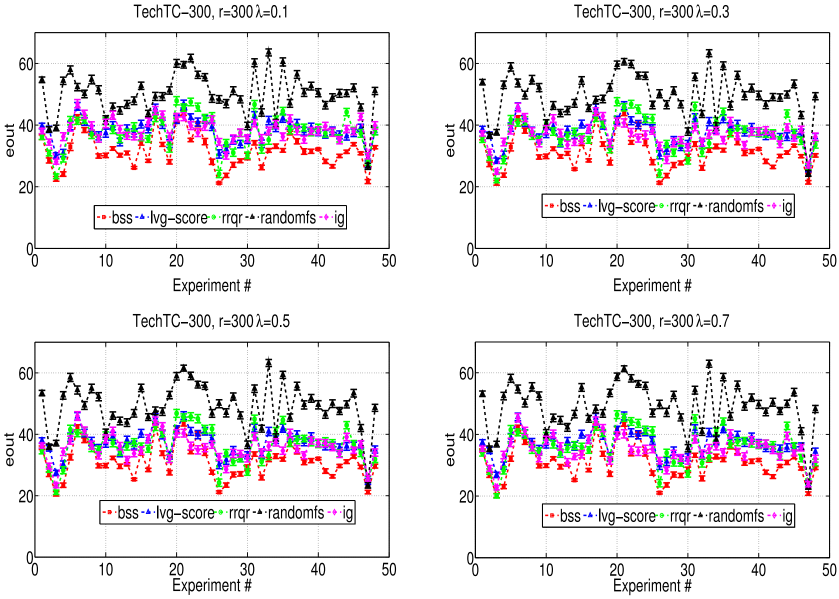
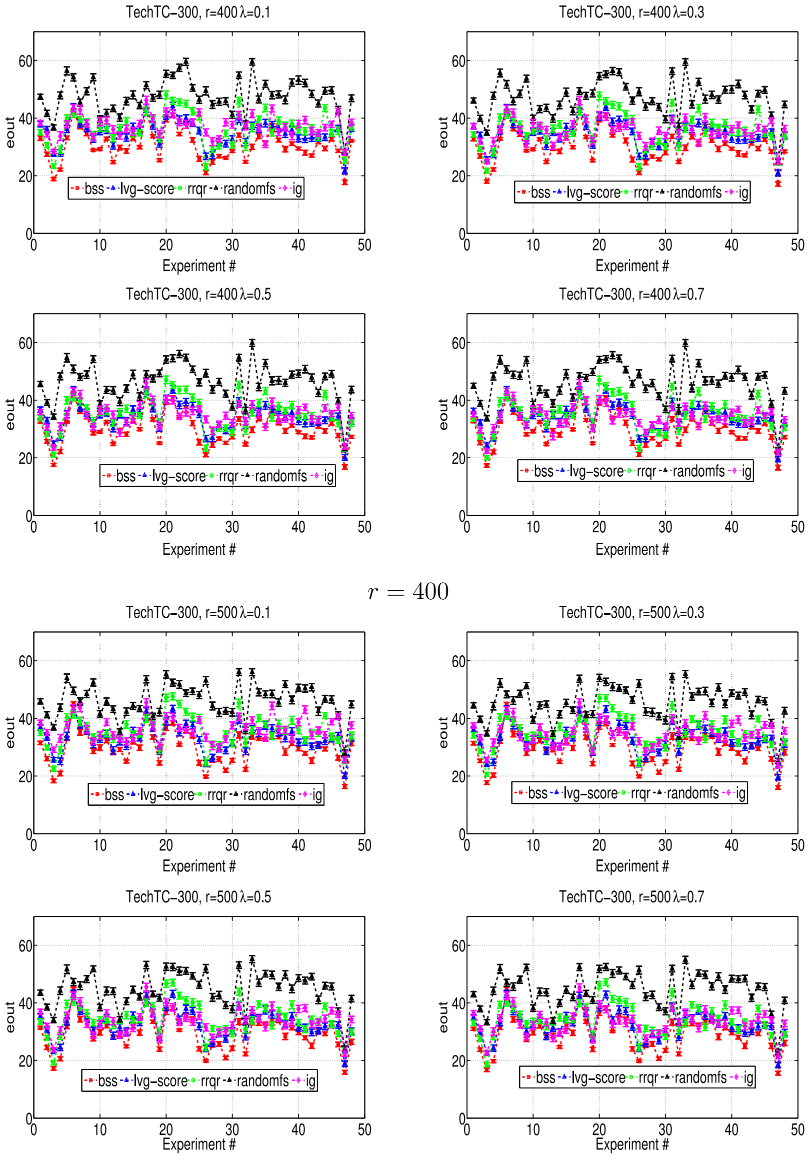


Figure 1: Out-of-sample error of 48 TechTC-300 documents averaged over 10 ten-fold cross-validation experiments for different values of regularization parameter λ and number of features $r = 300$. Vertical bars represent standard deviations.

closely related to the class values. Thus, we see that in using only 2% to 4% of all features, we are able to select relevant features and obtain good out-of-sample error. The top words selected by leverage-score sampling are *library, fishing, elliot, indiana, shelbyville, ontario*, which are closely related to the class.

Though feature selection is an offline task, we give a discussion of the running times of the different methods to highlight that BSS can be implemented in practice. We computed the running time of the different feature selection methods averaged over 10 ten-fold cross-validation experiments and over 48 data sets (see Table 2). The average time for feature selection by BSS is approximately over 1 minute, while the rest of the methods take less than 1 second. This shows that BSS can be implemented in practice and can scale up to reasonably large data sets with 20,000 to 50,000 features. For BSS and leverage-score sampling, the running time includes the time to compute the SVD of the matrix. BSS takes approximately 1 minute to select features but is at least 3% to 7% better in terms of out-of-sample error than the other methods. IG takes less than 1 second to select features, but it is 4% to 7% worse than BSS in terms of out-of-sample error.



$r = 400$

$r = 500$

Figure 2: Out-of-sample error of 48 TechTC-300 documents averaged over 10 ten-fold cross-validation experiments for different values of regularization parameter λ and number of features $r = 400$ and $r = 500$. Vertical bars represent standard deviation.

Table 4: A Subset of the TechTC Matrices of Our Study.

id1_id2	**id1**	**id2**
1092_789236	Arts:Music:Styles:Opera	US Navy:Decommisioned Submarines
17899_278949	US:Michigan:Travel & Tourism	Recreation:Sailing Clubs:UK
17899_48446	US:Michigan:Travel & Tourism	Chemistry:Analytical:Products
14630_814096	US:Colorado:Localities:Boulder	Europe:Ireland:Dublin:Localities
10539_300332	US:Indiana:Localities:S	Canada:Ontario:Localities:E
10567_11346	US:Indiana:Evansville	US:Florida:Metro Areas:Miami
10539_194915	US:Indiana:Localities:S	US:Texas:Localities:D

Table 5: Frequently Occurring Terms of the TechTC-300 Data Sets of Table 4 Selected by BSS.

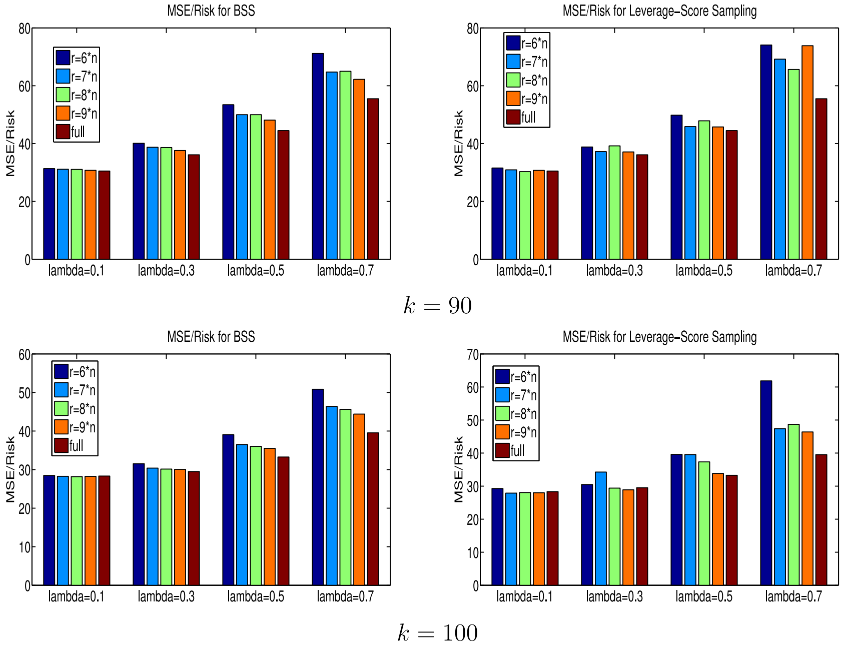
id1_id2	**Words**
1092_789236	naval,shipyard,submarine,triton,music,opera,libretto,theatre
17899_278949	sailing,cruising,boat,yacht,racing,michigan,leelanau,casino
17899_48446	vacation,lodging,michigan,asbestos,chemical,analytical,laboratory
14630_814096	ireland,dublin,boulder,colorado,lucan,swords,school,dalkey
10539_300332	ontario,fishing,county,elliot,schererville,shelbyville,indiana,bullet
10567_11346	florida,miami,beach,indiana,evansville,music,business,south
10539_194915	texas,dallas,plano,denton,indiana,schererville,gallery,north

Table 6: Frequently Occurring Terms of the TechTC-300 Data Sets of Table 4 Selected by Leverage-Score Sampling.

id1_id2	**Words**
1092_789236	sturgeon, seawolf, skate, triton, frame, opera, finback
17899_278949	sailing, yacht, laser, michigan,breakfast, county, clear
17899_48446	analysis, michigan, water, breakfast, asbestos, environmental, analytical
14630_814096	ireland, dublin, estate, lucan, dalkey, colorado, boulder
10539_300332	library, fishing, service, lodge, ontario, elliot, indiana, shelbyville
10567_11346	evansville, services, health, church, south, bullet, florida
10539_194915	dallas, texas, schererville, indiana, shelbyville, plano

6.4 Experiments on Ridge Regression in the Fixed Design Setting. In this section, we describe experiments on feature selection on ridge regression in the fixed design setting using synthetic and real data.

6.4.1 Synthetic Data. We generate the features of the synthetic data $\mathbf{X}$ in the same manner as described in section 6.3.1. We generate $\boldsymbol{\beta} \sim \mathcal{N}(0, 1)$ and $\mathbf{y} = \mathbf{X}^T \boldsymbol{\beta} + \boldsymbol{\omega}$, where $\boldsymbol{\omega} \in \mathbb{R}^n$ and $\boldsymbol{\beta} \in \mathbb{R}^d$. We set n to 30 and d to 1000.



$k = 90$

$k = 100$

Figure 3: MSE/Risk for synthetic data for $k = 90$ and $k = 100$ using different feature selection methods as a function of λ. The risk after feature selection is comparable to the risk of full data.

We set the number of relevant features k to 90 and 100 and ran two sets of experiments. We set the value of r, the number of features selected by BSS and leverage-score sampling, to $t * n$, where $t = 6, 7, 8, 9$ for both experiments. The value of λ was set to 0.1, 0.3, 0.5, and 0.7. We compared the risk of ridge regression using BSS and leverage-score sampling with the risk of full-feature selection and report the MSE/Risk in the fixed design setting as a measure of accuracy. Figure 3 shows the risk of synthetic data for both BSS and leverage-score sampling as a function of λ. The risk of the sampled data is comparable to the risk of the full data in most cases, which follows from our theory. We observe that for higher values of λ, the risk of sampled space becomes worse than that of full data for both BSS and leverage-score sampling. The risk in the sampled space is almost the same for both BSS and leverage-score sampling. The time to compute feature selection is less than 1 second for both methods (see Table 7).

6.4.2 TechTC-300. We use two TechTC-300 data sets, “10341-14525” and “10341-61792,” to illustrate our theory. We add gaussian noise to the labels. We set the value of r, the number of features to be selected, to 300, 400,

Table 7: Running Time of Various Feature Selection Methods (in Seconds).

	Synthetic Data	TechTC (10341-14525)	TechTC (10341-61792)
BSS	0.3368	68.8474	67.013
LVG	0.0045	0.3994	0.3909

Note: For synthetic data, the running time corresponds to the experiment when $r = 8n$. For TechTC-300, the running time corresponds to the experiment when $r = 400$.

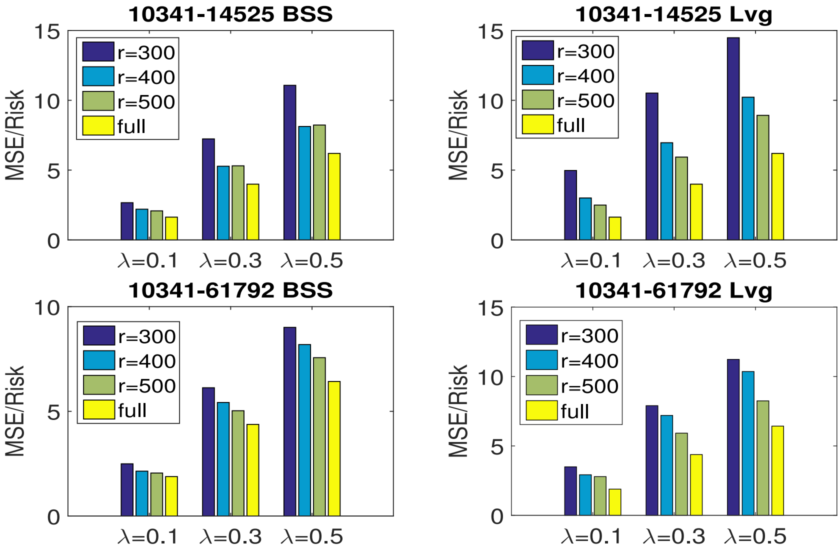


Figure 4: MSE/Risk for TechTC-300 data using different feature selection methods as a function of λ. The risk after feature selection is comparable to the risk of full data.

and 500. The value of λ was set to 0.1, 0.3 and 0.5. We compared the risk of ridge regression using BSS and leverage-score sampling with the risk of full-feature selection and report the MSE/Risk in the fixed design setting as a measure of accuracy. Figure 4 shows the risk of real data for both BSS and leverage-score sampling as a function of λ. The risk of the sampled data is comparable to the risk of the full data in most cases, which follows from our theory. The risk of the sampled data decreases with an increase in r. The time to perform feature selection is approximately 1 minute for BSS and less than 1 second for leverage-score sampling (see Table 7).

7 Conclusion

We present a provably accurate feature selection method for RLSC that works well empirically and also gives better generalization peformance than prior existing methods. The number of features required by BSS is of the order $O\left(n/\epsilon^2\right)$, which makes the result tighter than that obtained by leverage-score sampling. BSS has been used recently as a feature selection technique for *k*-means clustering (Boutsidis & Magdon-Ismail, 2013), and linear SVMs (Paul, Magdon-Ismail, & Drineas, 2015), and our work on RLSC helps to expand research in this direction. The risk of ridge regression in the sampled space is comparable to the risk of ridge regression in the full feature space in the fixed design setting, and we observe this in both theory and experiments. Interesting future work in this direction would be to include feature selection for nonlinear kernels with provable guarantees.

Acknowledgments

Most of the work was done when S.P. was a graduate student at RPI. This work is supported by NSF CCF 1016501 and NSF IIS 1319280.

References

Agarwal, D. (2002). Shrinkage estimator generalizations of proximal support vector machines. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 173–182). New York: ACM.

Avron, H., Sindhwani, V., & Woodruff, D. (2013). Sketching structured matrices for faster nonlinear regression. In C. J. C. Burges, L. Bottou, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems, 26* (pp. 2994–3002). Red Hook, NY: Curran.

Bach, F. (2013). Sharp analysis of low-rank kernel matrix approximations. In *Proceedings of the 26th Annual Conference on Learning Theory (COLT)* (pp. 185–209). JMLR.org.

Batson, J., Spielman, D., & Srivastava, N. (2009). Twice-Ramanujan sparsifiers. In *Proceedings of the 41st Annual ACM STOC* (pp. 255–262). New York: ACM.

Bhattacharyya, C. (2004). Second order cone programming formulations for feature selection. *JMLR*, *5*, 1417–1433.

Boutsidis, C., & Magdon-Ismail, M. (2013). Deterministic feature selection for *k*-means clustering. *IEEE Transactions on Information Theory*, *59*(9), 6099–6110.

Dasgupta, A., Drineas, P., Harb, B., Josifovski, V., & Mahoney, M. (2007). Feature selection methods for text classification. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 230–239). New York: ACM.

Davidov, D., Gabrilovich, E., & Markovitch, S. (2004). Parameterized generation of labeled datasets for text categorization based on a hierarchical directory. In *Proceedings of the 27th Annual International ACM SIGIR Conference* (pp. 250–257). New York: ACM. http://techtc.cs.technion.ac.il/techtc300/techtc300.html

Demmel, J., & Veselic, K. (1992). Jacobi's method is more accurate than QR. *SIAM Journal on Matrix Analysis and Applications*, *13*(4), 1204–1245.

Drineas, P., Mahoney, M., & Muthukrishnan, S. (2006). Sampling algorithms for l2 regression and applications. In *Proceedings of the 17th Annual ACM-SIAM SODA* (pp. 1127–1136). New York: ACM.

Fung, G., & Mangasarian, O. (2001). Proximal support vector machine classifiers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 77–86). New York: ACM.

Lu, Y., Dhillon, P., Foster, D., & Ungar, L. (2013). Faster ridge regression via the subsampled randomized hadamard transform. In C. J. C. Burges, L. Bottou, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems, 26* (pp. 369–377). Red Hook, NY: Curran.

Paul, S., Magdon-Ismail, M., & Drineas, P. (2015). Feature selection for linear SVM with provable guarantees. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics* (pp. 735–743). JMLR.org.

Poggio, T., & Smale, S. (2003). The mathematics of learning: Dealing with data. *Notices of the AMS*, *50*(5), 537–544.

Rifkin, R., Yeo, G., and Poggio, T. (2003). Regularized least-squares classification. *Nato Science Series Sub Series III Computer and Systems Sciences*, *190*, 131–154.

Rudelson, M., & Vershynin, R. (2007). Sampling from large matrices: An approach through geometric functional analysis. *J. ACM 54*(4).

Stewart, G., & Sun, J. (1990). *Matrix perturbation theory*. Boston: Academic Press.

Suykens, J., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, *9*(3), 293–300.

Yang, Y., & Pedersen, J. (1997). A comparative study on feature selection in text categorization. In *Proceedings of the International Conference on Machine Learning* (vol. 97, pp. 412–420). San Mateo, CA: Morgan Kaufmann.

Zhang, P., & Peng, J. (2004). SVM vs regularized least squares classification. In *Proceedings of the 17th International Conference on Pattern Recognition* (vol. 1, pp. 176–179). IEEE Computer Society.

Zhang, T., & Oles, F. (2001). Text categorization based on regularized linear classification methods. *Information Retrieval*, *4*(1), 5–31.

Received October 4, 2015; accepted November 15, 2015.