Deterministic Feature Selection for Regularized Least Squares Classification

Saurabh Paul and Petros Drineas

Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY, USA
pauls2@rpi.edu, drinep@cs.rpi.edu

Abstract. We introduce a deterministic sampling based feature selection technique for regularized least squares classification. 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 perform experiments on synthetic and real-world datasets, namely a subset of TechTC-300 datasets, to support our theory. Experimental results indicate that the proposed method performs better than the existing feature selection methods.

Keywords: Feature Selection, Sampling, Regularized Least Squares Classification.

1 Introduction

Regularized Least Squares Classifier (RLSC) is a simple classifier based on least squares and has a long history in machine learning [17,12,13,10,15,18,1]. RLSC has been known to perform comparably to the popular Support Vector Machines (SVM) [13,10,15,18]. RLSC can be solved by simple vector space operations and do not require quadratic optimization techniques like SVM. The main focus of this paper is on 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 [6] with provable guarantees which work well empirically. But the randomized methods have a failure probability and have to be re-run multiple times to get accurate results. Also, a randomized algorithm may not select the same features in different runs. A deterministic algorithm will select the same features irrespective 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 non-trivial worst-case performance bounds. The number of features selected 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.

T. Calders et al. (Eds.): ECML PKDD 2014, Part II, LNCS 8725, pp. 533–548, 2014.

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. Recently, Dasgupta et al. [6] 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 lesser than that required by the randomized algorithm. The leverage-score based algorithm has a sampling complexity of $O\left(\frac{n}{\epsilon^2}\log\left(\frac{n}{\epsilon^2\sqrt{\delta}}\right)\right)$, whereas single-set spectral sparsification requires $O\left(n/\epsilon^2\right)$ 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. Like in [6], 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.
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 [7] dataset. 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 time because feature selection is an offline task. The experimental results indicate that single-set spectral sparsification out-performs all the methods in terms of out-of-sample error for all 48 TechTC-300 datasets. 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

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 $\|\mathbf{A}\|_2 = \sigma_1$. σ_{max} and σ_{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 $\boldsymbol{\beta} \in \mathbb{R}^{n-d}$, i.e. $\mathbf{q}$ has one component along $\mathbf{A}$ and another component orthogonal to $\mathbf{A}$.

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, $\mathbf{S} \in \mathbb{R}^{r \times d}$ be a matrix such that $\mathbf{SA} \in \mathbb{R}^{r \times n}$ contains r rows of $\mathbf{A}$. 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 re-scaling matrices respectively. We will replace the sampling and re-scaling 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.

RLSC Basics. Consider a training data of n points in d dimensions with respective labels $y_i \in \{-1, +1\}$ for $i = 1, .., n$. The solution of binary classification problems via Tikhonov regularization in a Reproducing Kernel Hilbert Space (RKHS) using the squared loss function results in Regularized Least Squares Classification (RLSC) problem [13], which can be stated as:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{Kx} - \mathbf{y}\|_2^2 + \lambda \mathbf{x}^T \mathbf{Kx} \tag{1}$$

where $\mathbf{K}$ is the $n \times n$ kernel matrix defined over the training dataset, 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 Eqn. 1 in the full-dimensional space is

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

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:

$$f(\mathbf{q}) = \mathbf{x}_{opt}^T \mathbf{X}^T \mathbf{q}. \tag{3}$$

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 re-scales 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 dataset into r dimensions is given by $\tilde{\mathbf{X}} = \mathbf{RX}$ and the RLSC problem becomes

$$\min_{\mathbf{x} \in \mathbb{R}^n} \left\|\tilde{\mathbf{K}}\mathbf{x} - \mathbf{y}\right\|_2^2 + \lambda \mathbf{x}^T \tilde{\mathbf{K}} \mathbf{x}, \tag{4}$$

thus 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,

$$\tilde{f} = f(\tilde{\mathbf{q}}) = \tilde{\mathbf{x}}_{opt}^T \tilde{\mathbf{X}}^T \tilde{\mathbf{q}}. \tag{5}$$

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, i.e. $\|\boldsymbol{\beta}\|_2$ measures how much of $\mathbf{q}$ lies outside the subspace spanned by the training set.

Related Work. The work most closely related to ours is that of Dasgupta et al. [6] 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 SVD of the training set i.e. $O\left(n^2 d\right)$, whereas, for single-set spectral sparsification, it is $O\left(rdn^2\right)$. 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 et al. [2] 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.

4 Our Main Tool: Single-Set Spectral Sparsification

We describe the Single-Set Spectral Sparsification algorithm (**BSS**[1] for short) of [3] 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 $\mathbf{U}^T = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_d]$, 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 $L_\tau = \tau - \sqrt{r\ell}, \delta_L = 1, U_\tau = \delta_U \left(\tau + \sqrt{\ell r}\right)$ and $\delta_U = \left(1 + \sqrt{\ell/r}\right) / \left(1 - \sqrt{\ell/r}\right)$. For $U, L \in \mathbb{R}$ and $\mathbf{A} \in \mathbb{R}^{\ell \times \ell}$ a symmetric positive definite matrix with eigenvalues $\lambda_1, \lambda_2, ..., \lambda_\ell$, define

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

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 U and L respectively. We define $\mathcal{L}\left(\mathbf{u}, \delta_L, \mathbf{A}, L\right)$ and $\mathcal{U}\left(\mathbf{u}, \delta_U, \mathbf{A}, U\right)$ 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, ${t_\tau}^{-1} \leq \mathcal{L}\left(\mathbf{u}_{i_\tau}, \delta_L, \mathbf{A}, L\right)$ and ${t_\tau}^{-1} \geq \mathcal{U}\left(\mathbf{u}_{i_\tau}, \delta_U, \mathbf{A}, U\right)$. Thus, there will be at

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

Input: $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ...\mathbf{u}_d]^T \in \mathbb{R}^{d\times\ell}$ with $\mathbf{u}_i \in \mathbb{R}^\ell$ and $r > \ell$.
Output: Matrices $\mathbf{S} \in \mathbb{R}^{r\times d}, \mathbf{D} \in \mathbb{R}^{r\times r}$.

1. Initialize $\mathbf{A}_0 = \mathbf{0}_{\ell\times\ell}$, $\mathbf{S} = \mathbf{0}_{r\times d}, \mathbf{D} = \mathbf{0}_{r\times r}$.
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 $i_\tau \in \{1, 2, ..d\}$ and number $t_\tau > 0$ (See Section 4 for definitions of $\mathcal{U}, \mathcal{L}$)

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

   - Let $t_\tau^{-1} = \frac{1}{2}\left(\mathcal{U}\left(\mathbf{u}_{i_\tau}, \delta_U, \mathbf{A}_\tau, U_\tau\right) + \mathcal{L}\left(\mathbf{u}_{i_\tau}, \delta_L, \mathbf{A}_\tau, L_\tau\right)\right)$
   - Update $\mathbf{A}_{\tau+1} = \mathbf{A}_\tau + t_\tau \mathbf{u}_{i_\tau}\mathbf{u}_{i_\tau}^T$; set $\mathbf{S}_{\tau+1,i_\tau} = 1$ 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 $\mathbf{D}$.

Algorithm 1. Single-set Spectral Sparsification

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 $i = 1, .., d$, 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 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.

Lemma 1. ***BSS*** *[3]: Given* $\mathbf{U} \in \mathbb{R}^{d\times\ell}$ *satisfying* $\mathbf{U}^T\mathbf{U} = \mathbf{I}_\ell$ *and* $r > \ell$, *we can deterministically construct sampling and rescaling matrices* $\mathbf{S} \in \mathbb{R}^{r\times d}$ *and* $\mathbf{D} \in \mathbb{R}^{r\times r}$ *with* $\mathbf{R} = \mathbf{DS}$, *such that, for all* $\mathbf{y} \in \mathbb{R}^\ell$:

$$\left(1 - \sqrt{\ell/r}\right)^2 \|\mathbf{Uy}\|_2^2 \leq \|\mathbf{RUy}\|_2^2 \leq \left(1 + \sqrt{\ell/r}\right)^2 \|\mathbf{Uy}\|_2^2.$$

We now present a slightly modified version of Lemma 1 for our theorems.

Lemma 2. *Given* $\mathbf{U} \in \mathbb{R}^{d\times\ell}$ *satisfying* $\mathbf{U}^T\mathbf{U} = \mathbf{I}_\ell$ *and* $r > \ell$, *we can deterministically construct sampling and rescaling matrices* $\mathbf{S} \in \mathbb{R}^{r\times d}$ *and* $\mathbf{D} \in \mathbb{R}^{r\times r}$ *such that for* $\mathbf{R} = \mathbf{DS}$, $\left\|\mathbf{U}^T\mathbf{U} - \mathbf{U}^T\mathbf{R}^T\mathbf{RU}\right\|_2 \leq 3\sqrt{\ell/r}$

Proof. From Lemma 1, it follows, $\sigma_\ell\left(\mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U}\right) \geq \left(1-\sqrt{\ell/r}\right)^2$, $\sigma_1\left(\mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U}\right) \leq \left(1+\sqrt{\ell/r}\right)^2$. Thus, $\lambda_{max}\left(\mathbf{U}^T\mathbf{U}-\mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U}\right) \leq \left(1-\left(1-\sqrt{\ell/r}\right)^2\right) \leq 2\sqrt{\ell/r}$. Similarly, $\lambda_{min}\left(\mathbf{U}^T\mathbf{U}-\mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U}\right) \geq \left(1-\left(1+\sqrt{\ell/r}\right)^2\right) \geq 3\sqrt{\ell/r}$. 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. In the next section, we assume $\epsilon \in (0, 1/2]$.

5 Our Main Theorems

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

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

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

$$\boldsymbol{\Delta} = \boldsymbol{\Sigma}\mathbf{U}^T\mathbf{R}^T\mathbf{R}\mathbf{U}\boldsymbol{\Sigma} = \boldsymbol{\Sigma}\left(\mathbf{I} + \mathbf{E}\right)\boldsymbol{\Sigma}. \tag{6}$$

The optimal solution in the sampled space is given by,

$$\tilde{\mathbf{x}}_{opt} = \mathbf{V}\left(\boldsymbol{\Delta} + \lambda\mathbf{I}\right)^{-1}\mathbf{V}^T\mathbf{y} \tag{7}$$

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

$$\begin{aligned}\mathbf{q}^T\mathbf{X}\mathbf{x}_{opt} &= \boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{X}\mathbf{x}_{opt} + \boldsymbol{\beta}\mathbf{U}^{\perp T}\left(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\right)\mathbf{x}_{opt} \\ &= \boldsymbol{\alpha}^T\mathbf{V}\boldsymbol{\Sigma}^2\left(\boldsymbol{\Sigma}^2 + \lambda\mathbf{I}\right)^{-1}\mathbf{V}^T\mathbf{y} \qquad (8) \\ &= \boldsymbol{\alpha}^T\mathbf{V}\left(\mathbf{I} + \lambda\boldsymbol{\Sigma}^{-2}\right)^{-1}\mathbf{V}^T\mathbf{y}. \qquad (9)\end{aligned}$$

Eqn(8) follows because of the fact $\mathbf{U}^{\perp T}\mathbf{U} = \mathbf{0}$ and by substituting $\mathbf{x}_{opt}$ from Eqn.(2). Eqn.(9) follows from the fact that the matrices $\boldsymbol{\Sigma}^2$ and $\boldsymbol{\Sigma}^2 + \lambda\mathbf{I}$ are invertible. Now,

$$\begin{aligned}\left|\mathbf{q}^T\mathbf{X}\mathbf{x}_{opt} - \tilde{\mathbf{q}}^T\tilde{\mathbf{X}}\tilde{\mathbf{x}}_{opt}\right| &= \left|\mathbf{q}^T\mathbf{X}\mathbf{x}_{opt} - \mathbf{q}^T\mathbf{R}^T\mathbf{R}\mathbf{X}\tilde{\mathbf{x}}_{opt}\right| \\ &\leq \left|\mathbf{q}^T\mathbf{X}\mathbf{x}_{opt} - \boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{R}^T\mathbf{R}\mathbf{X}\tilde{\mathbf{x}}_{opt}\right| \quad (10) \\ &\quad + \left|\boldsymbol{\beta}^T\mathbf{U}^{\perp T}\mathbf{R}^T\mathbf{R}\mathbf{X}\tilde{\mathbf{x}}_{opt}\right| \quad (11)\end{aligned}$$

We bound (10) and (11) separately. Substituting the values of $\tilde{\mathbf{x}}_{opt}$ and $\boldsymbol{\Delta}$,

$$\begin{aligned}\boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{R}^T\mathbf{R}\mathbf{X}\tilde{\mathbf{x}}_{opt} &= \boldsymbol{\alpha}^T\mathbf{V}\boldsymbol{\Delta}\mathbf{V}^T\tilde{\mathbf{x}}_{opt} \\ &= \boldsymbol{\alpha}^T\mathbf{V}\boldsymbol{\Delta}\left(\boldsymbol{\Delta}+\lambda\mathbf{I}\right)^{-1}\mathbf{V}^T\mathbf{y} \\ &= \boldsymbol{\alpha}^T\mathbf{V}\left(\mathbf{I}+\lambda\boldsymbol{\Delta}^{-1}\right)^{-1}\mathbf{V}^T\mathbf{y} \\ &= \boldsymbol{\alpha}^T\mathbf{V}\left(\mathbf{I}+\lambda\boldsymbol{\Sigma}^{-1}\left(\mathbf{I}+\mathbf{E}\right)^{-1}\boldsymbol{\Sigma}^{-1}\right)^{-1}\mathbf{V}^T\mathbf{y} \\ &= \boldsymbol{\alpha}^T\mathbf{V}\left(\mathbf{I}+\lambda\boldsymbol{\Sigma}^{-2}+\lambda\boldsymbol{\Sigma}^{-1}\boldsymbol{\Phi}\boldsymbol{\Sigma}^{-1}\right)^{-1}\mathbf{V}^T\mathbf{y} \quad (12)\end{aligned}$$

The last line follows from Lemma 3 in Appendix, which states that $(\mathbf{I}+\mathbf{E})^{-1} = \mathbf{I}+\boldsymbol{\Phi}$, where $\boldsymbol{\Phi} = \sum_{i=1}^{\infty}(-\mathbf{E})^i$. The spectral norm of $\boldsymbol{\Phi}$ is bounded by,

$$\|\boldsymbol{\Phi}\|_2 = \left\|\sum_{i=1}^{\infty}(-\mathbf{E})^i\right\|_2 \leq \sum_{i=1}^{\infty}\|\mathbf{E}\|_2^i \leq \sum_{i=1}^{\infty}\epsilon^i = \epsilon/(1-\epsilon). \quad (13)$$

We now bound (10). Substituting (9) and (12) in (10),

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

The last line follows because of Lemma 4 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 only need to bound the spectral norm of $\left(\boldsymbol{\Sigma}^2+\lambda\mathbf{I}+\lambda\boldsymbol{\Phi}\right)^{-1}$ to bound the spectral norm of $\boldsymbol{\Psi}$. The spectral norm of the matrix $\left(\boldsymbol{\Sigma}^2+\lambda\mathbf{I}+\lambda\boldsymbol{\Phi}\right)^{-1}$ is the inverse of the smallest singular value of $\left(\boldsymbol{\Sigma}^2+\lambda\mathbf{I}+\lambda\boldsymbol{\Phi}\right)$. From perturbation theory of matrices [14] and (13), we get

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

Here, $\sigma_i(\mathbf{Q})$ represents the i^{th} 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 $\mathbf{X}$.

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

Thus, $\left\|\left(\mathbf{\Sigma}^2 + \lambda\mathbf{I} + \lambda\mathbf{\Phi}\right)^{-1}\right\|_2 = 1/\sigma_{min}\left(\mathbf{\Sigma}^2 + \lambda\mathbf{I} + \lambda\mathbf{\Phi}\right) \leq 1/\left(\sigma^2{}_{min} + (1-\epsilon)\lambda\right)$
Here, σ_{max} and σ_{min} denote the largest and smallest singular value of $\mathbf{X}$. Since $\|\mathbf{\Sigma}\|_2 \left\|\mathbf{\Sigma}^{-1}\right\|_2 = \sigma_{max}/\sigma_{min} \leq \kappa_{\mathbf{X}}$, (condition number of $\mathbf{X}$) we bound (10):

$$\left|\mathbf{q}^T\mathbf{X}\mathbf{x}_{opt} - \boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{R}^T\mathbf{R}\mathbf{X}\tilde{\mathbf{x}}_{opt}\right| \leq \frac{\epsilon\lambda\kappa_{\mathbf{X}}}{\sigma^2{}_{min} + (1-\epsilon)\lambda}\left\|\boldsymbol{\alpha}^T\mathbf{V}\left(\mathbf{I} + \lambda\mathbf{\Sigma}^{-2}\right)^{-1}\right\|_2 \left\|\mathbf{V}^T\mathbf{y}\right\|_2 \tag{14}$$

For $\lambda > 0$, the term $\sigma^2{}_{min} + (1-\epsilon)\lambda$ in Eqn.(14) is always larger than $(1-\epsilon)\,\lambda$, so it can be upper bounded by $2\epsilon\kappa_{\mathbf{X}}$ (assuming $\epsilon \leq 1/2$). Also,

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

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

$$\left|\mathbf{q}^T\mathbf{X}\mathbf{x}_{opt} - \boldsymbol{\alpha}^T\mathbf{X}^T\mathbf{R}^T\mathbf{R}\mathbf{X}\tilde{\mathbf{x}}_{opt}\right| \leq 2\epsilon\kappa_{\mathbf{X}}\|\boldsymbol{\alpha}\|_2\|\mathbf{y}\|_2. \tag{15}$$

We now bound (11). Expanding (11) using SVD and $\tilde{\mathbf{x}}_{opt}$,

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

The first inequality follows from $\boldsymbol{\beta} = \mathbf{U}^{\perp T}\mathbf{q}$; and the second inequality follows from Lemma 6 given in appendix. To conclude the proof, we bound the spectral norm of $\mathbf{\Sigma}\left(\mathbf{\Delta} + \lambda\mathbf{I}\right)^{-1}$. Note that from Eqn.(6), $\mathbf{\Sigma}^{-1}\mathbf{\Delta}\mathbf{\Sigma}^{-1} = \mathbf{I} + \mathbf{E}$ and $\mathbf{\Sigma}\mathbf{\Sigma}^{-1} = \mathbf{I}$,

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

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

$$\begin{aligned}
\left\|\left(\mathbf{I} + \lambda\mathbf{\Sigma}^{-2} + \mathbf{E}\right)^{-1}\mathbf{\Sigma}^{-1}\right\|_2 &\leq \frac{\sigma^2{}_{max}}{\left((1-\epsilon)\,\sigma^2{}_{max} + \lambda\right)\sigma_{min}} \\
&= \frac{\kappa_{\mathbf{X}}\sigma_{max}}{(1-\epsilon)\,\sigma^2{}_{max} + \lambda} \\
&\leq \frac{2\kappa_{\mathbf{X}}}{\sigma_{max}}
\end{aligned} \tag{16}$$

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

$$\left|\boldsymbol{\beta}^T \mathbf{U}^{\perp T} \mathbf{R}^T \mathbf{R} \mathbf{X} \tilde{\mathbf{x}}_{opt}\right| \leq \frac{2\epsilon\kappa_{\mathbf{X}}}{\sigma_{max}} \left\|\boldsymbol{\beta}\right\|_2 \left\|\mathbf{y}\right\|_2 . \tag{17}$$

Combining Eqns (15) and (17) we complete the proof for the case $\lambda > 0$. For $\lambda = 0$, Eqn.(14) 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, i.e. $\boldsymbol{\beta} = \mathbf{0}$.

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 $\mathbf{q} \in \mathbb{R}^d$ be the test point of the form $\mathbf{q} = \mathbf{X}\boldsymbol{\alpha}$, i.e. it lies entirely in the subspace spanned by the training set, and the two vectors $\mathbf{V}^T\mathbf{y}$ and $\left(\mathbf{I} + \lambda\boldsymbol{\Sigma}^{-2}\right)^{-1}\mathbf{V}^T\boldsymbol{\alpha}$ satisfy the property,*

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

for some constant ω. If we run RLSC after BSS, then $\left|\tilde{\mathbf{q}}^T\tilde{\mathbf{X}}\tilde{x}_{opt} - \mathbf{q}^T\mathbf{X}\mathbf{x}_{opt}\right| \leq 2\epsilon\omega\kappa_{\mathbf{X}}\left|\mathbf{q}^T\mathbf{X}\mathbf{x}_{opt}\right|$

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

6 Experiments

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

6.1 BSS Implementation Issues

The authors of [3] do not provide any implementation details of the **BSS** algorithm. Here we discuss several issues arising during the implementation.

Choice of Column Selection: At every iteration, there are multiple columns which satisfy the condition $\mathcal{U}\left(\mathbf{u}_i, \delta_U, \mathbf{A}_\tau, U_\tau\right) \leq \mathcal{L}\left(\mathbf{u}_i, \delta_L, \mathbf{A}_\tau, L_\tau\right)$. The authors of [3] suggest picking any column which 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.

Ill-conditioning: The second issue related to the implementation is ill conditioning. It is possible for $\mathbf{A}_\tau$ to be almost singular. At every iteration τ, we check the condition number of $\mathbf{A}_\tau$. If it is high, then we regularize $\mathbf{A}_\tau$ as follows : $\mathbf{A}_\tau = \mathbf{A}_\tau + \gamma\mathbf{I}$. We set $\gamma = 0.01$ in our experiments. Smaller values of γ resulted in large eigenvalues of ${\mathbf{A}_\tau}^{-1}$, which in turn, resulted in large values of t_τ causing bad-scaling of the columns of the input matrix.

6.2 Other Feature Selection Methods

In this section, we describe other feature-selection methods with which we compare BSS.

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

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.

Leverage Score Sampling: We describe the leverage-score sampling of [6]. Let $\mathbf{U}$ be the top-k left singular vectors of the training set $\mathbf{X}$. We create a carefully chosen probability distribution of the form $p_i = \frac{\|\mathbf{U}_i\|_2^2}{n}$. for $i = 1, 2, ..., d$, i.e. proportional to the squared Euclidean norms of the rows of the left-singular vectors and select r rows of $\mathbf{U}$ in i.i.d trials and re-scale the rows with $1/\sqrt{p_i}$. We repeat the sampling process five times to get around the randomness. In our experiments, k was set to the rank of $\mathbf{X}$.

Information Gain (IG): The Information Gain feature selection method [16] measures the amount of information obtained for binary class prediction by knowing the presence or absence of a feature in a dataset. The method is a supervised strategy, whereas the other methods used here are unsupervised.

Table 1. Most frequently selected features using the synthetic dataset

$r = 80$	$k = 90$	$k = 100$
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	88, 87, 86, 85, 84	100, 99, 98, 97, 95
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

6.3 Synthetic Data

We run our experiments on synthetic data where we control the number of relevant features in the dataset and demonstrate the working of Algorithm 1 on RLSC. We generate synthetic data in the same manner as given in [4]. The dataset 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 $y_i\mathcal{N}(-j, 1)$ 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 dataset has k relevant features and $(d-k)$ noisy features. By construction, among the first k features, the kth feature has the most discriminatory power, 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.

Table 2. Out-of-sample error of TechTC-300 datasets averaged over ten ten-fold cross-validation and over 48 datasets for three values of r. The first and second entry of each cell represents the mean and standard deviation. Items in bold indicate the best results.

$r = 300$	$\lambda = 0.1$	$\lambda = 0.3$	$\lambda = 0.5$	$\lambda = 0.7$
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

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 ten 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 dataset 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 supervised 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 ten different synthetic datasets and each time, the five most frequently selected features were from the set of relevant features. Thus, by selecting only 8%-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.

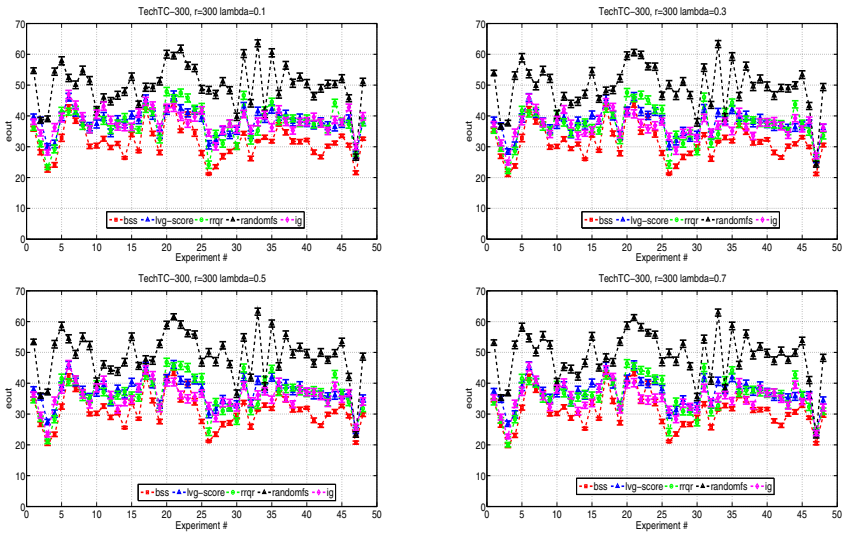


Fig. 1. Out-of-sample error of 48 TechTC-300 documents averaged over ten ten-fold cross validation experiments for different values of regularization parameter λ and number of features $r = 300$. Vertical bars represent standard deviation.

6.4 TechTC-300

We use the TechTC-300 data [7], consisting of a family of 295 document-term data matrices. The TechTC-300 dataset comes from the Open Directory Project (ODP), which is a large, comprehensive directory of the web, maintained by volunteer editors. Each matrix in the TechTC-300 dataset 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 [7]. Each data matrix consists of 150-280 documents, and each document is described with respect to 10,000-50,000 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 datasets. Next we grouped the datasets based on the categories and selected those datasets whose categories appeared at least thrice. There were 147 datasets, and we performed ten-fold

Table 3. A subset of the TechTC matrices of our study

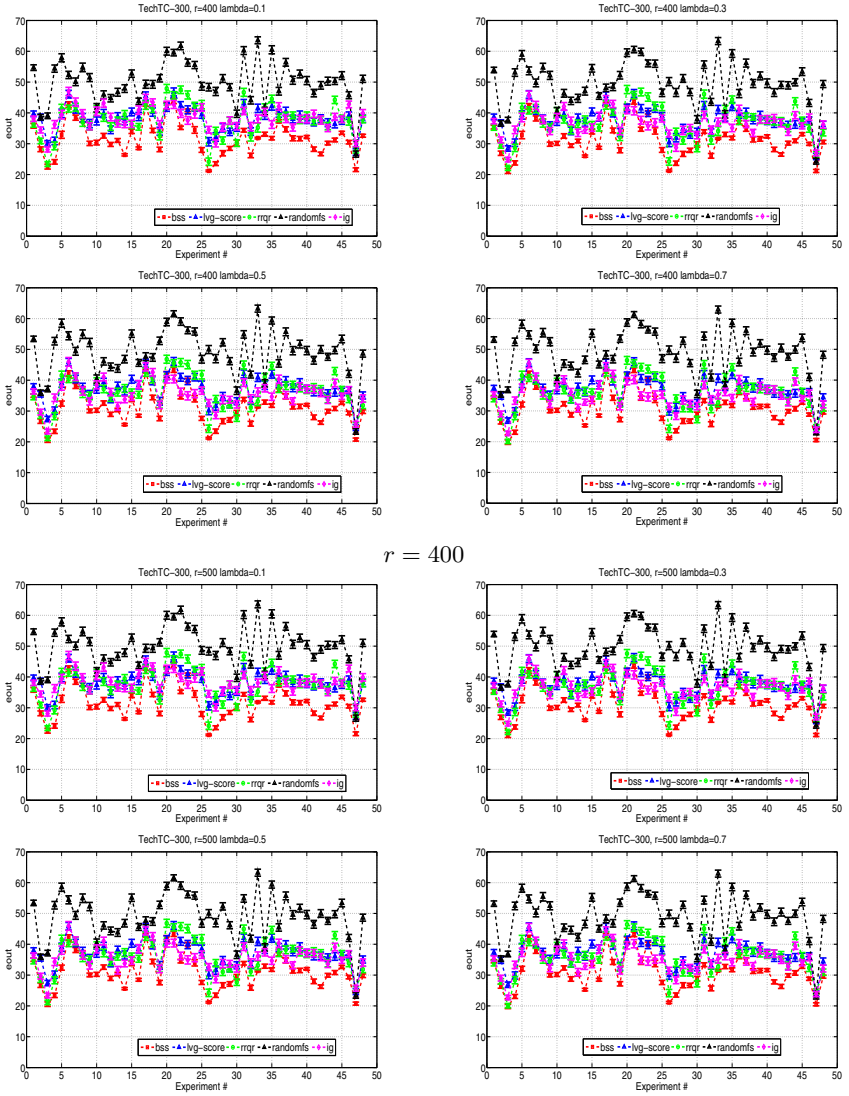
id1_id2	id1	id2
1092_789236	Arts:Music:Styles:Opera	US Navy:Decommisioned Submarines
17899_278949	US:Michigan:Travel & Tourism	Recreation:Sailing Clubs:UK
17899_48446	US:Michigan:Travel & Tourism	Chemistry:Analytical:Products
14630_814096	US:Colorado:Localities:Boulder	Europe:Ireland:Dublin:Localities
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

cross validation and repeated it ten times on 48 such datasets. We set the values of the regularization parameter of RLSC to $0.1, 0.3, 0.5$ and 0.7. We do not report running times because feature selection is an offline task. We set r to 300, 400 and 500. We report the out-of-sample error for all 48 datasets. BSS consistently outperforms Leverage-Score sampling, IG, RRQR and random feature selection on all 48 datasets for all values of the regularization parameter. Table 2 and Fig 1 shows the results. The out-of-sample error decreases with increase in 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%-7% better than the other methods when averaged over 48 document matrices. From Fig 1 and 2, it is evident that BSS is comparable to the other methods and often better on all 48 datasets. Leverage-score sampling requires greater number of samples to achieve the same out-of-sample error as BSS (See Table 2, $r = 500$ 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.

Due to space constraints, we list the most frequently occurring words selected by BSS for the $r = 300$ case for seven TechTC-300 datasets over 100 training sets used in the cross-validation experiments. Table 3 shows the names of the seven TechTC-300 document-term matrices. The words shown in Table 4

Table 4. Frequently occurring terms of the TechTC-300 datasets of Table 3 selected by BSS

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



$r = 400$

$r = 500$

Fig. 2. Out-of-sample error of 48 TechTC-300 documents averaged over ten 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.

were selected in all cross-validation experiments for these seven datasets. The words are closely related to the categories to which the documents belong, which shows that BSS selects 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. 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 closely related to the class values. Thus, we see that using only 2%-4% of all features we are able to select relevant features and obtain good out-of-sample error.

7 Conclusion

We present a provably accurate feature selection method for RLSC which 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 recently used as a feature selection technique for k-means clustering [5], linear SVMs [11] and our work on RLSC helps to expand research in this direction. An interesting future work in this direction would be to include feature selection for non-linear kernels with provable guarantees.

Acknowledgements. We thank the reviewers for their insightful comments. SP and PD are supported by NSF CCF 1016501 and NSF IIS 1319280.

References

1. Agarwal, D.: 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 (2002)
2. Avron, H., Sindhwani, V., Woodruff, D.: Sketching structured matrices for faster nonlinear regression. In: Advances in Neural Information Processing Systems, pp. 2994–3002 (2013)
3. Batson, J., Spielman, D., Srivastava, N.: Twice-ramanujan sparsifiers. In: Proceedings of the 41st Annual ACM STOC, pp. 255–262 (2009)
4. Bhattacharyya, C.: Second order cone programming formulations for feature selection. JMLR 5, 1417–1433 (2004)
5. Boutsidis, C., Magdon-Ismail, M.: Deterministic feature selection for k-means clustering. IEEE Transactions on Information Theory 59(9), 6099–6110 (2013)
6. Dasgupta, A., Drineas, P., Harb, B., Josifovski, V., Mahoney, M.: Feature selection methods for text classification. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 230–239 (2007)
7. Davidov, D., Gabrilovich, E., Markovitch, S.: 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 (2004), `http://techtc.cs.technion.ac.il/techtc300/techtc300.html`

8. Demmel, J., Veselic, K.: Jacobi's method is more accurate than qr. SIAM Journal on Matrix Analysis and Applications 13(4), 1204–1245 (1992)
9. Drineas, P., Mahoney, M., Muthukrishnan, S.: Sampling algorithms for l2 regression and applications. In: Proceedings of the 17th Annual ACM-SIAM SODA, pp. 1127–1136 (2006)
10. Fung, G., Mangasarian, O.: Proximal support vector machine classifiers. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 77–86 (2001)
11. Paul, S., Magdon-Ismail, M., Drineas, P.: Deterministic feature selection for linear svm with provable guarantees (2014), http://arxiv.org/abs/1406.0167
12. Poggio, T., Smale, S.: The mathematics of learning: Dealing with data. Notices of the AMS 50(5), 537–544 (2003)
13. Rifkin, R., Yeo, G., Poggio, T.: Regularized least-squares classification. Nato Science Series Sub Series III Computer and Systems Sciences 190, 131–154 (2003)
14. Stewart, G., Sun, J.: Matrix perturbation theory (1990)
15. Suykens, J., Vandewalle, J.: Least squares support vector machine classifiers. Neural Processing Letters 9(3), 293–300 (1999)
16. Yang, Y., Pedersen, J.: A comparative study on feature selection in text categorization. In: ICML, vol. 97, pp. 412–420 (1997)
17. Zhang, P., Peng, J.: SVM vs regularized least squares classification. In: Proceedings of the 17th International Conference on Pattern Recognition, vol. 1, pp. 176–179 (2004)
18. Zhang, T., Oles, F.: Text categorization based on regularized linear classification methods. Information Retrieval 4(1), 5–31 (2001)

8 Appendix

Lemma 3. *For any matrix* $\mathbf{E}$*, such that* $\mathbf{I}+\mathbf{E}$ *is invertible,* $(\mathbf{I}+\mathbf{E})^{-1} = \mathbf{I} + \sum_{i=1}^{\infty}(-\mathbf{E})^i$.

Lemma 4. *Let* $\mathbf{A}$ *and* $\tilde{\mathbf{A}} = \mathbf{A}+\mathbf{E}$ *be invertible matrices. Then* $\tilde{\mathbf{A}}^{-1} - \mathbf{A}^{-1} = -\mathbf{A}^{-1}\mathbf{E}\tilde{\mathbf{A}}^{-1}$.

Lemma 5. *Let* $\mathbf{D}$ *and* $\mathbf{X}$ *be matrices such that the product* $\mathbf{DXD}$ *is a symmetric positive definite matrix with matrix* $\mathbf{X}_{ii} = 1$*. Let the product* $\mathbf{DED}$ *be a perturbation such that,* $\|E\|_2 = \eta < \lambda_{min}(\mathbf{X})$*. Here* λ_{min} *corresponds to the smallest eigenvalue of* $\mathbf{X}$*. Let* λ_i *be the i-th eigenvalue of* $\mathbf{DXD}$ *and let* $\tilde{\lambda}_i$ *be the i-th eigenvalue of* $\mathbf{D}(\mathbf{X}+\mathbf{E})\mathbf{D}$*. Then,* $\left|\frac{\lambda_i - \tilde{\lambda}_i}{\lambda_i}\right| \leq \frac{\eta}{\lambda_{min}(\mathbf{X})}$.

The lemmas presented above are from matrix perturbation theory [14,8] and are used in the proof of our main theorem.

Lemma 6. *Let* $\epsilon \in (0, 1/2]$*. Then* $\left\|\mathbf{q}^T\mathbf{U}^{\perp}\mathbf{U}^{\perp T}\mathbf{R}^T\mathbf{R}\mathbf{U}\right\|_2 \leq \epsilon \left\|\mathbf{U}^{\perp}\mathbf{U}^{\perp T}\mathbf{q}\right\|_2$

The proof of this lemma is similar to Lemma 4.3 of [9].