Deterministic Feature Selection for Linear SVM with Provable Guarantees

Saurabh Paul ∗ Malik Magdon-Ismail † Petros Drineas ‡

June 3, 2014

Abstract

We introduce single-set spectral sparsification as a provably accurate deterministic sampling based feature-selection technique for linear SVM which can be used in both unsupervised and supervised settings. We develop a new supervised technique of feature selection from the support vectors based on the sampling method and prove theoretically that the margin in the feature space is preserved to within $\epsilon$-relative error by selecting features proportional to the number of support vectors. We prove that, in the case where the sampling method is used in an unsupervised manner, we preserve both the margin and radius of minimum enclosing ball in the feature space to within $\epsilon$-relative error, thus ensuring comparable generalization as in the original space. By using the sampling method in an unsupervised manner for linear SVM, we solve an open problem posed in [1]. We present extensive experiments on medium and large-scale real-world datasets to support our theory and to demonstrate that our method is competitive and often better than prior state-of-the-art, which did not come with provable guarantees.

1 Introduction

Support Vector Machines (SVM) [2] are one of the most popular classifiers used in machine learning. The main focus of this study is on feature selection for SVM with provable guarantees. There exist numerous feature selection techniques for SVM which work well empirically. One can perform feature selection in an unsupervised manner which selects features oblivious to the class or labels, or one can take into account the label information and perform supervised feature selection. In this work, we present a new feature selection technique for linear SVM with provable performance guarantees which can be used in both supervised and unsupervised settings.

We are the first to study a deterministic sampling-based feature selection strategy for SVM with provable performance guarantees on the margin. In the unsupervised setting, we perform deterministic sampling-based feature selection oblivious to the target and preserve both the margin and radius of minimum enclosing ball in the feature space upto $\epsilon$-relative error, thus solving an open problem posed in [1]. In the unsupervised setting, the number of features selected is proportional to the rank of the training set. The support vectors are a subset of the data which define the margin and they are dependent on the target values.

∗Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY, USA, pauls2@rpi.edu
†Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY, USA magdon@cs.rpi.edu
‡Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY, USA drinep@cs.rpi.edu
We introduce a novel method of supervised feature selection for SVM by deterministically selecting features from the support vector set and preserving the margin in the feature space up to $\epsilon$-relative error. In the supervised setting, the number of features selected is proportional to the number of support vectors. For the supervised feature selection strategy, we introduce a heuristic, which makes our algorithm scalable to large-scale datasets.

We first discuss SVM basics. For SVM classification, the training data set consists of $n$ points $x_i \in \mathbb{R}^d$, with respective labels $y_i \in \{-1, +1\}$ for $i = 1 \ldots n$. For linearly separable data, the primal form of the SVM learning problem is to construct a hyperplane $w^*$ which maximizes the geometric margin (the minimum distance of a data point to the hyperplane), while separating the data. For non-separable data the “soft” 1-norm margin is maximized. The dual lagrangian formulation of the classification problem leads to the following quadratic program:

$$\max_{\{\alpha_i\}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

subject to $\sum_{i=1}^{n} y_i \alpha_i = 0$, $0 \leq \alpha_i \leq C$, $i = 1 \ldots n$. (1)

In the above formulation, the unknown lagrange multipliers $\{\alpha_i\}_{i=1}^{n}$ are constrained to lie inside the “box constraint” $[0, C]^n$, where $C$ is part of the input. In order to measure the out-of-sample performance of the SVM classifier, we can use the VC-dimension of fat-separators. Assuming that the data lie in a ball of radius $B$, and that the hypothesis set consists of hyperplanes of width $\gamma$ (corresponding to the margin), then the VC-dimension of this hypothesis set is $O(B^2/\gamma^2)$ [3]. Now, given the in-sample error, we can obtain a bound for the out-of-sample error, which is monotonic in the VC-dimension [4].

1.1 Our Contributions

We introduce single-set spectral sparsification as the first provably accurate deterministic feature selection technique for linear SVM in both supervised and unsupervised settings with performance guarantees on the margin. We are the first to provide an empirical evaluation of the single-set spectral sparsification algorithm. We provide a simple method of extending unsupervised feature selection into a supervised one by selecting features from the support vectors. We use single-set spectral sparsification as a supervised feature selection method for SVM by selecting features from the support-vectors and the number of features selected is proportional to the number of support vectors. We prove theoretically that by solving the SVM optimization problem in the sampled space using the supervised technique, we preserve the margin up to relative error. In the transformed space, let the resulting margin after solving the optimization problem be $\tilde{\gamma}^*$. Then for suitably chosen values of $r = O(p/\epsilon^2)$, where $p$ is the number of support-vectors, the margin is preserved to relative error: $\tilde{\gamma}^* \geq (1 - \epsilon) \gamma^2$. We provide a heuristic for supervised feature selection using single-set spectral sparsification which makes the algorithm scale-up to large-scale datasets. Lastly, we prove theoretically that if we use single-set spectral sparsification in an unsupervised manner, i.e. select features from the training set, then we get relative-error bounds for out-of-sample error. In the transformed space, let the resulting margin after solving the optimization problem be $\tilde{\gamma}^*$, and assume that the smaller-dimension data have data radius $\tilde{B}$. Then we show that, for suitably chosen values of $r$, both the margin and the data
radius are preserved to relative error: $\tilde{\gamma}^2 \geq (1 - \epsilon) \gamma^{2}; \tilde{B}^2 \leq (1 + \epsilon) B^2$. While the main focus of this paper is theoretical, we compare both supervised and unsupervised versions of feature selection using single-set spectral sparsification with the corresponding supervised and unsupervised forms of Recursive Feature Elimination (RFE)\cite{8}, LPSVM\cite{9}, uniform sampling and rank-revealing QR factorization (RRQR) based method of column selection. Feature selection based on the single-set spectral sparsification technique is competitive and often better than RFE and LPSVM which did not come with provable guarantees, for both supervised and unsupervised versions.

2 Notation and Related Work

Notation: $A, B, \ldots$ denote matrices and $\alpha, \beta, \ldots$ denote column vectors; $e_i$ (for all $i = 1 \ldots n$) is the standard basis, whose dimensionality will be clear from context; and $I_n$ is the $n \times n$ identity matrix. The Singular Value Decomposition (SVD) of a matrix $A \in \mathbb{R}^{n \times d}$ of rank $\rho \leq \min\{n, d\}$ is equal to $A = U \Sigma V^T$, where $U \in \mathbb{R}^{n \times \rho}$ is an orthogonal matrix containing the left singular vectors, $\Sigma \in \mathbb{R}^{\rho \times \rho}$ is a diagonal matrix containing the singular values $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_\rho > 0$, and $V \in \mathbb{R}^{d \times \rho}$ is a matrix containing the right singular vectors. The spectral norm of $A$ is $\|A\|_2 = \sigma_1$.

Matrix Sampling Formalism: We now present the tools of feature selection. Let $A$ be the data matrix consisting of $n$ points and $d$ dimensions, $S \in \mathbb{R}^{d \times r}$ be a matrix such that $AS \in \mathbb{R}^{n \times r}$ contains $r$ columns of $A$. The matrix $S$ is called the sampling matrix as it samples $r$ columns of $A$. Let $D \in \mathbb{R}^{r \times r}$ be the diagonal matrix such that $ASD \in \mathbb{R}^{n \times r}$ rescales the columns of $A$ that are in $AS$. We will replace the sampling and re-scaling matrices by a single matrix $R \in \mathbb{R}^{d \times r}$, where $R = SD$ denotes the matrix specifying which of the $r$ columns of $A$ are to be sampled and how they are to be rescaled. Let $X^{tr} \in \mathbb{R}^{n \times d}$ be the matrix whose rows are the vectors $x_i^T$, $Y^{tr} \in \mathbb{R}^{n \times n}$ be the diagonal matrix with entries $Y^{tr}_{ii} = y_i$, and $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n] \in \mathbb{R}^n$ be the vector of lagrange multipliers to be determined by solving eqn. (2). The SVM optimization problem is

$$\max_\alpha 1^T \alpha - \frac{1}{2} \alpha^T Y^{tr} X^{tr} (X^{tr})^T Y^{tr} \alpha$$

subject to $1^T Y^{tr} \alpha = 0$; and $0 \leq \alpha \leq C$. \hfill (2)

(In the above, $1$, $0$, $C$ are vectors with the implied constant entry.) Let $\hat{\alpha^*}$ be an optimal solution of the above problem. The points $x_i$ for which $\hat{\alpha_i^*} > 0$, i.e., the points which appear in the expansion $w^*$, are the support vectors. The optimal separating hyperplane is given by $w^* = (X^{tr})^T Y^{tr} \hat{\alpha^*} = (X^{sv})^T Y^{sv} \alpha^*$, where $X^{sv}$, $Y^{sv}$ represent the support vector matrix and the corresponding labels respectively and $\alpha^*$ represents the lagrange multipliers corresponding to the support vectors. The geometric margin, $\gamma^*$, of this canonical optimal hyperplane is $\gamma^* = 1/\|w^*\|_2$, where $\|w^*\|_2^2 = \sum_{i=1}^{n} \alpha_i^*$. The data radius is $B = \min_{x^*} \max_{x_i} \|x_i - x^*\|_2$.

Our goal is to study how the SVM performs when sampling based feature selection algorithms are used to select features in both supervised and unsupervised settings. For supervised setting, we select features from the Support Vector matrix. Let $X^{sv}$ be the support vector matrix with $p$ support-vectors in $d$ dimensions and $Y^{sv} \in \mathbb{R}^{p \times p}$ be the class labels corresponding to the support vectors. Let $R \in \mathbb{R}^{d \times r_1}$ be the matrix that samples and re-scales $r_1$ columns of $X^{sv}$, thus reducing the dimensionality from $d$ to $r_1 \ll d$ and $r_1$ is proportional to the number of support vectors. The transformed data is the support vector
matrix with reduced dimension $X^{sv}R$. For the unsupervised setting, let $R \in \mathbb{R}^{d \times r_2}$ be the matrix that samples and re-scales $r_2$ columns of $X^tr$ thus reducing the dimensionality of the training set from $d$ to $r_2 \ll d$ and $r_2$ is proportional to the rank of the input matrix. The transformed dataset into $r_2$ dimensions is given by $\tilde{X}^tr = X^trR$, and the SVM optimization problem for classification becomes

$$\max_\alpha 1^T \hat{\alpha} - \frac{1}{2} \hat{\alpha}^T Y^{tr} X^{tr} R R^T (X^{tr})^T Y^{tr} \hat{\alpha},$$

subject to $1^T Y^{tr} \hat{\alpha} = 0; \quad 0 \leq \hat{\alpha} \leq C.$

**Related Work:** Guyon et al. [5] and Rakotomamonjy[7] proposed SVM based criteria to rank features based on the weight vector. Weston et al. [8] formulated a combinatorial optimization problem to select features by minimising the ratio of square radius of minimum enclosing ball and the margin. Weston et al. [9] used the zero norm to perform error minimization and feature selection in one step. A Newton based method of feature selection was proposed for a linear programming formulation of SVM in [6]. Tan et al. [10] formulated the $\ell_0$-norm Sparse SVM using a mixed integer programming problem. Do et al. [11] proposed R-SVM which performs feature selection and ranking by optimizing the radius-margin bound with a scaling factor. This work was further improved in [12] where they proposed a radius-margin based feature selection algorithm and formulated it using a standard quadratic convex optimization problem with linear or quadratic constraints. Another line of work includes the doubly regularised Support Vector Machine (DrSVM) [13] which uses a mixture of L2-norm and L1-norm penalties to solve the SVM optimization problem and automatically perform variable selection. Subsequent works on DrSVM involve reducing the computational bottleneck of solving this problem [14],[15]. Gilad-Bachrach et al.[16] formulate margin as a function of set of features and introduce an evaluation function which assigns a score to sets of features according to the margin they induce. Our work is different from these, since we present a provably accurate deterministic feature selection technique for linear SVM which can be used in both supervised and unsupervised settings.

### 3 Our main tool: Single-set Spectral Sparsification

We describe the Single-Set Spectral Sparsification algorithm (BSS\footnote{The name BSS comes from the authors Batson, Spielman and Srivastava of [17].} for short) of [17], (See Algorithm 1 in supplementary material), as a greedy technique that selects columns one at a time. The algorithm samples $r$ columns in deterministic time, hence the name deterministic sampling. Consider the input matrix as a set of $d$ column vectors $V^T = [v_1, v_2, ..., v_d]$, with $v_i \in \mathbb{R}^\ell (i = 1, \ldots, d).$ Given $\ell$ and $r > \ell$, we iterate over $\tau = 0, 1, 2, \ldots, r - 1$. Define the parameters $L_\tau = \tau - \sqrt{\ell/\tau}, \delta_L = 1, \delta_U = \left(1 + \sqrt{\ell/\tau}\right) / \left(1 - \sqrt{\ell/\tau}\right)$ and $U_\tau = \delta_U \left(\tau + \sqrt{\ell/\tau}\right).$ For $U, L \in \mathbb{R}$ and $A \in \mathbb{R}^{\ell \times \ell}$ a symmetric positive definite matrix with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_\ell$, define $\Phi(L, A) = \sum_{i=1}^\ell \frac{1}{\lambda_i - L}$ and $\hat{\Phi}(U, A) = \sum_{i=1}^\ell \frac{1}{U - \lambda_i}$ as the lower and upper potentials respectively. These potential functions measure how far the eigenvalues of $A$ are from the upper and lower barriers $U$ and $L$ respectively. We define $\mathcal{L}(v, \delta_L, A, L)$ and $\mathcal{U}(v, \delta_U, A, U)$ as follows:

$$\mathcal{L}(v, \delta_L, A, L) = \frac{v^T (A - (L + \delta_L) I_\ell)^{-1} v}{\hat{\Phi}(L + \delta_L, A) - \hat{\Phi}(L, A)} - \frac{v^T (A - (L + \delta_L) I_\ell)^{-1} v}{\Phi(L + \delta_L, A)} - \frac{v^T (A - (L + \delta_L) I_\ell)^{-1} v}{\Phi(L, A)}.$$
\[
\mathcal{U}(v, \delta_U, A, U) = v^T((U + \delta_U)I_\ell - A)^{-2}v + v^T((U + \delta_U)I_\ell - A)^{-1}v.
\]

At every iteration, there exists an index \(i_\tau\) and a weight \(t_\tau > 0\) such that, \(\tau = L(v_i, \delta_U, A, L)\) and \(t_{\tau-1} \geq \mathcal{U}(v_i, \delta_U, A, U)\). Thus, there will be at most \(r\) columns selected after \(\tau\) iterations. The running time of the algorithm is dominated by the search for an index \(i_\tau\) satisfying \(\mathcal{U}(v_i, \delta_U, A, \tau) \leq \mathcal{L}(v_i, \delta_U, A, L)\) and computing the weight \(t_\tau\). One needs to compute the upper and lower potentials \(\Phi(U, A)\) and \(\Phi(L, A)\) and hence the eigenvalues of \(A\). Cost per iteration is \(O(\ell^3)\) and the total cost is \(O(r\ell^3)\). For \(i = 1, \ldots, d\), we need to compute \(\mathcal{L}\) and \(\mathcal{U}\) for every \(v_i\) which can be done in \(O(d\ell^2)\) for every iteration, for a total of \(O(rd\ell^2)\). Thus total running time of the algorithm is \(O(rd\ell^2)\). We present the following lemma for the single-set spectral sparsification algorithm.

**Lemma 1. BSS [17]:** Given \(V \in \mathbb{R}^{d \times \ell}\) satisfying \(V^TV = I_\ell\) and \(r > \ell\), we can deterministically construct sampling and rescaling matrices \(S \in \mathbb{R}^{d \times r}\) and \(D \in \mathbb{R}^{r \times r}\) with \(R = SD\), such that, for all \(y \in \mathbb{R}^\ell\): \(\left(1 - \sqrt{\ell/r}\right)^2 \|Vy\|^2 \leq \|V^TRy\|^2 \leq \left(1 + \sqrt{\ell/r}\right)^2 \|Vy\|^2\).

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

**Lemma 2.** Given \(V \in \mathbb{R}^{d \times \ell}\) satisfying \(V^TV = I_\ell\) and \(r > \ell\), we can deterministically construct sampling and rescaling matrices \(S \in \mathbb{R}^{d \times r}\) and \(D \in \mathbb{R}^{r \times r}\) such that for \(R = SD\), \(\|V^TV - V^TRR^TV\|^2 \leq 3\sqrt{\ell/r}\).

**Proof.** From Lemma 1 it follows, \(\sigma_\ell(V^TRR^TV) \geq \left(1 - \sqrt{\ell/r}\right)^2\) and \(\sigma_1(V^TRR^TV) \leq \left(1 + \sqrt{\ell/r}\right)^2\). Thus, \(\lambda_{\max}(V^TV - V^TRR^TV) \leq \left(1 - \left(1 - \sqrt{\ell/r}\right)^2\right) \leq 2\sqrt{\ell/r}\). Similarly, \(\lambda_{\min}(V^TV - V^TRR^TV) \geq \left(1 - \left(1 + \sqrt{\ell/r}\right)^2\right) \geq 3\sqrt{\ell/r}\). Combining these two results, we have \(\|V^TV - V^TRR^TV\|^2 \leq 3\sqrt{\ell/r}\).

### 4 Margin is preserved by Supervised BSS

We now state and prove our main result, namely that solving the SVM optimization problem in the supervised setting using BSS results in comparable margin as in the original space.

**Theorem 1.** Let \(r_1 = O(p/\epsilon^2)\), where \(p > 0\) is an accuracy parameter, \(p\) is the number of support vectors and \(r_1\) is the number of features selected. Let \(R \in \mathbb{R}^{d \times r_1}\) be a matrix as defined in Lemma 2 satisfying \(\|V^TV - V^TRR^TV\|^2 \leq \epsilon\), where \(V\) is the matrix of right singular vectors of the support vector matrix \(X^{sv}\). Let \(\gamma^*\) and \(\tilde{\gamma}^*\) be the margins obtained by solving the SVM problems using the support vector matrices \(X^{sv}\) and \(X^{sv}R\), respectively. Then, \(\tilde{\gamma}^* \geq (1 - \epsilon)\gamma^*\).

**Proof.** Let \(X^{tr} \in \mathbb{R}^{n \times d}\), \(Y^{tr} \in \mathbb{R}^{n \times n}\) be the feature matrix and class labels of the training set (as defined in Section 2) and let \(\hat{\alpha}^* = [\hat{\alpha}_1^*, \hat{\alpha}_2^*, \ldots, \hat{\alpha}_n^*]^T \in \mathbb{R}^n\) be the vector achieving the optimal solution for the problem of eqn. (2). Then,

\[
Z_{opt} = \sum_{j=1}^{n} \hat{\alpha}_j^* - \frac{1}{2} \hat{\alpha}^* Y^{tr} X^{tr} (X^{tr})^T Y^{tr} \hat{\alpha}^*.
\]
Let $p \leq n$ be the support vectors with $\alpha_j > 0$. Let $\alpha^* = [\alpha_1^*, \alpha_2^*, \ldots, \alpha_p^*]^T \in \mathbb{R}^p$ be the vector achieving the optimal solution for the problem of eqn. (4). Let $X^{sv} \in \mathbb{R}^{p \times d}$, $Y^{sv} \in \mathbb{R}^{p \times p}$ be the support vector matrix and the corresponding labels respectively. Let $E = V^T V - V^T R R^T V$. Then, we can write eqn (4) in terms of support vectors as,

$$Z_{opt} = \sum_{i=1}^{p} \alpha_i^* - \frac{1}{2} \alpha^* Y^{sv} X^{sv} (X^{sv})^T Y^{sv} \alpha^*$$  \hspace{1cm} (5)

$$= \sum_{i=1}^{p} \alpha_i^* - \frac{1}{2} \alpha^* Y^{sv} U \Sigma V^T V \Sigma U^T Y^{sv} \alpha^*$$

$$= \sum_{i=1}^{p} \alpha_i^* - \frac{1}{2} \alpha^* Y^{sv} U \Sigma V^T R R^T V \Sigma U^T Y^{sv} \alpha^* - \frac{1}{2} \alpha^* Y^{sv} U \Sigma E U^T Y^{sv} \alpha^*$$ \hspace{1cm} (6)

Let $\tilde{\alpha}^* = [\tilde{\alpha}_1^*, \tilde{\alpha}_2^*, \ldots, \tilde{\alpha}_p^*]^T \in \mathbb{R}^p$ be the vector achieving the optimal solution for the dimensionally-reduced SVM problem of eqn. (5) using $\tilde{X}^{sv} = X^{sv} R$. Using the SVD of $X^{sv}$,

$$Z_{opt} = \sum_{i=1}^{p} \tilde{\alpha}_i^* - \frac{1}{2} \tilde{\alpha}^* Y^{sv} U \Sigma V^T R R^T V \Sigma U^T Y^{sv} \tilde{\alpha}^*$$ \hspace{1cm} (7)

Since the constraints on $\alpha^*$, $\tilde{\alpha}^*$ do not depend on the data it is clear that $\tilde{\alpha}^*$ is a feasible solution for the problem of eqn. (5). Thus, from the optimality of $\alpha^*$, and using eqn. (7), it follows that
We can rewrite the above inequality as

\[ Z_{opt} = \sum_{i=1}^{p} \alpha_i^* - \frac{1}{2} \alpha^T Y^{sv} U \Sigma V^T R R^T V \Sigma U^T Y^{sv} \alpha^* - \frac{1}{2} \alpha^T Y^{sv} U \Sigma E \Sigma U^T Y^{sv} \alpha^* \]

\[ \geq \sum_{i=1}^{p} \tilde{\alpha}_i^* - \frac{1}{2} \tilde{\alpha}^T Y^{sv} U \Sigma V^T R R^T V \Sigma U^T Y^{sv} \tilde{\alpha}^* - \frac{1}{2} \tilde{\alpha}^T Y^{sv} U \Sigma E \Sigma U^T Y^{sv} \tilde{\alpha}^* \]

\[ = \tilde{Z}_{opt} - \frac{1}{2} \tilde{\alpha}^T Y^{sv} U \Sigma E \Sigma U^T Y^{sv} \tilde{\alpha}^*. \tag{8} \]

Towards that end, we bound the difference:

\[ \frac{1}{2} \tilde{\alpha}^T Y^{sv} U \Sigma E \Sigma U^T Y^{sv} \tilde{\alpha}^* \leq \frac{1}{2} \| Q \|_2 \| E \|_2 \| Q^T \|_2 = \frac{1}{2} \| E \|_2 \| Q \|_2^2 = \frac{1}{2} \| E \|_2 \| \tilde{\alpha}^T Y^{sv} X^{sv} \|_2^2 \tag{9} \]

Combining eqns. (8) and (9), we get

\[ Z_{opt} \geq \tilde{Z}_{opt} - \frac{1}{2} \| E \|_2 \| \tilde{\alpha}^T Y^{sv} X^{sv} \|_2^2. \tag{10} \]

We now proceed to bound the second term in the right-hand side of the above equation. Towards that end, we bound the difference:

\[ \left| \tilde{\alpha}^T Y^{sv} X^{sv} + \sum_{i=1}^{p} \gamma_i \alpha_i^* - \tilde{\alpha}^T Y^{sv} X^{sv} \right| \leq \| E \|_2 \| \tilde{\alpha}^T Y^{sv} U \Sigma V^T \|_2^2 = \| E \|_2 \| \tilde{\alpha}^T Y^{sv} U \Sigma V^T \|_2 \]

We can rewrite the above inequality as

\[ \| \tilde{\alpha}^T Y^{sv} X^{sv} R \|_2^2 - \| \tilde{\alpha}^T Y^{sv} X^{sv} \|_2^2 \leq \| E \|_2 \| \tilde{\alpha}^T Y^{sv} X^{sv} \|_2 \]

\[ \text{thus, } \| \tilde{\alpha}^T Y^{sv} X^{sv} \|_2 \leq \frac{1}{1 - \| E \|_2^2} \| \tilde{\alpha}^T Y^{sv} X^{sv} R \|_2^2. \]

Combining eqn. (10), we get

\[ Z_{opt} \geq \tilde{Z}_{opt} - \frac{1}{2} \left( \frac{\| E \|_2^2}{1 - \| E \|_2^2} \right) \| \tilde{\alpha}^T Y^{sv} X^{sv} R \|_2^2. \tag{11} \]

Now recall that \( w^T = \alpha^T Y^{sv} X^{sv} \), \( \tilde{w}^T = \tilde{\alpha}^T Y^{sv} X^{sv} R \), \( \| w^* \|_2 = \sum_{i=1}^{p} \alpha_i^* \), and \( \| \tilde{w}^* \|_2 = \sum_{i=1}^{p} \tilde{\alpha}_i^* \). Then, the optimal solutions \( Z_{opt} \) and \( \tilde{Z}_{opt} \) can be expressed as follows:

\[ Z_{opt} = \| w^* \|_2^2 - \frac{1}{2} \| w^* \|_2^2 = \frac{1}{2} \| w^* \|_2^2. \tag{12} \]

\[ \tilde{Z}_{opt} = \| \tilde{w}^* \|_2^2 - \frac{1}{2} \| \tilde{w}^* \|_2^2 = \frac{1}{2} \| \tilde{w}^* \|_2^2. \tag{13} \]

Combining eqns. (11), (12) and (13), we get

\[ \| w^* \|_2^2 \geq \| \tilde{w}^* \|_2^2 - \left( \frac{\| E \|_2^2}{1 - \| E \|_2^2} \right) \| \tilde{w}^* \|_2^2 \]

\[ = \left( 1 - \frac{\| E \|_2^2}{1 - \| E \|_2^2} \right) \| \tilde{w}^* \|_2^2. \tag{14} \]

Let \( \gamma^* = \| w^* \|_2^2 \) be the geometric margin of the problem of eqn. (5) and let \( \tilde{\gamma}^* = \| \tilde{w}^* \|_2^2 \) be the geometric margin of the problem of eqn. (7). Then, the above equation implies:
\[
\gamma'^2 \leq \left(1 - \frac{\|E\|_2}{1 - \|E\|_2}\right)^{-1}\gamma^2 \Rightarrow \tilde{\gamma}'^2 \geq \left(1 - \frac{\|E\|_2}{1 - \|E\|_2}\right)\gamma^2.
\] (15)

5 Geometry is preserved by Unsupervised BSS

In an unsupervised setting, we preserve both the margin and radius of the minimum enclosing ball, thus ensuring comparable generalization bounds, in terms of the ratio of radius of minimum enclosing ball and the margin, in the sampled space.

**Theorem 2.** Let \( r_2 = O(n/\epsilon^2) \), where \( \epsilon > 0 \) is an accuracy parameter, \( n \) is the number of training points and \( r_2 \) is the number of features selected. Let \( R \in \mathbb{R}^{d \times r_2} \) be the matrix, as defined in Lemma 2 satisfying \( \| V^T \Sigma - V^T R R^T V \|_2 \leq \epsilon \), where \( V \) is the right-singular vector matrix of the training set \( \mathbf{X}^{tr} \). Let \( \gamma^* \) and \( \tilde{\gamma}'^2 \) be the margins obtained by solving the SVM problems in full-dimensional space and by using BSS in an unsupervised manner i.e. by using \( \mathbf{X}^{tr} \) and \( \mathbf{X}^{tr} R \) respectively. Then, \( \tilde{\gamma}'^2 \geq (1 - \epsilon) \gamma^* \).

Theorem 2 can be proved in a similar manner as Theorem 1.

**Theorem 3.** Let \( r_2 = O(n/\epsilon^2) \), where \( \epsilon > 0 \) is an accuracy parameter, \( n \) is the number of training points and \( r_2 \) is the number of features selected. Let \( B \) be the radius of the minimum ball enclosing all points in the full-dimensional space, and let \( \tilde{B} \) be the radius of the ball enclosing all points in the sampled subspace obtained by using BSS in an unsupervised manner. For \( R \) as in Lemma 2, \( \tilde{B}^2 \leq (1 + \epsilon) B^2 \).

Proof. (of Theorem 3) We consider the matrix \( \mathbf{X}_B \in \mathbb{R}^{(n+1) \times d} \) whose first \( n \) rows are the rows of \( \mathbf{X}^{tr} \) and whose last row is the vector \( \mathbf{x}_B \); here \( \mathbf{x}_B \) denotes the center of the minimum radius ball enclosing all \( n \) points. Then, the SVD of \( \mathbf{X}_B \) is equal to \( \mathbf{X}_B = U_B \Sigma_B V_B^T \), where \( U_B \in \mathbb{R}^{(n+1) \times \rho_B} \), \( \Sigma_B \in \mathbb{R}^{\rho_B \times \rho_B} \), and \( V \in \mathbb{R}^{\rho \times \rho_B} \). Here \( \rho_B \) is the rank of the matrix \( \mathbf{X}_B \) and clearly \( \rho_B \leq \rho + 1 \). (Recall that \( \rho \) is the rank of the matrix \( \mathbf{X}^{tr} \).) Let \( B \) be the radius of the minimal radius ball enclosing all \( n \) points in the original space. Then, for any \( i = 1, \ldots, n \),

\[
B^2 \geq \| \mathbf{x}_i - \mathbf{x}_B \|_2^2 = \| (\mathbf{e}_i - \mathbf{e}_{n+1})^T \mathbf{X}_B \|_2^2.
\] (16)

Now consider the matrix \( \mathbf{X}_B R \) and notice that

\[
\left\| (\mathbf{e}_i - \mathbf{e}_{n+1})^T \mathbf{X}_B \right\|_2^2 - \left\| (\mathbf{e}_i - \mathbf{e}_{n+1})^T \mathbf{X}_B R \right\|_2^2
\]

\[
= (\mathbf{e}_i - \mathbf{e}_{n+1})^T (\mathbf{X}_B^T \mathbf{X}_B - \mathbf{X}_B R \mathbf{R}^T \mathbf{X}_B) (\mathbf{e}_i - \mathbf{e}_{n+1})
\]

\[
= (\mathbf{e}_i - \mathbf{e}_{n+1})^T \mathbf{U}_B \Sigma_B \mathbf{E}_B \Sigma_B \mathbf{U}_B^T (\mathbf{e}_i - \mathbf{e}_{n+1})
\]

\[
\leq \| \mathbf{E}_B \|_2 \left\| (\mathbf{e}_i - \mathbf{e}_{n+1})^T \mathbf{U}_B \Sigma_B \right\|_2 \| (\mathbf{e}_i - \mathbf{e}_{n+1}) \mathbf{X}_B \|_2^2.
\]

In the above, we let \( \mathbf{E}_B \in \mathbb{R}^{\rho_B \times \rho_B} \) be the matrix that satisfies \( \mathbf{V}_B^T \mathbf{V}_B = \mathbf{V}_B^T R \mathbf{R}^T \mathbf{V}_B + \mathbf{E}_B \), and we also used \( \mathbf{V}_B^T \mathbf{V}_B = \mathbf{I} \). Now consider the ball whose center is the \((n+1)\)-th row of
the matrix \( X_B R \) (essentially, the center of the minimal radius enclosing ball for the original points in the sampled space). Let \( \hat{i} = \arg\max_{i=1 \ldots n} \left\| (e_i - e_{n+1})^T X_B R \right\|_2^2 \); then, using the above bound and eqn. \ref{eq:bound}, we get
\[
\left\| (e_i - e_{n+1})^T X_B R \right\|_2^2 \leq (1 + \|E_B\|_2) \left\| (e_i - e_{n+1})^T X_B \right\|_2^2 \leq (1 + \|E_B\|_2)B^2.
\]
Thus, there exists a ball centered at \( e_{n+1}X_B R \) (the projected center of the minimal radius ball in the original space) with radius at most \( \sqrt{1 + \|E_B\|_2}B \) that encloses all the points in the sampled space. Recall that \( \tilde{B} \) is defined as the radius of the minimal radius ball that encloses all points in sampled subspace; clearly, \( \tilde{B}^2 \leq (1 + \|E_B\|_2)B^2 \). We can now use Lemma \ref{lem:lemma2} on \( V_B \) to conclude that (using \( \rho_B \leq \rho + 1 \)) \( \|E_B\|_2 \leq \epsilon \).

**Theorem 4.** Let \( r_2 = O\left(\frac{n}{\epsilon^2}\right) \), where \( \epsilon > 0 \) is an accuracy parameter, \( n \) is the number of training points and \( r_2 \) is the number of features selected. Let \( R \in \mathbb{R}^{d \times r_2} \) be a matrix as in Lemma \ref{lem:lemma4} satisfying \( \|V^TV - V^TR \sigma \sigma^T V\|_2 \leq \epsilon \), where \( V \) is the right-singular vector matrix of the training set \( X^t \). Let \( \gamma^* \) and \( \tilde{\gamma}^* \) be the margins obtained by solving the SVM problems using data matrices \( X^t \) and \( X^t R \) respectively. Let \( B \) be the radius of the minimum ball enclosing all points in the full-dimensional space (rows of \( X^t \)) and let \( \tilde{B} \) be the radius of the ball enclosing all points in the dimensionally-reduced space (rows of \( X^t R \)). Then,
\[
\frac{\tilde{B}^2}{\tilde{\gamma}^*} \leq \frac{(1 + \epsilon) B^2}{(1 - \epsilon) \gamma^*^2}.
\]

The proof follows directly by combining the proofs of Theorems \ref{thm:thm2} and \ref{thm:thm3}.

### 5.1 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. Let \( A \) be a \( n \times d \) matrix with \( (n < d) \) and an integer \( k \) \((k < d)\) and assume partial QR factorizations of the form
\[
AP = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix},
\]
where \( Q \in \mathbb{R}^{n \times n} \) is an orthogonal matrix, \( P \in \mathbb{R}^{d \times d} \) is a permutation matrix, \( R_{11} \in \mathbb{R}^{k \times k}, R_{12} \in \mathbb{R}^{k \times (d-k)}, R_{22} \in \mathbb{R}^{(d-k) \times (d-k)} \) The above factorization is called a RRQR factorization if \( \sigma_{\min}(R_{11}) \geq \sigma_k(A)/p(k,d), \sigma_{\max}(R_{22}) \leq \sigma_{\min}(A)p(k,d) \), where \( p(k,d) \) is a function bounded by a low-degree polynomial in \( k \) and \( d \). The important columns are given by \( A_1 = Q \begin{pmatrix} R_{11} \\ 0 \end{pmatrix} \) and \( \sigma_i(A_1) = \sigma_i(R_{11}) \) 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.

**Recursive Feature Elimination:** Recursive Feature Elimination (RFE), \cite{5} tries to find the best subset of features which leads to the largest margin of class separation using SVM. At each iteration, the algorithm greedily removes the feature that decreases the margin the least, until the required number of features remain. At each step, it computes the weight
vector and removes the feature with smallest weight. RFE is computationally expensive for high-dimensional datasets. Therefore, at each iteration, multiple features are removed to avoid the computational bottleneck.

**LPSVM:** The feature selection problem for SVM can be formulated in the form of a linear program. LPSVM [6] uses a fast Newton method to solve this problem and obtains a sparse solution of the weight vector, which is used to select the features.

### 6 Experiments

We compared BSS with RFE [5], LPSVM [6], rank-revealing QR factorization (RRQR), random feature selection and full-data without feature selection on synthetic and real-world datasets. For the supervised case, we first run SVM on the training set, then run a feature selection method on the support-vector set and recalibrate the model using the support vector-set. For unsupervised feature selection, we perform feature selection on the training set. For LPSVM, we were not able to control the number of features and report the out-of-sample error using the features output by the algorithm. We did not extrapolate the values of out-of-sample error for LPSVM. We repeated random feature selection five times. We performed ten-fold cross-validation and repeated it ten times. For medium-scale datasets like TechTC-300 we do not perform approximate BSS. For large-scale datasets like Reuters-CCAT [18] we use the approximate BSS method as described in Section 6.4. We used LIBSVM [19] as our SVM solver for medium-scale datasets and LIBLINEAR [20] for large-scale datasets. We do not report running times in our experiments, since feature selection is an offline task. We implemented all our algorithms in MATLAB R2013b on an Intel i-7 processor with 16GB RAM. BSS is better than LPSVM and RRQR and comparable to RFE on 49 TechTC-300 datasets.

### 6.1 BSS Implementation Issues:

The authors of [17] 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 \( U(v_i, \delta_U, A_\tau, U_\tau) \leq L(v_i, \delta_L, A_\tau, L_\tau) \). The authors of [17] suggest picking any column which satisfies this constraint. Selecting a column naively leaves out important features required for classification. It is also possible that the same column may be selected by the algorithm multiple times because it satisfies the constraint. Therefore, we choose the column \( v_i \) which has not been selected in previous iterations and whose Euclidean-norm is highest among the candidate set. A column with low-Euclidean norm will imply that it is sparse, which in turn means that the column is present across a small number of training points and thus may not be relevant. Columns with zero Euclidean norm never get selected by the algorithm. In the inner loop of Algorithm 1 \( U \) and \( L \) 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 \( A_\tau \) to be almost singular. At every iteration \( \tau \), we check the condition number of \( A_\tau \). If it is high, then we regularize \( A_\tau \) as follows : \( A_\tau = A_\tau + \lambda I \). We set \( \lambda = 0.01 \) in our experiments. Smaller values of \( \lambda \) resulted in large eigenvalues of \( A_\tau^{-1} \), which in turn, resulted in large values of \( t_\tau \) causing bad-scaling of the columns of the input matrix.
6.2 Experiments on Supervised Feature Selection

**Synthetic Data:** We generate synthetic data as described in [21], where we control the number of relevant features in the dataset. 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 $x_i$ are the relevant features and are drawn from $y_i N(-j, 1)$ distribution, where $N(\mu, \sigma^2)$ is a random normal distribution with mean $\mu$ and variance $\sigma^2$ and $j$ varies from 1 to $k$. The remaining $(d - k)$ features are chosen from a $N(0, 1)$ distribution and are noisy features. By construction, among the first $k$ features, the $k^{th}$ feature has the most discriminatory power, followed by $(k - 1)^{th}$ feature and so on.

We set $n$ to 200 and $d$ to 1000. We set $k$ to 40 and 50 and ran two sets of experiments. We set the value of $r_1$, i.e. the number of features selected by BSS to 30 and 40 for all experiments. We performed ten-fold cross-validation and repeated it ten times. We used LIBSVM with default settings and set $C = 1$. We compared with the other methods. The mean out-of-sample error was 0 for all methods for both $k = 40$ and $k = 50$. Table 1 shows the set of five most frequently selected features by the different methods for one such synthetic dataset. 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. 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 3% - 4% of all features, we show that we are able to obtain the most discriminatory features along with good out-of-sample error using BSS.

|       | $r_1 = 30$ |           | $r_1 = 40$ |
|-------|------------|-----------|------------|
|       | $k = 40$   | $k = 50$  | $k = 40$   | $k = 50$  |
| BSS   | 40, 39, 34, 36, 37 | 50, 49, 48, 47, 46 | 40, 39, 34, 36, 37 | 50, 49, 48, 47, 46 |
| RFE   | 40, 39, 38, 37, 36 | 50, 49, 48, 47, 46 | 40, 39, 38, 37, 36 | 50, 49, 48, 47, 46 |
| LPSVM | 40, 39, 38, 37, 34 | 50, 49, 48, 43, 40 | 40, 39, 38, 37, 34 | 50, 49, 48, 43, 40 |
| RRQR  | 40, 30, 29, 28, 27 | 50, 30, 29, 28, 27 | 40, 39, 38, 37, 36 | 50, 40, 39, 38, 37 |

**TechTC-300:** For our first real dataset, we use the TechTC-300 data, 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 [22]. Each data matrix consists of 150-280 documents (the rows of the data matrix), and each document is described with respect to 10,000-50,000 words (features are columns of the matrix). Thus, TechTC-300 provides a diverse collection of data sets for a systematic study of the performance of the SVM 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. We were left with 147 datasets, and we ran ten-fold cross validation and repeated it ten times on
Table 2: A subset of the TechTC matrices of our study

|   | id1, id2 | id1                                      | id2                                      |
|---|---------|------------------------------------------|------------------------------------------|
| (i) | 1092, 135724 | Arts: Music: Styles: Opera               | Arts: Education: Language: Reading Instructions |
| (ii) | 1092, 789236 | Arts: Music: Styles: Opera               | US Navy: Decommissioned Attack Submarines |
| (iii) | 17899, 278949 | US: Michigan: Travel & Tourism           | Recreation: Sailing Clubs: UK            |
| (iv) | 17899, 48446 | US: Michigan: Travel & Tourism           | Science: Chemistry: Analytical: Products  |
| (v)  | 14630, 14096 | US: Colorado: Localities: Boulder        | Europe: Ireland: Dublin: Localities       |
| (vi) | 10539, 300332 | US: Indiana: Localities: S               | Canada: Ontario: Localities: E           |
| (vii) | 10762, 524208 | US: Minnesota: Localities: I             | Games: Video Games: Action: Downloads: Free |
| (viii) | 10567, 11346 | USA: Indiana: Evansville                 | US: Florida: Metro Areas: Miami          |
| (ix)  | 10539, 194915 | US: Indiana: Localities: S               | US: Texas: Localities: D                 |
| (x)   | 10835, 14525 | US: Pennsylvania: Business & Economy     | US: Arkansas: Localities: M              |

49 such datasets. We set the parameter $C = 1$ in LIBSVM and used default settings. We tried different values of $C$ for the full-dataset and the out-of-sample error averaged over 49 TechTC-300 documents did not change much, so we report the results of $C = 1$.

We set the number of features to 200, 300, 400 and 500. Fig 1 shows the out-of-sample error for the 49 datasets. For the supervised feature selection, BSS is comparable to RFE and better than RRQR, LPSVM, full-data and uniform sampling in terms of out-of-sample error. For LPSVM, the number of selected features averaged over 49 datasets was greater than 500, but it performed worse than BSS.

We list the most frequently occurring words selected by supervised BSS for the $r_1 = 200$ case for ten TechTC-300 datasets averaged over 100 training sets. Table 2 shows the names of the ten TechTC-300 document-term matrices. The words shown in Table 3 were selected in all cross-validation experiments for these ten datasets. The words are closely related to the categories to which the documents belong, which shows that BSS selects important features from the support-vector matrix. 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: Decommissioned Attack Submarines”, the BSS algorithm selects submarine, shipmate, hullnumber, opera which are closely related to the two classes. Another example is the document-pair 10539, 300332, where 10539 belongs to the category of “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 which are closely related to the class values. Thus, we see that using only 2%-4% of all features we are able to obtain good out-of-sample error.

6.3 Experiments on Unsupervised Feature Selection

For the unsupervised feature selection case, we performed experiments on the same 49 TechTC-300 datasets and set $r_2$ to 300, 400 and 500 as seen in Fig 1. For LPSVM, the number of selected features averaged over 49 datasets was close to 300. In the unsupervised case, BSS is comparable to the other methods RRQR, LPSVM and RFE. These methods are better than random feature selection and full-data without feature selection. This shows that unsupervised BSS is a competitive feature selection algorithm. Supervised feature selection is better than unsupervised feature selection for BSS and RFE, while unsupervised RRQR and LPSVM are better than their supervised versions. Running BSS on the support-vector set is equivalent to running BSS on the training data. However,
Table 3: Frequently occurring terms of the ten TechTC-300 datasets of Table 2 selected by supervised BSS

(i) reading, education, opera, programs, spacer
(ii) submarine, shipmate, hullnumber, opera, spacer
(iii) sailing, yacht, michigan, travel, vacation
(iv) analysis, asbestos, environmental, michigan, vacation
(v) ireland, dublin, boulder, products, swords
(vi) ontario, elliot, shelbyville, indiana, lodge
(vii) games, graphics, download, version, minnesota
(viii) florida, evansville, music, events, chapter
(ix) dallas, indiana, estate, homes, church
(x) pennsylvania, arkansas, business, baptist, company

RRQR and LPSVM are primarily used as unsupervised feature selection techniques and so they perform well in that setting. RFE is a heuristic based on SVM and running RFE on the support-vectors is equivalent to running RFE on the training data.

Figure 1: Plots of out-of-sample error of Supervised and Unsupervised BSS compared with other methods for 49 TechTC-300 documents averaged over ten ten-fold cross validation experiments. Vertical bars represent standard deviation.
6.4 Approximate BSS

We describe a heuristic to make supervised BSS scalable to large-scale datasets. For datasets with large number of support vectors, we premultiply the support vector matrix $X$ with a random gaussian matrix $G \in \mathbb{R}^{t \times p}$ to obtain $\tilde{X} = GX$ and then use BSS to select features from the right singular vectors of $\tilde{X}$. The right singular vectors of $\tilde{X}$ closely approximates the right singular vectors of $X$. Hence the columns selected from $\tilde{X}$ will be approximately same as the columns selected from $X$. The algorithm is presented as Algorithm 2.

### Algorithm 2: Approximate BSS

**Input:** Support vector matrix $X \in \mathbb{R}^{p \times d}$, $t, r$.

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

1. Generate a random Gaussian matrix, $G \in \mathbb{R}^{t \times p}$.
2. Compute $\tilde{X} = GX$.
3. Compute right singular vectors $V$ of $\tilde{X}$ using SVD.
4. Run Algorithm 1 using $V$ and $r$ as inputs and get matrices $S$ and $D$ as outputs.
5. Return $S$ and $D$.

We performed experiments on a subset of Reuters Corpus dataset, namely reuters-CCAT, which contains binary classification task. We used the L2-regularized L2-loss SVM formulation in the dual form in LIBLINEAR and set the value of $C$ to 10. We experimented with different values of $C$ on the full-dataset, and since there was small change in classification accuracy among the different values of $C$, we chose $C = 10$ for our experiments. We pre-multiplied the support vector matrix with a random gaussian matrix of size $t \times p$, where $p$ is the number of support vectors and $t$ was set to 128 and 256. We repeated our experiments five times using five different random gaussian matrices to get around the randomness. We set the value of $r_1$ in BSS to 1024 and 2048. LPSVM selects 1898 features for CCAT. Table 4 shows the results of our experiments. We observe that the out-of-sample error using approx-BSS is close to that of RRQR and comparable to RFE, LPSVM and full-data. The out-of-sample error of approx-BSS decreases with an increase in the value of $t$. This shows that we get a good approximation of the right singular vectors of the support vector matrix with an increase in number of projections.

Table 4: Results of Approximate BSS. CCAT (train / test): (23149 / 781265), d=47236. Mean and standard deviation (in parenthesis) of out-of-sample error. Eout of full-data is 8.66 ± 0.54.

|   |     | BSS ($t = 128$) | BSS ($t = 256$) | RRQR | RFE | LPSVM |
|---|-----|----------------|-----------------|------|-----|-------|
| CCAT | 1024 | 10.53 (0.59) | 10.35 (0.64) | 9.97 (0.62) | 8.92 (0.57) | 9.97 (0.55) |
| CCAT | 2048 | 11.13 (0.66) | 10.63 (0.62) | 10.04 (0.66) | 8.56 (0.54) | 9.97 (0.55) |
7 Conclusions

We describe a provably accurate feature-selection method for linear SVMs which works well empirically. We present a simple method of extending an unsupervised feature selection method into a supervised one for SVM. Supervised BSS is better than unsupervised BSS based feature selection. BSS is comparable and often better than prior state-of-the-art feature selection methods for SVM in both supervised and unsupervised settings which did not come with provable guarantees. BSS has been used as a feature selection method for k-means clustering [23], for finding core-sets in linear regression [24] and our work helps to expand research in this direction.

References

[1] A. Dasgupta, P. Drineas, B. Harb, V. Josifovski, and M.W. Mahoney. Feature selection methods for text classification. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 230–239, 2007.

[2] N. Cristianini and J. Shawe-Taylor. Support Vector Machines and other kernel-based learning methods. Cambridge University Press, 2000.

[3] V.N. Vapnik. Statistical Learning Theory. Theory of Probability and its Applications, 16:264–280, 1998.

[4] V.N. Vapnik and A. Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to their Probabilities. Theory of Probability and its Applications, 16:264–280, 1971.

[5] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. Machine Learning, 46(1-3):389–422, 2002.

[6] M. Glenn Fung and O.L. Mangasarian. A feature selection newton method for support vector machine classification. Comput. Optim. Appl., 28(2):185–202, 2004.

[7] A. Rakotomamonjy. Variable selection using svm based criteria. JMLR, 3:1357–1370, 2003.

[8] J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for svms. In NIPS, volume 12, pages 668–674, 2000.

[9] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping. Use of the zero norm with linear models and kernel methods. JMLR, 3:1439–1461, 2003.

[10] M. Tan, L. Wang, and I.W. Tsang. Learning sparse svm for feature selection on very high dimensional datasets. In Proceedings of the 27th International Conference on Machine Learning (ICML), pages 1047–1054, 2010.

[11] H. Do, A. Kalousis, and M. Hilario. Feature weighting using margin and radius based error bound optimization in svms. In European Conference on Machine Learning (ECML), pages 315–329, 2009b.

[12] A. Kalousis and H.T. Do. Convex formulations of radius-margin based support vector machines. In Proceedings of the 30th International Conference on Machine Learning (ICML), pages 169–177, 2013.

[13] L. Wang, J. Zhu, and H. Zou. The doubly regularized support vector machine. Statistica Sinica, 16(2):589, 2006.

[14] L. Wang, J. Zhu, and H. Zou. Hybrid huberized support vector machines for microarray classification and gene selection. Bioinformatics, 24(3):412–419, 2008.
[15] G. Ye, Y. Chen, and X. Xie. Efficient variable selection in support vector machines via the alternating direction method of multipliers. In AISTATS, pages 832–840, 2011.

[16] R. Gilad-Bachrach, A. Navot, and N. Tishby. Margin based feature selection-theory and algorithms. In Proceedings of the twenty-first international conference on Machine learning (ICML), page 43, 2004.

[17] J.D. Batson, D.A. Spielman, and N. Srivastava. Twice-ramanujan sparsifiers. In Proceedings of the 41st annual ACM STOC, pages 255–262, 2009.

[18] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. JMLR, pages 361–397, 2004.

[19] C-C. Chang and C-J. Lin. Libsvm: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[20] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. JMLR, pages 1871–1874, 2008.

[21] C. Bhattacharyya. Second order cone programming formulations for feature selection. JMLR, 5:1417–1433, 2004.

[22] D. Davidov, E. Gabrilovich, and S. Markovitch. Parameterized generation of labeled datasets for text categorization based on a hierarchical directory. In Proceedings of the 27th Annual International ACM SIGIR Conference, pages 250–257, 2004. http://techtc.cs.technion.ac.il/techtc300/techtc300.html

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

[24] C. Boutsidis, P. Drineas, and M. Magdon-Ismail. Near-optimal coresets for least-squares regression. IEEE transactions on information theory, 59(10):6880–6892, 2013.