JMLR: Workshop and Conference Proceedings 10: 86-96 The Fourth Workshop on Feature Selection in Data Mining

Increasing Feature Selection Accuracy for L_1 Regularized Linear Models in Large Datasets

Abhishek Jaiantilal ABHISHEK.JAIANTILAL@COLORADO.EDU
Department of Computer Science
University of Colorado
Boulder, CO, 80309, USA

Gregory Grudic GREG.GRUDIC@FLASHBACKTECHNOLOGIES.COM
Flashback Technologies, LLC
Longmont, CO, 80503, USA

Editor: Huan Liu, Hiroshi Motoda, Rudy Setiono, and Zheng Zhao

Abstract

L_1 (also referred to as the 1-norm or *Lasso*) penalty based formulations have been shown to be effective in problem domains when noisy features are present. However, the L_1 penalty does not give favorable asymptotic properties with respect to feature selection, and has been shown to be inconsistent as a feature selection estimator; e.g. when noisy features are correlated with the relevant features. This can affect the estimation of the correct feature set, in certain domains like robotics, when both the number of examples and the number of features are large. The weighted lasso penalty by (Zou, 2006) has been proposed to rectify this problem of correct estimation of the feature set. This paper proposes a novel method for identifying problem specific L_1 feature weights by utilizing the results from (Zou, 2006) and (Rocha et al., 2009) and is applicable to regression and classification algorithms. Our method increases the accuracy of L_1 penalized algorithms through randomized experiments on subsets of the training data as a fast pre-processing step. We show experimental and theoretical results supporting the efficacy of the proposed method on two L_1 penalized classification algorithms.

Keywords: Feature selection, L_1 penalized algorithms

1. Introduction

Feature selection using the L_1 penalty (also referred as 1-norm or *Lasso* penalty) has been shown to perform well when there are spurious features mixed with relevant features and this property has been extensively discussed in (Efron et al., 2004), (Tibshirani, 1996) and (Zhu et al., 2003). In this paper, we focus on feature selection via the L_1 penalty for classification, addressing open problems related to feature selection accuracy and large datasets. This paper is organized as follows, Section-2 presents motivation and background, primarily focusing on the fact that asymptotically L_1 penalty based method might include spurious features. Based on the work in (Zou, 2006), we show that random sampling can find a set of weights that improves accuracy over the unweighted (which is normally used) L_1 penalty methods and we detail this in Section-3. In Section-4, we show results on two different classification algorithms and compare the weighted method proposed in (Zou, 2007) with the random sampling method described in our paper. Our method differs from Zou's

method as it hinges on random sampling to find the weight vector instead of using the L_2 penalty. The proposed method is shown to give significant improvement in accuracy over a number of data sets. Section 5 summarizes the results and concludes with future work.

The contribution of our work is as follows: we show that a fast pre-processing step can be used to increase the accuracy of L_1 regularized models and is a good fit when the number of examples are large; we connect the theoretical results from (Rocha et al., 2009) showing the viability of our method on various L_1 penalized algorithms and also show empirical results supporting our claim.

2. Background Information and Motivation

Consider the following setup in which information about n examples, each with p dimensions, is represented in a n x p design matrix denoted by X , with $y \in R^n$ representing target values/labels, and $\beta \in R^p$ representing a set of model parameters to be estimated. For our paper, we consider classification based linear models with a convex loss function and a penalty term (a regularizer). In (1), we show a regularized formulation that can be used to generally describe many machine learning algorithms. The metric or loss, $L(X, y, \beta)$, may represent various loss functions including 'hinge loss' for classification based Support Vector Machines (SVMs) and 'squared error loss' for regression.

$$\beta = arg\min_{\beta} L(X, y, \beta) + \lambda J(\beta) \tag{1}$$

$$\text{where} \quad L(X, y, \beta) = \text{ loss function }, \; J(\beta) = \text{ penalty function } \text{ and } \lambda \geq 0$$

Popular forms of the penalty functions ($J(\beta)$) are by using the L_2 and L_1 norm on β and are termed Ridge and Lasso penalty respectively in literature (refer to (Tibshirani, 1996)).

2.1 Asymptotic properties of L_1 penalty

Many papers including (Tibshirani, 1996), (Efron et al., 2004) and (Zhu et al., 2003) discuss the merits of the L_1 penalty. The L_1 penalty has been shown to be efficient in producing sparse models (models with many of the β's set to 0) and this feature selecting ability makes it robust against noisy features. In addition, the L_1 penalty is a convex penalty and when used in conjunction with convex loss functions, the resultant formulation has a global minimum.

As the L_1 penalty is used for simultaneous feature selection and correct estimation, a topic of interest is to understand whether sparsity holds when $n \to \infty$, n=number of examples. Intuitively, given enough samples, the estimated parameters β_n should approach the true parameters β_0.

$$y = X\beta_0 + \epsilon \tag{2}$$

Assume that the data is generated as shown in (2), with ϵ being gaussian noise of zero mean and β_0 being the true generating model parameters. Also, β_k^j represents the j^{th} feature for β_k. If A_0=$\{j \mid \beta_0^j \neq 0\}$ is the true model and A_n is the model found for $n \to \infty$. For **consistency** in feature selection, we need A_n=$\{ \; j \mid \beta_n^j \neq 0\}$ and $\lim_{n\to\infty} P(A_n = A_0)$= 1, that is we find the correct set of features A_0 asymptotically. (Zou, 2006) showed that lasso estimator is consistent (in terms of $\beta_N \to \beta_0$) but can be inconsistent as a feature selecting estimator in presence of correlated noisy features.

2.1.1 Hybrid SVM

(Zou, 2006) showed that weighted lasso penalty as shown in (3) and which is termed as the weighted lasso regression, can be used for simulataneous feature selection and creating accurate models. In (Zou, 2007), the same properties are applied in case of classification and referred to as '*Improved 1-norm SVM*' or 'Hybrid SVM'. The weighted lasso formulations for regression and classification are shown in (3) and (4) respectively. In (3), $\beta(OLS)$ denotes the weights found via least squares regression. For the weighted lasso penalty, the formulations in (3) and (4) are still convex and will require almost no modification to the (unweighted) lasso penalty based algorithms. Refer to (Zou, 2006) for the modifications that are needed. Intuitively, the weights found via the L_2 penalty are inversely proportional to the true model parameter β_0. If those weights are lower (i.e. the true model magnitude is higher) then in the weighted lasso penalty we are penalizing those corresponding features lesser and thereby encouraging those features to have higher magnitude in the weighted L_1 models and vice-versa for noisy features.

$$\text{Weighted Lasso Regression: } \min_{\beta} ||y - X\beta||^2 + \lambda \sum_j W_j|\beta_j| \quad \text{s.t. } W_j = |\beta(OLS)_j|^{-\gamma}, \gamma > 0 \tag{3}$$

$$\text{Improved 1-norm SVM: } \min_{\beta,\beta_0} \sum_i [1 - y_i(x_{:,i}\beta + \beta_0)]_+ + \lambda \sum_j W_j|\beta_j|, \tag{4}$$

$$\text{where } W_j = |\beta(l_2)_j|^{-\gamma}, \gamma > 0, \qquad \beta(l_2) = \arg\min_{\beta,\beta_0} \sum_i [1 - y_i(x_{:,i}\beta + \beta_0)]_+ + \lambda_2 \sum_j ||\beta_j||_2^2$$

$$\text{Improved SVM2: } \min_{\beta,\beta_0} \sum_i [1 - y_i(x_{:,i}\beta + \beta_0)]_+^2 + \lambda \sum_j W_j|\beta_j|, \tag{5}$$

$$\text{where } W_j = |\beta(l_2)_j|^{-\gamma}, \gamma > 0, \qquad \beta(l_2) = \arg\min_{\beta,\beta_0} \sum_i [1 - y_i(x_{:,i}\beta + \beta_0)]_+^2 + \lambda_2 \sum_j ||\beta_j||_2^2$$

$\{x_{:,i}, y_i\}$ represents an example, λ, λ_2 are regularizing parameters. $v_+ = max(v, 0)$ in the above equations.

2.2 Motivation for our Method

The weighted lasso penalty is dependent on obtaining suitable weights 'W'. (Zou, 2006, 2007) shows that the ordinary least squares estimates and the estimates from SVM with the L_2 norm penalty can be used to find the weights as shown in (3) and (4). For our paper, we obtain these weights via feature selection on randomized subsets of the training data. If the accuracy is higher than the unweighted case, it means that the features are appropriately (and correctly) weighted.

One of our goals was to see the translation of results from (Zou, 2006) to other linear formulations and thus we also experimented on the weighted **SVM2** formulation shown in (5) (unweighted formulation is shown in (7)). The SVM2 formulation is referred to in literature as Quadratic loss SVM (but with L_2 penalty) or 2-norm SVM (refer to (Shawe-Taylor and Cristianini, 2004)). It is squared hinge loss coupled with the L_1 penalty.

2.2.1 Efficient Algorithms to solve formulations with L_1 norm penalty

(Efron et al., 2004) showed an efficient algorithm for lasso regression called Least Angle Regression (LARS), that can solve for all values of λ, that is $0 \leq \lambda \leq \infty$. In (Rosset and

Zhu, 2007), a generic algorithm, for which LARS is a special case, is documented that can be used for all double differentiable losses with the L_1 penalty. For our experiments, we resort to specific linear SVM based formulations for which entire regularization paths can be constructed. (6) is the penalized formulation for **'1-norm SVM'**. (Zhu et al., 2003) showed a simple piecewise algorithm to solve for $0 \leq \lambda \leq \infty$ in the 1-norm SVM. As the loss and the penalty function are both singly differentiable, a piecewise path cannot be constructed as efficiently as in LARS but linear programming can be employed to calculate the step size. (7) is an equivalent to (6) and similar to the formulation seen in literature except with the L_2 loss function instead of the L_1 loss function. (7) is the penalized formulation for squared hinge loss (or Quadratic loss SVM) with the L_1 penalty. As the loss function is doubly differentiable, via the method described by (Rosset and Zhu, 2007), an efficient piecewise algorithm that be constructed to solve for $0 \leq \lambda \leq \infty$. Our vested interest in using such piecewise algorithms, is to help understand whether better (entire) regularization paths are created or not for the weighted L_1 penalty.

$$\text{1-norm SVM:} \quad \min_{\beta,\beta_0} \sum_i [1 - y_i(x_{:,i}\beta + \beta_0)]_+ + \lambda \sum_j |\beta_j|, \tag{6}$$

$$\text{Equivalent to (6)} : \min_{\beta,\beta_0} \; ||\beta||_1 + C \sum_i \xi_i, \text{s.t. } y_i(x_{:,i}\beta + \beta_0) \geq 1 - \xi_i, \; \xi_i \geq 0$$

$$\text{SVM2:} \quad \min_{\beta,\beta_0} \sum_i [1 - y_i(x_{:,i}\beta + \beta_0)]_+^2 + \lambda \sum_j |\beta_j|, \tag{7}$$

$$v_+ = max(v, 0) \text{ in the above equations.}$$

3. Randomized Sampling (RS) Method to Create Weight Vector

Our randomized sampling method depends on a small random subset of training data. We assume that the subset of the training data is small, i.e. it is computationally cheap to act on such a set in a reasonable time. Also, such randomized sampling is done multiple times.

3.1 Randomized Sampling (RS) Method

Our randomized sampling algorithm is described below in **Algorithm-1: Randomized Sampling Method**. The algorithm can be explained as follows: We choose a subset of m examples out of the presented n examples such that $m << n$. We train a L_1 penalty based algorithm (e.g. 1-norm SVM (Zhu et al., 2003), SVM2, etc.) so that we can find a set of relevant features. We keep a note of the features that we found in that particular experiment. After many such randomized experiments, the counts of the number of times a feature was found in these randomized experiments is summed up and normalized and denoted by V.

This count vector, denoted by V, is then inverted and used as weights for the weighted version of the algorithm; i.e. weights used in the weighted formulations are $W = 1/V$. Intuitively, if a feature is important and is found multiple times via the RS method, then the corresponding weight for the feature is less and thus it is penalized lesser, encouraging higher magnitude for the feature.

Algorithm-1: Randomized Sampling (RS) Method:

Input: $\boldsymbol{n}$ examples each with $\boldsymbol{p}$ features, $\boldsymbol{K}$ randomized experiments, $\boldsymbol{B}$ (Block) number of examples used to train model in each randomized experiment.

Output: Count vector $\boldsymbol{W}$ (1xD vector) representing number of times features were selected in K randomized experiment.

Divide N examples into K randomized sets each of size B and denote them as $Ntrn_i, i = 1 \dots K$ and *let* $V \longleftarrow 0$

for $i = 1 \dots K$ **do**

  Get $Ntrn_i$, construct $Ntst_i$ and $Nval_i$ set.

  Train Model$_i$ = L_1 Algorithm($Ntrn_i$, $Ntst_i$, $Nval_i$)

  S_i = selected features in Model$_i$ via validation data.

  $V \leftarrow V + \{x, x \in R^D | x_j = 1 \text{ if } j \in S_i \text{ else } x_j = 0\}$

end

3.2 Consistency of choosing a set of Features from Randomized Sampling (RS) Experiments

Our method is dependent on finding some set of relevant features and their counts for a given dataset via the RS method. Our experimental results are restricted to the weighted and unweighted formulation for SVM2 and 1-norm SVM, but our theoretical results are applicable to all linear models with twice differentiable loss function with the L_1 penalty. We next mention results, regarding the asymptotic consistency and normality properties in n (number of examples) for L_1 penalized algorithms, which can help understand the consistency of our method.

Lemma 1: This result is from Theorem-5 in (Rocha et al., 2009). If the loss function $L(X, y, \beta)$ shown in (1) is bounded, unique and a convex function, with $E|L(X, y, \beta)| < \infty$ and furthermore $L(X, y, \beta)$ is twice differentiable with a positive hessian H matrix, then the following consistency condition defined for the L_1 penalty when using the formulation in (1) and true model in (2):

$$||H_{A^c,A}[H_{A,A} - H_{A,\beta_0}H^{-1}_{\beta_0,\beta_0}H_{\beta_0,A}]^{-1}sign(\beta_A)||_\infty \leq 1, \text{ where } H_{x,y} = \frac{d^2L(X, y, \beta)}{dxdy} \quad (8)$$

Where $A_c = \{j \in 1...p | \beta_j = 0\}$, $A = \{j \in 1...p | \beta_j \neq 0\}$ and β_0 is an intercept.

- if λ_n is a sequence of non-negative real numbers such that $\lambda_n n^{-1} \to 0$ and $\lambda_n n^{-(1+c)/2} \to \lambda > 0$ for some $0 < c < 1/2$ as $n \to \infty$ and the condition (8) is satisfied then $P[sign(\beta_n(\lambda_n) = sign(\beta)] \geq 1 - exp[-n^c]$. β_n is parameter found for number of examples=n .

- If the condition in (8) is not satisfied then for any sequence of non-negative numbers λ_n $\lim_{n\to\infty} P[sign(\beta_n(\lambda_n)) = sign(\beta)] < 1$. The probability of choosing incorrect variables is bounded to $exp(-Dn^c)$, where D is a positive constant (shown in the proof of Theorem 5 of (Rocha et al., 2009)).

If the condition in (8) is fulfilled, it means that the interactions between relevant and noisy features are distinguishable and the L_1 penalty can correctly identify the signs in β. If

the condition in (8) is not fulfilled, then noisy features will be added to the model with a probability away from 1. Also, note that the above conditions are applicable for 1-norm SVM, as shown in (Rocha et al., 2009).

Lemma 2: We use b to specify the size of the subset and assuming $b \to \infty$, then from Lemma-1, when condition of consistency (8) is satisfied then $P[sign(\beta_b(\lambda_b) = sign(\beta)] \geq 1 - exp[-b^c] \approx 1$, where β_b and λ_b represent the parameters for the subset of size b. For k such subsets V, as depicted in the algorithm in Section 3.1, is bounded to $k(1-exp[-b^c]) \approx k,\ b \to \infty$. When the condition in (8) is not satisfied then the probability of choosing noisy variables in a subset is upperbounded to $exp(-Db^c)$ and for k subsets, $sum(V_j) \leq k \cdot exp(-Db^c)$ and $V_j \approx 0,\ b \to \infty$, (where V_j are indices of noisy variables). Thus, the noisy variables have a probability of having a low count in V and a large weight in W, thus penalizing the noisy features heavily .

Table 1: Mean ± Std. Deviation of Error Rates in % on Models 1 & 4 by SVM2

q	p	2-norm SVM2	1-norm SVM2	Hybrid (Zou)	RS(20%)	RS(30%)	RS(40%)
2	14	9.64±2.30	7.92±1.89	7.88±2.09	7.69±1.71	7.67±1.66	7.68±1.69
4	27	10.90±2.41	8.01±1.84	7.88±2.09	7.73±1.59	7.73±1.59	7.71±1.60
6	44	12.17±2.64	7.93±1.79	7.79±1.69	7.64±1.60	7.64±1.59	7.64±1.52
8	65	13.45±2.96	8.13±2.10	7.87±1.84	7.82±1.85	7.83±1.85	7.81±1.83
12	119	16.91±3.24	8.11±1.95	8.05±1.94	7.78±1.71	7.78±1.70	7.76±1.66
16	189	17.93±3.32	7.87±1.78	8.29±2.41	7.66±1.57	7.66±1.63	7.66±1.63
20	275	19.31±3.32	8.06±2.14	8.04±2.01	7.69±1.81	7.74±1.89	7.77±1.87

Random Sampling is Subsampling: To better quantify our random sampling method, we explain it in terms of subsampling (refer to (Politos et al., 1999)). Subsampling is a method of sampling m examples from n total examples with $m < n$, unlike bootstrap that samples n times with replacement from n samples. Let estimator θ be a general function of i.i.d data generated from some probability distribution P. In our case of feature selection, this estimator is the feature set. We are interested in finding an estimator and it's confidence region based on the probability P of the data and we define it as $\theta(P)$. When P is large then we can construct an empirical estimator $\hat{\theta}_n$ of $\theta(P)$ such that $\hat{\theta}_n = \theta(\hat{P}_n)$, where P_n is the empirical distribution; that is estimate the true feature set empirically. We define a root of form $\tau_n(\hat{\theta} - \theta)$, where τ_n is some sequence (like $\sqrt{n}$ or n) increasing with n (number of examples), and we are looking at the difference between the empirical estimator $\hat{\theta}_n$ and the true estimator θ. We define $J_n(P)$ to be the sampling distribution of $\tau_n(\hat{\theta} - \theta(F))$ based on a sample size of n from P and define the CDF as

$$J_n(x, P) = \text{Probability}_P\{\tau_n(\hat{\theta}_n - \theta) \leq x\},\ \ x \in R \tag{9}$$

Lemma 3: From (Politos et al., 1999), for data generated via i.i.d., there is a limiting law $J(P)$ such that $J_n(P)$ converges weakly (in probability) to $J(P)$ and $\tau_b(\theta_n - \theta) \to 0$ as $n \to \infty$ with the conditions that $\tau_b/\tau_n \to 0$, $b \to \infty$ and $b/n \to 0$, where b is the number of examples in the subsample experiment and n is the total number of available examples.

Lemma 3, has remarkably weak conditions for subsampling and it requires that the root has some limiting distribution and the sample size b is not too large (but still going to infinity) compared to n . In our case, the subsets are of size $b \to \infty, b << n$ and for the rate of estimation at $\tau_n \propto n^c, \tau_b \propto b^c, 0 < c \leq 1$, then $\tau_b/\tau_n \to 0$. For the RS method, we create weight vector whose index for a feature is non-zero if that feature was found in

Table 2: Mean ± Std. Deviation of Error Rates in % on on Models 1 & 4 by 1-norm SVM

q	p	2-norm SVM	1-norm SVM	Hybrid (Zou)	RS(20%)	RS(30%)	RS(40%)
2	14	8.74±1.30	7.64±0.09	7.64±1.02	7.63±1.02	7.64±1.01	7.53±0.09
4	27	9.76±1.75	7.85±1.14	7.95±1.34	7.83±1.28	7.79±1.24	7.69±1.19
6	44	10.57±1.95	7.85±1.01	7.92±1.12	7.79±1.12	7.77±1.18	7.69±1.23
8	65	11.47±2.31	7.81±0.99	7.99±1.36	7.75±1.13	7.74±1.15	7.63±1.09
12	119	13.27±2.48	7.91±0.98	8.04±1.35	7.77±1.16	7.82±1.21	7.63±1.00
16	189	15.58±2.94	7.94±1.15	7.87±1.21	7.74±1.31	7.75±1.23	7.64±1.14
20	275	17.14±2.96	7.90±1.00	7.85±1.11	7.77±1.20	7.80±1.27	7.69±1.19

a particular experiment. $\hat{\theta}_n$ is the sample mean of n such RS experiments weights, having mean converging to $\theta(P)$ (due to Lemma-2). Thus estimating the true feature set on basis of random sampling of subsets of data is weakly convergent. (Zou, 2006) used a root-n-consistent estimator's weight (from the L_2 penalty) but mentions that the conditions can be further weakened and if there is an a_n such that $a_n \to \infty$ and $a_n(\hat{\theta} - \theta) = O(1)$ then such an estimator can also be used. By Lemma-3, our RS estimator is one such consistent estimator and thus can be used as a valid estimator for usage with the weighted lasso penalty.

4. Algorithms and Experiments

In this paper, we limit ourselves to an empirical study of data block sizes for the RS estimator. We replicate the experiments from *'An Improved 1-norm SVM for Simultaneous Classification and Variable Selection'* by (Zou, 2007) and report on 1-norm SVM and SVM2. **Method for choosing Weights (for Hybrid and RS) and Validation data (for RS):** For the Hybrid SVM, in order to find the optimal weights via L_2 penalty, we use the method described by (Zou, 2007). We first find the best SVM (or SVM2) algorithm model weights ($\beta(l_2)$) with the L_2 penalty via a parameteric search over costs C={0.1, 0.5, 1, 2, 5, 10}. We then create entire piecewise paths for various weight values $|\beta(l_2)|^{-\gamma}, \gamma = \{1, 2, 4\}$; choose the best performing model on validation data and then report on the test dataset. Description on how we chose training set for the RS method is given in individual experiments. Our RS experiments need validation data to help choose the relevant features for each of the RS training set. We do the following: if n is the size of the training set and we choose m of those examples for the current RS training set, we just use the left out $n - m$ examples (as validation) for choosing the best features from the piecewise paths generated by the L_1 algorithm on the m examples. In case, if a held out validation set was present, we use that instead.

4.1 Synthetic Datasets

We simulate two synthetic datasets, one akin to "orange data" described in (Zhu et al., 2003) and another a bernoulli distribution based dataset. The following notation is used for some of the tables: We use **"C"** and **"IC"** to denote the mean number of **correctly** and **incorrectly** selected features, respectively. Also, we resort to reporting to mean and std. deviation as the median of the incorrectly selected features was 0 for many experiments. **"PPS"** stands for the **probability of perfect selection**, i.e the probability of only choosing the correct feature set.

Models 1 and 4 from (Zou, 2007): The "orange data" has two classes, one inside the other like the core inside the skin of the orange. The first class has two independent standard normals x_1 and x_2. The second class also has two independent standard normals x_1 and x_2 but is conditioned on $4.5 \leq x_1^2 + x_2^2 \leq 8$. To simulate the effects of noise, there are 'q' independent standard normals. The Bayes rule is 1-2$I(4.5 \leq x_1^2 + x_2^2 \leq 8)$, where $I()$ is an indicator function and the Bayes error is about 4%. We resort to an enlarged dictionary $D = \{\sqrt{2}x_j, \sqrt{2}x_j x_k, x_j^2, j, k = 1, 2, \ldots, 2+q\}$ as the original space is not linear. We have independent sets of 100 validation examples and 20000 test examples and. 'q' is set to 2, 4, 6, 8, 12, 16, 20 and we report on 500 experiments.

For the RS method, block sizes were set to 20%, 30% and 40% of the total training size and performed 10/(%size of each block/100) total experiments; i.e. for 20% we generated 10/0.2=20 total randomized training sets each of size 0.2*(total training data). The weighted vector was created via the RS method described earlier and then used to train the weighted 1-norm and SVM2 algorithms.

Table 3: Variable Selection Results on Models 1 & 4 using SVM2

q		6	8	12	16	20
p		44	65	119	189	275
1-norm SVM2	IC	1.5±2.59	1.42±2.44	1.67±3.4	1.58±2.95	1.71 ±3.52
	PPS	0.554	0.544	0.536	0.564	0.592
Hybrid SVM2	IC	1.05±1.87	1.03±1.79	1.19±2.05	1.35±2.51	1.13±2.21
	PPS	0.596	0.598	0.554	0.576	0.596
RS(20%)	IC	0.65±1.15	0.62±1.15	0.8±1.48	0.61±1.17	0.54±1.04
	PPS	0.636	0.646	0.600	0.686	0.666
RS(30%)	IC	0.69±1.18	0.73±1.15	0.70±1.27	0.63±1.25	0.56±1.06
	PPS	0.626	0.604	0.626	0.666	0.662
RS(40%)	IC	0.62±1.05	0.61±1.01	0.68±1.29	0.66±1.25	0.55±1.02
	PPS	0.644	0.636	0.650	0.668	0.672
RS(50%)	IC	0.67±1.11	0.65±1.19	0.69±1.31	0.62±1.36	0.59±1.14
	PPS	0.628	0.628	0.630	0.670	0.670

*C (mean of Correct features)=2 for all above experiments

We depict error rates in Table 1 & 2 for SVM2 and 1-norm SVM respectively. q depicts the number of noise features in original space and p represents the number of features in the new space via the dictionary D. The L_2 algorithm version, in the 3^{rd} column, show increasing error rates as the number of noisy features increase. The L_1 algorithm version, in the 4^{th} column is much more robust to noise and the error rates do not degrade at all. Hybrid SVM perform usually better than the unweighted 1-norm SVM (except for couple of times for in Table 2). For all different block sizes, the RS method performs best. The feature selecting ability of individual algorithm is depicted in Table 3 (Note: 1-norm SVM results were omitted for space constraints and the results were similar to those of SVM2). We can see that the probability of finding the best model is high for all the algorithms. Hybrid is better at that compared to the 1-norm and the RS method performs best.

Models 2, 3 and 5 from (Zou, 2007): Models 2, 3 and 5 are simulated from the model $y \sim \text{Bernoulli}\{p(u)\}$ where $p(u) = exp(x^T\beta + \beta_0 + \epsilon)/(1 + exp(x^T\beta + \beta_0) + \epsilon)\}$ with ϵ being

a standard normal representing error. We create 100 training examples, 100 validation examples, 20,000 test examples and report on 500 randomized experiments.

Model 2 (Sparse Model): We set $\beta_0 = 0$ and $\beta = \{3, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 3\}$. The features $x_1, ...x_{12}$ are standard normals and experiments are done with correlation between x_i and x_j set to $\rho = \{0, 0.5\}$. The Bayes rule is to assign classes to $2I(x_1 + x_6 + x_{12}) - 1$.

Model 3 (Sparse Model): We use $\beta_0 = 1$ and $\beta = \{3, 2, 0, 0, 0, 0, 0, 0, 0, 0\}$. The features $x_1, ...x_{12}$ are standard normals and experiments are done with correlation between x_i and x_j set to $\rho = \{0, 0.5\}^{|i-j|}$. The Bayes rule is to assign classes to $2I(3x_1 + 2x_2 + 1) - 1$.

Model 5 (Noisy features): We use $\beta_0 = 1$ and $\beta = \{3, 2, 0, 0, 0, 0, 0, 0, 0, 0\}$. The features $x_1, ...x_{12}$ are standard normals and experiments are done with correlation set to $\rho = 0.5^{|i-j|}$. We added 300 independent normal variables as noise features to get a total of 309 features.

Table 4: Mean Error rates in % for Models 2, 3 & 5 using SVM2

Exp. Name	Correlation	Bayes	2-norm	1-norm	Hybrid	RS(20%)	RS(30%)	RS(40%)
Model 2	$\rho = 0$	6.04	9.77	8.14	7.46	7.51	7.53	7.55
	$\rho = 0.5$	4.35	7.74	6.43	5.96	5.97	5.86	5.86
Model 3	$\rho = 0$	8.48	11.04	9.79	9.54	9.46	9.46	9.45
	$\rho = 0.5$	7.03	8.49	9.51	8.45	8.17	8.17	8.20
Model 5	$\rho = 0.5^{\|i-j\|}$	6.88	31.31	9.32	8.5	8.6	8.56	8.22

*range of std. deviation in accuracy for above table was between 1.02 to 1.96.

For Models 2, 3 and 5: error rates are reported in Table-4 for SVM2 (results for 1-norm SVM were similar and hence skipped). Note, weighted models are consistently better than both of their 1-norm and 2-norm unweighted counterparts. The RS method has equal or greater accuracy than the Hybrid version.

4.2 Real World Datasets

UCI datasets: In Table 5, results on the Spam, WDBC and Ionosphere datasets from UCI repository, by (Asuncion and Newman, 2007), are reported. For WDBC and Ionosphere dataset, we split the data into 3 parts with 2 parts used for training (and validation) and the 3^{rd} remaining part for testing. For the Spam dataset, indicators for test (1536 examples) set and training set can be obtained from `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`. For our RS method we generated smaller datasets from the training set as follows: If the training set size is N and size for individual RS set is K, then the number of datasets generated are $10 * N/K$. We also show the size of the RS training set as Block in the table. For Hybrid SVM, the best parameter for γ and C are chosen as described earlier in Section 4. We report on 50 randomized experiments. In Table 5, error rates for both SVM2 and 1-norm SVM are shown. The use of weights via Hybrid and RS method, always increases the accuracy from the unweighted case. Also, as seen on both synthetic and real world datasets, RS blocksize does not create that much variability in the results.

Robotic Dataset: We now discuss a novel use of our subsampling method on robotic datasets (Procopio, 2007). These datasets are created by hand labeling 100 images obtained from running the DARPA LAGR robot in varied outdoor environments. The classes labeled are robot traverseable path and obstacles. The authors provide pre-extracted color histogram features for the dataset at (Procopio et al., 2009). We used a subset (12,000

Table 5: Mean ± Std. Deviation of Error Rates on Real world Datasets

Dataset	Algorithm (Name/Block)	Without Weighting	Randomized Sampling Weighting	Hybrid SVM	2-norm SVM
WDBC	1-norm (100)	3.66 ± 1.17	2.79 ± 0.93	2.89 ± 0.79	4.05 ± 1.36
	1-norm (150)		2.79 ± 0.90	3.16 ± 1.22	
	SVM2 (100)	3.55 ± 1.81	2.78 ± 1.03	2.73 ± 1.01	
	SVM2 (150)		2.90 ± 1.15	2.91 ± 1.13	
SPAM	1-norm (200)	9.09 ± 0.878	8.18 ± 0.49	8.31 ± 0.61	7.06 ± 0.04
	1-norm (1000)		7.53 ± 0.17	8.19 ± 0.73	
	SVM2 (200)	8.45 ± 3.43	7.38 ± 0.52	7.39 ± 0.30	
	SVM2 (1000)		7.70 ± 2.73	7.48 ± 0.52	
Ionosphere	1-norm (50)	12.38 ± 2.04	11.52 ± 1.39	11.84 ± 1.38	13.03 ± 2.86
	1-norm (75)		11.25 ± 1.98	11.56 ± 1.73	
	1-norm (100)		11.29 ± 1.65	11.72 ± 1.23	
	SVM2 (50)	12.69 ± 2.82	11.43 ± 2.52	11.21 ± 2.66	
	SVM2 (75)		11.61 ± 2.50	11.22 ± 2.58	
	SVM2 (100)		11.37 ± 2.67	11.26 ± 2.68	

Table 6: Avg. Error rate on Robotic Datasets from (Procopio, 2007)

	DS1A	DS2A	DS3A
Unweighted SVM2	8.92	4.36	1.24
Weighted SVM2	6.41	4.13	1.15

examples) of the available data for each of the 100 frames. Each example is 15 dimensional. We set our experimentation as follows: for each frame F_i, i is the index of the frame, we divide the obtained examples (12K examples) into 8 folds (9.6K examples) for training, 1 fold (1.2K examples) for validation and 1 fold (1.2K examples) for testing. We train/validate/test on the unweighted SVM2 algorithm. For the weighted experiment, we train via our RS method, by dividing the training into 10 subsets (each 960 examples) and finding the weight vector. This weight vector is then used to create the weighted SVM2 models and we report on the test set. Now, instead of discarding weights when a new frame arrives, we use the weights found in frame F_i again in F_{i+1}, i.e. if weights in frame F_i are noted as W_i then:

$W_{i+1} \leftarrow \{W_i + \text{weight results of RS for frame } F_{i+1}\}$

This is one experimental environment, where creating L_2 models for the entire data is not feasible and the RS estimator is a potential approach. Also, propagating feature importance between frames is an advantage for the RS estimator. In Table-6, we show overall results for 100 frames for 3 datasets done 10 times. We propagate the weights for the weighted SVM2 between frames. As shown, there is a drop in error rates (between 5-28%) for the weighted SVM2 compared to the unweighted SVM2. The overhead of computing the weights via RS was $< 10\%$ that of computing a model for the entire training set.

5. Conclusions and Future work

A Random Sampling framework is presented and is empirically shown to give effective feature weights to the lasso penalty, resulting in both increased model accuracy and feature selection accuracy. The proposed framework is at least as effective (and at times more

effective than) the Hybrid SVM, with the added benefit of significantly lower computational cost. In addition, unlike the Hybrid SVM which must see all the data at once, Random Sampling is shown to be effective in an on-line setting where predictions must be made based on only partially available data (as in data taken from the robotics domain). In this paper the framework is demonstrated on two types of linear classification algorithms, and theoretical support is presented showing its applicability, in general, to sparse algorithms.

References

A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. URL `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

Dimitris N. Politos, Joseph P. Romano, and Michael Wolf. *Subsampling*. Springer, 1999.

Michael J. Procopio. Hand-labeled DARPA LAGR datasets. Available at `http://ml.cs.colorado.edu/~procopio/labeledlagrdata/`, 2007.

Michael J. Procopio, Jane Mulligan, and Greg Grudic. Learning terrain segmentation with classifier ensembles for autonomous robot navigation in unstructured environments. *Journal of Field Robotics*, 26(2):145–175, 2009. doi: http://dx.doi.org/10.1002/rob.20279.

Guilherme V. Rocha, Xing Wang, and Bin Yu. Asymptotic distribution and sparsistency for l1-penalized parametric m-estimators with applications to linear svm and logistic regression, 2009. URL `http://www.citebase.org/abstract?id=oai:arXiv.org:0908.1940`.

Saharon Rosset and Ji Zhu. Piecewise linear regularized solution paths. *Annals of Statistics*, (35), 2007.

John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani. 1-norm support vector machines. In *Neural Information Processing Systems*, page 16. MIT Press, 2003.

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429(12), December 2006.

Hui Zou. An improved 1-norm svm for simultaneous classification and variable selection. *AISTATS*, 2:675–681, 2007.