2006; Shi et al., 2012) make a assumption that all examples in the original space are separated with a positive margin (with a high probability). Another analysis in (Zhang et al., 2014) assumes the weight vector for classification is sparse. These assumptions are too strong to hold in many real applications.

Contributions. To address these limitations, we propose dual-sparse regularized randomized reduction methods referred to as **DSRR** by leveraging the (near) sparsity of dual solutions for large-scale high-dimensional (**LSHD**) classification problems (i.e., the number of (effective) support vectors is small compared to the total number of examples). In particular, we add a dual-sparse regularizer into the reduced dual problem. We present a novel theoretical analysis of the recovery error of the dual variables and the primal variable and study its implication for different randomized reduction methods (e.g., random projection, random hashing and random sampling).

Novelties. Compared with previous works (Blum, 2005; Balcan et al., 2006; Shi et al., 2012; Paul et al., 2013), our theoretical analysis demands a mild assumption about the data and directly provides guarantee on a small recovery error of the obtained model, which is critical for subsequent analysis, e.g., feature selection (Guyon et al., 2002; Brank et al., 2002) and model interpretation (Rätsch et al., 2005; Sonnenburg & Franc, 2010; Rtsch et al., 2005; Sonnenburg et al., 2007; Ben-Hur et al., 2008). For example, when exploiting a linear model to classify people into sick or not sick based on genomic markers, the learned weight vector is important for understanding the effect of different genomic markers on the disease and for designing effective medicine (Jostins & Barrett, 2011; Kang & Cho, 2011). In addition, the recovery could also increase the predictive performance, in particular when there exists noise in the original features (Goldberger et al., 2005).

Compared with (Zhang et al., 2014) that proposes to recover a linear model in the original feature space by dual recovery, i.e., constructing a weight vector using the dual variables learned from the reduced problem and the original feature vectors, our methods are better in that (i) we rely on a more realistic assumption of the sparsity of dual variables (e.g., in support vector machine (SVM)); (ii) we analyze both smooth loss functions and non-smooth loss functions (they focused on smooth functions); (iii) we study different randomized reduction methods in the same framework not just the random projection.

In numerical experiments, we present an empirical study on a real data set to support our analysis and we also demonstrate a novel application of the reduction and recovery framework in distributed learning from LSHD data, which combines the benefits of the two complementary techniques for addressing big data problems. Distributed learning/optimization recently receives significant interest in solving big data problems (Jaggi et al., 2014; Li et al., 2014; Yang, 2013; Agarwal et al., 2011). However, it is notorious for high communication cost, especially when the dimensionality of data is very high. By solving a dimensionality reduced data problem and using the recovered solution as an initial solution to the distributed optimization on the original data, we can reduce the number of iterations and the communication cost. In practice, we employ the recently developed distributed stochastic dual coordinate ascent algorithm (Yang, 2013), and observe that using the recovered solution as an initial solution we are able to attain almost the same performance with only one or two communications of high dimensional vectors among multiple machines.

2. Preliminaries

Let $(\mathbf{x}_i, y_i), i = 1, \ldots, n$ denote a set of training examples, where $\mathbf{x}_i \in \mathbb{R}^d, y_i \in \{1, -1\}$. Assume both n and d are very large. The goal of classification is to solve the following optimization problem:

$$\mathbf{w}_* = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}^\top \mathbf{x}_i y_i) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 \quad (1)$$

where $\ell(z)$ is a convex loss function and λ is a regularization parameter. Using the conjugate function, we can turn the problem into a dual problem:

$$\alpha_* = \arg\max_{\alpha\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n} -\ell_i^*(-\alpha_i) - \frac{1}{2\lambda n^2}\alpha^\top X^\top X\alpha \quad (2)$$

where $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ is the data matrix and $\ell_i^*(\cdot)$ is the convex conjugate function of $\ell(z y_i)$. Given the optimal dual solution α_*, the optimal primal solution can be computed by $\mathbf{w}_* = \frac{1}{\lambda n} X\alpha_*$. For LSHD problems, directly solving the primal problem (1) or the dual problem (2) could be very expensive. We aim to address the challenge by randomized reduction methods. Let $A(\cdot) : \mathbb{R}^d \to \mathbb{R}^m$ denote a randomized reduction operator that reduces a d-dimensional feature vector into m-dimensional feature vector. Let $\hat{\mathbf{x}} = A(\mathbf{x})$ denote the reduced feature vector. With the reduced feature vectors $\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_n$ of the training examples, a conventional approach is to solve the following reduced primal problem

$$\mathbf{u}_* = \arg\min_{\mathbf{u}\in\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{u}^\top \hat{\mathbf{x}}_i y_i) + \frac{\lambda}{2}\|\mathbf{u}\|_2^2 \quad (3)$$

or its the dual problem

$$\hat{\alpha}_* = \arg\max_{\alpha\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n} -\ell_i^*(-\alpha_i) - \frac{1}{2\lambda n^2}\alpha^\top \hat{X}^\top \hat{X}\alpha \quad (4)$$

where $\hat{X} = (\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_n) \in \mathbb{R}^{m \times n}$. Previous studies have analyzed the reduced problems for random projection methods and proved the preservation of margin (Blum, 2005; Shi et al., 2012) and the preservation of minimum

enclosing ball (Paul et al., 2013). Zhang et al. (2014) proposed a dual recovery approach that constructs a recovered solution by $\hat{w} = \frac{1}{n}\sum_{i=1}^{n}[\hat{\ }]_i x_i$ and proved the recovery error for random projection under the assumption of low-rank data matrix or sparse w. In addition, they also showed that the naive recovery by $A^\top u$ (when $A(x) = Ax$) has a large recovery error.

One deficiency with the simple dual recovery approach is that due to the reduction in the feature space, many non-support vectors for the original optimization problem will become support vectors, which could result in the corruption in the recovery error. As a result, the original analysis of dual recovery method requires a strong assumption of data (i.e., the low rank assumption). In this work, we plan to address this limitation in a different way, which allows us to relax the assumption significantly.

3. DSRR and its Guarantee

To reduce the number of or the contribution of training instances that are non-support vectors in the original optimization problem and are transformed into support vectors due to the reduction of the feature space, we employ a simple trick that adds a dual-sparse regularization to the reduced dual problem. In particular, we solve the following problem:

$$\tilde{\ } = \arg\max_{2\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n} \grave{}_i(\ _i) \quad \frac{1}{2\ n^2}\ ^T\widehat{X}^\top\widehat{X} \quad \frac{1}{n}R(\ ) \tag{5}$$

where $R(\ ) = \ k\ k_1$, and $\ > 0$ is a regularization parameter, whose theoretical value will be revealed later.

To further understand the added dual-sparse regularizer, we consider SVM, where the loss function can be either the hinge loss (a non-smooth function) $\grave{}(zy) = \max(0; 1\ \ zy)$ or the squared hinge loss (a smooth function) $\grave{}(zy) = \max(0; 1\ \ zy)^2$. We first consider the hinge loss, where $\grave{}_i(\ _i) = \ _i y_i$ for $\ _i y_i\ 2\ [\ \ 1; 0]$. Then the new dual problem is equivalent to

$$\max_{y2[\ \ 1;0]^n} \frac{1}{n}\sum_{i=1}^{n} \ _i y_i \quad \frac{1}{2\ n^2}\ ^T\widehat{X}^\top\widehat{X} \quad \frac{1}{n}k\ \ k_1$$

Using variable transformation $\ _i y_i \ \ !\ \ _i$, the above problem is equivalent to

$$\max_{2[0;1]^n} \frac{1}{n}\sum_{i=1}^{n} \ _i(1 \quad\ ) \quad \frac{1}{2\ n^2}(\quad y)^T\widehat{X}^\top\widehat{X}(\quad y)$$

Changing into the primal form, we have

$$\max_{u2\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^{n} \grave{}_1^{\ *}(u^\top\hat{x}_i y_i) + \frac{}{2}kuk_2^2 \tag{6}$$

where $\grave{}^*(z) = \max(0; \quad z)$ is a max-margin loss with margin given by . It can be understood that adding the $\grave{}_1$ regularization in the reduced problem of SVM is equivalent to using a max-margin loss with a smaller margin, which is intuitive because examples become difficult to separate after dimensionality reduction and is consistent with several previous studies that the margin is reduced in the reduced feature space (Blum, 2005; Shi et al., 2012). Similarly for squared hinge loss, the equivalent primal problem is

$$\max_{u2\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^{n} \grave{}_1^{\ 2}(u^\top\hat{x}_i y_i) + \frac{}{2}kuk_2^2 \tag{7}$$

where $\grave{}^2(z) = \max(0; \quad z)^2$.

Although adding a dual-sparse regularizer is intuitive and can be motivated from previous results, we emphasize that the proposed dual-sparse formulation provides a new perspective and bounding the dual recovery error $k\tilde{\ } \quad k$ is a non-trivial task, which is a major contribution of this paper. We first state our main result in Theorem 1 for smooth loss functions.

Theorem 1. *Let $\tilde{\ }$ be the optimal dual solution to (5). Assume is s-sparse with the support set given by S. If $\frac{2}{n}k(X^\top X \quad \widehat{X}^\top\widehat{X})\ \ k_1$, then we have*

$$k[\tilde{\ }]_{S^c}k_1 \quad 3k[\tilde{\ }]_S \quad [\ \ ]_S k_1 \tag{8}$$

Furthermore, if $\grave{}(z)$ is a L-smooth loss function [2], we have

$$k\tilde{\ } \quad k_2 \quad 3\ L\sqrt{s};\ \ k\tilde{\ } \quad k_1 \quad 12\ Ls \tag{9}$$

$$k[\tilde{\ }]_S \quad [\ \ ]_S k_1 \quad 3\ Ls;\ \ k[\tilde{\ }]_{S^c}k_1 \quad 9\ Ls \tag{10}$$

where S^c is the complement of S, and $[\ \ ]_S$ is a vector that only contains the elements of in the set S.

Remark 1: The proof is presented at the end of Section 4. It can be seen that the dual recovery error is proportional to the value of which is dependent on $k(X^\top X \quad \widehat{X}^\top\widehat{X})\ \ k_1$, *which we can bound without using any assumption about the data matrix or the optimal dual variable* . In contrast, previous bounds (Zhang et al., 2013; 2014; Paul et al., 2013) depend on $kX^\top X \quad \widehat{X}^\top\widehat{X}k_2$, which requires the low rank assumption on X. In next section, we provide an upper bound of $\frac{1}{n}k(X^\top X \quad \widehat{X}^\top\widehat{X})\ \ k_1$ that will allow us to understand how the reduced dimensionality m affects the recovery error. Essentially, the results indicate that for random projection, randomized Hadamard transform and random hashing, $\frac{1}{n}k(X^\top X \quad \widehat{X}^\top\widehat{X})\ \ k_1 \quad O(\sqrt{\frac{\log(n=\ )}{m}})kw\ k_2$ with a high probability $1 \quad$, and thus the recovery error will be scaled as $\sqrt{1=m}$ in terms of m - the same order of recovery error as in (Zhang et al., 2013; 2014) that assumes low rank of the data matrix.

Remark 2: We would like to make a connection with LASSO for sparse signal recovery. In sparse signal recovery under noise measurements $f = Uw + e$, where e denotes the noise in measurements, if a LASSO

[2] A function is L-smooth if its gradient is L-Lipschitz continuous.

$\min_{\mathbf{w}} \frac{1}{2}kU\mathrm{w} \quad \mathrm{f}k_2^2 + \quad k\mathrm{w}k_1$ is solved for the solution, then the regularization parameter is required to be larger than the quantity $kU^{>}\mathrm{e}k_1$ that depends on the noise in order to have an accurate recovery (Eldar & Kutyniok, 2012). Similarly in our formulation, the added $`_1$ regularization $k \ k_1$ is to counteract the noise in $\widehat{X}\widehat{X}^{>}$ as compared with $XX^{>}$ and the value of is dependent on the noise.

To present the theoretical result on the non-smooth loss functions, we need to introduce restricted eigen-value conditions similar to those used in the sparse recovery analysis for LASSO (Bickel et al., 2009; Xiao & Zhang, 2013). In particular, we introduce the following definition of restricted eigen-value condition.

Definition 2. *Given an integer $s > 0$, we define*

$$K_{n;s} = f \quad 2 \mathbb{R}^n : k \ k_2 \quad 1; k \ k_1 \quad \overline{p}\overline{s}g:$$

We say that X satisfies the restricted eigenvalue condition at sparsity level s if there exist positive constants $^{+}_{s}$ and $_s$ such that

$$^{+}_{s} = \sup_{2K_{n;s}} \frac{^{>}X^{>}X}{n}; \qquad _s = \inf_{2K_{n;s}} \frac{^{>}X^{>}X}{n}:$$

We also define another quantity that measures the restricted eigen-value of $X^{>}X \quad \widehat{X}^{>}\widehat{X}$, namely

$$_s = \sup_{2K_{n;s}} \frac{j \ ^{>}(X^{>}X \quad \widehat{X}^{>}\widehat{X}) \ j}{n}: \tag{11}$$

Theorem 3. *Let $\tilde{}$ be the optimal dual solution to (5). Assume is s-sparse with the support set given by S. If $\frac{2}{n}k(X^{>}X \quad \widehat{X}^{>}\widehat{X}) \quad k_1$, then we have*

$$k[\tilde{} \ ]_{S^c}k_1 \quad 3k[\tilde{} \ ]_S \quad [\ ]_Sk_1$$

Assume the data matrix X satisfies the restricted eigenvalue condition at sparsity level $16s$ and $_{16s} < _{16s}$, we have

$$k\tilde{} \quad k_2 \quad \frac{3}{2(\ _{16s} \quad _{16s})} \ \overline{p}\overline{s}$$

$$k\tilde{} \quad k_1 \quad \frac{6}{(\ _{16s} \quad _{16s})} \ s$$

Remark 3: The proof appears in the full version of the paper [3]. Compared to smooth loss functions, the conditions that guarantee a small recovery for non-smooth loss functions are more restricted. In next section, we will provide a bound on $_{16s}$ to further understand the condition of $_{16s} \quad _{16s}$, which essentially implies that m $\Omega\left(\left(\frac{^{+}_{16s}}{_{16s}}\right)^2 s\log(n/s)\right)$.

Last but not least, we provide a theoretical result on the recovery error for the nearly sparse optimal dual variable . We state the result for smooth loss functions. To quantify the near sparsity, we let $^s \ 2 \mathbb{R}^n$ denote a vector that zeros all entries in except for the top-s elements in magnitude and assume ^{s} satisfies the following condition:

$$\left\| r\grave{} \ (\ ^s) + \frac{1}{n}X^{>}X \ ^s \right\|_1 \tag{12}$$

where $r\grave{} \ (\ ) = (r\grave{}_1(\ _1); : : : ; r\grave{}_n(\ _n))^{>}$. The above condition can be considered as a sub-optimality condition (Boyd & Vandenberghe, 2004) of ^{s} measured in the infinite norm. For the optimal solution , we have $r\grave{} \ (\ ) + \frac{1}{n}X^{>}X \quad = 0$.

Theorem 4. *Let $\tilde{}$ be the optimal dual solution to (5). Assume is nearly s-sparse such that (12) holds with the support set of ^{s} given by S. If $\frac{2}{n}k(X^{>}X \quad \widehat{X}^{>}\widehat{X}) \quad k_1 + 2$, then we have*

$$k[\tilde{} \ ]_{S^c}k_1 \quad 3k[\tilde{} \ ]_S \quad [\ ]_Sk_1$$

Furthermore, if $`(z)$ is a L-smooth loss function, we have

$$k\tilde{} \quad ^sk_2 \quad 3 \ L\overline{p}\overline{s}; \quad k\tilde{} \quad ^sk_1 \quad 12 \ Ls \tag{13}$$

$$k[\tilde{} \ ]_S \quad [\ ]_Sk_1 \quad 3 \ Ls; \quad k[\tilde{} \ ]_{S^c}k_1 \quad 9 \ Ls \tag{14}$$

Remark 4: The proof appears in the full version of the paper. Compared to Theorem 1 for exactly sparse optimal dual solution, the dual recovery error bound for nearly sparse optimal dual solution is increased by $6L\overline{p}\overline{s}$ for $`_2$ norm and by $24Ls$ for $`_1$ norm.

Finally, we note that with the recovery error bound for the dual solution, we can easily derive an error bound for the primal solution $\tilde{\mathrm{w}} = \frac{1}{n}X\tilde{}$. Below we present a theorem for smooth loss functions. One can easily extend the result to non-smooth loss functions.

Theorem 5. *Let $\tilde{\mathrm{w}}$ be the recovered primal solution using $\tilde{}$ the optimal dual solution to (5). Assume is s-sparse and $`(z)$ is a L-smooth loss function. If $\frac{2}{n}k(X^{>}X \quad \widehat{X}^{>}\widehat{X}) \quad k_1$ then we have*

$$k\tilde{\mathrm{w}} \quad \mathrm{w} \ k_2 \quad \frac{1}{n}3L \ \overline{p}\overline{s}$$

where $_1$ is the maximum singular value of X. Furthermore if $\frac{1}{n}X^{>}X$ has a restricted eigen-value $^{+}_{16s}$ at sparsity level $16s$, then

$$k\tilde{\mathrm{w}} \quad \mathrm{w} \ k_2 \quad \frac{\sqrt{^{+}_{16s}}}{\overline{p}\overline{n}}3L \ \overline{p}\overline{s}$$

Remark 5: Since $^{+}_{16s}$ is always less than $^2_1/n$, the second result if the restricted eigen-value condition holds is always better than the first result. With the bound of as revealed later, we can see that the error of $\tilde{\mathrm{w}}$ scales as $O(\sqrt{\frac{s}{m}}k\mathrm{w} \ k_2)$ in terms of sparsity s of , the reduced dimensionality m and the magnitude of w. A similar order of error bound was established in (Zhang et al., 2014) assuming w is s-sparse and X is approximately low rank. In contrast, we do not assume X is approximately low rank.

4. Analysis

In this section, we first provide upper bound analysis of $\frac{2}{n}k(X^{>}X \quad \widehat{X}^{>}\widehat{X}) \quad k_1$ and $_s$, and then present the

[3] http://arxiv.org/abs/1504.03991

proof of Theorem 1 for smooth loss functions. To facilitate our analysis, we define

$$\Delta = \frac{1}{n}(\widehat{X}^\top \widehat{X} \quad X^\top X)$$

4.1. Bounding $k\Delta k_1$

A critical condition in both Theorem 1 and Theorem 3 is $> k\Delta k_1$. In order to reveal the theoretical value of and its implication for various randomized reduction methods, we need to bound $k\Delta k_1$. We first provide a general analysis and then study its implication for various randomized reduction methods separately. The analysis is based on the following assumption, which essentially is indicated by Johnson-Lindenstrauss (JL)-type lemmas.

Assumption 1 (A1). *Let $A(\mathsf{x}) = A\mathsf{x}$ be a linear projection operator where $A \in \mathbb{R}^{m \quad d}$ such that for any given $\mathsf{x} \in \mathbb{R}^d$ with a high probability $1 \quad$, we have*

$$\left|kA\mathsf{x}k_2^2 \quad k\mathsf{x}k_2^2\right| \quad {}_{A;}\, k\mathsf{x}k_2^2$$

where ${}_{A;}$ depends on m, and possibly d.

With this assumption, we have the following theorem regarding the upper bound of $k\Delta k_1$.

Theorem 6. *Suppose $A \in \mathbb{R}^{m \quad d}$ satisfies Assumption* **A**, *then with a high probability $1 \quad 2$ we have*

$$k\Delta k_1 \quad R k\mathsf{w}\, k_2\, {}_{A;\, =n}$$

where $R = \max_i k\mathsf{x}_i k_2$.

Proof.

$$\frac{1}{n}(\widehat{X}^\top \widehat{X} \quad X^\top X) \quad = \frac{1}{n}(X^\top A^\top A X \quad X^\top X)$$

$$= \frac{1}{n} X^\top (A^\top A \quad I) X \quad = X^\top (I \quad A^\top A)\mathsf{w}$$

where we use the fact $\mathsf{w} \quad = \quad \frac{1}{n} X \quad$. Then

$$\frac{1}{n}[(\widehat{X}^\top \widehat{X} \quad X^\top X) \quad]_i = \mathsf{x}_i^\top (I \quad A^\top A)\mathsf{w}$$

Therefore in order to bound $k\Delta k_1$, we need to bound $\mathsf{x}_i^\top (I \quad A^\top A)\mathsf{w}$ for all $i \in [n]$. We first bound for individual i and then apply the union bound. Let $\widetilde{\mathsf{x}}_i$ and $\widetilde{\mathsf{w}}$ be normalized version of x_i and w, i.e., $\widetilde{\mathsf{x}}_i = \mathsf{x}_i / k\mathsf{x}_i k_2$ and $\widetilde{\mathsf{w}} = \mathsf{w} / k\mathsf{w}\, k_2$. Suppose Assumption **A** is satisfied, then with a probability $1 \quad$,

$$\widetilde{\mathsf{x}}_i^\top A^\top A\widetilde{\mathsf{w}} \quad \widetilde{\mathsf{x}}_i^\top \widetilde{\mathsf{w}} \;=\; \frac{kA(\widetilde{\mathsf{x}}_i + \widetilde{\mathsf{w}}_i)k_2^2 \quad kA(\widetilde{\mathsf{x}}_i \quad \widetilde{\mathsf{w}})k_2^2}{4}$$

$$\widetilde{\mathsf{x}}_i^\top \widetilde{\mathsf{w}} \quad \frac{{}_{A;}}{2}(k\widetilde{\mathsf{x}}_i k_2^2 + k\widetilde{\mathsf{w}}\, k_2^2) \quad {}_{A;}$$

Similarly with a probability $1 \quad$,

$$\widetilde{\mathsf{x}}_i^\top A^\top A\widetilde{\mathsf{w}} \quad \widetilde{\mathsf{x}}_i^\top \widetilde{\mathsf{w}} \;=\; \frac{kA(\widetilde{\mathsf{x}}_i + \widetilde{\mathsf{w}})k_2^2 \quad kA(\widetilde{\mathsf{x}}_i \quad \widetilde{\mathsf{w}})k_2^2}{4}$$

$$\widetilde{\mathsf{x}}_i^\top \widetilde{\mathsf{w}} \quad \frac{{}_{A;}}{2}(k\widetilde{\mathsf{x}}\, k_2^2 + k\widetilde{\mathsf{w}}\, k_2^2) \quad {}_{A;}$$

Therefore with a probability $1 \quad 2$, we have

$$j\mathsf{x}_i^\top A^\top A\mathsf{w} \quad \mathsf{x}_i^\top \mathsf{w}\, j$$

$$k\mathsf{x}_i k_2 k\mathsf{w}\, k_2 j\widetilde{\mathsf{x}}_i^\top A^\top A\widetilde{\mathsf{w}} \quad \widetilde{\mathsf{x}}^\top \widetilde{\mathsf{w}}\, j \quad k\mathsf{x}_i k_2 k\mathsf{w}\, k_2\, {}_{A;}$$

Then applying union bound, we complete the proof.

Next, we discuss four classes of randomized reduction operators, namely random projection, randomized Hadamard transform, random hashing and random sampling, and study the corresponding ${}_{A;}$ and their implications for the recovery error.

Random Projection. Random projection has been employed widely for dimension reduction. The projection operator A is usually sampled from sub-Gaussian distributions with mean 0 and variance $1/m$, e.g., (i) Gaussian distribution: $A_{ij} \quad \mathcal{N}(0; 1/m)$, (ii) Rademacher distribution: $\Pr(A_{ij} = \quad 1/\sqrt{m}) = 0.5$, (iii) discrete distribution: $\Pr(A_{ij} = \quad \sqrt{3/m}) = 1/6$ and $\Pr(A_{ij} = 0) = 2/3$. The last two distributions for dimensionality reduction were proposed and analyzed in (Achlioptas, 2003). The following lemma is the general JL-type lemma for A with sub-Gaussian entries, which reveals the value of ${}_{A;}$ in Assumption **A**.

Lemma 1. *(Nelson) Let $A \in \mathbb{R}^{m \quad d}$ be a random matrix with subGaussian entries of mean 0 and variance $1/m$. For any given x with a probability $1 \quad$, we have*

$$\left|kA\mathsf{x}k_2^2 \quad k\mathsf{x}k_2^2\right| \quad c\sqrt{\frac{\log(1/ \quad)}{m}}\, k\mathsf{x}k_2^2$$

where c is some small universal constant.

Randomized Hadamard Transform. Randomized Hadamard transform was introduced to speed-up random projection, reducing the computational time [4] of random projection from $O(dm)$ to $O(d \log d)$ or even $O(d \log m)$. The projection matrix A is of the form $A = PHD$, where

- $D \in \mathbb{R}^{d \quad d}$ is a diagonal matrix with $D_{ii} = \quad 1$ with equal probabilities.
- H is the $d \quad d$ Hadamard matrix (assuming d is a power of 2), scaled by $1/\sqrt{d}$.
- $P \in \mathbb{R}^{m \quad d}$ is typically a sparse matrix that facilities computing $P\mathsf{x}$. Several choices of P are possible (Nelson; Ailon & Chazelle, 2009; Tropp, 2011). Below we provide a JL-type lemma for a randomized Hadamard transform with $P \in \mathbb{R}^{m \quad d}$ that samples m coordinates from $\sqrt{\frac{d}{m}} HD\mathsf{x}$ with replacement.

Lemma 2. *(Nelson) Let $A = \sqrt{\frac{d}{m}} PHD \in \mathbb{R}^{m \quad d}$ be a randomized Hadamard transform with P being a random sampling matrix. For any given x with a probability $1 \quad$, we have*

$$\left|kA\mathsf{x}k_2^2 \quad k\mathsf{x}k_2^2\right| \quad c\sqrt{\frac{\log(1/ \quad)\log(d/ \quad)}{m}}\, k\mathsf{x}k_2^2$$

where c is some small universal constant.

Remark 6: Compared to random projection, there is an additional $\sqrt{\log(d/ \quad)}$ factor in ${}_{A;}$. However, it can

[4] refers to the running time of computing $A\mathsf{x}$.

be removed by applying an additional random projection. In particular, if we let $A = \sqrt{\frac{d}{m}} P' P H D \in \mathbb{R}^{m \times d}$, where $P \in \mathbb{R}^{t \times d}$ is a random sampling matrix with $t = m \log(d/\ )$ and $P' \in \mathbb{R}^{m \times t}$ is a random projection matrix that satisfies Lemma 1, then we have the same order of $_{A;}$. Please refer to (Nelson) for more details.

Random Hashing. Another line of work to speed-up random projection is random hashing which makes the projection matrix A

rive at the following upper bound for $_s$.

Theorem 7. *With a probability* 1 *, we have*

$$_s \quad O\left(\ ^{+}_{s}\sqrt{\frac{(\log(1=\ )+s\log(n=s))}{m}} \right)$$

Remark 9: With above result, we can further understand the condition $_{16s}$ $_{16s}$, which amounts to

$$O\left(\ ^{+}_{16s}\sqrt{\frac{(\log(1=\ )+s\log(n=s))}{m}} \right) \quad _{16s};$$

i.e., m $\Omega(\ ^{2}_{16s}(\log(1=\ )+s\log(n=s)))$ where $_{16s} = \ ^{+}_{16s}= \ _{16s}$ is the restricted condition number of the data matrix.

4.3. Proof Sketch of Theorem 1

We present a proof sketch of Theorem 1. Due to limitation of space, other proofs are provided in the supplement. Let $\widehat{F}(\ )$ be defined as

$$\widehat{F}(\ ) = \frac{1}{n}\sum_{i=1}^{n} `_i(\ _i) + \frac{1}{2\ n^2}\ ^{\intercal}\widehat{X}^{>}\widehat{X}\ \ + \frac{}{n}k\ \ k_1$$

Since $\tilde{\ } = \arg\min \widehat{F}(\ )$ therefore for any $g\ \ 2\ @k\ \ k_1$

$$\begin{aligned} 0 \quad & \hat{F}(\tilde{\ }) \quad \hat{F}(\ \ ) \\ & (\tilde{\ } \quad)^{>}\left(\frac{1}{n}r`(\ \ ) + \frac{1}{n^2}\widehat{X}^{>}\widehat{X}\ \ \right) \\ & + \frac{}{n}(\tilde{\ } \quad)^{>}g\ + \frac{1}{2nL}k\tilde{\ } \quad k_2^2 \end{aligned}$$

where we used the strong convexity of $`_i$ and its strong convexity modulus $1=L$. By the optimality condition of , we can have

$$0 \quad (\quad \tilde{\ })^{>}\left(\frac{1}{n}r`(\ \ ) + \frac{1}{n^2}X^{>}X\ \ \right) \qquad (16)$$

Combining the above two inequalities we have

$$0 \quad (\tilde{\ } \quad)^{>}\frac{1}{n}\Delta + \frac{}{n}(\tilde{\ } \quad)^{>}g\ + \frac{1}{2nL}k\tilde{\ } \quad k_2^2$$

Since the above inequality holds for any $g\ \ 2\ @k\ \ k_1$, if we choose $[g\ ]_i = sign([\tilde{\ }\ ]_i); i\ 2\ S^c$, then we have

$$(\tilde{\ } \quad)^{>}g \quad k[\tilde{\ }\ ]_S \quad [\ \ ]_S k_1 + k[\tilde{\ }\ ]_{S^c}k_1$$

Combining the above inequalities leads to

$$\begin{aligned} (\ + k\Delta k_1)k[\tilde{\ }\ ]_S \quad [\ \ ]_S k_1 \quad & (\quad k\Delta k_1)k[\tilde{\ }\ ]_{S^c}k_1 \\ & + \frac{1}{2L}k\tilde{\ } \quad k_2^2 \end{aligned}$$

Assuming $2k\Delta k_1$, we have

$$\begin{aligned} & k\tilde{\ } \quad k_2^2 \quad 3\ Lk[\tilde{\ }\ ]_S \quad [\ \ ]_S k_1 \\ & k[\tilde{\ }\ ]_{S^c}k_1 \quad 3k[\tilde{\ }\ ]_S \quad [\ \ ]_S k_1 \end{aligned} \qquad (17)$$

Therefore,

$$k[\tilde{\ } \quad]_S k_1^2 \quad sk\tilde{\ } \quad k_2^2 \quad 3\ Lsk[\tilde{\ }\ ]_S \quad [\ \ ]_S k_1$$

leading to the result

$$k[\tilde{\ }\ ]_S \quad [\ \ ]_S k_1 \quad 3\ Ls:$$

Combing this inequality with inequalities in (17) we have

$$k[\tilde{\ }\ ]_{S^c}k_1 \quad 9\ Ls; \quad k\tilde{\ } \quad k_2 \quad 3\ L\sqrt{s}:$$

5. Numerical Experiments

In this section, we provide a case study in support of DSRR and the theoretical analysis, and a demonstration of the application of DSRR to distributed optimization.

A case study on text classification. We use the RCV1-binary data (Lewis et al., 2004) to conduct a case study. The data contains $697; 641$ documents and $47; 236$ features. We use a splitting $677; 399=20; 242$ for training and testing. The feature vectors were normalized such that the $`_2$ norm is equal to 1. We only report the results using random hashing since it is the most efficient, while other randomized reduction methods (except for random sampling) have similar performance. For the loss function, we use both the squared hinge loss (smooth) and the hinge loss (non-smooth). We aim to examine two questions related to our analysis and motivation (i) how does the value of affect the recovery error? (ii) how does the number of samples m affect the recovery error?

We vary the value of among $0; 0{:}1; 0{:}2; : : : ; 0{:}9$, the value of m among $1024; 2048; 4096; 8192$, and the value of among $0{:}001; 0{:}00001$. Note that $= 0$ corresponds to the randomized reduction approach without the sparse regularizer. The results averaged over 5 random trials are shown in Figure 1 for the squared hinge loss and in Figure 2 for the hinge loss. We first analyze the results in Figure 1. We can observe that when increases the ratio of $\frac{k[\tilde{\ }_*]_{S^c}k_1}{k[\tilde{\ }_*]_S \quad [\ \ _*]_S k_1}$ decreases indicating that the magnitude of dual variables for the original non-support vectors decreases. This is intuitive and consistent with our motivation. The recovery error of the dual solution (middle) first decreases and then increases. This can be partially explained by the theoretical result in Theorem 1. When the value of becomes larger than a certain threshold making $> k\Delta k_1$ hold, then Theorem 1 implies that a larger will lead to a larger error. On the other hand, when is less than the threshold, the dual recovery error will decrease as increases. In addition, the figures exhibit that the thresholds for larger m are smaller which is consistent with our analysis of $k\Delta k_1 = O(\sqrt{1=m})$. The difference between $= 0{:}001$ and $= 0{:}00001$ is because that smaller will lead to larger kw k_2. In terms of the hinge loss, we observe similar trends, however, the recovery is much more difficult than that for squared hinge loss especially when the value of is small.

An application to distributed learning. Although in some cases the solution learned in the reduced space can provide sufficiently good performance, it usually performs worse than the optimal solution that solves the original problem and sometimes the performance gap between them can not be ignored as seen in following experiments. To address this issue, we combine the benefits of distributed learning and the proposed randomized reduction methods for solv-

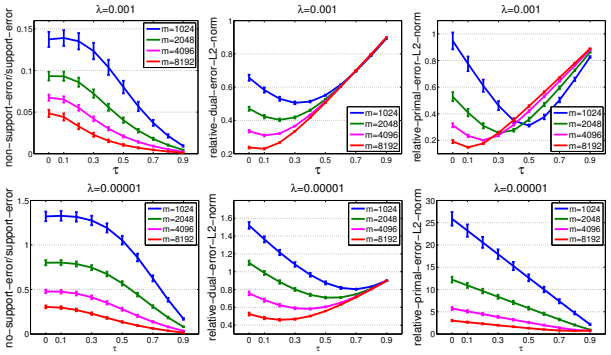

Figure 1. Recovery error for squared hinge loss. From left to right: $\frac{\|[\bar{\alpha}_*]_{\mathcal{S}^c}\|_1}{\|[\bar{\alpha}_*]_{\mathcal{S}}-[\alpha_*]_{\mathcal{S}}\|_1}$ vs , $\frac{\|\bar{\alpha}_*-\alpha_*\|_2}{\|\alpha_*\|_2}$ vs , and $\frac{\|\bar{\mathbf{w}}_*-\mathbf{w}_*\|_2}{\|\mathbf{w}_*\|_2}$ vs .

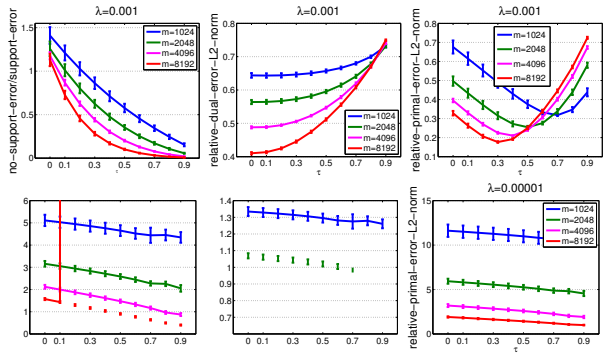

Figure 2. Same curves as above but for non-smooth hinge loss.

ing big data problems. When data is too large and sits on multiple machines, distributed learning can be employed to solve the optimization problem. In distributed learning, individual machines iteratively solve sub-problems associated with the subset of data on them and communicate some global variables (e.g., the primal solution w *2* $\mathbb{R}^d$) among them. When the dimensionality *d* is very large, the total communication cost could be very high. To reduce the total communication cost, we propose to first solve the reduced data problem and then use the found solution as the initial solution to the distributed learning for the original data.

Below, we demonstrate the effectiveness of DSRR for the recently proposed distributed stochastic dual coordinate ascent (DisDCA) algorithm (Yang, 2013). The procedure is (1) reduce original high-dimensional data to very low dimensional space on individual machines; (2) use DisDCA to solve the reduced problem; (3) use the optimal dual solution to the reduce problem as an initial solution to DisDCA for solving the original problem. We record the running time for randomized reduction in step 1 and optimization of the reduced problem in step 2, and the optimization of the original problem in step 3. We compare the performance of four methods (i) the **DSRR** method that uses the model of the reduced problem solved by DisDCA to make predictions, (ii) the method that uses the recovered model

Table 1. Statistics of datasets

Name	#Training	#Testing	#Features	#Nodes
RCV1	677,399	20,242	47, 236	5
KDD	8,407,752	748,401	29,890,095	10

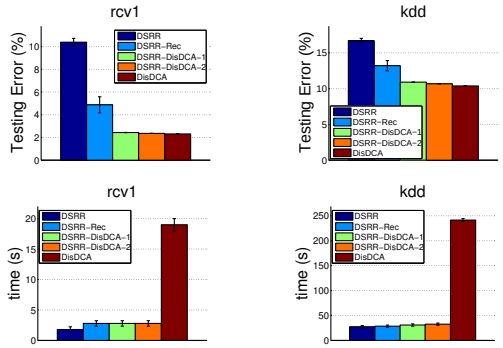

Figure 3. Top: Testing error for different methods. Bottom: Training time for different methods. The value of $= 10^{-5}$ and the value of $= 0.9$. The high-dimensional features are reduced to $m = 1024$-dimensional space using random hashing. The loss function is the squared hinge loss.

in the original space, referred to as **DSRR-Rec**; (iii) the method that uses the dual solution to the reduced problem as an initial solution of DisDCA and runs it for the original problem with $k = 1$ or 2 communications (the number of updates before each communication is set to the number of examples in each machine), referred to as **DSRR-DisDCA-*k***; and (iv) the distributed method that directly solves the original problem by **DisDCA**. For DisDCA to solve the original problem, we stop running when its performance on the testing data does not improve. Two data sets are used, namely RCV1-binary, KDD 2010 Cup data. For KDD 2010 Cup data, we use the one available on LibSVM data website. The statistics of the two data sets are summarized in Table 1. The results averaged over 5 trials are shown in Figure 3, which exhibit that the performance of DSRR-DisDCA-1/2 is remarkable in the sense that it achieves almost the same performance of directly training on the original data (DisDCA) and uses much less training time. In addition, DSRR-DisDCA performs much better than DSRR and has small computational overhead.

6. Conclusions

In this paper, we have proposed dual-sparse regularized randomized reduction methods for classification. We presented rigorous theoretical analysis of the proposed dual-sparse randomized reduction methods in terms of recovery error under a mild condition that the optimal dual variable is (nearly) sparse for both smooth and non-smooth loss functions, and for various randomized reduction approaches. The numerical experiments validate our theoretical analysis and also demonstrate that the proposed reduction and recovery framework can benefit distributed optimization by providing a good initial solution.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful and insightful comments. T. Yang was supported in part by NSF (IIS-1463988). R. Jin was partially supported by NSF IIS-1251031 and ONR N000141410631.

References

Achlioptas, Dimitris. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of Computer and System Sciences.*, 66:671–687, 2003.

Agarwal, Alekh, Chapelle, Olivier, Dudk, Miroslav, and Langford, John. A reliable effective terascale linear learning system. *CoRR*, 2011.

Ailon, Nir and Chazelle, Bernard. The fast johnson–lindenstrauss transform and approximate nearest neighbors. *SIAM J. Comput.*, 39(1):302–322, 2009.

Balcan, Maria-Florina, Blum, Avrim, and Vempala, Santosh. Kernels as features: On kernels, margins, and low-dimensional mappings. *Machine Learning*, 65:79–94, 2006.

Bartz, Daniel, Hatrick, Kerr, Hesse, Christian W., Müller, Klaus-Robert, and Lemm, Steven. Directional variance adjustment: improving covariance estimates for high-dimensional portfolio optimization. *arXiv:1109.3069*, 2011.

Ben-Hur, Asa, Ong, Cheng Soon, Sonnenburg, Sören, Schölkopf, Bernhard, and Rätsch, Gunnar. Support vector machines and kernels for computational biology. *PLoS Comput Biology*, 4:e1000173, 2008.

Bickel, Peter J., Ritov, Ya'acov, and Tsybakov, Alexandre B. Simultaneous analysis of lasso and dantzig selector. *ANNALS OF STATISTICS*, 37(4), 2009.

Blum, Avrim. Random projection, margins, kernels, and feature-selection. In *Proceedings of the 2005 International Conference on Subspace, Latent Structure and Feature Selection*, volume 3940, pp. 52–68. Springer, 2005.

Boyd, Stephen and Vandenberghe, Lieven. *Convex Optimization*. Cambridge University Press, 2004.

Brank, Janez, Grobelnik, Marko, Milić-Frayling, Natasa, and Mladenić, D. Feature selection using support vector machines. In *Proceedings of the International Conference on Data Mining Methods and Databases for Engineering, Finance, and Other Fields*, pp. 84–89, 2002.

Dasgupta, Anirban, Kumar, Ravi, and Sarlós, Tamás. A sparse johnson–lindenstrauss transform. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pp. 341–350, 2010.

Drineas, Petros, Mahoney, Michael W., and Muthukrishnan, S. Sampling algorithms for l2 regression and applications. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1127–1136, 2006.

Drineas, Petros, Mahoney, Michael W., and Muthukrishnan, S. Relative-error cur matrix decompositions. *SIAM Journal Matrix Analysis Applications*, 30:844–881, 2008.

Drineas, Petros, Mahoney, Michael W., Muthukrishnan, S., and Sarlós, Tamàs. Faster least squares approximation. *Numerische Mathematik*, 117(2):219–249, February 2011.

Eldar, Yonina C. and Kutyniok, Gitta. *Compressed Sensing: Theory and Applications*. Compressed Sensing: Theory and Applications. Cambridge University Press, 2012. ISBN 9781107005587.

Goldberger, Jacob, Roweis, Sam, Hinton, Geoffrey, and Salakhutdinov, Ruslan. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 513–520, 2005.

Guyon, Isabelle, Weston, Jason, Barnhill, Stephen, and Vapnik, Vladimir. Gene selection for cancer classification using support vector machines. *Machine Learning (ML)*, 46:389–422, 2002.

Jaggi, Martin, Smith, Virginia, Takác, Martin, Terhorst, Jonathan, Krishnan, Sanjay, Hofmann, Thomas, and Jordan, Michael I. Communication-efficient distributed dual coordinate ascent. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 3068–3076, 2014.

Jostins, Luke and Barrett, Jeffrey C. Genetic risk prediction in complex disease. *Human Molecular Genetics*, 20 (R2):R182–R188, 2011.

Kane, Daniel M. and Nelson, Jelani. Sparser johnson-lindenstrauss transforms. *Journal of the ACM*, 61:4:1–4:23, 2014.

Kang, J., Kugathasan S. Georges M. Zhao H. and Cho, J.H. Improved risk prediction for crohn's disease with a multi-locus approach. *Human Molecular Genetics*, 20: 2435–2442, 2011.

Lewis, David D., Yang, Yiming, Rose, Tony G., and Li, Fan. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research (JMLR)*, 5:361–397, 2004.

Li, Mu, Andersen, David G, Smola, Alex J, and Yu, Kai. Communication efficient distributed machine learning with the parameter server. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 19–27. 2014.

Ma, Ping, Mahoney, Michael W., and Yu, Bin. A statistical perspective on algorithmic leveraging. In *Proceedings of the 31th International Conference on Machine Learning (ICML)*, pp. 91–99, 2014.

Mahoney, Michael W. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

Mahoney, Michael W. and Drineas, Petros. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.

Mitchell, Tom M., Hutchinson, Rebecca, Niculescu, Radu S., Pereira, Francisco, Wang, Xuerui, Just, Marcel, and Newman, Sharlene. Learning to decode cognitive states from brain images. *Machine Learning*, 57(1-2): 145–175, 2004.

Nelson, Jelani. Johnson-lindenstrauss notes. Technical report, MIT.

Paul, Saurabh, Boutsidis, Christos, Magdon-Ismail, Malik, and Drineas, Petros. Random projections for support vector machines. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 498–506, 2013.

Plan, Yaniv and Vershynin, Roman. One-bit compressed sensing by linear programming. *CoRR*, abs/1109.4299, 2011.

Rätsch, G., Sonnenburg, S., and Schölkopf, B. RASE: recognition of alternatively spliced exons in c.elegans. *Bioinformatics*, 21:i369–i377, 2005.

Rtsch, Gunnar, Sonnenburg, Sren, and Schlkopf, Bernhard. Rase: recognition of alternatively spliced exons in c.elegans. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology (ISMB) (Supplement of Bioinformatics)*, pp. 369–377, 2005.

Sánchez, Jorge, Perronnin, Florent, Mensink, Thomas, and Verbeek, Jakob J. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245, 2013.

Shi, Qinfeng, Petterson, James, Dror, Gideon, Langford, John, Smola, Alex, and Vishwanathan, S.V.N. Hash kernels for structured data. *Journal of Machine Learning Research (JMLR)*, 10:2615–2637, 2009a.

Shi, Qinfeng, Petterson, James, Dror, Gideon, Langford, John, Smola, Alexander J., Strehl, Alexander L., and Vishwanathan, Vishy. Hash kernels. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 496–503, 2009b.

Shi, Qinfeng, Shen, Chunhua, Hill, Rhys, and van den Hengel, Anton. Is margin preserved after random projection? In *Proceedings of the International Conference on Machine Learning (ICML)*, 2012.

Simianer, Patrick, Riezler, Stefan, and Dyer, Chris. Joint feature selection in distributed stochastic learning for large-scale discriminative training in smt. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 11–21, 2012.

Sonnenburg, Sren and Franc, Vojtech. Coffin: A computational framework for linear svms. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 999–1006, 2010.

Sonnenburg, Sren, Schweikert, Gabriele, Philips, Petra, Behr, Jonas, and Rtsch, Gunnar. Accurate splice site prediction using support vector machines. *BMC Bioinformatics*, 8, 2007.

Tropp, Joel A. Improved analysis of the subsampled randomized hadamard transform. *Advances in Adaptive Data Analysis*, 3(1-2):115–126, 2011.

Weinberger, Kilian Q., Dasgupta, Anirban, Langford, John, Smola, Alexander J., and Attenberg, Josh. Feature hashing for large scale multitask learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1113–1120, 2009.

Xiao, Lin and Zhang, Tong. A proximal-gradient homotopy method for the sparse least-squares problem. *SIAM Journal on Optimization*, 23(2):1062–1091, 2013.

Yang, Tianbao. Trading computation for communication: Distributed stochastic dual coordinate ascent. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 629–637, 2013.

Zhang, Lijun, Mahdavi, Mehrdad, Jin, Rong, Yang, Tianbao, and Zhu, Shenghuo. Recovering the optimal solution by dual random projection. In *Proceedings of Conference of Learning Theory*, pp. 135–157, 2013.

Zhang, Lijun, Mahdavi, Mehrdad, Jin, Rong, Yang, Tianbao, and Zhu, Shenghuo. Random projections for classification: A recovery approach. *IEEE Transactions on Information Theory (IEEE TIT)*, 60(11):7300–7316, 2014.