Machine Learning (2020) 109:899–938
https://doi.org/10.1007/s10994-019-05865-4

High-dimensional model recovery from random sketched data by exploring intrinsic sparsity

Tianbao Yang[1] · Lijun Zhang[2] · Qihang Lin[3] · Shenghuo Zhu[4] · Rong Jin[4]

Received: 28 August 2018 / Revised: 9 August 2019 / Accepted: 11 December 2019 /
Published online: 7 January 2020

Abstract
Learning from large-scale and high-dimensional data still remains a computationally challenging problem, though it has received increasing interest recently. To address this issue, randomized reduction methods have been developed by either reducing the dimensionality or reducing the number of training instances to obtain a small sketch of the original data. In this paper, we focus on recovering a high-dimensional classification/regression model from random sketched data. We propose to exploit the intrinsic sparsity of optimal solutions and develop novel methods by increasing the regularization parameter before the sparse regularizer. In particular, (i) for high-dimensional classification problems, we leverage randomized reduction methods to reduce the dimensionality of data and solve a dual formulation on the random sketched data with an introduced sparse regularizer on the dual solution; (ii) for high-dimensional sparse least-squares regression problems, we employ randomized reduction methods to reduce the scale of data and solve a formulation on the random sketched data with an increased regularization parameter before the sparse regularizer. For both classes of problems, by exploiting the intrinsic sparsity of the optimal dual solution or the optimal primal solution we provide formal theoretical guarantee on the recovery error of learned models in comparison with the optimal models that are learned from the original data. Compared with previous studies on randomized reduction for machine learning, the present work enjoy several advantages: (i) the proposed formulations enjoys intuitive geometric explanations; (ii) the theoretical guarantee does not rely on any stringent assumptions about the original data (e.g., low-rankness of the data matrix or the data are linearly separable); (iii) the theory covers both smooth and non-smooth loss functions for classification; (iv) the analysis is applicable to a broad class of randomized reduction methods as long as the reduction matrices admit the Johnson–Lindenstrauss type of lemma. We also present empirical studies to support the proposed methods and the presented theory.

Keywords Classification · Regression · Large-scale · High dimension · Sparsity · Randomized reduction · JL transform

Editor: Pradeep Ravikumar.

Extended author information available on the last page of the article

1 Introduction

As the scale and dimensionality of data continue to grow in many applications (e.g., bioinformatics, finance, computer vision, medical informatics) (Sánchez et al. 2013; Mitchell et al. 2004; Simianer et al. 2012; Bartz et al. 2011), it becomes critical to develop efficient and effective algorithms to solve big data machine learning problems. Randomized reduction methods for large-scale and high-dimensional (LSHD) data analytics have received a great deal of attention in recent years (Mahoney and Drineas 2009; Shi et al. 2012; Paul et al. 2013; Weinberger et al. 2009; Mahoney 2011). By either reducing the dimensionality (*referred to as feature reduction*) or reducing the number of training instances (*referred to as instance reduction*), the resulting problem has a smaller size of training data that is not only memory-efficient but also computation-efficient. While randomized instance reduction has been studied a lot for least-squares regression (Drineas et al. 2008, 2006, 2011; Ma et al. 2014; Zhou et al. 2007), randomized feature reduction is more popular for linear classification (Blum 2005; Shi et al. 2009a, 2012; Paul et al. 2013; Weinberger et al. 2009) (e.g., random hashing is a noticeable built-in tool in Vowpal Wabbit,[1] a fast learning library, for solving high-dimensional problems.). In this paper, we focus on the applications of randomized reduction methods to high-dimensional classification and sparse least-squares regression problems.

Previous results are limited on theoretical properties of randomized reduction methods for high-dimensional classification problems and sparse least-squares regression problems. In particular, for high-dimensional classification problems, previous results on the preservation of margin (Blum 2005; Balcan et al. 2006; Shi et al. 2012; Paul et al. 2013) and the recovery error of the model (Zhang et al. 2014) rely on strong assumptions about the data. For example, both Paul et al. (2013) and Zhang et al. (2014) assume the data matrix is of low-rank, and Blum (2005), Balcan et al. (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) imposes an additional assumption that the weight vector for classification is sparse. These assumptions are too strong to hold in many real applications. Zhou et al. (2007) studied randomized reduction for ℓ_1 regularized least-squares problem and analyzed the sparsitency (i.e., the recovery of the support set) and the persistency (i.e., the generalization performance). However, their result is *asymptotic*, which only holds when the number of instances approaches infinity, and requires strong assumptions about the data matrix and other parameters.

To address these limitations, we propose to explore the intrinsic sparsity of the problem, i.e., the sparsity of the optimal dual solution for classification problems and the sparsity of the optimal model for sparse regression problems. To leverage the sparsity, we propose novel formulations for both classes of problems by introducing or strengthening the sparse regularizer. In particular, for LSHD classification problems we propose a sparse regularized dual formulation on the dimensionality-reduced data by leveraging the (near) sparsity of the optimal dual solution (i.e., the number of (effective) support vectors is small compared to the total number of examples). For large-scale sparse least-squares regression problems, we analyze a modified formulation on the scale-reduced data by increasing the regularization parameter before the sparse regularizer of the model. We develop novel theoretical analysis on the recovery error of learned high-dimensional models.

Compared with previous works (Blum 2005; Balcan et al. 2006; Shi et al. 2012; Paul et al. 2013; Zhang et al. 2014; Zhou et al. 2007), the presented theoretical results enjoy several

[1] http://hunch.net/~vw/.

advantages: (i) our analysis demands a milder assumption about the data; (ii) our analysis covers both smooth and non-smooth loss functions for classification and both the ℓ_1 regularizer and the elastic net regularizer for sparse regression; (iii) our analysis is applicable to a broad class of randomized reduction methods unlike that most previous works focus on random Gaussian projection; (iv) our results directly provide 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. 2005a; Sonnenburg and Franc 2010; Ratsch et al. 2005b; Sonnenburg et al. 2007; Ben-Hur et al. 2008). To elaborate on the last point, 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 and Barrett 2011; Kang et al. 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).

The remainder of the paper is organized as follows. We review more related work in Sect. 2. We present preliminaries in Sect. 3 and the main results in Sect. 4. We present randomized reduction methods in Sect. 5, and discuss optimization and time complexity in Sect. 6. We present the proofs in Sect. 7. Experimental results are presented in Sect. 9. Finally, we conclude in Sect. 10. Proofs of some technical lemmas are presented in the "Appendix".

2 Related work

2.1 Random projection for high-dimensional learning

Random projection has been employed for addressing the computational challenge of high-dimensional learning problems, including the classification problems (Balcan et al. 2006) and the regression problems (Maillard and Munos 2009). If let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ denote a set of instances, by random projection we can reduce the high-dimensional features into a low dimensional feature space by $\widehat{\mathbf{x}}_i = A\mathbf{x}_i \in \mathbb{R}^m$, where $A \in \mathbb{R}^{m\times d}$ is a random projection matrix. Several works have studied some theoretical properties of classification in the low dimensional space. For example, Paul et al. (2013) considered the following problem and its random sketched (RS) counterpart:

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

$$\text{RS:} \quad \min_{\mathbf{u}\in\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^n \ell(\mathbf{u}^\top \widehat{\mathbf{x}}_i, y_i) + \frac{\lambda}{2}\|\mathbf{u}\|_2^2,$$

where $y_i \in \{1, -1\}$ and $\ell(z, y) = \max(0, 1 - yz)$ is the hinge loss. Paul et al. showed that the margin and minimum enclosing ball in the reduced feature space are preserved to within a small relative error provided that the data matrix $X \in \mathbb{R}^{n\times d}$ is of low-rank. Zhang et al. (2013) studied the problem of recovering the original optimal solution $\mathbf{w}_*$ and proposed a dual recovery approach, i.e., using the learned dual variable in the reduced feature space to recover the model in the original feature space. They analyzed a recovery error under the low-rank assumption of the data matrix. In the extended version Zhang et al. (2014) also considered a case when the optimal solution $\mathbf{w}_*$ is sparse and the data matrix is approximately low rank. They established a recovery error in the order of $O(\sqrt{s/m}\|\mathbf{w}_*\|_2)$, where s is either

the rank of the data matrix or the sparsity of the optimal primal solution. In Blum (2005), Balcan et al. (2006), Shi et al. (2012), the authors have shown that the classification margin in the reduced feature space is approximately preserved provided that the original data is linearly separable by a large margin (with a high probability). However, in practice these conditions for classification problems are usually too strong to hold, i.e., the optimal model $\mathbf{w}_*$ is not necessarily sparse and the data matrix is not necessarily low rank and the data is not necessarily linearly separable.

Recently, Pilanci and Wainwright (2015) studied a random sketched quadratic convex program over an arbitrary convex set, which is closely related to the present work. The original problem and the random sketched problem are given by

$$\mathbf{w}_* = \arg\min_{\mathbf{w}\in\mathcal{C}} f(\mathbf{w}) \triangleq \|X\mathbf{w} - \mathbf{y}\|_2^2, \quad \text{RS: } \widehat{\mathbf{w}}_* = \arg\min_{\mathbf{w}\in\mathcal{C}} \|A(X\mathbf{w} - \mathbf{y})\|_2^2.$$

The framework includes least-squares regression, sparse constrained least-squares regression, a dual formulation of a particular form of SVM, and etc. The established a relative approximation error in terms of the objective function, i.e., $f(\widehat{\mathbf{w}}_*) \leq (1+\delta)^2 f(\mathbf{w}_*)$ by leveraging the Gaussian width of the intersection between the transformed tangent cone and the Euclidean ball. For sparse constrained least-squares regression and a particular form of SVM, they also exploit the sparsity of the optimal solution for developing the Gaussian width. There are some key differences between our work and Pilanci and Wainwright (2015): (i) we focus on regularized formulations instead of constraint formulations; Regularized formulations are attractive because they can be solved with less efforts than the constrained formulations (e.g., the ℓ_1 regularizer can be handled by soft-thresholding with $O(d)$ time complexity with d being the dimensionality of the variable, while projection onto the ℓ_1 constraint needs $O(d\log(d))$ time complexity); (ii) we propose modified formulations on the random sketched data by introducing or strengthening the sparse regularizer, which yields improved recovery results; (iii) we also explore the strong convexity of the objective function to develop stronger theoretical guarantee; (iv) our analysis focuses on the recovery error of high-dimensional models instead of the relative optimization error in terms of the objective function. Notably, a small optimization error in the objective value does not necessarily indicate a small recovery error on the solution.

2.2 Approximate least-squares regression

In numerical linear algebra, one important problem is the over-constrained least-squares problem, i.e., finding a vector $\mathbf{w}_*$ such that the Euclidean norm of the residual error $\|X\mathbf{w} - \mathbf{y}\|_2$ is minimized, where the data matrix $X \in \mathbb{R}^{n\times d}$ has $n \gg d$. The exact solver takes $O(nd^2)$ time complexity. Tremendous studies have been devoted to developing and analyzing randomized algorithms for finding an approximate solution to the above problem in $o(nd^2)$ (Drineas et al. 2006, 2008, 2011; Boutsidis et al. 2009; Mahoney 2011; Halko et al. 2011). These works share the same paradigm by applying an appropriate random matrix $A \in \mathbb{R}^{m\times n}$ to both X and $\mathbf{y}$ and solving the induced subproblem, i.e., $\widehat{\mathbf{w}}_* = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \|A(X\mathbf{w} - \mathbf{y})\|_2$. Relative-error bounds for $\|\mathbf{y} - X\widehat{\mathbf{w}}_*\|_2$ and $\|\mathbf{w}_* - \widehat{\mathbf{w}}_*\|_2$ have been developed. Nevertheless, sparse regularized least-squares regression with random sketched data has not been studied much except for Zhou et al. (2007) and Pilanci and Wainwright (2016). Zhou et al. (2007) focuses on asymptotic analysis of sparsitency (i.e., the recovery of the support set) and the persistency (i.e., the generalization performance). Pilanci and Wainwright (2016) studied the unconstrained and sparse-constrained least-squares problems under sketched data. In particular, they established a lower bound on the statistical error of any estimator based on

the sketched data $(AX, A\mathbf{y})$, which is sub-optimal compared with the estimator that is directly learned from $(X, \mathbf{y})$. Hence, they proposed an iterative Hessian sketch method based on the data $(AX, X^\top \mathbf{y})$ to learn a solution that is statistically optimal. It is worth mentioning that the difference between this work and Pilanci and Wainwright (2016) is that we focus on learning from the sketched data $(AX, A\mathbf{y})$, and the statistical error of the model learned from the sketched data $(AX, A\mathbf{y})$ by our method matches the lower bound established in Pilanci and Wainwright (2016). It indicates that our method is statistically optimal among all methods that learn a sparse solution based on the sketched data $(AX, A\mathbf{y})$.

2.3 Sparse recovery analysis

The LASSO problem has been one of the core problems in statistics and machine learning, which is essentially to learn a high-dimensional sparse vector $\mathbf{u}_* \in \mathbb{R}^d$ from (potentially noise) linear measurements $\mathbf{y} = X\mathbf{u}_* + \xi \in \mathbb{R}^n$. A rich theoretical literature (Tibshirani 1996; Zhao and Yu 2006; Wainwright 2009) describes the consistency, in particular the sign consistency, of various sparse regression techniques. A stringent "irrepresentable condition" has been established to achieve sign consistency. To circumvent the stringent assumption, several studies (Jia and Rohe 2012; Paul et al. 2008) have proposed to precondition the data matrix X and/or the target vector $\mathbf{y}$ by PX and $P\mathbf{y}$ before solving the LASSO problem with a particular preconditioning matrix $P \in \mathbb{R}^{n\times n}$. The oracle inequalities of the solution to LASSO (Bickel et al. 2009) and other sparse estimators (e.g., the Dantzig selector of Candes and Tao 2007) have also been established under restricted eigen-value conditions of the data matrix X and the Gaussian noise assumption of ξ. The focus in these studies is on when the number of measurements n is much less than the number of features, i.e., $n \ll d$. Different from these work, we consider that both n and d are significantly large[2] and design fast algorithms for solving the sparse least-squares problem approximately by using random sketched data $AX \in \mathbb{R}^{m\times d}$ and $A\mathbf{y} \in \mathbb{R}^{m\times 1}$ with $m \ll n$. The analysis is centered on the recovery error of the learned model with respect to the optimal solution to the original problem. For problems that are not strongly convex, we also use the restricted eigen-value conditions of the data matrix to establish the recovery error.

2.4 Randomized reduction by Johnson–Lindenstrauss (JL) transforms

The JL transforms refer to a class of transforms that obey the JL lemma (Johnson and Lindenstrauss 1984), which states that any N points in Euclidean space can be embedded into $O(\epsilon^2 \log N)$ dimensions so that all pairwise Euclidean distances are preserved upto $1 \pm \epsilon$. Since the original Johnson–Lindenstrauss result, many transforms have been designed to satisfy the JL lemma, including Gaussian random matrices (Dasgupta and Gupta 2003), sub-Gaussian random matrices (Achlioptas 2003), randomized Hadamard transform (Ailon and Chazelle 2006), sparse JL transforms by random hashing (Dasgupta et al. 2010; Kane and Nelson 2014). The analysis presented in this work builds upon the JL lemma and therefore our method can enjoy the computational benefits of sparse JL transforms, i.e., less memory and fast computation. More details of these transforms will be given later.

[2] This setting recently receives increasing interest (Yen et al. 2014).

3 Preliminaries

Let $\|\cdot\|_2$ and $\|\cdot\|_1$ denote the Euclidean norm (ℓ_2 norm) and ℓ_1 norm, respectively. Denote by $\partial f(\mathbf{w})$ and $\nabla f(\mathbf{w})$ the subgradient and gradient of a non-smooth function and a smooth function, respectively. A function $f(\mathbf{w}) : \mathbb{R}^d \to \mathbb{R}$ is a G-Lipschitz continuous function w.r.t $\|\cdot\|_2$, if

$$|f(\mathbf{w}_1) - f(\mathbf{w}_2)| \leq G\|\mathbf{w}_1 - \mathbf{w}_2\|_2, \forall \mathbf{w}_1, \mathbf{w}_2 \in \mathbb{R}^d.$$

A convex function $f(\mathbf{w}) : \mathcal{D} \to \mathbb{R}$ is β-strongly convex w.r.t $\|\cdot\|_2$, if for any $\alpha \in [0, 1]$ and $\forall \mathbf{w}_1, \mathbf{w}_2 \in \mathcal{D}$

$$f(\alpha\mathbf{w}_1 + (1-\alpha)\mathbf{w}_2) \leq \alpha f(\mathbf{w}_1) + (1-\alpha) f(\mathbf{w}_2) - \frac{1}{2}\alpha(1-\alpha)\beta\|\mathbf{w}_1 - \mathbf{w}_2\|_2^2,$$

where β is called the strong convexity modulus of f. The strong convexity is also equivalent to

$$f(\mathbf{w}_1) \geq f(\mathbf{w}_2) + \langle \partial f(\mathbf{w}_2), \mathbf{w}_1 - \mathbf{w}_2\rangle + \frac{\beta}{2}\|\mathbf{w}_1 - \mathbf{w}_2\|_2^2, \forall \mathbf{w}_1, \mathbf{w}_2 \in \mathcal{D}.$$

A function $f(\mathbf{w}) : \mathcal{D} \to \mathbb{R}$ is L-smooth w.r.t $\|\cdot\|_2$, if it is differentiable and its gradient is L-Lipschitz continuous, i.e.,

$$\|\nabla f(\mathbf{w}_1) - \nabla f(\mathbf{w}_2)\|_2 \leq L\|\mathbf{w}_1 - \mathbf{w}_2\|_2, \forall \mathbf{w}_1, \mathbf{w}_2 \in \mathcal{D},$$

or equivalently

$$f(\mathbf{w}_1) \leq f(\mathbf{w}_2) + \langle \nabla f(\mathbf{w}_2), \mathbf{w}_1 - \mathbf{w}_2\rangle + \frac{L}{2}\|\mathbf{w}_1 - \mathbf{w}_2\|_2^2, \forall \mathbf{w}_1, \mathbf{w}_2 \in \mathcal{D}.$$

Given a convex function $f(\mathbf{w})$, its convex conjugate function $f^*(\mathbf{u})$ is defined as

$$f^*(\mathbf{u}) = \max_{\mathbf{w}\in\mathcal{D}} \mathbf{w}^\top\mathbf{u} - f(\mathbf{u}).$$

It is known that the convex conjugate of a L-smooth function is $1/L$ strongly convex, and vice versa (Kakade et al. 2009).

We write $f = O(g)$ if there exists a constant $C > 0$ such that $f \leq Cg$, write $f = \Omega(g)$ if there exists a constant $c > 0$ such that $f \geq cg$, and write $f = \Theta(g)$ if there exist constants $0 < c \leq C$ such that $cg \leq f \leq Cg$. A vector $\mathbf{w} \in \mathbb{R}^d$ is said to be s-sparse if it has at most s non-zero entries.

Let $(\mathbf{x}_i, y_i), i = 1, \ldots, n$ denote a set of training examples, where $\mathbf{x}_i \in \mathbb{R}^d$ denotes the feature vector, y_i denotes the label. Assume both n and d are very large. For classification, we consider $y_i \in \{1, -1\}$, and for regression we consider $y_i \in \mathbb{R}$. Let $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^\top = (\bar{\mathbf{x}}_1, \ldots, \bar{\mathbf{x}}_d) \in \mathbb{R}^{n\times d}$ denote the data matrix and $\mathbf{y} = (y_1, \ldots, y_n)^\top$. In this paper, we focus on learning a linear model $\mathbf{w} \in \mathbb{R}^d$ from training examples that makes a prediction by $\mathbf{w}^\top\mathbf{x}$. For classification, the problem of interest can be cast into a regularized minimization:

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

where $\ell(z)$ is a convex loss function suited for classification (e.g., hinge loss $\ell(z) = \max(0, 1 - zy)$ or the squared hinge loss $\ell(z) = \max(0, 1 - z)^2$) and λ is a regularization parameter. Using the conjugate function, we can turn the problem into a dual problem:

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

where $\ell_i^*(\alpha)$ is the convex conjugate function of $\ell(z)$. Given the optimal dual solution α_*, the optimal primal solution can be computed by $\mathbf{w}_*^c = -\frac{1}{\lambda n}X^\top\alpha_*$.

For regression, we consider the sparse regularized least-squares regression (SLSR) problem:

$$\mathbf{w}_*^r = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2n}\|X\mathbf{w}-\mathbf{y}\|_2^2 + R(\mathbf{w}), \tag{3}$$

where $R(\mathbf{w})$ is a sparsity-inducing norm. In the sequel, we consider two widely used sparsity-inducing norms: (i) $R(\mathbf{w}) = \gamma\|\mathbf{w}\|_1$, the ℓ_1 norm that leads to a formulation also known as LASSO (Tibshirani 1996); (ii) $R(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \gamma\|\mathbf{w}\|_1$, the mixture of ℓ_1 and ℓ_2 norm that leads to a formulation known as the Elastic Net (Zou and Hastie 2003).

Several remarks are in order. Firstly, we will abuse the same notation $\mathbf{w}_*$ to denote the optimal solution to the classification problem or the SLSR problem when it is clear from the context. Secondly, the dual formulation in (2) for classification and the SLSR problem (3) share a similar structure that consists of a quadratic term and a (strongly) convex term, which allows us to derive similar theoretical guarantee on the recovery error of α_* for classification and of $\mathbf{w}_*^r$ for SLSR. In particular, when the loss function $\ell(z, y)$ is a smooth function (e.g., the squared hinge loss), its conjugate function is strongly convex. When $R(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \tau\|\mathbf{w}\|_1$ with $\lambda > 0$, it is strongly convex. Thirdly, we assume α_* and $\mathbf{w}_*^r$ are sparse. The sparsity of α_* is implicit in many large-scale high-dimensional classification problems, which is indicated by that the number of the support vectors is usually much less than the total number of examples. The sparsity of $\mathbf{w}_*^r$ is explicit induced by the sparsity-inducing regularizer $R(\mathbf{w})$. We will also consider the case when α_* is nearly sparse.

Let $A \in \mathbb{R}^{m\times N}$ denote a random reduction matrix from a distribution that satisfies the JL lemma stated below.

Lemma 1 (JL Lemma) *For any integer $N > 0$, and any $0 < \delta < 1/2$, there exists a probability distribution on $m \times N$ real matrices A such that for any fixed $\mathbf{x} \in \mathbb{R}^N$ with a probability at least $1-\delta$, we have*

$$\left|\|A\mathbf{x}\|_2^2 - \|\mathbf{x}\|_2^2\right| \le c\sqrt{\frac{\log(1/\delta)}{m}}\|\mathbf{x}\|_2^2, \tag{4}$$

where c is a universal constant.

We will discuss several random matrices A that satisfy the above assumption in more details in Sect. 5.

4 Main results

Although the analysis for the classification problem share many similarities to that for the SLSR problem, we will state the results and analysis separately for the sake of clarity.

4.1 Classification

For LSHD data, directly solving (1) or (2) is very expensive. We address the computational challenge by employing randomized reduction methods to reduce the dimensionality. Let

$\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_n$ denote the random sketched data of the training examples, where $\widehat{\mathbf{x}}_i = A\mathbf{x}_i \in \mathbb{R}^m$. With the random sketched data, a conventional approach is to solve the following reduced primal problem

$$\min_{\mathbf{u}\in\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^n \ell(y_i \mathbf{u}^\top \widehat{\mathbf{x}}_i) + \frac{\lambda}{2}\|\mathbf{u}\|_2^2, \tag{5}$$

or its the dual problem

$$\widehat{\alpha}_* = \arg\max_{\alpha\in\mathbb{R}^n} -\frac{1}{n}\sum_{i=1}^n \ell_i^*(\alpha_i) - \frac{1}{2\lambda n^2}\alpha^T \widehat{X}\widehat{X}^\top \alpha, \tag{6}$$

where $\widehat{X} = (\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_n)^\top \in \mathbb{R}^{n\times m}$. Zhang et al. (2014) proposed a dual recovery approach that constructs a recovered solution by $\widehat{\mathbf{w}}_* = -\frac{1}{\lambda n}\sum_{i=1}^n [\widehat{\alpha}_*]_i \mathbf{x}_i$ and proved the recovery error for random projection under the assumption of low-rank data matrix or sparse $\mathbf{w}_*$. One deficiency with the simple dual recovery approach is that it fails to capture that data in the reduced feature space could become difficult to be separated, especially when the reduced dimensionality m is small. As a result, the magnitude of the dual solution may become larger, causing many non-support vectors (of the original problem) to be support vectors, which could further result in the corruption in the recovery error. That is also the reason that 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.

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

$$\widetilde{\alpha}_* = \arg\max_{\alpha\in\mathbb{R}^n} -\frac{1}{n}\sum_{i=1}^n \ell_i^*(\alpha_i) - \frac{1}{2\lambda n^2}\alpha^T \widehat{X}\widehat{X}^\top \alpha - \frac{1}{n}\tau\|\alpha\|_1, \tag{7}$$

where $\tau > 0$ is a regularization parameter, whose theoretical value will be revealed later. Given $\widetilde{\alpha}_*$, we can recover a model in the original high-dimensional space by

$$\widetilde{\mathbf{w}}_* = -\frac{1}{\lambda n} X^\top \widetilde{\alpha}_*. \tag{8}$$

Geometric Explanation by Reduced Margin: To further understand the added sparse regularizer, we consider SVM, where the loss function can be either the hinge loss (a non-smooth function) $\ell(z) = \max(0, 1-z)$ or the squared hinge loss (a smooth function) $\ell(z) = \max(0, 1-z)^2$. We first consider the hinge loss, where $\ell_i^*(\alpha_i) = \alpha_i y_i$ for $\alpha_i y_i \in [-1, 0]$. Then the new dual problem is equivalent to

$$\max_{\alpha\circ\mathbf{y}\in[-1,0]^n} \frac{1}{n}\sum_{i=1}^n -\alpha_i y_i - \frac{1}{2\lambda n^2}\alpha^T \widehat{X}\widehat{X}^\top \alpha - \frac{\tau}{n}\|\alpha\|_1.$$

Using variable transformation $-\alpha_i y_i \to \beta_i$, the above problem is equivalent to

$$\max_{\beta\in[0,1]^n} \frac{1}{n}\sum_{i=1}^n \beta_i(1-\tau) - \frac{1}{2\lambda n^2}(\beta\circ\mathbf{y})^T \widehat{X}\widehat{X}^\top (\beta\circ\mathbf{y}).$$

Changing into the primal form, we have

$$\max_{\mathbf{u}\in\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^{n} \ell_{1-\tau}(\mathbf{u}^\top \widehat{\mathbf{x}}_i y_i) + \frac{\lambda}{2}\|\mathbf{u}\|_2^2, \tag{9}$$

where $\ell_\gamma(z) = \max(0, \gamma - z)$ is a max-margin loss with margin given by γ. It can be understood that adding the ℓ_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

$$\max_{\mathbf{u}\in\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^{n} \ell^2_{1-\tau}(\mathbf{u}^\top \widehat{\mathbf{x}}_i y_i) + \frac{\lambda}{2}\|\mathbf{u}\|_2^2, \tag{10}$$

where $\ell^2_\gamma(z) = \max(0, \gamma - z)^2$. Although adding a sparse regularizer on the dual variable is intuitive and can be motivated from previous results, we emphasize that the proposed sparse regularized dual formulation provides a new perspective and bounding the dual recovery error $\|\widetilde{\alpha}_* - \alpha_*\|$ is a non-trivial task. We first state our main result in Theorem 1 for smooth loss functions.

Theorem 1 *Let $A \in \mathbb{R}^{m\times d}$ be a random matrix sampled from a distribution that satisfies the JL lemma. Let $\widetilde{\alpha}_*$ be the optimal dual solution to (7). Assume α_* is s-sparse, $\max_i \|\mathbf{x}_i\|_2 \le R$ and $\ell(z)$ is L-smooth. If we set*

$$\tau = \Theta\left(R\|\mathbf{w}_*\|_2\sqrt{\frac{\log(2n/\delta)}{m}}\right) \ge 2cR\|\mathbf{w}_*\|_2\sqrt{\frac{\log(2n/\delta)}{m}},$$

then we have

$$\|\widetilde{\alpha}_* - \alpha_*\|_2 \le O\left(LR\|\mathbf{w}_*\|_2\sqrt{s}\sqrt{\frac{\log(2n/\delta)}{m}}\right), \tag{11}$$

$$\|\widetilde{\alpha}_* - \alpha_*\|_1 \le O\left(LR\|\mathbf{w}_*\|_2 s\sqrt{\frac{\log(2n/\delta)}{m}}\right), \tag{12}$$

where c is the quantity that appears in the Lemma 1.

Remark 1 The smooth loss function that can yield a sparse α_* includes the squared hinge loss $\ell(z) = \max(0, 1 - z)^2$ and the ϵ-insensitive loss. From the proof, we will see that the value of τ depends on the order of $\|(XX^\top - \widehat{X}\widehat{X}^\top)\alpha_*\|_\infty$, *which we can bound without using any assumption about the data matrix*. In contrast, previous bounds (Zhang et al. 2013, 2014; Paul et al. 2013) depend on $\|XX^\top - \widehat{X}\widehat{X}^\top\|_2$, which requires the low rank assumption on X. The result in the above theorem indicates the recovery error of the obtained dual variable 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 $\mathbf{f} = U\mathbf{w}_* + \mathbf{e}$, where $\mathbf{e}$ denotes the noise in measurements, if a LASSO $\min_{\mathbf{w}} \frac{1}{2}\|U\mathbf{w} - \mathbf{f}\|_2^2 + \lambda\|\mathbf{w}\|_1$ is solved for the solution, then the regularization parameter λ is required to be larger than the quantity $\|U^\top \mathbf{e}\|_\infty$ that depends on the noise in order to have an accurate recovery (Eldar and Kutyniok 2012). Similarly in

our formulation, the added ℓ_1 regularization $\tau\|\alpha\|_1$ is to counteract the noise in $\widehat{X}\widehat{X}^\top$ as compared with $XX^\top$ and the value of τ is dependent on the noise.

Next, we present the results for non-smooth loss functions. Compared with a smooth loss function that renders the objective function (7) strongly convex, the convex conjugate of a non-smooth function is not strongly convex. To address this limitation, we can explore the restricted strong convexity of the quadratic term with respect to sparse solutions. To this end, we introduce the restricted eigen-value conditions similar to those used in the sparse recovery analysis for LASSO (Bickel et al. 2009; Xiao and Zhang 2013). In particular, we introduce the following definition of restricted eigen-values.

Definition 1 Given an integer $s > 0$, we define

$$\mathcal{K}_{n,s} = \{\alpha \in \mathbb{R}^n : \|\alpha\|_2 \leq 1, \|\alpha\|_1 \leq \sqrt{s}\}.$$

We say that $XX^\top \in \mathbb{R}^{n\times n}$ satisfies the restricted eigenvalue condition at sparsity level s if there exist positive constants $\rho^+_{n,s}$ and $\rho^-_{n,s}$ such that

$$\rho^+_{n,s} = \sup_{\alpha\in\mathcal{K}_{n,s}} \frac{\alpha^\top XX^\top\alpha}{n}, \quad \rho^-_{n,s} = \inf_{\alpha\in\mathcal{K}_{n,s}} \frac{\alpha^\top XX^\top\alpha}{n}. \tag{13}$$

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

$$\sigma_{n,s} = \sup_{\alpha\in\mathcal{K}_{n,s}} \frac{|\alpha^\top(XX^\top - \widehat{X}\widehat{X}^\top)\alpha|}{n}. \tag{14}$$

The lemma below shows that $\sigma_{n,s}$ can be bounded by $\widetilde{O}(\rho^+_{n,s}\sqrt{s/m})$.

Lemma 2 *Let $A \in \mathbb{R}^{m\times d}$ be a random matrix sampled from a distribution that satisfies the JL lemma. With a probability $1-\delta$, we have*

$$\sigma_{n,s} \leq 16c\rho^+_{n,s}\sqrt{\frac{(\log(2/\delta) + 2s\log(27n/s))}{m}},$$

where c is the quantity that appears in Lemma 1.

Theorem 2 *Let $A \in \mathbb{R}^{m\times d}$ be a random matrix sampled from a distribution that satisfies the JL lemma. Let $\widetilde{\alpha}_*$ be the optimal dual solution to (7). Assume α_* is s-sparse, $\max_i \|\mathbf{x}_i\|_2 \leq R$, the data matrix $XX^\top$ satisfies the restricted eigen-value condition at sparsity level $16s$ and $\sigma_{n,16s} < \rho^-_{n,16s}$. If we set*

$$\tau = \Theta\left(2cR\|\mathbf{w}_*\|_2\sqrt{\frac{\log(2n/\delta)}{m}}\right) \geq 2cR\|\mathbf{w}_*\|_2\sqrt{\frac{\log(2n/\delta)}{m}},$$

then we have

$$\|\widetilde{\alpha}_* - \alpha_*\|_2 \leq O\left(\frac{\lambda}{(\rho^-_{n,16s} - \sigma_{n,16s})}R\|\mathbf{w}_*\|_2\sqrt{s}\sqrt{\frac{\log(2n/\delta)}{m}}\right),$$

$$\|\widetilde{\alpha}_* - \alpha_*\|_1 \leq O\left(\frac{\lambda}{(\rho^-_{n,16s} - \sigma_{n,16s})}R\|\mathbf{w}_*\|_2 s\sqrt{\frac{\log(2n/\delta)}{m}}\right),$$

where c is the quantity that appears in Lemma 1.

Remark 3 Compared to smooth loss functions, the smoothness constant L is replaced by $\widehat{L} = \frac{\lambda}{(\rho^-_{n,16s} - \sigma_{n,16s})}$ for the non-smooth loss functions, and there is an extra condition $\sigma_{n,16s} < \rho^-_{n,16s}$. In light of Lemma 2, the condition implies that $m \geq \Omega\left(\left(\frac{\rho^+_{n,16s}}{\rho^-_{n,16s}}\right)^2 s \log(n/s)\right)$.

Finally, we present the recovery error of the recovered high-dimensional model $\widetilde{\mathbf{w}}_*$.

Theorem 3 *Let $A \in \mathbb{R}^{m\times d}$ be a random matrix sampled from a distribution that satisfies the JL lemma. Let $\widetilde{\mathbf{w}}_*$ be the recovered primal solution in (8). Define $\tilde{L} = L$ if the loss function is L-smooth, otherwise $\tilde{L} = \frac{\lambda}{(\rho^-_{n,16s} - \sigma_{n,16s})}$. Assume α_* is s-sparse, $\max_i \|\mathbf{x}_i\|_2 \leq R$, and $\sigma_{n,16s} < \rho^-_{n,16s}$ if the loss function is non-smooth. If we set*

$$\tau = \Theta\left(2cR\|\mathbf{w}_*\|_2\sqrt{\frac{\log(n/\delta)}{m}}\right) \geq 2cR\|\mathbf{w}_*\|_2\sqrt{\frac{\log(n/\delta)}{m}},$$

then we have

$$\|\widetilde{\mathbf{w}}_* - \mathbf{w}_*\|_2 \leq O\left(\frac{\sqrt{\rho^+_{n,16s}}}{\lambda\sqrt{n}}\tilde{L}R\|\mathbf{w}_*\|_2\sqrt{s}\sqrt{\frac{\log(n/\delta)}{m}}\right),$$

where c is the quantity that appears in Lemma 1.

Remark 4 To understand the order of the recovery error, we consider a smooth loss function with L being a constant. According to the learning theory, the optimal λ can be in the order of $1/\sqrt{n}$ (Sridharan et al. 2008). In addition, $\sqrt{\rho_{n,16s}} \leq R$. Thus, the recovery error is in the order of $O(R^2\|\mathbf{w}_*\|_2\sqrt{s/m})$ for a smooth loss function. We can compare this recovery error with that derived in Zhang et al. (2014) for smooth loss functions, which is in the order of $O(\sqrt{r/m}\|\mathbf{w}_*\|_2)$ with r being the rank of the data matrix. The two errors have the same scaling in terms of m. The error bound in Zhang et al. (2014) depends on the rank of the data matrix, while that in Theorem 3 depends on the sparsity of the optimal dual solution α_* to the original problem. For real LSHD data, the sparsity of optimal solution α_* is usually much less than n, however, the rank of the data matrix is not necessarily much less than n in particular when the dimensionality is very high, which renders the proposed dual-sparse recovery more attractive. In experiments, we will show that the proposed dual-sparse recovery gives smaller error than the ordinary dual recovery approach on real datasets.

4.2 Sparse regularized least-squares regression (SLSR)

Recall the problem of SLSR

$$\mathbf{w}_* = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2n}\|X\mathbf{w} - \mathbf{y}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \gamma\|\mathbf{w}\|_1, \tag{15}$$

where $\lambda \geq 0$. Although many optimization algorithms have been developed for solving (15), they could still suffer from high computational complexities for large-scale high-dimensional data due to (i) an $O(nd)$ memory complexity and (ii) an $O(nd)$ iteration complexity.

To alleviate the two complexities, we consider using the JL transform to reduce the scale of data. In particular, we let $A \in \mathbb{R}^{m\times n}$ denote a random reduction matrix corresponding to a JL transform, then we compute a sketched data by $\widehat{X} = AX \in \mathbb{R}^{m\times d}$ and $\widehat{\mathbf{y}} = A\mathbf{y} \in \mathbb{R}^m$, and then solve the following problem:

$$\widehat{\mathbf{w}}_* = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2n}\|\widehat{X}\mathbf{w} - \widehat{\mathbf{y}}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + (\gamma+\tau)\|\mathbf{w}\|_1, \tag{16}$$

where $\tau > 0$ is added to strengthen the sparse regularizer, whose theoretical value is exhibited later. We emphasize that to obtain a bound on the optimization error of $\widehat{\mathbf{w}}_*$, i.e., $\|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|$, it is important to increase the value of the regularization parameter before the ℓ_1 norm. Intuitively, after compressing the data the optimal solution may become less sparse, hence increasing the regularization parameter can pull the solution towards closer to the original optimal solution.

Geometric Interpretation by Optimality Condition. We can explain the added parameter τ from a *geometric viewpoint*, which sheds insights on its theoretical value. Without loss of generality, we consider $\lambda = 0$. Since $\mathbf{w}_*$ is the optimal solution to the original problem, then there exists a sub-gradient $g \in \partial\|\mathbf{w}_*\|_1$ such that

$$\frac{1}{n}X^\top(X\mathbf{w}_* - \mathbf{y}) + \gamma g = 0.$$

Since $\|g\|_\infty \leq 1$, therefore $\mathbf{w}_*$ must satisfy $\frac{1}{n}\|X^\top(X\mathbf{w}_* - \mathbf{y})\|_\infty \leq \gamma$. Similarly, the compressed problem (16) also defines a domain of the optimal solution $\widehat{\mathbf{w}}_*$, i.e.,

$$\widehat{\mathcal{D}}_{\mathbf{w}} = \left\{\mathbf{w} \in \mathbb{R}^d : \frac{1}{n}\|\widehat{X}^\top(\widehat{X}\mathbf{w} - \widehat{\mathbf{y}})\|_\infty \leq \tau + \gamma\right\}. \tag{17}$$

It turns out that σ is added to ensure that the original optimal solution $\mathbf{w}_*$ lies in $\widehat{\mathcal{D}}_{\mathbf{w}}$ provided that σ is set appropriately, which can be verified as follows:

$$\begin{aligned}
\frac{1}{n}\left\|\widehat{X}^\top(\widehat{X}\mathbf{w}_* - \widehat{\mathbf{y}})\right\|_\infty &= \frac{1}{n}\left\|X^\top(X\mathbf{w}_* - \mathbf{y}) + \widehat{X}^\top(\widehat{X}\mathbf{w}_* - \widehat{\mathbf{y}}) - X^\top(X\mathbf{w}_* - \mathbf{y})\right\|_\infty \\
&\leq \frac{1}{n}\|X^\top(X\mathbf{w}_* - \mathbf{y})\|_\infty + \frac{1}{n}\left\|\widehat{X}^\top(\widehat{X}\mathbf{w}_* - \widehat{\mathbf{y}}) - X^\top(X\mathbf{w}_* - \mathbf{y})\right\|_\infty \\
&\leq \gamma + \frac{1}{n}\|X^\top(A^\top A - I)(X\mathbf{w}_* - \mathbf{y})\|_\infty.
\end{aligned}$$

Hence, if we set $\tau \geq \frac{1}{n}\|X^\top(A^\top A - I)(X\mathbf{w}_* - \mathbf{y})\|_\infty$, it is guaranteed that $\mathbf{w}_*$ also lies in $\widehat{\mathcal{D}}_{\mathbf{w}}$. Lemma 5 in Sect. 7 provides an upper bound of $\frac{1}{n}\|X^\top(A^\top A - I)(X\mathbf{w}_* - \mathbf{y})\|_\infty$, therefore exhibits a theoretical value of τ.

Next, we present the theoretical guarantee on the recovery error of the obtained solution $\widehat{\mathbf{w}}_*$. We use the notation $\mathbf{e}$ to denote $X\mathbf{w}_* - \mathbf{y} = \mathbf{e}$ and assume $\|\mathbf{e}\|_2 \leq \eta$. We abuse the notation R to denote the upper bound of column vectors in X, i.e., $\max_{1\leq j\leq d}\|\bar{\mathbf{x}}_j\|_2 \leq R$ where $\bar{\mathbf{x}}_j$ denotes the j-th column of X. We first present the result for the elastic net regularizer, which is a strongly convex function.

Theorem 4 *Let $A \in \mathbb{R}^{m\times d}$ be a random matrix sampled from a distribution that satisfies the JL lemma. Let $\mathbf{w}_*$ and $\widehat{\mathbf{w}}_*$ be the optimal solutions to (15) and (16) for $\lambda > 0$, respectively. Assume $\mathbf{w}_*$ is s-sparse and $\max_j \|\bar{\mathbf{x}}_j\|_2 \leq R$. If we set*

$$\tau = \Theta\left(\frac{\eta R}{n}\sqrt{\frac{\log(2d/\delta)}{m}}\right) \geq \frac{2c\eta R}{n}\sqrt{\frac{\log(2d/\delta)}{m}}.$$

Then with a probability at least $1-\delta$, we have

$$\|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|_2 \leq O\left(\frac{\eta R\sqrt{s}}{n\lambda}\sqrt{\frac{\log(2d/\delta)}{m}}\right), \quad \|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|_1 \leq O\left(\frac{\eta Rs}{n\lambda}\sqrt{\frac{\log(2d/\delta)}{m}}\right),$$

where c is the quantity that appears in Lemma 1.

Remark The order of τ is derived by bounding $\frac{1}{n}\|X^\top(A^\top A - I)\mathbf{e}\|_\infty$ that will be exhibited later. The upper bound of the recovery error exhibits several interesting properties: (i) the term of $\sqrt{\frac{s^{2/p}\log(d/\delta)}{m}}$ for p-norm error bound occurs commonly in theoretical results of sparse recovery (Koltchinskii 2011); (ii) the term of R/λ is related to the condition number of the optimization problem (15), which reflects the intrinsic difficulty of optimization; and (iii) the term of η/n is related to the empirical error of the optimal solution $\mathbf{w}_*$. This term makes sense because if $\eta = 0$ indicating that the optimal solution $\mathbf{w}_*$ satisfies $X\mathbf{w}_* - \mathbf{y} = 0$, then it is straightforward to verify that $\mathbf{w}_*$ also satisfies the optimality condition of (16) for $\tau = 0$. Due to the uniqueness of the optimal solution to (15), thus $\widehat{\mathbf{w}}_* = \mathbf{w}_*$.

Next, we present the result for random sketched LASSO. Since the objective is not strongly convex, we explore the restricted strong convexity with respect to sparse vectors, i.e, the restricted eigen-value condition. Different from that the classification problem, here we assume the restricted eigen-value condition for the matrix $X^\top X$.

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

$$\mathcal{K}_{d,s} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\|_2 \leq 1, \|\mathbf{w}\|_1 \leq \sqrt{s}\}.$$

We say that $X^\top X \in \mathbb{R}^{d\times d}$ satisfies the restricted eigenvalue condition at sparsity level s if there exist positive constants $\rho^+_{d,s}$ and $\rho^-_{d,s}$ such that

$$\rho^+_{d,s} = \sup_{\mathbf{w}\in\mathcal{K}_{d,s}} \frac{\mathbf{w}^\top X^\top X\mathbf{w}}{n}, \quad \rho^-_{d,s} = \inf_{\mathbf{w}\in\mathcal{K}_{d,s}} \frac{\mathbf{w}^\top X^\top X\mathbf{w}}{n}. \tag{18}$$

Note that the above definitions of $\rho^+_{d,s}$ and $\rho^-_{d,s}$ are little different from $\rho^+_{n,s}$ and $\rho^-_{n,s}$. Similar to before, we define another quantity that measures the restricted eigen-value of $X^\top X - \widehat{X}^\top\widehat{X}$, namely

$$\sigma_{d,s} = \sup_{\mathbf{w}\in\mathcal{K}_{d,s}} \frac{|\mathbf{w}^\top(X^\top X - \widehat{X}^\top\widehat{X})\mathbf{w}|}{n}. \tag{19}$$

A similar lemma to Lemma 2 hold for $\sigma_{d,s}$ and $\rho^+_{d,s}$ defined here.

Lemma 3 *Let $A \in \mathbb{R}^{m\times d}$ be a random matrix sampled from a distribution that satisfies the JL lemma. With a probability $1-\delta$, we have*

$$\sigma_{d,s} \leq 16c\rho^+_{d,s}\sqrt{\frac{(\log(2/\delta) + 2s\log(27d/s))}{m}},$$

where c is the quantity that appears in Lemma 1.

Theorem 5 *Let $A \in \mathbb{R}^{m\times d}$ be a random matrix sampled from a distribution that satisfies the JL lemma. Assume $X^\top X$ satisfies the restricted eigen-value condition at sparsity level $16s$. Let $\mathbf{w}_*$ and $\widehat{\mathbf{w}}_*$ be the optimal solutions to (15) and (16) with $\lambda = 0$, respectively, and $\Lambda = \rho^-_{d,16s} - 2\sigma_{d,16s}$. If we set*

$$\tau = \Theta\left(\frac{\eta R}{n}\sqrt{\frac{\log(2d/\delta)}{m}}\right) \geq \frac{2c\eta R}{n}\sqrt{\frac{\log(2d/\delta)}{m}},$$

Assume $\Lambda > 0$, then with a probability at least $1-\delta$, we have

$$\|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|_2 \leq O\left(\frac{\eta R\sqrt{s}}{n\Lambda}\sqrt{\frac{\log(2d/\delta)}{m}}\right), \quad \|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|_1 \leq O\left(\frac{\eta Rs}{n\Lambda}\sqrt{\frac{\log(2d/\delta)}{m}}\right),$$

where c is the quantity that appears in Lemma 1.

Remark Note that λ in Theorem 4 is replaced by Λ in Theorem 5. In order to make the result to be valid, we must have $\Lambda > 0$, i.e., $m \geq \Omega\left(\left(\frac{\rho^+_{d,16s}}{\rho^-_{d,16s}}\right)^2 s\log(d/s)\right)$. In addition, if the conditions in Theorem 5 hold, the result in Theorem 4 can be made stronger by replacing λ with $\lambda + \Lambda$.

Remark Under a standard statistical model that the noise in the observation follows a Gaussian distribution the above result indicates the proposed method based on the sketched data $(AX, A\mathbf{y})$ is statistically optimal. In particular, if we assume there exists $\mathbf{u}_*$ such that $y = \mathbf{u}_*^\top \mathbf{x} + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. It was proved that under appropriate conditions (restricted isometry property of the data matrix X), any sparse estimator $\widetilde{\mathbf{w}}$ based on the sketched data $(AX, A\mathbf{y})$ cannot be better than $O(\frac{s\log(d/s)}{m})$ (Pilanci and Wainwright 2016). Under the condition that the entries in X is bounded and the Gaussian noise in the observations $\mathbf{e}$ we have $R \sim O(\sqrt{n})$ and $\eta \sim O(\sqrt{n})$. Thus, the above result indicates that $\|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|_2^2 \leq O(\frac{s\log(d)}{m})$, which further implies that the statistical error of $\widehat{\mathbf{w}}_*$ will be dominated by $O(\frac{s\log(d)}{m})$ since the statistical error of $\mathbf{w}_*$ is $O(\frac{s\log(d)}{n})$ (Bickel et al. 2009). Therefore, the proposed method is statistically optimal up to a logarithmic factor among all methods that learn sparse solution from the sketched data $(AX, A\mathbf{y})$.

5 Randomized reduction by JL transforms

In this section, we review several classes of random reduction matrices $A \in \mathbb{R}^{m\times N}$ and their JL-type lemmas.

5.1 subGaussian random projection

subGaussian 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: $A_{ij} \sim \mathcal{N}(0, 1/m)$, (ii) Rademacher distribution: $\Pr(A_{ij} = \pm 1/\sqrt{m}) = 0.5$, (iii) discrete distribution: $\Pr(A_{ij} = \pm\sqrt{3/m}) = 1/6$ 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.

Lemma 4 (Nelson 2015) *Let $A \in \mathbb{R}^{m\times N}$ be a random matrix with subGaussian entries of mean 0 and variance $1/m$. For any given $\mathbf{x} \in \mathbb{R}^N$ with a probability $1-\delta$, we have*

$$\left|\|A\mathbf{x}\|_2^2 - \|\mathbf{x}\|_2^2\right| \leq c\sqrt{\frac{\log(1/\delta)}{m}}\|\mathbf{x}\|_2^2.$$

where c is some small universal constant.

5.2 Subsampled randomized Hadamard transform (SRHT)

Randomized Hadamard transform was introduced to speed up random projection, reducing the computational time[3] of random projection from $O(Nm)$ to $O(N \log N)$ or even $O(N \log m)$. The projection matrix A is of the form $A = PHD$, where

- $D \in \mathbb{R}^{N\times N}$ is a diagonal matrix with $D_{ii} = \pm 1$ with equal probabilities.
- H is the $N \times N$ Hadamard matrix (assuming N is a power of 2), scaled by $1/\sqrt{N}$.
- $P \in \mathbb{R}^{m\times N}$ is typically a sparse matrix that facilities computing $P\mathbf{x}$. Several choices of P are possible (Ailon and Chazelle 2009; Nelson 2015; Tropp 2011). Below we provide a JL-type lemma for subsampled randomized Hadamard transform.

Lemma 5 (Boutsidis and Gittens 2013) *Let $A = \sqrt{\frac{N}{m}} PHD \in \mathbb{R}^{m\times N}$ be a subsampled randomized Hadamard transform with P being a random sampling matrix with or without replacement. For any given $\mathbf{x}$ with a probability $1 - \delta$, we have*

$$\left| \|A\mathbf{x}\|_2^2 - \|\mathbf{x}\|_2^2 \right| \leq c\sqrt{\frac{\log(1/\delta)\log(N/\delta)}{m}} \|\mathbf{x}\|_2^2,$$

where c is some small universal constant.

Remark 5 Compared to subGaussian random projection, there is an additional $\sqrt{\log(N/\delta)}$ factor in the upper bound. So directly using the SRHT transform will blow up the recovery error by a logarithmic factor of $\sqrt{\log(d/\delta)}$ for classification and a logarithmic factor of $\sqrt{\log(n/\delta)}$ for SLSR. However, this additional factor can be removed by applying an additional random projection. In particular, if we let $A = \sqrt{\frac{N}{m}} P'PHD \in \mathbb{R}^{m\times N}$, where $P \in \mathbb{R}^{t\times N}$ is a random sampling matrix with $t = m \log(N/\delta)$ and $\hat{P} \in \mathbb{R}^{m\times t}$ is a random projection matrix that satisfies Lemma 4, then we have the same order as that in Lemma 4. Note that the projection by $\hat{P}$ only involves marginal computation when $m \ll N$. Please refer to Nelson (2015) for more details.

5.3 Random hashing

Another line of work to speed up random projection is random hashing which makes the reduction matrix A much sparser and takes advantage of the sparsity of original high-dimensional vectors. It was introduced in Shi et al. (2009b) for dimensionality reduction and later was improved to an unbiased version by Weinberger et al. (2009) with some theoretical analysis. Dasgupta et al. (2010) provided a rigorous analysis of the unbiased random hashing. Recently, Kane and Nelson (2014) proposed two new random hashing algorithms with a slightly sparser random matrix A. Here we provide a JL-type lemma for the random hashing algorithm in Kane and Nelson (2014). Let $h_k(i) : [n] \to [m/s], k = 1, \ldots, s$ denote s independent random hashing functions (assuming m is a multiple of s) and let

$$A = \begin{pmatrix} H_1 D_1 \\ H_2 D_2 \\ \cdot \\ \cdot \\ H_s D_s \end{pmatrix} \in \mathbb{R}^{m\times N} \tag{20}$$

[3] Refers to the running time of computing $A\mathbf{x}$.

be a random matrix with a block of s random hashing matrices, where $D_k \in \mathbb{R}^{N\times N}$ is a diagonal matrix with each entry sampled from $\{-1,+1\}$ with equal probabilities, and $H_k \in \mathbb{R}^{m/s,N}$ with $[H_k]_{j,i} = \delta_{j,h_k(i)}$. Note that A is a sparse matrix with the sparsity of each column being s. The lemma below states the JL-type result for the random hashing matrix given in (20).

Lemma 6 *Let A be given in (20). If $s = \Theta(\sqrt{m\log(1/\delta)})$ with a probability $1-\delta$, we have*

$$\left|\|A\mathbf{x}\|_2^2 - \|\mathbf{x}\|_2^2\right| \le c\sqrt{\frac{\log(1/\delta)}{m}}\|\mathbf{x}\|_2^2.$$

If $s = 1$, the with a probability $2/3$ we have

$$\left|\|A\mathbf{x}\|_2^2 - \|\mathbf{x}\|_2^2\right| \le c\sqrt{\frac{1}{m}}\|\mathbf{x}\|_2^2,$$

where c is a universal constant.

Remark 6 The first inequality is a result in Kane and Nelson (2014). The second inequality is proved in Nelson and Nguyen (2012). If using one block of random hashing matrix for A, then the high probability error bounds become a constant probability error bounds. In practice, we find that using one block random hashing performs as well as Gaussian random projection.

5.4 Random sampling

Last we discuss random sampling and compare with the aforementioned randomized reduction methods.

Lemma 7 *Let $A = \sqrt{\frac{d}{m}}P \in \mathbb{R}^{m\times d}$ be a scaled random sampling matrix where $P \in \mathbb{R}^{m\times d}$ samples m coordinates with or without replacement. Then with a probability $1-\delta$, we have*

$$\left|\|A\mathbf{x}\|_2^2 - \|\mathbf{x}\|_2^2\right| \le c\frac{\sqrt{d}\|\mathbf{x}\|_\infty}{\|\mathbf{x}\|_2}\sqrt{\frac{\log(1/\delta)}{m}}\|\mathbf{x}\|_2^2,$$

where c is a universal constant.

Remark 7 Compared with other three randomized reduction methods, there is an additional $\frac{\|\mathbf{x}\|_\infty}{\|\mathbf{x}\|_2}\sqrt{d}$ factor in the upper bound, which could result in a much larger recovery error. That is why the randomized Hadamard transform was introduced to make this additional factor close to a constant.

6 Optimization and time complexity

In this section, we discuss the optimization for solving the original problems and the random sketched problems, and the time complexities for the proposed randomized methods. There are many optimization algorithms that can be used to solve the original formulations in (1) or (2) or (3), including stochastic dual coordinate ascent for solving the dual formulation (SDCA) (Shalev-Shwartz and Zhang 2013b), stochastic variance reduced gradient (SVRG) (Johnson and Zhang 2013), stochastic average gradient algorithm (SAGA) (Defazio et al. 2014), accelerated stochastic proximal coordinate gradient method (APCG) (Lin et al. 2014),

and many other variants (Shalev-Shwartz and Zhang 2013a, 2014; Xiao and Zhang 2014). Next, we focus on the APCG algorithm since (i) it can be applied to solving the sparse regularized dual formulation in (7) and the SLSR formulation in (16); (ii) it is applicable for both strongly convex objective functions and non-strongly convex objective functions; (iii) it achieves the state-of-the art iteration complexities for both problems. We restrict the discussion to the strongly convex objective functions (i.e., the loss functions are smooth for classification and the regularizer for SLSR includes the ℓ_2 norm). In particular, we assume the loss functions for classification $\ell(z)$ is L-smooth. In the following discussion, we make no assumption about the sparsity of the data.

First, we note that the optimization problems in (2), (7), (15) and (16) can be written as the following general form:

$$\min_{\mathbf{u}\in\mathbb{R}^N} F(\mathbf{u}) \triangleq \underbrace{\frac{1}{2n}\mathbf{u}^\top C^\top C\mathbf{u} + \frac{\beta}{2}\|\mathbf{u}\|_2^2}_{f(\mathbf{u})} + \underbrace{\sum_{i=1}^N \phi_i(u_i)}_{\Psi(\mathbf{u})}, \tag{21}$$

where $C \in \mathbb{R}^{M\times N}$ and $\phi_i(u_i)$ is a convex function. In particular,

- For (2), we have $N = n, M = d, \mathbf{u} = \alpha \in \mathbb{R}^n$, $C = X^\top \in \mathbb{R}^{d\times n}$, $\beta = \frac{\lambda}{L}$, and $\phi_i(\alpha_i) = \lambda(\ell_i^*(\alpha_i) - \frac{1}{L}\alpha_i^2)$
- For (7), we have $N = n, M = m, \mathbf{u} = \alpha \in \mathbb{R}^n$, $C = \widehat{X}^\top \in \mathbb{R}^{m\times n}$, $\beta = \frac{\lambda}{L}$, and $\phi_i(\alpha_i) = \lambda(\ell_i^*(\alpha_i) - \frac{1}{L}\alpha_i^2 + \tau|\alpha_i|)$
- For (15), we have $N = d, M = n, \mathbf{u} = \mathbf{w} \in \mathbb{R}^d$, $C = X \in \mathbb{R}^{n\times d}$, $\beta = \lambda$, and $\phi_i(w_i) = \gamma|w_i| - \frac{1}{n}w_i[X^\top\mathbf{y}]_i$
- For (16), we have $N = d, M = m, \mathbf{u} = \mathbf{w} \in \mathbb{R}^d$, $C = \widehat{X} \in \mathbb{R}^{m\times d}$, $\beta = \lambda$, and $\phi_i(w_i) = (\gamma+\tau)|w_i| - \frac{1}{n}w_i[\widehat{X}^\top\widehat{\mathbf{y}}]_i$

Assume that $\max_j \|C_{*j}\|_2 \le R_c$ where C_{*j} denotes the j-th column. By the specific form of $f(\mathbf{u})$ in (21), we can show that

$$|\nabla_i f(\mathbf{u} + \mathbf{e}_i\delta) - \nabla_i f(\mathbf{u})| \le \left(\frac{R_c^2}{n} + \beta\right)|\delta|, \quad \forall\,\delta\in\mathbb{R}, \quad i = 1,\ldots,n, \quad \mathbf{u}\in\mathbb{R}^N,$$

where $\mathbf{e}_i$ is the basis vector whose all coordinates are zero except that the i-th coordinate is 1. By definition, this means the coordinate-wise Lipschitz continuity constant for each coordinate of $f(\mathbf{u})$ is $\frac{R_c^2}{n} + \beta$. Knowing this constant and the strong convexity parameter β of $f(\mathbf{u})$, APCG method can be formulated as Algorithm 1.

The direct implementation of APCG requires updating full-dimensional vectors such as $\mathbf{v}^{(k)}$, $\mathbf{w}^{(k+1)}$ and $\mathbf{u}^{(k+1)}$ at each iteration. Moreover, the i_kth coordinate of the gradient $\nabla_i f(\mathbf{u}) = \frac{C_{*i}^T C\mathbf{u}}{n} + \beta u_i$ so that its direct computation requires taking a full-dimensional inner product of vectors per iteration. Hence, the complexity of each iteration Algorithm 1 is $O(N)$ at least if the matrix C^TC can be pre-computed and stored in memory. In order to avoid these full-dimensional updates, the authors in Lin et al. (2014) proposed a change of variables scheme so that APCG can be represented equivalently by transformed variables which can be updated in only one dimension at each iteration and allows a low-cost computation of $\nabla_i f(\mathbf{u})$. We present this efficient implementation of APCG in Algorithm 2.

According to Lin et al. (2014), the transformed variables $\mathbf{x}^{(k)}$ and $\mathbf{y}^{(k)}$ in Algorithm 2 are related to the original variables in Algorithm 1 as follows

$$\mathbf{u}^{(k)} = \rho^k\mathbf{x}^{(k)} + \mathbf{y}^{(k)}, \quad \mathbf{v}^{(k)} = \rho^{k+1}\mathbf{x}^{(k)} + \mathbf{y}^{(k)}, \quad \mathbf{w}^{(k)} = -\rho^k\mathbf{x}^{(k)} + \mathbf{y}^{(k)},$$

Algorithm 1 APCG for $\beta > 0$

input: $\mathbf{u}^{(0)} \in \mathrm{dom}(\Psi)$, convexity parameter $\beta > 0$ and a constant R_c such that $\max_j \|C_{*j}\|_2 \le R_c$.

initialize: set $\mathbf{w}^{(0)} = \mathbf{u}^{(0)}$ and $\theta = \frac{1}{N}\sqrt{\frac{n\beta}{R_c^2+n\beta}}$.

iterate: repeat for $k = 0, 1, 2, \ldots$

1. Compute $\mathbf{v}^{(k)} = \frac{\mathbf{u}^{(k)}+\theta\mathbf{w}^{(k)}}{1+\theta}$.
2. Choose $i_k \in \{1, \ldots, n\}$ uniformly at random and compute

$$\mathbf{w}^{(k+1)} = \mathrm{argmin}_{\mathbf{u}\in\mathbb{R}^N} L(\mathbf{u}), \text{ where}$$
$$L(\mathbf{u}) = \left\{\frac{N\theta}{2}\left(\frac{R_c^2}{n}+\beta\right)\left\|\mathbf{u}-(1-\theta)\mathbf{w}^{(k)}-\theta\mathbf{v}^{(k)}\right\|^2 + \nabla_{i_k} f(\mathbf{v}^{(k)})(u_{i_k} - v_{i_k}^{(k)}) + \phi_{i_k}(u_{i_k})\right\}.$$

3. Set $\mathbf{u}^{(k+1)} = \mathbf{v}^{(k)} + N\theta(\mathbf{w}^{(k+1)} - \mathbf{w}^{(k)}) + N\theta^2(\mathbf{w}^{(k)} - \mathbf{v}^{(k)})$.

Algorithm 2 Efficient implementation of APCG for $\beta > 0$

input: $\mathbf{u}^{(0)} \in \mathrm{dom}(\Psi)$, convexity parameter $\beta > 0$ and a constant R_c such that $\max_j \|C_{*j}\|_2 \le R_c$.

initialize: set $\mathbf{x}^{(0)} = 0$, $\mathbf{y}^{(0)} = \mathbf{u}^{(0)}$, $\mathbf{p}^{(0)} = 0$, $\mathbf{q}^{(0)} = C\mathbf{u}^{(0)}$, $\theta = \frac{1}{N}\sqrt{\frac{n\beta}{R_c^2+n\beta}}$ and $\rho = \frac{1-\theta}{1+\theta}$.

iterate: repeat for $k = 0, 1, 2, \ldots$

1. Choose $i_k \in \{1, \ldots, n\}$ uniformly at random and compute the coordinate gradient

$$\nabla_{i_k}^{(k)} = \frac{1}{n}\left(\rho^{k+1}C_{*i_k}^T\mathbf{p}^{(k)} + C_{*i_k}^T\mathbf{q}^{(k)}\right) + \beta\left(\rho^{k+1}x_{i_k}^{(k)} + y_{i_k}^{(k)}\right).$$

2. Compute

$$h_{i_k}^{(k)} = \mathrm{argmin}_{h\in\mathbb{R}}\left\{\frac{N\theta}{2}\left(\frac{R_c^2}{n}+\beta\right)h^2 + \nabla_{i_k}^{(k)}h + \phi_{i_k}\left(-\rho^{k+1}x_{i_k}^{(k)} + y_{i_k}^{(k)} + h\right)\right\}.$$

3. Let $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)}$ and $\mathbf{y}^{(k+1)} = \mathbf{y}^{(k)}$, and update

$$x_{i_k}^{(k+1)} = x_{i_k}^{(k)} - \frac{1-N\theta}{2\rho^{k+1}}h_{i_k}^{(k)}, \qquad y_{i_k}^{(k+1)} = y_{i_k}^{(k)} + \frac{1+N\theta}{2}h_{i_k}^{(k)},$$
$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} - \frac{1-N\theta}{2\rho^{k+1}}C_{*i_k}h_{i_k}^{(k)}, \qquad \mathbf{q}^{(k+1)} = \mathbf{q}^{(k)} + \frac{1+N\theta}{2}C_{*i_k}h_{i_k}^{(k)}.$$

output: $\mathbf{u}^{(k+1)} = \rho^{k+1}\mathbf{x}^{(k+1)} + \mathbf{y}^{(k+1)}$

where ρ is defined as in Algorithm 2. The auxiliary vectors $\mathbf{p}^{(k)}$ and $\mathbf{q}^{(k)}$ are introduced in order to compute $\nabla_{i_k} f(\mathbf{v}^{(k)})$ efficiently. In fact, according to Lin et al. (2014), $\mathbf{p}^{(k)}$ and $\mathbf{q}^{(k)}$ satisfy

$$\mathbf{p}^{(k)} = C\mathbf{u}^{(k)}, \qquad \mathbf{q}^{(k)} = C\mathbf{v}^{(k)},$$

in each iteration. Since $\mathbf{v}^{(k)} = \rho^{k+1}\mathbf{x}^{(k)} + \mathbf{y}^{(k)}$, we have

$$\nabla_{i_k} f(\mathbf{v}^{(k)}) = \frac{1}{n}C_{*i_k}^T C(\rho^{k+1}\mathbf{x}^{(k)} + \mathbf{y}^{(k)}) + \beta(\rho^{k+1}x_{i_k}^{(k)} + y_{i_k}^{(k)}) = \nabla_{i_k}^{(k)},$$

where $\nabla_{i_k}^{(k)}$ is defined as in Algorithm 2. These observations verify that Algorithm 2 and Algorithm 1 are essentially equivalent.

Note that the complexity for computing $\nabla_{i_k}^{(k)}$ and updating $\mathbf{p}^{(k)}$ and $\mathbf{q}^{(k)}$ are both $O(M)$ while the complexity for computing $h_{i_k}^{(k)}$ and updating $\mathbf{x}^{(k)}$ and $\mathbf{y}^{(k)}$ are only $O(1)$. Hence, the per-iteration complexity of Algorithm 2 is only $O(M)$, lower than that of Algorithm 1.

The iteration complexity of APCG for solving (21) is stated in the following proposition.

Proposition 1 *Assume that* $\max_j \|C_{*j}\|_2 \leq R_c$ *where* C_{*j} *denotes the* j*-th column. Then the iteration complexity of APCG for finding* $\mathbf{u}$ *such that* $\mathrm{E}[F(\mathbf{u})] - F_* \leq \epsilon$ *is bounded by*

$$O\left(N + N\sqrt{\frac{R_c^2}{n\beta}}\right)\log(1/\epsilon),$$

and the time complexity of APCG is bounded by

$$O\left(NM + NM\sqrt{\frac{R_c^2}{n\beta}}\right)\log(1/\epsilon).$$

Suppose the loss function ℓ in (5) is not L-smooth. Its conjugate function ℓ^* is not $\frac{1}{L}$-strongly convex. Then, (2) and (7) can be still formulated as (21) but with $\beta = 0$ and $\phi_i(\alpha_i) = \lambda\ell_i^*(\alpha_i)$. Besides, if the parameter $\lambda = 0$ in (15) and (16), we will $\beta = 0$ in the corresponding formulation of (21). In either case, (21) is no longer a strongly convex optimization problem. However, both Algorithm 1 and Algorithm 2 can be still adapted for solving (21) with similar updating steps except that the parameter θ will vary with the iteration number k. As showed in Lin et al. (2014), if $\beta = 0$, the iteration complexity of APCG for finding $\mathbf{u}$ such that $\mathrm{E}[F(\mathbf{u})] - F_* \leq \epsilon$ is bounded by

$$O\left(N\sqrt{\frac{R_c^2}{n\epsilon}}\right),$$

and the per-iteration cost with an implementation similar to Algorithm 2 is still $O(M)$.

Next, we analyze the total time complexities for the optimizing original formulations (2), (15) and for the proposed formulations (7) and (16).

6.1 Optimization time complexity for classification

We first consider the time complexity for optimizing the original formulation (2) and the proposed formulation (7) for classification. For the original formulation, $R_c = R \triangleq \max_{1\leq i\leq n} \|\mathbf{x}_i\|_2$, the time complexity is

$$O\left(nd + nd\sqrt{\frac{R^2L}{n\lambda}}\right)\log(1/\epsilon). \tag{22}$$

In contrast, for the proposed formulation (7) with a high probability $1-\delta$,

$$\begin{aligned} R_c &= \max_{1\leq i\leq n} \|\widehat{\mathbf{x}}_i\|_2 = \max_{1\leq i\leq n} \|A\mathbf{x}_i\|_2 = \max_{1\leq i\leq n} \|A\mathbf{x}_i\|_2 \leq \max_{1\leq i\leq n} \sqrt{(1+\epsilon_m)}\|\mathbf{x}_i\|_2 \\ &= \sqrt{(1+\epsilon_m)}R, \end{aligned}$$

where $\epsilon_m = c\sqrt{\log(n/\delta)/m}$ according to Assumption 1. By assuming that m is sufficiently large such that $\epsilon_m \leq 1$, then the time complexity is

$$O\left(nm + nm\sqrt{\frac{R^2L}{n\lambda}}\right)\log(1/\epsilon) = O\left(\frac{m}{d}\left[nd + nd\sqrt{\frac{R^2L}{n\lambda}}\right]\right)\log(1/\epsilon). \tag{23}$$

Compared to time complexity in (22) for solving the original formulation, the time complexity for solving the proposed formulation on the random sketched data is scaled roughly by $m/d \ll 1$ with a high probability.

6.2 Optimization time complexity for SLSR

Next, we analyze and compare the time complexity of optimization for (15) and (16). For (15), $R_c = R \triangleq \max_{1\leq j\leq d} \|\bar{\mathbf{x}}_j\|_2$, where $\bar{\mathbf{x}}_j$ denotes the j-th column of the data matrix X. Therefore the time complexity of APCG becomes

$$O\left(nd + nd\sqrt{\frac{R^2}{n\lambda}}\right)\log(1/\epsilon). \tag{24}$$

For (16), with high probability $1-\delta$, we have

$$R_c = \max_{1\leq j\leq d} \|A\bar{\mathbf{x}}_j\|_2 \leq \max_{1\leq j\leq d} \sqrt{1+\epsilon_m}\|\bar{\mathbf{x}}_j\|_2 = \sqrt{1+\epsilon_m}R,$$

where $\epsilon_m = O(\sqrt{\log(d/\delta)/m})$. Let m be sufficiently large, we can conclude that R_c for $\widehat{X}$ is $O(R)$. Therefore, the time complexity of APCG for solving (16) is

$$O\left(md + md\sqrt{\frac{R^2}{n\lambda}}\right)\log(1/\epsilon) = O\left(\frac{m}{n}\left[nd + nd\sqrt{\frac{R^2}{n\lambda}}\right]\right)\log(1/\epsilon)$$

Hence, we can see that the optimization time complexity of APCG for solving (16) can be reduced upto a factor of $1-\frac{m}{n}$, which is substantial when $m \ll n$.

6.3 Total amortized time complexity and parallel computation

The total time complexity for the proposed randomized methods consists of the optimization time for the reduced formulations, and the extra time for randomized reduction and dual recovery if a high-dimensional model in the original feature space is required in the classification problem, i.e.,

$$\text{Total Time} = \text{time}_{proc} + \text{time}_{opt},$$

where time_{opt} refers to the optimization time for solving the reduced formulations and time_{proc} refers to the extra processing time of reduction and recovery if necessary. The recovery time is $O(nd)$ for the proposed randomized algorithm for classification. Among all randomized reduction methods, the transformation using the Gaussian random matrices is the most expensive that takes $O(mnd)$ time complexity when applied to $X \in \mathbb{R}^{n\times d}$, while subsampled randomized Hadamard transform and random hashing can reduce it to $\widetilde{O}(nd)$ and $O(nnz(X))$, respectively, where $\widetilde{O}(\cdot)$ suppresses only a logarithmic factor and $nnz(X)$ denotes the number of non-zeros entries in X. Although the extra computational time still

scales as nd in the worst case comparable to that for optimizing the original optimization problems, the computational benefit of the proposed randomized algorithms can become more prominent when we consider the amortizing time complexity and parallel computation to speed up reduction and recovery. In particular, in machine learning, we usually need to tune the regularization parameters (aka cross-validation) to achieve a better generalization performance. Let B denote the number of settings for the regularization parameters, and K denote the number of nodes (cores) for performing the reduction (and recovery if necessary), then the total amortized time complexity of one node (core) becomes

$$\text{Total Amortized Time} = \frac{\text{time}_{proc}}{K} + B \cdot \text{time}_{opt}.$$

In comparison, the total amortized time complexity for solving the original formulations is $B \cdot \text{Time}_{opt}$. According to the analysis in previous subsections, time_{opt} is $m/n \ll 1$ or $m/d \ll 1$ times of Time_{opt}, hence the total amortized time complexity of the proposed randomized methods using parallel computation can be substantially reduced. Note that here we do not consider the parallel optimization that includes communication time between different nodes (cores), rendering the analysis of optimization time much more involved.

7 Proofs

In this section, we present proofs for the main results.

7.1 Proof of Theorem 1

Denote by $\mathcal{S}$ the support set of α_* and by $\mathcal{S}^c$ its complement. Define

$$\Delta = \frac{1}{\lambda n}(XX^\top - \widehat{X}\widehat{X}^\top)\alpha_*. \tag{25}$$

Let $\hat{F}(\alpha)$ be defined as

$$\hat{F}(\alpha) = \frac{1}{n}\sum_{i=1}^{n} \ell_i^*(\alpha_i) + \frac{1}{2\lambda n^2}\alpha^T \widehat{X}\widehat{X}^\top \alpha + \frac{\tau}{n}\|\alpha\|_1.$$

Since $\widetilde{\alpha}_* = \arg\min \hat{F}(\alpha)$ therefore for any $g_* \in \partial\|\alpha_*\|_1$

$$0 \geq \hat{F}(\widetilde{\alpha}_*) - \hat{F}(\alpha_*) \geq (\widetilde{\alpha}_* - \alpha_*)^\top \left(\frac{1}{n}\nabla \ell^*(\alpha_*) + \frac{1}{\lambda n^2}\widehat{X}\widehat{X}^\top \alpha_*\right) + \frac{\tau}{n}(\widetilde{\alpha}_* - \alpha_*)^\top g_* + \frac{1}{2nL}\|\widetilde{\alpha}_* - \alpha_*\|_2^2,$$

where we used the strong convexity of ℓ_i^* and its strong convexity modulus $1/L$. By the optimality condition of α_*, we can have

$$0 \geq (\alpha_* - \widetilde{\alpha}_*)^\top \left(\frac{1}{n}\nabla \ell^*(\alpha_*) + \frac{1}{\lambda n^2}XX^\top \alpha_*\right). \tag{26}$$

Combining the above two inequalities we have

$$0 \geq (\widetilde{\alpha}_* - \alpha_*)^\top \frac{1}{n}\Delta + \frac{\tau}{n}(\widetilde{\alpha}_* - \alpha_*)^\top g_* + \frac{1}{2nL}\|\widetilde{\alpha}_* - \alpha_*\|_2^2.$$

Since the above inequality holds for any $g_* \in \partial\|\alpha_*\|_1$, if we choose $[g_*]_i = sign([\widetilde{\alpha}_*]_i)$, $i \in \mathcal{S}^c$, then we have

$$(\widetilde{\alpha}_* - \alpha_*)^\top g_* \geq -\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1 + \|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1.$$

Combining the above inequalities leads to

$$(\tau + \|\Delta\|_\infty)\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1 \geq (\tau - \|\Delta\|_\infty)\|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1 + \frac{1}{2L}\|\widetilde{\alpha}_* - \alpha_*\|_2^2. \tag{27}$$

Assuming $\tau \geq 2\|\Delta\|_\infty$, we have

$$\frac{3\tau}{2}\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1 \geq \frac{\tau}{2}\|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1 + \frac{1}{2L}\|\widetilde{\alpha}_* - \alpha_*\|_2^2. \tag{28}$$

Then

$$\begin{aligned} &\|\widetilde{\alpha}_* - \alpha_*\|_2^2 \leq 3\tau L\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1 \\ &\|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1 \leq 3\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1. \end{aligned} \tag{29}$$

Therefore,

$$\|[\widetilde{\alpha}_* - \alpha_*]_{\mathcal{S}}\|_1^2 \leq s\|\widetilde{\alpha}_* - \alpha_*\|_2^2 \leq 3\tau Ls\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1,$$

leading to the result

$$\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1 \leq 3\tau Ls.$$

Combining this inequality with inequalities in (29) we have

$$\begin{aligned} &\|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1 \leq 9\tau Ls \\ &\|\widetilde{\alpha}_* - \alpha_*\|_2 \leq 3\tau L\sqrt{s}. \end{aligned} \tag{30}$$

and

$$\|\widetilde{\alpha}_* - \alpha_*\|_1 \leq \|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1 + \|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1 \leq 12\tau Ls. \tag{31}$$

To complete the proof of Theorem 1, we need to bound $\|\Delta\|_\infty$, i.e., the value of τ.

Lemma 8 *Let $A \in \mathbb{R}^{m\times d}$ be a random matrix such that Assumption 1. Then with a high probability $1-\delta$ we have*

$$\|\Delta\|_\infty \leq cR\|\mathbf{w}_*\|_2\sqrt{\frac{\log(2n/\delta)}{m}},$$

where $R = \max_i \|\mathbf{x}_i\|_2$.

Proof Let $\epsilon_{m,\delta} = c\sqrt{\log(1/\delta)/m}$.

$$\begin{aligned} &\frac{1}{\lambda n}(\widehat{X}\widehat{X}^\top - XX^\top)\alpha_* = \frac{1}{\lambda n}(XA^\top AX^\top - XX^\top)\alpha_* \\ &= \frac{1}{\lambda n}X(A^\top A - I)X^\top\alpha_* = X(I - A^\top A)\mathbf{w}_*, \end{aligned}$$

where we use the fact $\mathbf{w}_* = -\frac{1}{\lambda n}X^\top\alpha_*$. Then

$$\frac{1}{\lambda n}[(\widehat{X}\widehat{X}^\top - XX^\top)\alpha_*]_i = \mathbf{x}_i^\top(I - A^\top A)\mathbf{w}_*.$$

Therefore in order to bound $\|\Delta\|_\infty$, we need to bound $\mathbf{x}_i^\top (I - A^\top A)\mathbf{w}_*$ for all $i \in [n]$. We first bound for individual i and then apply the union bound. Let $\widetilde{\mathbf{x}}_i$ and $\widetilde{\mathbf{w}}_*$ be normalized version of $\mathbf{x}_i$ and $\mathbf{w}_*$, i.e., $\widetilde{\mathbf{x}}_i = \mathbf{x}_i/\|\mathbf{x}_i\|_2$ and $\widetilde{\mathbf{w}}_* = \mathbf{w}_*/\|\mathbf{w}_*\|_2$. By Assumption 1, with a probability $1-\delta$,

$$(1-\epsilon_{m,\delta})\|\widetilde{\mathbf{x}}_i + \widetilde{\mathbf{w}}_*\|_2^2 \le \|A(\widetilde{\mathbf{x}}_i + \widetilde{\mathbf{w}}_*)\|_2^2 \le (1+\epsilon_{m,\delta})\|\widetilde{\mathbf{x}}_i + \widetilde{\mathbf{w}}_*\|_2^2,$$

and with a probability $1-\delta$,

$$(1-\epsilon_{m,\delta})\|\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{w}}_*\|_2^2 \le \|A(\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{w}}_*)\|_2^2 \le (1+\epsilon_{m,\delta})\|\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{w}}_*\|_2^2.$$

Then with a probability $1-2\delta$, we have

$$\begin{aligned}
\widetilde{\mathbf{x}}_i^\top A^\top A\widetilde{\mathbf{w}}_* - \widetilde{\mathbf{x}}_i^\top \widetilde{\mathbf{w}}_* &= \frac{\|A(\widetilde{\mathbf{x}}_i + \widetilde{\mathbf{w}}_*)\|_2^2 - \|A(\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{w}}_*)\|_2^2}{4} - \widetilde{\mathbf{x}}_i^\top \widetilde{\mathbf{w}}_* \\
&\le \frac{(1+\epsilon_{m,\delta})\|\widetilde{\mathbf{x}}_i + \widetilde{\mathbf{w}}_*\|_2^2 - (1-\epsilon_{m,\delta})\|\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{w}}_*\|_2^2}{4} - \widetilde{\mathbf{x}}_i^\top \widetilde{\mathbf{w}}_* \\
&\le \frac{\epsilon_{m,\delta}}{2}(\|\widetilde{\mathbf{x}}_i\|_2^2 + \|\widetilde{\mathbf{w}}_*\|_2^2) \le \epsilon_{m,\delta},
\end{aligned}$$

and

$$\begin{aligned}
\widetilde{\mathbf{x}}_i^\top A^\top A\widetilde{\mathbf{w}} - \widetilde{\mathbf{x}}_i^\top \widetilde{\mathbf{w}}_* &= \frac{\|A(\widetilde{\mathbf{x}}_i + \widetilde{\mathbf{w}}_*)\|_2^2 - \|A(\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{w}}_*)\|_2^2}{4} - \widetilde{\mathbf{x}}_i^\top \widetilde{\mathbf{w}}_* \\
&\ge \frac{(1-\epsilon_{m,\delta})\|\widetilde{\mathbf{x}}_i + \widetilde{\mathbf{w}}_*\|_2^2 - (1+\epsilon_{m,\delta})\|\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{w}}_*\|_2^2}{4} - \widetilde{\mathbf{x}}_i^\top \widetilde{\mathbf{w}}_* \\
&\ge -\frac{\epsilon_{m,\delta}}{2}(\|\widetilde{\mathbf{x}}_i\|_2^2 + \|\widetilde{\mathbf{w}}_*\|_2^2) \ge -\epsilon_{m,\delta}.
\end{aligned}$$

Therefore with a probability $1-2\delta$, we have

$$|\mathbf{x}_i^\top A^\top A\mathbf{w}_* - \mathbf{x}_i^\top \mathbf{w}_*| \le \|\mathbf{x}_i\|_2\|\mathbf{w}_*\|_2|\widetilde{\mathbf{x}}_i^\top A^\top A\widetilde{\mathbf{w}}_* - \widetilde{\mathbf{x}}^\top \widetilde{\mathbf{w}}_*| \le \|\mathbf{x}_i\|_2\|\mathbf{w}_*\|_2\epsilon_{m,\delta}.$$

Then applying union bound, with a probability $1-2n\delta$,

$$\|\Delta\|_\infty \le cR\|\mathbf{w}_*\|_2\sqrt{\frac{\log(1/\delta)}{m}},$$

or with a probability $1-\delta$

$$\|\Delta\|_\infty \le cR\|\mathbf{w}_*\|_2\sqrt{\frac{\log(2n/\delta)}{m}}.$$

□

To finish the proof of Theorem 1, we can combine the above Lemma with the inequalities in (31) and (30) by noting the value of τ.

7.2 Proof of Theorem 2

Following the same proof of Theorem 1, we first notice that inequality (27) holds for $L = \infty$, i.e.,

$$(\tau + \|\Delta\|_\infty)\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1 \ge (\tau - \|\Delta\|_\infty)\|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1.$$

Therefore if $\tau \geq 2\|\Delta\|_\infty$, we have

$$\|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1 \leq 3\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1.$$

As a result,

$$\frac{\|\widetilde{\alpha}_* - \alpha_*\|_1}{\|\widetilde{\alpha}_* - \alpha_*\|_2} \leq \frac{\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1 + \|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1}{\|\widetilde{\alpha}_* - \alpha_*\|_2} \leq \frac{4\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1}{\|\widetilde{\alpha}_* - \alpha_*\|_2} \leq 4\sqrt{s}.$$

By the definition of $\mathcal{K}_{n,s}$, we have $\dfrac{\widetilde{\alpha}_* - \alpha_*}{\|\widetilde{\alpha}_* - \alpha_*\|_2} \in \mathcal{K}_{n,16s}$. To proceed the proof, there exists $\widetilde{g}_* \in \partial|\widetilde{\alpha}_*|_1$ such that

$$0 \geq (\widetilde{\alpha}_* - \alpha_*)^\top \left(\frac{1}{n}\nabla\ell^*(\widetilde{\alpha}_*) + \frac{1}{\lambda n^2}\widehat{X}^\top\widehat{X}\widetilde{\alpha}_*\right) + \frac{\tau}{n}(\widetilde{\alpha}_* - \alpha_*)^\top\widetilde{g}_*.$$

Adding the above inequality to (26), we have

$$\begin{aligned}
0 \geq\ & (\alpha_* - \widetilde{\alpha}_*)^\top \left(\frac{1}{n}\nabla\ell^*(\alpha_*) - \frac{1}{n}\nabla\ell^*(\widetilde{\alpha}_*)\right) \\
& + (\alpha_* - \widetilde{\alpha}_*)^\top \left(\frac{1}{\lambda n^2}X^\top X\alpha_* - \frac{1}{\lambda n^2}\widehat{X}^\top\widehat{X}\widetilde{\alpha}_*\right) \\
& + \frac{\tau}{n}\|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1 - \frac{\tau}{n}\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1.
\end{aligned}$$

By convexity of ℓ^* we have

$$(\alpha_* - \widetilde{\alpha}_*)^\top \left[\frac{1}{n}\nabla\ell^*(\alpha_*) - \frac{1}{n}\nabla\ell^*(\widetilde{\alpha}_*)\right] \geq 0.$$

Thus, we have

$$\begin{aligned}
\tau\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1 \geq\ & \tau\|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1 + (\alpha_* - \widetilde{\alpha}_*)^\top \left(\frac{1}{\lambda n}X^\top X - \frac{1}{\lambda n}\widehat{X}^\top\widehat{X}\right)\alpha_* \\
& - (\alpha_* - \widetilde{\alpha}_*)^\top \left(\frac{1}{\lambda n}X^\top X - \frac{1}{\lambda n}\widehat{X}^\top\widehat{X}\right)(\alpha_* - \widetilde{\alpha}_*) \\
& + \frac{1}{\lambda n}(\alpha_* - \widetilde{\alpha}_*)^\top X^\top X(\alpha_* - \widetilde{\alpha}_*).
\end{aligned}$$

Since

$$(\alpha_* - \widetilde{\alpha}_*)^\top\Delta \geq -\|\Delta\|_\infty\|\alpha_* - \widetilde{\alpha}_*\|_1,$$

and $\tau \geq 2\|\Delta\|_\infty$ and by the definition of ρ_s^-, σ_s, we have

$$\frac{3\tau}{2}\|[\widetilde{\alpha}_* - \alpha_*]_{\mathcal{S}}\|_1 \geq \frac{\tau}{2}\|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1 + \frac{\rho_{16s}^- - \sigma_{16s}}{\lambda}\|\widetilde{\alpha}_* - \alpha_*\|_2^2.$$

This is similar to the inequality (28). Then the conclusion follows the same analysis as before.

7.3 Proof of Theorem 4

Define

$$\mathbf{q} = \frac{1}{n}X^\top(A^\top A - I)\mathbf{e}, \qquad \mathbf{e} = X\mathbf{w}_* - \mathbf{y}. \tag{32}$$

First, we note that

$$\begin{aligned}\widehat{\mathbf{w}}_* &= \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2n}\|\widehat{X}\mathbf{w}-\widehat{\mathbf{y}}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + (\gamma+\tau)\|\mathbf{w}\|_1 \\ &= \arg\min_{\mathbf{w}\in\mathbb{R}^d} \underbrace{\frac{1}{2n}\left(\mathbf{w}^\top\widehat{X}^\top\widehat{X}\mathbf{w} - 2\mathbf{w}^\top\widehat{X}^\top\mathbf{y}\right) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + (\gamma+\tau)\|\mathbf{w}\|_1}_{\widehat{F}(\mathbf{w})},\end{aligned}$$

and

$$\mathbf{w}_* = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2n}\|X\mathbf{w}-\mathbf{y}\|_2^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2 + \gamma\|\mathbf{w}\|_1.$$

By optimality of $\widehat{\mathbf{w}}_*$ and the strong convexity of $\widehat{F}(\mathbf{w})$, for any $g \in \partial\|\mathbf{w}_*\|_1$ we have

$$\begin{aligned}0 \geq \widehat{F}(\widehat{\mathbf{w}}_*) - \widehat{F}(\mathbf{w}_*) \geq (\widehat{\mathbf{w}}_*-\mathbf{w}_*)^\top\left(\frac{1}{n}\widehat{X}^\top\widehat{X}\mathbf{w}_* - \frac{1}{n}\widehat{X}^\top\widehat{\mathbf{y}} + \lambda\mathbf{w}_*\right) \\ + (\gamma+\tau)(\widehat{\mathbf{w}}_*-\mathbf{w}_*)^\top g + \frac{\lambda}{2}\|\widehat{\mathbf{w}}_*-\mathbf{w}_*\|_2^2. \qquad (33)\end{aligned}$$

By the optimality condition of $\mathbf{w}_*$, there exists $h \in \partial\|\mathbf{w}_*\|_1$ such that

$$\frac{1}{n}X^\top X\mathbf{w}_* - \frac{1}{n}X^\top\mathbf{y} + \lambda\mathbf{w}_* + \gamma h = 0. \qquad (34)$$

By utilizing the above equation in (33), we have

$$0 \geq (\widehat{\mathbf{w}}_*-\mathbf{w}_*)^\top\mathbf{q} + (\widehat{\mathbf{w}}_*-\mathbf{w}_*)^\top\left[(\gamma+\tau)g - \gamma h\right] + \frac{\lambda}{2}\|\widehat{\mathbf{w}}_*-\mathbf{w}_*\|_2^2. \qquad (35)$$

Let $\mathcal{S}$ denote the support set of $\mathbf{w}_*$ and $\mathcal{S}_c$ denote its complement set. Since g could be any sub-gradient of $\|\mathbf{w}\|_1$ at $\mathbf{w}_*$, we define g as $g_i = \begin{cases} h_i, & i \in \mathcal{S} \\ sign(\widehat{w}_{*i}), & i \in \mathcal{S}_c \end{cases}$. Then we have

$$\begin{aligned}(\widehat{\mathbf{w}}_*-\mathbf{w}_*)^\top\left[(\gamma+\tau)g-\gamma h\right] &= \sum_{i\in\mathcal{S}}(\widehat{w}_{*i}-w_{*i})(\tau h_i) \\ &\quad + \sum_{i\in\mathcal{S}^c}(\widehat{w}_{*i}-w_{*i})(\tau sign(\widehat{w}_{*i}) + \gamma(sign(\widehat{w}_{*i}) - h_i)) \\ &\geq -\tau\|[\widehat{\mathbf{w}}_*-\mathbf{w}_*]_{\mathcal{S}}\|_1 + \sum_{i\in\mathcal{S}_c}\tau sign(\widehat{w}_{*i})\widehat{w}_{*i} \\ &\quad + \sum_{i\in\mathcal{S}_c}\gamma(sign(\widehat{w}_{*i}) - h_i)\widehat{w}_{*i} \\ &\geq -\tau\|[\widehat{\mathbf{w}}_*-\mathbf{w}_*]_{\mathcal{S}}\|_1 + \tau\|[\widehat{\mathbf{w}}_*]_{\mathcal{S}_c}\|_1,\end{aligned}$$

where the last inequality uses $|h_i| \leq 1$ and $\sum_{i\in\mathcal{S}_c}(sign(\widehat{w}_{*i}) - h_i)\widehat{w}_{*i} \geq 0$. Combining the above inequality with (35), we have

$$0 \geq -\|\widehat{\mathbf{w}}_*-\mathbf{w}_*\|_1\|\mathbf{q}\|_\infty - \tau\|[\widehat{\mathbf{w}}_*-\mathbf{w}_*]_{\mathcal{S}}\|_1 + \tau\|[\widehat{\mathbf{w}}_*]_{\mathcal{S}_c}\|_1 + \frac{\lambda}{2}\|\widehat{\mathbf{w}}_*-\mathbf{w}_*\|_2^2.$$

By splitting $\|\widehat{\mathbf{w}}_*-\mathbf{w}_*\|_1 = \|[\widehat{\mathbf{w}}_*-\mathbf{w}_*]_{\mathcal{S}}\|_1 + \|[\widehat{\mathbf{w}}_*-\mathbf{w}_*]_{\mathcal{S}_c}\|_1$ and reorganizing the above inequality we have

$$\frac{\lambda}{2}\|\widehat{\mathbf{w}}_*-\mathbf{w}_*\|_2^2 + (\tau - \|\mathbf{q}\|_\infty)\|[\widehat{\mathbf{w}}_*]_{\mathcal{S}^c}\|_1 \leq (\tau + \|\mathbf{q}\|_\infty)\|[\widehat{\mathbf{w}}_*-\mathbf{w}_*]_{\mathcal{S}}\|_1. \qquad (36)$$

If $\tau \geq 2\|\mathbf{q}\|_\infty$, then we have

$$\frac{\lambda}{2}\|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|_2^2 \leq \frac{3\tau}{2}\|[\widehat{\mathbf{w}}_* - \mathbf{w}_*]_{\mathcal{S}}\|_1 \tag{37}$$

$$\|[\widehat{\mathbf{w}}_*]_{\mathcal{S}^c}\|_1 \leq 3\|[\widehat{\mathbf{w}}_* - \mathbf{w}_*]_{\mathcal{S}}\|_1. \tag{38}$$

Note that the inequality (38) hold regardless the value of λ. Since

$$\|[\widehat{\mathbf{w}}_* - \mathbf{w}_*]_{\mathcal{S}}\|_1 \leq \sqrt{s}\|[\widehat{\mathbf{w}}_* - \mathbf{w}_*]_{\mathcal{S}}\|_2,$$
$$\|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|_2 \geq \max(\|[\widehat{\mathbf{w}}_* - \mathbf{w}_*]_{\mathcal{S}}\|_2, \|[\widehat{\mathbf{w}}_*]_{\mathcal{S}^c}\|_2),$$

by combining the above inequalities with (37), we can get

$$\|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|_2 \leq \frac{3\tau}{\lambda}\sqrt{s}, \quad \|[\widehat{\mathbf{w}}_* - \mathbf{w}_*]_{\mathcal{S}}\|_1 \leq \frac{3\tau}{\lambda}s,$$

and

$$\begin{aligned}\|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|_1 &\leq \|[\widehat{\mathbf{w}}_*]_{\mathcal{S}^c}\|_1 + \|[\widehat{\mathbf{w}}_* - \mathbf{w}_*]_{\mathcal{S}}\|_1 \\ &\leq 3\|[\widehat{\mathbf{w}}_* - \mathbf{w}_*]_{\mathcal{S}}\|_1 + \|[\widehat{\mathbf{w}}_* - \mathbf{w}_*]_{\mathcal{S}}\|_1 \\ &\leq \frac{12\tau}{\lambda}s.\end{aligned}$$

We can then complete the proof of Theorem 2 by noting the upper bound of $\|\mathbf{q}\|_\infty$ in the following lemma and by setting γ according to the Theorem.

Lemma 9 *Let $A \in \mathbb{R}^{m\times n}$ be a random matrix such that Assumption 1. With a probability at least $1-\delta$, we have*

$$\|\mathbf{q}\|_\infty \leq \frac{c\eta R}{n}\sqrt{\frac{\log(2d/\delta)}{m}},$$

where $R = \max_j \|\bar{\mathbf{x}}_j\|$ and c is the quantity that appears in Assumption 1.

The lemma can be proved similarly as the Lemma 8.

7.4 Proof of Theorem 5

When $\lambda = 0$, the reduced problem becomes

$$\widehat{\mathbf{w}}_* = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \underbrace{\frac{1}{2n}\|\widehat{X}\mathbf{w} - \widehat{\mathbf{y}}\|_2^2 + (\gamma+\tau)\|\mathbf{w}\|_1}_{\widehat{F}(\mathbf{w})}. \tag{39}$$

Recall the original optimization problem:

$$\mathbf{w}_* = \arg\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2n}\|X\mathbf{w} - \mathbf{y}\|_2^2 + \gamma\|\mathbf{w}\|_1.$$

From the proof of Theorem 4, we have

$$\|[\widehat{\mathbf{w}}_*]_{\mathcal{S}^c}\|_1 \leq 3\|[\widehat{\mathbf{w}}_* - \mathbf{w}_*]_{\mathcal{S}}\|_1, \quad \text{and} \quad \frac{\|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|_1}{\|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|_2} = \frac{4\|[\widehat{\mathbf{w}}_* - \mathbf{w}_*]_{\mathcal{S}}\|_1}{\|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|_2} \leq 4\sqrt{s}.$$

Thus $\frac{\widehat{\mathbf{w}}_*-\mathbf{w}_*}{\|\widehat{\mathbf{w}}_*-\mathbf{w}_*\|_2} \in \mathcal{K}_{d,16s}$. Then we proceed our proof as follows. Since $\mathbf{w}_*$ optimizes $F(\mathbf{w}_*)$, we have for any $g \in \partial\|\widehat{\mathbf{w}}_*\|_1$

$$0 \geq F(\mathbf{w}_*) - F(\widehat{\mathbf{w}}_*) \geq (\mathbf{w}_* - \widehat{\mathbf{w}}_*)^\top \left(\frac{1}{n}X^\top X\widehat{\mathbf{w}}_* - \frac{1}{n}X^\top \mathbf{y} + \gamma g\right) + \frac{1}{2n}(\mathbf{w}_* - \widehat{\mathbf{w}}_*)^\top X^\top X(\mathbf{w}_* - \widehat{\mathbf{w}}_*).$$

Since $\widehat{\mathbf{w}}_*$ optimizes $\widehat{F}(\mathbf{w})$, by the first order optimality condition there exists $h \in \partial\|\widehat{\mathbf{w}}_*\|_1$ such that

$$0 \geq (\widehat{\mathbf{w}}_* - \mathbf{w}_*)^\top \left(\frac{1}{n}\widehat{X}^\top \widehat{X}\widehat{\mathbf{w}}_* - \frac{1}{n}\widehat{X}^\top \widehat{\mathbf{y}}\right) + (\gamma + \tau)(\widehat{\mathbf{w}}_* - \mathbf{w}_*)^\top h.$$

Combining the two inequalities above we have

$$\begin{aligned}
0 \geq\ & (\mathbf{w}_* - \widehat{\mathbf{w}}_*)^\top \left(\frac{1}{n}X^\top X\widehat{\mathbf{w}}_* - \frac{1}{n}X^\top \mathbf{y} - \frac{1}{n}\widehat{X}^\top \widehat{X}\widehat{\mathbf{w}}_* + \frac{1}{n}\widehat{X}^\top \widehat{\mathbf{y}}\right) \\
& + (\widehat{\mathbf{w}}_* - \mathbf{w}_*)^\top (\gamma h + \tau h - \gamma g) \\
& + \frac{1}{2n}(\mathbf{w}_* - \widehat{\mathbf{w}}_*)^\top X^\top X(\mathbf{w}_* - \widehat{\mathbf{w}}_*) \\
=\ & (\mathbf{w}_* - \widehat{\mathbf{w}}_*)^\top \left(\frac{1}{n}X^\top X\mathbf{w}_* - \frac{1}{n}X^\top \mathbf{y} - \frac{1}{n}\widehat{X}^\top \widehat{X}\mathbf{w}_* + \frac{1}{n}\widehat{X}^\top \widehat{\mathbf{y}}\right) \\
& + (\widehat{\mathbf{w}}_* - \mathbf{w}_*)^\top (\gamma h + \tau h - \gamma g) \\
& + \frac{1}{2n}(\mathbf{w}_* - \widehat{\mathbf{w}}_*)^\top X^\top X(\mathbf{w}_* - \widehat{\mathbf{w}}_*) \\
& + (\mathbf{w}_* - \widehat{\mathbf{w}}_*)^\top \left(\frac{1}{n}X^\top X(\widehat{\mathbf{w}}_* - \mathbf{w}_*) - \frac{1}{n}\widehat{X}^\top \widehat{X}(\widehat{\mathbf{w}}_* - \mathbf{w}_*)\right) \\
=\ & (\mathbf{w}_* - \widehat{\mathbf{w}}_*)^\top \left(\frac{1}{n}X^\top X\mathbf{w}_* - \frac{1}{n}X^\top \mathbf{y} - \frac{1}{n}\widehat{X}^\top \widehat{X}\mathbf{w}_* + \frac{1}{n}\widehat{X}^\top \widehat{\mathbf{y}}\right) \\
& + (\widehat{\mathbf{w}}_* - \mathbf{w}_*)^\top (\gamma h + \tau h - \gamma g) \\
& + \frac{1}{2n}(\mathbf{w}_* - \widehat{\mathbf{w}}_*)^\top X^\top X(\mathbf{w}_* - \widehat{\mathbf{w}}_*) \\
& + (\mathbf{w}_* - \widehat{\mathbf{w}}_*)^\top \left(\frac{1}{n}X^\top X - \frac{1}{n}\widehat{X}^\top \widehat{X}\right)(\widehat{\mathbf{w}}_* - \mathbf{w}_*).
\end{aligned}$$

By setting $g_i = h_i, i \in \mathcal{S}$ and following the same analysis as in the proof of Theorem 4, we have

$$(\widehat{\mathbf{w}}_* - \mathbf{w}_*)^\top (\gamma h + \tau h - \gamma g) \geq -\tau\|[\widehat{\mathbf{w}}_* - \mathbf{w}_*]_{\mathcal{S}}\|_1 + \tau\|[\widehat{\mathbf{w}}_*]_{\mathcal{S}^c}\|_1.$$

As a result,

$$0 \geq -\|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|_1\|\mathbf{q}\|_\infty - \tau\|[\widehat{\mathbf{w}}_* - \mathbf{w}_*]_{\mathcal{S}}\|_1 + \tau\|[\widehat{\mathbf{w}}_*]_{\mathcal{S}^c}\|_1 + \frac{\rho^-_{d,16s}}{2}\|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|_2^2 - \sigma_{d,16s}\|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|_2^2.$$

Then we arrive at the same inequality as in (36) with λ replaced by $\rho^-_{d,16s} - 2\sigma_{d,16s}$. Then the same analysis will follow to prove Theorem 5.

Table 1 Statistics of datasets

Name	#Training	#Testing	#Features
RCV1	677,399	20,242	47,236
KDD	8,407,752	748,401	29,890,095
Splice	1,000,000	4,627,840	12,495,340
E2006-tfidf	16,087	3308	150,360

8 Extensions

8.1 Recovery error with nearly sparse dual solution for classification

In this section, we provide a theoretical result on the recovery error for the nearly sparse optimal dual variable α_*. We state the result for smooth loss functions. To quantify the near sparsity, we let $\alpha_*^s \in \mathbb{R}^n$ denote a vector that zeros all entries in α_* except for the top-s elements in magnitude and assume α_*^s satisfies the following condition:

$$\left\| \nabla \ell^*(\alpha_*^s) + \frac{1}{\lambda n} X X^\top \alpha_*^s \right\|_\infty \leq \xi, \tag{40}$$

where $\nabla \ell^*(\alpha) = (\nabla \ell_1^*(\alpha_1), \ldots, \nabla \ell_n^*(\alpha_n))^\top$. The above condition can be considered as a sub-optimality condition (Boyd and Vandenberghe 2004) of α_*^s measured in the infinite norm. For the optimal solution α_* that lies in the interior of its domain, we have $\nabla \ell^*(\alpha_*) + \frac{1}{\lambda n} X X^\top \alpha_* = 0$.

Theorem 6 *Let $A \in \mathbb{R}^{m \times d}$ be a random matrix sampled from a distribution that satisfies the JL lemma. Let $\widetilde{\alpha}_*$ be the optimal dual solution to (7). Assume α_* is nearly s-sparse such that (40) holds, $\max_i \|\mathbf{x}_i\|_2 \leq R$ and $\ell(z)$ is L-smooth. If we set $\tau \geq \frac{2}{\lambda n} \|(X X^\top - \widehat{X} \widehat{X}^\top) \alpha_*\|_\infty + 2\xi$, then we have*

$$\|\widetilde{\alpha}_* - \alpha_*^s\|_2 \leq 3\tau L \sqrt{s}, \quad \|\widetilde{\alpha}_* - \alpha_*^s\|_1 \leq 12 \tau L s.$$

Remark 8 The proof appears in "Appendix A". Compared to Theorem 1 for exactly sparse optimal dual solution, the dual recovery error bound for nearly sparse optimal dual solution is increased by $6L\sqrt{s}\xi$ for ℓ_2 norm and by $24Ls\xi$ for ℓ_1 norm.

9 Numerical experiments

In this section, we provide experimental results to complement the theoretical analysis. We use four datasets for our experiments, whose statistics are summarized in Table 1. Among them, RCV1, KDD and Splice are for the classification task and E2006-tfidf is for the regression task.

9.1 Classification

We first study the recovery error for classification. We use the RCV1-binary data (Lewis et al. 2004) to conduct the 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 ℓ_2 norm is equal to 1. We only report the results using random hashing as in (20)

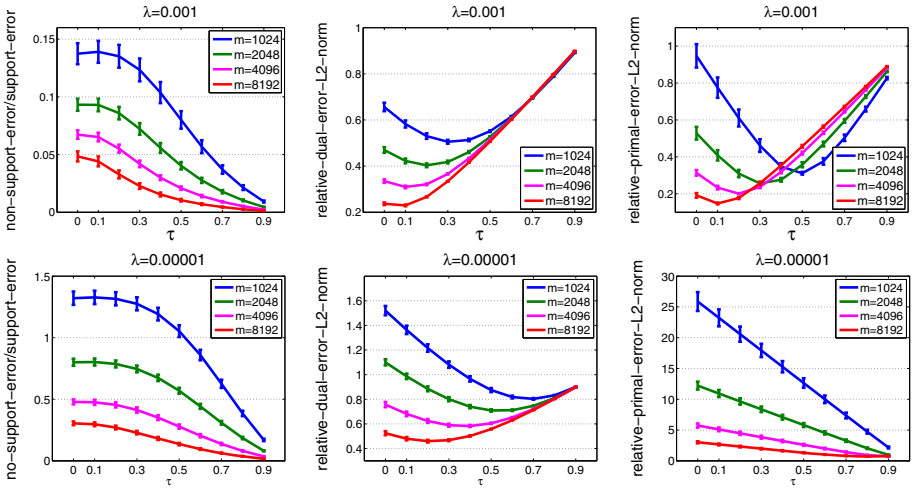

Fig. 1 Recovery error for squared hinge loss. From left to right: $\frac{\|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1}{\|[\widetilde{\alpha}_*]_{\mathcal{S}}-[\alpha_*]_{\mathcal{S}}\|_1}$ versus τ, $\frac{\|\widetilde{\alpha}_*-\alpha_*\|_2}{\|\alpha_*\|_2}$ versus τ, and $\frac{\|\widetilde{\mathbf{w}}_*-\mathbf{w}_*\|_2}{\|\mathbf{w}_*\|_2}$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

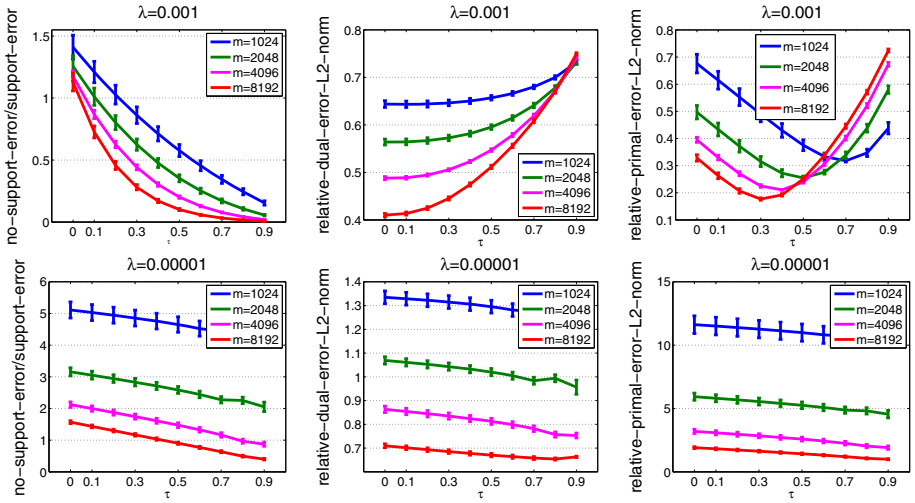

Fig. 2 Same curves as in Fig. 1 but for non-smooth hinge loss

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 τ affect the recovery error? (ii) how does the number of samples m affect the recovery error? We vary the value of τ among $0, 0.1, 0.2, \ldots, 0.9$, the value of m among 1024, 2048, 4096, 8192, and the value of λ among 0.001, 0.00001. Note that $\tau = 0$ corresponds to the dual recovery approach proposed in Zhang et al. (2013). The results averaged over 5 random trials are shown in Fig. 1 for the squared hinge loss and in Fig. 2 for the hinge loss. We first analyze the results in Fig. 1. We can observe

that when τ increases the ratio of $\frac{\|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1}{\|[\widetilde{\alpha}_*]_{\mathcal{S}}-[\alpha_*]_{\mathcal{S}}\|_1}$ 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 τ becomes larger than a certain threshold making $\tau > \|\Delta\|_\infty$ hold, then Theorem 1 implies that a larger τ will lead to a larger error. On the other hand, when τ is less than the threshold, the dual recovery error will decrease as τ increases, verifying that adding a sparse regularizer is very important. In addition, the figures exhibit that the thresholds for larger m are smaller which is consistent with our analysis of $\|\Delta\|_\infty = O(\sqrt{1/m})$. The difference between $\lambda = 0.001$ and $\lambda = 0.00001$ is because that smaller λ will lead to larger $\|\mathbf{w}_*\|_2$. 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 λ is small.

Next, we compare the classification performance of different randomized methods. We compare the proposed randomized method by solving a sparse regularized dual formulation and recovering a high-dimensional model by dual recovery (referred to as DSRR), and the previous random projection with dual recovery (referred to as RPDR), and the standard random projection approach (referred to as RP) that learns a low-dimensional model from random sketched data. For each method, we use three randomized reduction methods, namely random Gaussian projection (RG), random hashing (RH) and random sampling (RS). Note comparing to RS that does not satisfy the JL lemma in general allows us to verify that the JL property is very important for maintaining good performance. The loss function is fixed to the hinge loss, the regularization parameter is set to 10^{-5}, and the sparse regularization parameter in DSRR is set to 0.9. We test on two data sets, namely RCV1 used in the first experiment and Splice site data (Sonnenburg and Franc 2010). For Splice site data, we use a subset of data that contains 10^6 training examples, 4,627,840 testing examples and 12,495,340 features, and we evaluate the classification performance by the measure of area under precision recall curve (auPRC)[4] as in Sonnenburg and Franc (2010) due to that the data is highly imbalanced. We show the results in Fig. 3 for two values of the reduced dimensionality $m = 500$ and $m = 1000$. From the results we can see that (i) RS usually performs worse than RG and RH, which verifies that satisfying the JL property is very important for the randomized reduction methods; (ii) there is no clear winner between RG and RH; however RH is much more efficient RG; (iii) DSRR performs similarly to RPDR on RCV1 data and better than RPDR on Splice site data, implying that DSRR is favorable than RPDR; (iv) both DSRR and RPDR performs better than RP, verifying that recovering an accurate high-dimensional model benefits the prediction performance.

Runtime Speedup by Distributed Learning: Although in some cases the recovered solution or 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 solving 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 $\mathbf{w} \in \mathbb{R}^d$) 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

[4] The higher the auPRC the better the performance.

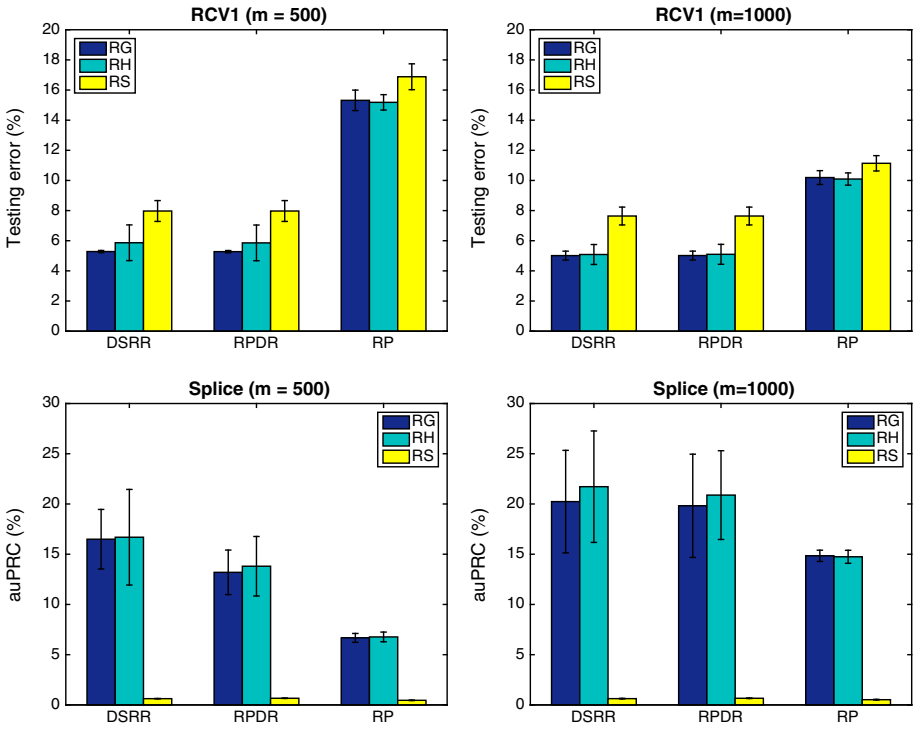

Fig. 3 Classification performance of different randomized methods on RCV1 ad Splice data

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 method that only uses the low-dimensional model learned by solving the reduced problem (7) with DisDCA, which is referred to as DSRR-ld, (ii) the method that uses the recovered model in the original space with the dual solution learned by solving the reduced problem (7) with DisDCA, referred to as DSRR-hd; (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 dual 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. RCV1 data is evenly distributed over 5 machines and KDD data is evenly distributed over 10 machines. The results averaged over 5 trials are shown in Fig. 4, which exhibit that the

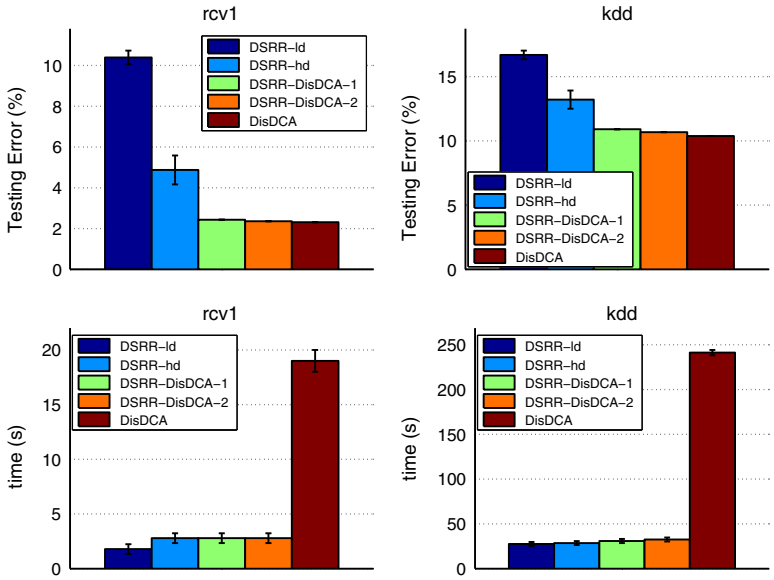

Fig. 4 Top: Testing error for different methods. Bottom: Training time for different methods. The value of $\lambda = 10^{-5}$ and the value of $\tau = 0.9$. The high-dimensional features are reduced to $m = 1024$-dimensional space using random hashing. The loss function is the squared hinge loss

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 with about 5 times speedup. In addition, DSRR-DisDCA performs much better than DSRR-ld and has small computational overhead.

9.2 SLSR

In this subsection, we present some numerical experiments for SLSR. We conduct experiments on two datasets, a synthetic dataset and a real dataset. The synthetic data is generated similar to previous studies on sparse signal recovery (Xiao and Zhang 2013). In particular, we generate a random matrix $X \in \mathbb{R}^{n\times d}$ with $n = 10^4$ and $d = 10^5$. The entries of the matrix X are generated independently with the uniform distribution over the interval $[-1, +1]$. A sparse vector $\mathbf{u}_* \in \mathbb{R}^d$ is generated with the same distribution at 100 randomly chosen coordinates. The noise $\xi \in \mathbb{R}^n$ is a dense vector with independent random entries with the uniform distribution over the interval $[-\sigma, \sigma]$, where σ is the noise magnitude and is set to 0.1. We scale the data matrix X such that all entries have a variance of $1/n$ and scale the noise vector ξ accordingly. Finally the vector $\mathbf{y}$ was obtained as $\mathbf{y} = X\mathbf{u}_* + \xi$. For elastic net on the synthetic data, we try two different values of λ, 10^{-8} and 10^{-5}. The value of γ is set to 10^{-5} for both elastic net and lasso. Note that these values are not intended to optimize the performance of elastic net and lasso on the synthetic data. The real data used in the experiment is E2006-tfidf dataset. We use the version available on libsvm website.[5] There are a total of $n = 16,087$ training instances and $d = 150,360$ features and 3308 testing instances. We normalize the training data such that each dimension has mean zero

[5] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

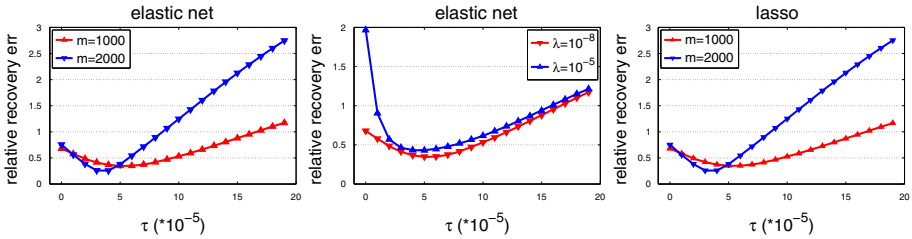

Fig. 5 Recovery error of elastic net and lasso under different settings on the synthetic data

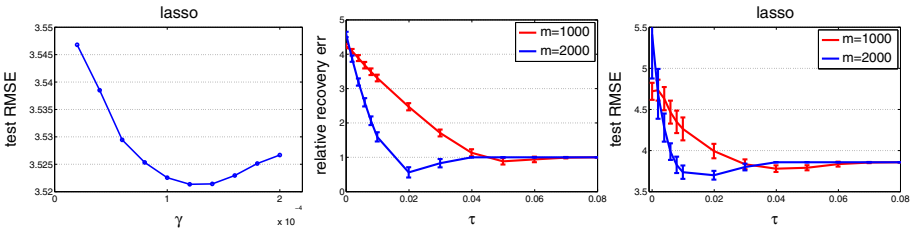

Fig. 6 Recovery or Regression error of lasso under different settings on the E2006-tfidf

and variance $1/n$. The testing data is normalized using the statistics computed on the training data. For JL transform, we use the random matrix with one block of random hashing. The experimental results on the synthetic data under different settings are shown in Fig. 5. In the left plot, we compare the recovery error for elastic net with $\lambda = 10^{-8}$ and two different values of m, i.e., $m = 1000$ and $m = 2000$. The horizontal axis is the value of σ, the added regularization parameter. We can observe that adding a slightly larger additional ℓ_1 norm to the compressed data problem indeed reduces the recovery error. When the value of τ is larger than some threshold, the error will increase, which is consistent with our theoretical results. In particular, we can see that the threshold value for $m = 2000$ is smaller than that for $m = 1000$. In the middle plot, we compare the recovery error for elastic net with $m = 1000$ and two different values of the regularization parameter λ. Similar trends of the recovery error versus σ are also observed. In addition, it is interesting to see that the recovery error for $\lambda = 10^{-8}$ is less than that for $\lambda = 10^{-5}$, which seems to contradict to the theoretical results at the first glance due to the explicit inverse dependence on λ. However, the optimization error also depends on $\|\mathbf{e}\|_2$, which measures the empirical error of the corresponding optimal model. We find that with $\lambda = 10^{-8}$ we have a smaller $\|\mathbf{e}\|_2 = 0.95$ compared to 1.34 with $\lambda = 10^{-5}$, which explains the result in the middle plot. For the right plot, we repeat the same experiments for lasso as in the left plot for elastic net, and observe similar results.

The experimental results on E2006-tfidf dataset for lasso are shown in Fig. 6. In the left plot, we show the root mean square error (RMSE) on the testing data of different models learned from the original data with different values of γ. In the middle and right plots, we fix the value of $\gamma = 10^{-4}$ and increase the value of τ and plot the relative recovery error and the RMSE on the testing data. Again, the empirical results are consistent with the theoretical results and verify that with random sketched data a larger ℓ_1 regularizer could yield a better performance.

10 Conclusions

In this paper, we have proposed new theory of high-dimensional model recovery from random sketched data. We have studied two important problems in machine learning, i.e., classification and sparse least-squares regression. We propose to explore the intrinsic sparsity of the problem and develop new formulations on the sketched data and novel analysis of the recovery error. The numerical experiments validate our theoretical analysis and also demonstrate that the proposed randomized methods are favorable in comparison with previous randomized algorithms.

A Proof of Theorem 6

Let $\hat{F}(\alpha)$ be defined as

$$\hat{F}(\alpha) = \frac{1}{n}\sum_{i=1}^{n} \ell_i^*(\alpha_i) + \frac{1}{2\lambda n^2}\alpha^T \widehat{X}\widehat{X}^\top \alpha + \frac{\tau}{n}\|\alpha\|_1,$$

and $F(\alpha)$ be defined as

$$F(\alpha) = \frac{1}{n}\sum_{i=1}^{n} \ell_i^*(\alpha_i) + \frac{1}{2\lambda n^2}\alpha^T XX^\top \alpha.$$

Since $\widetilde{\alpha}_* = \arg\min \hat{F}(\alpha)$ therefore for any $g_* \in \partial\|\alpha_*^s\|_1$

$$0 \geq \hat{F}(\widetilde{\alpha}_*) - \hat{F}(\alpha_*^s) \geq (\widetilde{\alpha}_* - \alpha_*^s)^\top \left(\frac{1}{n}\nabla \ell^*(\alpha_*^s) + \frac{1}{\lambda n^2}\widehat{X}\widehat{X}^\top \alpha_*^s\right) + \frac{\tau}{n}(\widetilde{\alpha}_* - \alpha_*^s)^\top g_* + \frac{1}{2nL}\|\widetilde{\alpha}_* - \alpha_*^s\|_2^2,$$

where we used the strong convexity of ℓ_i^* and its strong convexity modulus $1/L$. Due to the sub-optimality of α_*^s, we have

$$\frac{1}{n}\|\alpha_*^s - \widetilde{\alpha}_*\|_1 \xi \geq (\widetilde{\alpha}_* - \alpha_*^s)^\top \left[\frac{1}{n}\nabla \ell^*(\alpha_*^s) + \frac{1}{\lambda n^2} XX^\top \alpha_*^s\right].$$

Combining the above two inequalities we have

$$\frac{1}{n}\|\alpha_*^s - \widetilde{\alpha}_*\|_1 \xi \geq (\widetilde{\alpha}_* - \alpha_*^s)^\top \left(\frac{1}{\lambda n^2}(\widehat{X}\widehat{X}^\top - XX^\top)\alpha_*^s\right) + \frac{\tau}{n}(\widetilde{\alpha}_* - \alpha_*^s)^\top g_* + \frac{1}{2nL}\|\widetilde{\alpha}_* - \alpha_*^s\|_2^2.$$

Since the above inequality holds for any $g_* \in \partial\|\alpha_*^s\|_1$, if we choose $[g_*]_i = sign([\widetilde{\alpha}_*]_i)$, $i \in \mathcal{S}^c$, then we have

$$(\xi + \tau)\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1 \geq -\|\Delta\|_\infty \|\widetilde{\alpha}_* - \alpha_*^s\|_1 + (\tau - \xi)\|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1 + \frac{1}{2L}\|\widetilde{\alpha}_* - \alpha_*^s\|_2^2. \tag{41}$$

Thus

$$(\tau + \xi + \|\Delta\|_\infty)\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1 \geq (\tau - \xi - \|\Delta\|_\infty)\|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1 + \frac{1}{2L}\|\widetilde{\alpha}_* - \alpha_*^s\|_2^2.$$

Assuming $\tau \geq 2(\|\Delta\|_\infty + \xi)$, we have

$$\begin{aligned} \|\widetilde{\alpha}_* - \alpha_*^s\|_2^2 &\leq 3\tau L \|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1 \\ \|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1 &\leq 3\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1. \end{aligned} \tag{42}$$

Therefore,

$$\frac{\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1^2}{s} \leq \|\widetilde{\alpha}_* - \alpha_*^s\|_2^2 \leq 3\tau L \|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1.$$

leading to the result

$$\|[\widetilde{\alpha}_*]_{\mathcal{S}} - [\alpha_*]_{\mathcal{S}}\|_1 \leq 3\tau L s.$$

Combing above inequality with inequalities in (42) we have

$$\|[\widetilde{\alpha}_*]_{\mathcal{S}^c}\|_1 \leq 9\tau L s, \quad \|\widetilde{\alpha}_* - \alpha_*^s\|_2 \leq 3\tau L\sqrt{s}.$$

B Proof of Lemmas 2 and 3

Since the two lemmas are equivalent, we use the notation in Lemma 2.

The key idea is to use the convex relaxation of $\mathcal{K}_{n,s}$. Define $\mathcal{S}_{n,s} = \{\alpha \in \mathbb{R}^n : \|\alpha\|_2 \leq 1, \|\alpha\|_0 \leq s\}$. It was shown in Plan and Vershynin (2011) that $conv(\mathcal{S}_{n,s}) \subset \mathcal{K}_{n,s} \subset 2conv(\mathcal{S}_{n,s})$, where $conv(\mathcal{S})$ is the convex hull of the set $\mathcal{S}$. Then for any $\alpha \in \mathcal{K}_{n,s}$, we can write it as $\alpha = 2\sum_i \lambda_i \beta_i$ where $\beta_i \in \mathcal{S}_{n,s}$, $\sum_i \lambda_i = 1$ and $\lambda_i \geq 0$. Thus, we have

$$\begin{aligned} |(X^\top\alpha)^\top(I - A^\top A)(X^\top\alpha)| &\leq 4\left|\left(X\sum_i \lambda_i\beta_i\right)^\top (I - A^\top A)\left(X\sum_i \lambda_i\beta_i\right)\right| \\ &\leq 4\sum_{ij}\lambda_i\lambda_j|(X^\top\beta_i)^\top(I - A^\top A)(X^\top\beta_j)| \\ &\leq 4\max_{\alpha_1,\alpha_2\in\mathcal{S}_{n,s}}|(X^\top\alpha_1)^\top(I - A^\top A)(X^\top\alpha_2)|\sum_{ij}\lambda_i\lambda_j \\ &= 4\max_{\alpha_1,\alpha_2\in\mathcal{S}_{n,s}}|(X^\top\alpha_1)^\top(I - A^\top A)(X^\top\alpha_2). \end{aligned}$$

Therefore

$$\sigma_{n,s} = \max_{\alpha\in\mathcal{K}_{n,s}}\frac{1}{n}|(X^\top\alpha)^\top(I - A^\top A)(X^\top\alpha)| \leq 4\max_{\alpha_1,\alpha_2\in\mathcal{S}_{n,s}}\frac{1}{n}|(X\alpha_1)^\top(I - A^\top A)(X\alpha_2)|.$$

Let $\mathbf{u}_1 = X^\top\alpha_1$ and $\mathbf{u}_2 = X^\top\alpha_2$. Following the proof of Theorem 5, for any fixed $\alpha_1, \alpha_2 \in \mathcal{S}_{n,s}$, with a probability $1 - 2\delta$ we have

$$\frac{1}{n}|(X^\top\alpha_1)^\top(I - A^\top A)(X^\top\alpha_2)| \leq \frac{1}{n}\|X^\top\alpha_1\|_2\|X^\top\alpha_2\|_2 c\sqrt{\frac{\log(1/\delta)}{m}} \leq \rho_{n,s}^+ c\sqrt{\frac{\log(1/\delta)}{m}},$$

where we use

$$\max_{\alpha\in\mathcal{S}_{n,s}}\frac{\|X^\top\alpha\|_2}{\sqrt{n}} \leq \max_{\alpha\in\mathcal{K}_{n,s}}\frac{\|X^\top\alpha\|_2}{\sqrt{n}} = \sqrt{\rho_{n,s}^+}.$$

In order to extend the inequity to all $\alpha_1, \alpha_2 \in \mathcal{S}_{n,s}$. We consider the ϵ proper-net of $\mathcal{S}_{n,s}$ (Plan and Vershynin 2011) denoted by $\mathcal{S}_{n,s}(\epsilon)$. Lemma 3.3 in Plan and Vershynin (2011) shows that the entropy of $\mathcal{S}_{n,s}$, i.e., the cardinality of $\mathcal{S}_{n,s}(\epsilon)$ denoted $N(\mathcal{S}_{n,s}, \epsilon)$ is bounded by

$$\log N(\mathcal{S}_{n,s}, \epsilon) \leq s \log\left(\frac{9n}{\epsilon s}\right).$$

Then by using the union bound, we have with a probability $1 - 2\delta$, we have

$$\begin{aligned}\max_{\substack{\alpha_1 \in \mathcal{S}_{n,s}(\epsilon) \\ \alpha_2 \in \mathcal{S}_{n,s}(\epsilon)}} \frac{1}{n} |(X^\top \alpha_1)^\top (I - A^\top A)(X^\top \alpha_2)| &\leq c\rho_{n,s}^+ \sqrt{\frac{\log(N^2(\mathcal{S}_{n,s}, \epsilon)/\delta)}{m}} \\ &\leq c\rho_{n,s}^+ \sqrt{\frac{\log(1/\delta) + 2s\log(9n/\epsilon s)}{m}}. \end{aligned} \tag{43}$$

To proceed the proof, we need the following lemma.

Lemma 10 *Let* $U = \frac{1}{n} X(I - A^\top A)X^\top$, *and Let*

$$\begin{aligned}\mathcal{E}_s(\alpha_2) &= \max_{\alpha_1 \in \mathcal{S}_{n,s}} |\alpha_1^\top U \alpha_2|, \\ \mathcal{E}_s(\alpha_2, \epsilon) &= \max_{\alpha_1 \in \mathcal{S}_{n,s}(\epsilon)} |\alpha_1^\top U \alpha_2|.\end{aligned}$$

For $\epsilon \in (0, 1/\sqrt{2})$, *we have*

$$\mathcal{E}_s(\alpha_2) \leq \left(\frac{1}{1 - \sqrt{2}\epsilon}\right) \mathcal{E}_s(\alpha_2, \epsilon).$$

Proof Following Lemma 9.2 of Koltchinskii (2011), for any $\alpha, \alpha' \in \mathcal{S}_{n,s}$, we can always find two vectors β, β' such that

$$\alpha - \alpha' = \beta - \beta', \ \|\beta\|_0 \leq s, \ \|\beta'\|_0 \leq s, \ \beta^\top \beta' = 0.$$

Let

$$\begin{aligned}\mathcal{E}_s(\alpha_2) &= \max_{\alpha_1 \in \mathcal{S}_{n,s}} |\alpha_1^\top U \alpha_2|, \\ \mathcal{E}_s(\alpha_2, \epsilon) &= \max_{\alpha_1 \in \mathcal{S}_{n,s}(\epsilon)} |\alpha_1^\top U \alpha_2|.\end{aligned}$$

Thus

$$\begin{aligned}&|\langle \alpha - \alpha', U\alpha_2 \rangle| \leq |\langle \beta, U\alpha_2 \rangle| + |\langle -\beta', U\alpha_2 \rangle| \\ &= \|\beta\|_2 \left|\left\langle \frac{\beta}{\|\beta\|_2}, U\alpha_2 \right\rangle\right| + \|\beta'\|_2 \left|\left\langle \frac{-\beta'}{\|\beta'\|_2}, U\alpha_2 \right\rangle\right| \\ &\leq (\|\beta\|_2 + \|\beta'\|_2)\mathcal{E}_s(\alpha_2) \leq \mathcal{E}_s(\alpha_2)\sqrt{2}\sqrt{\|\beta\|_2^2 + \|\beta'\|_2^2} \\ &= \mathcal{E}_s(\alpha_2)\sqrt{2}\|\beta - \beta'\|_2 = \mathcal{E}_s(\alpha_2)\sqrt{2}\|\beta - \beta'\|_2 = \mathcal{E}_s(\alpha_2)\sqrt{2}\|\alpha - \alpha'\|_2.\end{aligned}$$

Then, we have

$$\begin{aligned}\mathcal{E}_s(\alpha_2) = \max_{\alpha \in \mathcal{S}_{n,s}} |\alpha^\top U \alpha_2| &\leq \max_{\alpha \in \mathcal{S}_{n,s}(\epsilon)} |\alpha^\top U \alpha_2| + \sup_{\substack{\alpha \in \mathcal{S}_{n,s} \\ \alpha' \in \mathcal{S}_{n,s}(\epsilon), \|\alpha - \alpha'\|_2 \leq \epsilon}} \langle \alpha - \alpha', U\alpha_2 \rangle \\ &\leq \mathcal{E}_s(\alpha_2, \epsilon) + \sqrt{2}\epsilon \mathcal{E}_s(\alpha_2),\end{aligned}$$

which implies

$$\mathcal{E}_s(\alpha_2) \leq \frac{\mathcal{E}_s(\alpha_2, \epsilon)}{1 - \sqrt{2}\epsilon}.$$

□

Lemma 11 *Let*

$$\mathcal{E}_s(\epsilon) = \max_{\alpha_2 \in \mathcal{S}_{n,s}} \mathcal{E}_s(\alpha_2, \epsilon) = \max_{\substack{\alpha_1 \in \mathcal{S}_{n,s} \\ \alpha_2 \in \mathcal{S}_{n,s}(\epsilon)}} |\alpha_1^\top U \alpha_2|,$$

$$\mathcal{E}_s(\epsilon, \epsilon) = \max_{\alpha_2 \in \mathcal{S}_{n,s}(\epsilon)} \mathcal{E}_s(\alpha_2, \epsilon) = \max_{\alpha_1, \alpha_2 \in \mathcal{S}_{n,s}(\epsilon)} |\alpha_1^\top U \alpha_2|.$$

For $\epsilon \in (0, 1/\sqrt{2})$*, we have*

$$\mathcal{E}_s(\epsilon) \leq \left(\frac{1}{1 - \sqrt{2}\epsilon}\right) \mathcal{E}_s(\epsilon, \epsilon).$$

The proof of the above lemma follows the same analysis as that of Lemma 10. By combining Lemmas 10 and 11, we have

$$\begin{aligned}
\sigma_{n,s} &\leq 4 \max_{\alpha_2 \in \mathcal{S}_{n,s}} \mathcal{E}_s(\alpha_2) \leq \frac{\max_{\alpha_2 \in \mathcal{S}_{n,s}} \mathcal{E}_s(\alpha_2, \epsilon)}{1 - \sqrt{2}\epsilon} \\
&= 4 \frac{1}{1 - \sqrt{2}\epsilon} \mathcal{E}_s(\epsilon) \leq 4 \left(\frac{1}{1 - \sqrt{2}\epsilon}\right)^2 \mathcal{E}_s(\epsilon, \epsilon) \\
&= 4 \left(\frac{1}{1 - \sqrt{2}\epsilon}\right)^2 \max_{\alpha_1, \alpha_2 \in \mathcal{S}_{n,s}(\epsilon)} |\alpha_1^\top U \alpha_2|.
\end{aligned}$$

By combing the above inequality with inequality (43), we have

$$\sigma_{n,s} \leq 4c \left(\frac{1}{1 - \sqrt{2}\epsilon}\right)^2 \rho_{n,s}^+ \sqrt{\frac{\log(1/\delta) + 2s \log(9n/\epsilon s)}{m}}.$$

If we set $\epsilon = 1/(2\sqrt{2})$, we can complete the proof.

References

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

Ailon, N. & Chazelle, B. (2006). Approximate nearest neighbors and the fast Johnson–Lindenstrauss transform. In *Proceedings of the ACM symposium on theory of computing (STOC)* (pp. 557–563).

Ailon, N., & Chazelle, B. (2009). The fast Johnson-Lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on Computing*, *39*(1), 302–322.

Balcan, M. F., Blum, A., & Vempala, S. (2006). Kernels as features: On kernels, margins, and low-dimensional mappings. *Machine Learning*, *65*(1), 79–94.

Bartz, D. Hatrick, K. Hesse, C. W. Müller, K. R. & Lemm, S. (2011). Directional variance adjustment: Improving covariance estimates for high-dimensional portfolio optimization. arXiv:1109.3069

Ben-Hur, A., Ong, C. S., Sonnenburg, S., Schölkopf, B., & Rätsch, G. (2008). Support vector machines and kernels for computational biology. *PLOS Computational Biology*, *4*, e1000173.

Bickel, P. J., Ritov, Y., & Tsybakov, A. B. (2009). Simultaneous analysis of Lasso and Dantzig selector. *Annals of Statistics*, *37*(4), 1705–1732.

Blum, A. (2005) Random projection, margins, kernels, and feature-selection. In *Proceedings of the 2005 international conference on subspace, latent structure and feature selection* (vol. 3940, pp. 52–68). Springer.

Boutsidis, C., & Gittens, A. (2013). Improved matrix algorithms via the subsampled randomized Hadamard transform. *SIAM Journal on Matrix Analysis and Applications*, *34*(3), 1301–1340.

Boutsidis, C., Mahoney, M. W., & Drineas, P. (2009). An improved approximation algorithm for the column subset selection problem. In *Proceedings of the twentieth annual ACM-SIAM symposium on discrete algorithms* (pp. 968–977).

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.

Brank, J., Grobelnik, M., Milić-Frayling, N., & Mladenić, D. (2002). 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).

Candes, E., & Tao, T. (2007). The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, *35*(6), 2313–2351.

Dasgupta, A., Kumar, R., & Sarlós, T. (2010). A sparse Johnson–Lindenstrauss transform. In *Proceedings of the 42nd ACM symposium on theory of computing, STOC '10* (pp. 341–350).

Dasgupta, S., & Gupta, A. (2003). An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, *22*(1), 60–65.

Defazio, A., Bach, F. R., & Lacoste-Julien, S. (2014). SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems (NIPS)* (pp. 1646–1654).

Drineas, P., Mahoney, M. W., & Muthukrishnan, S. (2006). Sampling algorithms for l2 regression and applications. In *ACM-SIAM symposium on discrete algorithms (SODA)* (pp. 1127–1136).

Drineas, P., Mahoney, M. W., & Muthukrishnan, S. (2008). Relative-error cur matrix decompositions. *SIAM Journal Matrix Analysis Applications*, *30*, 844–881.

Drineas, P., Mahoney, M. W., Muthukrishnan, S., & Sarlós, T. (2011). Faster least squares approximation. *Numerische Mathematik*, *117*(2), 219–249.

Eldar, Y. C., & Kutyniok, G. (2012). *Compressed sensing: Theory and applications*. Cambridge: Cambridge University Press.

Goldberger, J., Roweis, S., Hinton, G. & Salakhutdinov, R. (2005). Neighbourhood components analysis. In *Advances in neural information processing systems (NIPS)* (pp. 513–520).

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning (ML)*, *46*, 389–422.

Halko, N., Martinsson, P. G., & Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, *53*(2), 217–288.

Jia, J., & Rohe, K. (2015). Preconditioning the Lasso for sign consistency. *Electronic Journal of Statistics*, *9*(1), 1150–1172. https://doi.org/10.1214/15-EJS1029.

Johnson, R., & Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems (NIPS)* (pp. 315–323).

Johnson, W., & Lindenstrauss, J. (1984). Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Connecticut, 1982)* (vol. 26, pp. 189–206).

Jostins, L., & Barrett, J. C. (2011). Genetic risk prediction in complex disease. *Human Molecular Genetics*, *20*(R2), R182–R188.

Kakade, S. M., Shalev-Shwartz, S., & Tewari, A. (2009). *On the duality of strong convexity and strong smoothness: learning applications and matrix regularization*. Toyota Technological Institute: Technical report.

Kane, D. M., & Nelson, J. (2014). Sparser Johnson–Lindenstrauss transforms. *Journal of the ACM*, *61*, 4:1–4:23.

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

Koltchinskii, V. (2011). *Oracle inequalities in empirical risk minimization and sparse recovery problems*. Berlin: Springer.

Koltchinskii, V. (2011). *Oracle inequalities in empirical risk minimization and sparse recovery problems: École D'Été de Probabilités de Saint-Flour XXXVIII-2008*. Ecole d'été de probabilités de Saint-Flour: Springer.

Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research (JMLR)*, *5*, 361–397.

Lin, Q., Lu, Z., & Xiao, L. (2014). An accelerated proximal coordinate gradient method. In *NIPS* (pp. 3059–3067).

Ma, P., Mahoney, M. W., & Yu, B. (2014). A statistical perspective on algorithmic leveraging. In *Proceedings of the 31th International conference on machine learning (ICML)* (pp. 91–99).

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

Mahoney, M. W., & Drineas, P. (2009). Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, *106*(3), 697–702.
Maillard, O., & Munos, R. (2009). Compressed least-squares regression. In *NIPS* (pp. 1213–1221).
Mitchell, T. M., Hutchinson, R., Niculescu, R. S., Pereira, F., Wang, X., Just, M., et al. (2004). Learning to decode cognitive states from brain images. *Machine Learning*, *57*(1–2), 145–175.
Nelson, J. (2015). *Johnson–Lindenstrauss notes*. Technical report, MIT.
Nelson, J., & Nguyen, H. L. (2012). OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. CoRR. Retreived 2018. arXiv:abs/1211.1002.
Paul, D., Bair, E., Hastie, T., & Tibshirani, R. (2008). Preconditioning for feature selection and regression in high-dimensional problems. *The Annals of Statistics*, *36*, 1595–1618.
Paul, S., Boutsidis, C., Magdon-Ismail, M. & Drineas, P. (2013). Random projections for support vector machines. In *AISTATS* (pp. 498–506).
Pilanci, M., & Wainwright, M. J. (2015). Randomized sketches of convex programs with sharp guarantees. *IEEE Transactions on Information Theory*, *61*(9), 5096–5115.
Pilanci, M., & Wainwright, M. J. (2016). Iterative hessian sketch: Fast and accurate solution approximation for constrained least-squares. *Journal of Machine Learning Research*, *17*(1), 1842–1879.
Plan, Y., & Vershynin, R. (2011). One-bit compressed sensing by linear programming. CoRR. Retreived 2018. arXiv:abs/1109.4299.
Rätsch, G., Sonnenburg, S., & Schölkopf, B. (2005a). RASE: Recognition of alternatively spliced exons in C. elegans. *Bioinformatics*, *21*, i369–i377.
Ratsch, G., Sonnenburg, S., & Scholkopf, B. (2005b). 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).
Sánchez, J., Perronnin, F., Mensink, T., & Verbeek, J. J. (2013). Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, *105*(3), 222–245.
Shalev-Shwartz, S., & Zhang, T. (2013a). Accelerated mini-batch stochastic dual coordinate ascent. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 378–385).
Shalev-Shwartz, S., & Zhang, T. (2013b). Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research (JMLR)*, *14*, 567–599.
Shalev-Shwartz, S., & Zhang, T. (2014). Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *ICML* (pp. 64–72).
Shi, Q., Petterson, J., Dror, G., Langford, J., Smola, A., & Vishwanathan, S. (2009a). Hash kernels for structured data. *Journal of Machine Learning Research (JMLR)*, *10*, 2615–2637.
Shi, Q., Petterson, J., Dror, G., Langford, J., Smola, A. J., Strehl, A. L., & Vishwanathan, V. (2009b). Hash kernels. In *Proceedings of the international conference on artificial intelligence and statistics (AISTATS)* (pp. 496–503).
Shi, Q., Shen, C., Hill, R., & van den Hengel, A. (2012). Is margin preserved after random projection? In *Proceedings of the international conference on machine learning (ICML)*.
Simianer, P., Riezler, S., & Dyer, C. (2012). 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).
Sonnenburg, S., & Franc, V. (2010). Coffin: A computational framework for linear SVMS. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 999–1006).
Sonnenburg, S., Schweikert, G., Philips, P., Behr, J., & Ratsch, G. (2007). Accurate splice site prediction using support vector machines. *BMC Bioinformatics*, *8*, S7.
Sridharan, K., Shalev-Shwartz, S., & Srebro, N. (2008). Fast rates for regularized objectives. In *Advances in neural information processing systems (NIPS)* (pp. 1545–1552).
Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, *58*, 267–288.
Tropp, J. A. (2011). Improved analysis of the subsampled randomized hadamard transform. *Advances in Adaptive Data Analysis*, *3*(1–2), 115–126.
Wainwright, M. J. (2009). Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting. *IEEE Transactions on Information Theory*, *55*(12), 5728–5741.
Weinberger, K. Q., Dasgupta, A., Langford, J., Smola, A. J., & Attenberg, J. (2009). Feature hashing for large scale multitask learning. In *Proceedings of the international conference on machine learning (ICML)* (pp. 1113–1120).
Xiao, L., & Zhang, T. (2013). A proximal-gradient homotopy method for the sparse least-squares problem. *SIAM Journal on Optimization*, *23*(2), 1062–1091.
Xiao, L., & Zhang, T. (2014). A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, *24*(4), 2057–2075.

Yang, T. (2013). Trading computation for communication: Distributed stochastic dual coordinate ascent. In *Advances in neural information processing systems (NIPS)* (pp. 629–637).
Yen, I. E., Lin, T., Lin, S., Ravikumar, P. K., & Dhillon, I. S. (2014). Sparse random feature algorithm as coordinate descent in Hilbert space. In *NIPS* (pp. 2456–2464).
Zhang, L., Mahdavi, M., Jin, R., Yang, T., & Zhu, S. (2013). Recovering the optimal solution by dual random projection. In *Proceedings of the conference on learning theory (COLT)* (pp. 135–157).
Zhang, L., Mahdavi, M., Jin, R., Yang, T., & Zhu, S. (2014). Random projections for classification: A recovery approach. *IEEE Transactions on Information Theory (IEEE TIT)*, *60*(11), 7300–7316.
Zhao, P., & Yu, B. (2006). On model election consistency of lasso. *JMLR*, *7*, 2541–2563.
Zhou, S., Lafferty, J. D., & Wasserman, L. A. (2007). Compressed regression. In *NIPS* (pp. 1713–1720).
Zou, H., & Hastie, T. (2003). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*, 301–320.

Affiliations

Tianbao Yang[1] · Lijun Zhang[2] · Qihang Lin[3] · Shenghuo Zhu[4] · Rong Jin[4]

✉ Tianbao Yang
tianbao-yang@uiowa.edu

Lijun Zhang
zhanglj@lamda.nju.edu.cn

Qihang Lin
qihang-lin@uiowa.edu

Shenghuo Zhu
shenghuo@gmail.com

Rong Jin
jinrong.jr@alibaba-inc.com

[1] Department of Computer Science, The University of Iowa, Iowa City, IA 52242, USA

[2] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

[3] Department of Business Analytics, The University of Iowa, Iowa City, IA 52242, USA

[4] Alibaba Group, Seattle, WA, USA