Doubly Greedy Primal-Dual Coordinate Descent for Sparse Empirical Risk Minimization

Qi Lei [1] Ian E.H. Yen [2] Chao-yuan Wu [3] Inderjit S. Dhillon [1 3 4] Pradeep Ravikumar [2]

Abstract

We consider the popular problem of sparse empirical risk minimization with linear predictors and a large number of both features and observations. With a convex-concave saddle point objective reformulation, we propose a Doubly Greedy Primal-Dual Coordinate Descent algorithm that is able to exploit sparsity in both primal and dual variables. It enjoys a low cost per iteration and our theoretical analysis shows that it converges linearly with a good iteration complexity, provided that the set of primal variables is sparse. We then extend this algorithm further to leverage active sets. The resulting new algorithm is even faster, and experiments on large-scale Multi-class data sets show that our algorithm achieves up to 30 times speedup on several state-of-the-art optimization methods.

1. Introduction

Regularized empirical risk minimization with linear predictors is a key workhorse in machine learning. It has the following general form:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d}\left\{P(\boldsymbol{x})\overset{\text{def}}{=}\frac{1}{n}\sum_{i=1}^{n}\phi_i(\boldsymbol{a}_i^\top\boldsymbol{x})+g(\boldsymbol{x})\right\} \tag{1}$$

where $\boldsymbol{a}_i\in\mathbb{R}^d$ is one of the n data samples with d features. $\phi_i:\mathbb{R}\to\mathbb{R}$ is a convex loss function of the linear predictor $\boldsymbol{a}_i^\top\boldsymbol{x}$, for $i=1,\cdots,n$, and $g:\mathbb{R}^d\to\mathbb{R}$ is a convex regularization function for the coefficient vector $\boldsymbol{x}\in\mathbb{R}^d$. The loss function ϕ_i assigns a cost to the difference between the linear predictor $\boldsymbol{a}_i^\top\boldsymbol{x}$ and the associated label b_i.

With continuous and discrete b_i, (1) captures regression and classification problems respectively. As a popular instance, when $\phi_i(z)=\max\{0,1-b_iz\}$ and $g(\boldsymbol{x})=\mu/2\|\boldsymbol{x}\|_2^2$, (1) reduces to the linear SVM (support vector machine) classification problem. While setting $\phi_i(z)=\log(1+\exp(-b_iz))$, we obtain the logistic regression problem.

We are interested in developing efficient algorithms for solving this general problem (1) for the setting where the coefficient vector $\boldsymbol{x}$ is assumed to be sparse. Applications where such a sparsity is natural include large-scale multi-class/multi-label classification, low-degree polynomial data mapping (Chang et al., 2010), n-gram feature mapping (Sonnenburg & Franc, 2010), and random feature kernel machines (Rahimi & Recht, 2007), specifically with a sparsity constraint on the random features (Yen et al., 2014).

Our paper is organized as follows: In Section 2 we review existing algorithms to solve the primal, dual as well as primal-dual formulations of the problem (1). In Section 3, we present our Doubly Greedy Primal-Dual Coordinate Descent method for the convex-concave saddle point formulation of the problem (1). We propose an alternative method that is more efficient in practice with the use of incrementally increased active sets in both primal and dual variables. In Section 4 we show linear convergence for our proposed algorithm, and demonstrate the advantages of greedy methods with sparse variables. Finally in Section 5 we compare the performance of our method with other state-of-the-art methods on some real-world datasets, both with respect to time and iterations.

2. Formulation and related work

Notations: We use A to denote the data matrix, with rows $A_i=\boldsymbol{a}_i$ corresponding to samples, and the column A^j corresponding to features. We use $[n]$ to compactly denote $\{1,2,\cdots n\}$. Throughout the paper, $\|\cdot\|$ denotes l_2-norm unless otherwise specified.

Assumptions: In order to establish equivalence of the primal, dual problem and the convex-concave saddle point formulation, we make the following assumptions.

- g, the regularization for primal variable, is assumed to be μ-strongly convex, formally,

  $$g(\boldsymbol{y})\ge g(\boldsymbol{x})+\langle\nabla g(\boldsymbol{x}),\boldsymbol{y}-\boldsymbol{x}\rangle+\frac{\mu}{2}\|\boldsymbol{y}-\boldsymbol{x}\|^2,$$

  for any sub-gradient $\nabla g(\boldsymbol{x})\in\partial g(\boldsymbol{x}),\boldsymbol{x},\boldsymbol{y}\in\mathbb{R}^d$. We

[1] Department of ICES, University of Texas, Austin [2] Department of CS, Carnegie Mellon University, Pittsburgh [3] Department of CS, University of Texas, Austin [4] Amazon/A9, Palo Alto. Correspondence to: Qi Lei <leiqi@ices.utexas.edu>, Ian E.H. Yen <eyan@cs.cmu.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

also assume that g has decomposable structure, i.e., $g(\boldsymbol{x}) = \sum_i g_i(x_i)$.

- ϕ_i is $\frac{1}{\gamma}$-smooth, for $i \in [n]$:
  $\phi_i(y) \le \phi_i(x) + \phi_i'(x)(y-x) + \frac{\gamma}{2}(y-x)^2, x, y \in \mathbb{R}$
  or equivalently, ϕ_i' is Lipschitz continuous, i.e., $|\phi_i'(x) - \phi_i'(y)| \le \frac{1}{\gamma}|x-y|$.

2.1. Primal, dual and primal-dual formulations

Under the assumption of strongly convex regularization g and smooth loss function ϕ_i, minimizing (1) is equivalent to maximizing its dual formulation:

$$\max_{y \in \mathbb{R}^n} \left\{ D(\boldsymbol{y}) \equiv -g^*(-\frac{A^\top \boldsymbol{y}}{n}) - \frac{1}{n}\sum_{i=1}^n \phi_i^*(y_i) \right\} \quad (2)$$

or the unique solution for the following convex-concave saddle point problem:

$$\max_{y \in \mathbb{R}^n} \min_{\boldsymbol{x} \in \mathbb{R}^d} \left\{ \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) = g(\boldsymbol{x}) + \frac{1}{n}\boldsymbol{y}^\top A\boldsymbol{x} - \frac{1}{n}\sum_{i=1}^n \phi_i^*(y_i) \right\} \quad (3)$$

Note that $\phi_i(\boldsymbol{a}_i^\top \boldsymbol{x})$ in (1) is also smooth with respect to $\boldsymbol{x}$, since $\nabla_x \phi_i(\boldsymbol{a}_i^\top \boldsymbol{x}) = \phi_i'(\boldsymbol{a}_i^\top \boldsymbol{x})\boldsymbol{a}_i$, therefore $\phi_i(\boldsymbol{a}_i^\top \boldsymbol{x})$ is R^2/γ-smooth with respect to $\boldsymbol{x}$, where R is defined as $R = \max_i \|\boldsymbol{a}_i\|_2$. (Zhang & Xiao, 2014) thus defined the condition number for the primal-dual form as:

$$\kappa \overset{\text{def}}{=} \frac{R^2}{\mu\gamma}.$$

We share this definition in this paper. The commonly used condition number for the gradient descent of the primal form is simply $(R^2/\gamma + \mu)/\mu = 1 + \kappa$, see (Nesterov, 2004).

2.2. Related work

There has been a long line of work over the years to derive fast solvers for the generic optimization problem (1). In Table 1, we review the time complexity to achieve ϵ error with respect to either primal, dual or primal-dual optimality for existing methods.

Primal (accelerated) gradient descent (Nesterov, 2004; 2005) require $\mathcal{O}((1+\kappa)\log(1/\epsilon))$ (or $\mathcal{O}((1+\sqrt{\kappa})\log(1/\epsilon))$ if accelerated) iterations to achieve primal error less than ϵ. Note that $1+\kappa$ is the condition number of (1). Since each iteration takes $\mathcal{O}(nd)$ operations, the overall time complexity is $\mathcal{O}(nd\,(1+\kappa)\,\log(1/\epsilon))$ (or $\mathcal{O}(nd\,(1+\sqrt{\kappa})\,\log(1/\epsilon))$ if accelerated). Due to the large per iteration cost for large n, stochastic variants that separately optimize each ϕ_i have proved more popular in big data settings. Examples include SGD (Bottou, 2010), SAG (Schmidt et al., 2013), SVRG (Johnson & Zhang, 2013), SAGA (Defazio et al., 2014), MISO (Mairal, 2015) and their accelerated versions (Xiao & Zhang, 2014). The stochastic scheme of optimizing individual ϕ_i is similar to updating each dual coordinate individually. Their time complexity thus matches that of dual coordinate descent methods (Hsieh et al., 2008; Shalev-Shwartz & Zhang, 2013b; Yang, 2013; Qu et al., 2014), which enjoy a time complexity of $\mathcal{O}(nd\,(1+\kappa/n)\log(1/\epsilon))$, and a further acceleration step (Shalev-Shwartz & Zhang, 2016; 2013a) will improve the complexity to $\mathcal{O}(nd\,(1+\sqrt{\kappa/n})\log(1/\epsilon))$. These stochastic methods have a lower per iteration cost of $O(d)$, but each step optimizes terms that are much less well-conditioned, and consequently have a larger iteration complexity, for instance of $O(n\,(1+\sqrt{\kappa/n})\log(1/\epsilon))$ in the accelerated case.

With the primal-dual formulation, (Zhang & Xiao, 2014) introduce a novel stochastic primal-dual coordinate method (SPDC), which with acceleration achieves a time complexity of $\mathcal{O}(nd\,(1+\sqrt{\kappa/n})\log(1/\epsilon))$, matching that of accelerated stochastic dual coordinate descent methods.

However, in practice, SPDC could lead to more expensive computations for sparse data matrices due to dense updates. For some special choices of the model, (Zhang & Xiao, 2014) provided efficient implementation for sparse feature structures, but the average update time for each coordinate is still much longer than that of dual coordinate descent. Moreover, they cannot exploit intermediate sparse iterates by methods such as shrinking technique (Hsieh et al., 2008). We note moreover that acceleration is not always practically useful in many real-world settings, other than in extremely ill-conditioned situations. In particular, κ is typically of the order of $\sqrt{n}$ or n as shown in (Bousquet & Elisseeff, 2002; Zhang & Xiao, 2014), and therefore the conditioning of $\mathcal{O}(1+\sqrt{\kappa/n})$ is not necessarily much better than $\mathcal{O}(1+\kappa/n)$. Our experiments also corroborate this, showing that vanilla dual coordinate descent under reasonable precision or condition number is not improved upon by SDPC.

Therefore we raise the following question: *Does the primal-dual formulation have other good properties that could be leveraged to improve optimization performance?*

For instance, some recent work with the primal-dual formulation updates stochastically sampled coordinates (Yu et al., 2015), which has a reduced cost per iteration, provided the data admits a low-rank factorization or when the proximal mapping for primal and dual variables are relatively computational expensive, which however may not hold in practice, so that the the noise caused by this preprocessing could hurt test performance. Moreover, even when their assumptions hold, their low-rank matrix factorization step itself may dominate the total computation time.

2.3. Our Contribution

In this paper, we try to address the key question above in the setting of empirical risk minimization problems with very large n and d, and where the set of primal (and/or dual) variables are assumed to be sparse. We then show that the primal-dual formulation of the problem allows for naturally leveraging available primal and/or dual sparsity.

Table 1. Basic summary of running-time complexity of existing methods and our method (DGPD). n is the number of samples, d is the dimension of samples and primal variables, κ is the condition number for primal-dual coordinate algorithms. For our method, s is the upper bound of sparsity in its primal variables; For DSPDC(Yu et al., 2015), A is assumed to factorized as $UV, U \in \mathbb{R}^{n\times k}, V \in \mathbb{R}^{k\times d}$, and $\kappa_1 = \frac{\max_i \|\boldsymbol{a}_i\|_\infty}{\gamma\mu} \in [\frac{\kappa}{d}, \kappa]$.

	Time complexity	Extra assumption
GD	$\mathcal{O}(dn(1+\kappa)\log\frac{1}{\epsilon})$	–
AGD	$\mathcal{O}(dn(1+\sqrt{\kappa})\log\frac{1}{\epsilon})$	–
SGD	$\mathcal{O}(d(1+\kappa)\frac{1}{\epsilon})$	–
MISO SDCA SVRG SAG(A)	$\mathcal{O}(dn(1+\frac{\kappa}{n})\log\frac{1}{\epsilon})$, or $\mathcal{O}(dn(1+\sqrt{\frac{\kappa}{n}})\log\frac{1}{\epsilon})$ if accelerated	–
SPDC	$\mathcal{O}(dn(1+\sqrt{\frac{\kappa}{n}})\log\frac{1}{\epsilon})$	–
DSPDC	$\mathcal{O}(kn(1+d\sqrt{\frac{\kappa_1}{n}})\log\frac{1}{\epsilon})$	A is factorized
ours	$\mathcal{O}(s(d+n)(1+\frac{\kappa}{n})\log\frac{1}{\epsilon})$	$\boldsymbol{x}$ is sparse

In Table 1, we review the total time complexity to achieve ϵ accuracy. We can see that all algorithms that achieve a linear convergence rate require running time that has a factor nd, and in particular, none of their convergence rates involve the sparsity of the primal or dual variables.

There have been some attempts to modify existing primal or dual coordinate approaches in order to exploit sparsity of either primal or dual variables, but these do not perform very well in practice. One popular approach uses a shrinking heuristic in updating dual coordinates (Hsieh et al., 2008), which however still requires complexity linear to the number of coordinates d and does not guarantee rate of convergence. (Nutini et al., 2015) consider the idea of searching more important active coordinates to update in each iteration. Their greedy updates yield an iteration complexity linear in $1/\mu_1$ instead of d/μ, where μ and μ_1 are the parameters of strong convexity with respect to L_2 and L_1 norms respectively. However, with the commonly used L_2 regularization term $\mu\|\cdot\|^2$ that is used to ensure μ-strong convexity, the term is exactly $\mu_1 = \frac{\mu}{d}$ l_1-strongly convex. Moreover, in practice, searching active coordinates causes considerable overhead. While there have been some strategies proposed to address this such as (Dhillon et al., 2011) that leverages nearest neighbor search to reduce the searching time, these have further requirements on the data structure used to store the data. Overall, it thus remains to more carefully study the optimization problem in order to facilitate the use of greedy approaches to exploit primal or dual sparsity.

In this paper, we propose a Doubly Greedy Primal-Dual (DGPD) Coordinate method that greedily selects and updates both primal and dual variables. This method enjoys an overall low time complexity under a sparsity assumption on the primal variables:

Theorem 2.1. Main result: (informal) *For the empirical risk minimization problem* (1) *with $l_1 + l_2$ regularization, there exists an algorithm (DGPD) that achieves ϵ error in $\mathcal{O}(s(n+d)(1+\frac{\kappa}{n})\log\frac{1}{\epsilon}))$ time, where s is an upper bound of the sparsity of the primal variables.*

3. The Doubly Greedy Primal-Dual (DGPD) Coordinate Descent method

Coordinate-wise updates are most natural when g is separable, as is assumed for instance in the Stochastic Primal-Dual Coordinate method of (Zhang & Xiao, 2014). In this paper, to exploit sparsity in primal variables, we additionally focus on the case where $g(\boldsymbol{x}) = \frac{\mu}{2}\|\boldsymbol{x}\|^2 + \lambda\|\boldsymbol{x}\|_1$. With respect to the loss function ϕ, it is assumed to be $\frac{1}{\gamma}$-smooth and convex. For instance, setting ϕ_i as the smooth hinge loss(Shalev-Shwartz & Zhang, 2013b):

$$\phi_i(z) = \begin{cases} 0 & \text{if } b_i z \geq 1 \\ \frac{1}{2} - b_i z & \text{if } b_i z \leq 0 \\ (\frac{1}{2} - b_i z)^2 & \text{otherwise,} \end{cases}$$

the smoothness parameter $\gamma = \frac{1}{2}$. For the logit function $\phi_i(z) = \log(1 + \exp(-b_i z)$, the smoothness parameter $\gamma = 4$.

When iterates are sparse, it is more efficient to perform greedy coordinate descent. We will provide a brief theoretical vignette of this phenomenon in Section 4.1. With this motivation, our proposed method Doubly Greedy Primal-Dual Coordinate Descent (DGPD) greedily selects and updates both the primal and dual variables, one coordinate a time. Our overall method is detailed in Algorithm 1.

In Algorithm 1, we start from all zero vectors $\boldsymbol{x}^{(0)}, \boldsymbol{z}^{(0)} \in \mathbb{R}^n$, and $\boldsymbol{y}^{(0)}, \boldsymbol{w}^{(0)} \in \mathbb{R}^d$, where $\boldsymbol{x}^{(0)}$, and $\boldsymbol{y}^{(0)}$ are the iterates for primal and dual variables, and $\boldsymbol{w}^{(0)}$ and $\boldsymbol{z}^{(0)}$ are two auxiliary vectors, maintained as $\boldsymbol{w} \equiv A\boldsymbol{x}$ and $z \equiv A^\top \boldsymbol{y}$ to cache and reduce computations.

Primal Updates. In each iteration, we first compute the optimal primal variable $\bar{x}^{(t-1)}$ for the current $\boldsymbol{y}^{(t-1)}$, i.e.,

$$\bar{\boldsymbol{x}}^{(t-1)} = \arg\min_x \mathcal{L}(\boldsymbol{x}, \boldsymbol{y}^{(t-1)}) \Rightarrow \text{Eqn.(4)}$$

Then, we only update the coordinate $j^{(t)}$ that will decrease $\mathcal{L}(\boldsymbol{x}, \boldsymbol{y})$ the most, i.e.,

$$j^{(t)} = \arg\min_{k\in[d]} \mathcal{L}(\boldsymbol{x}^{(t)} + (\bar{x}_k^{(t-1)} - x_k^{(t)})\boldsymbol{e}_k, \boldsymbol{y}^{(t-1)}) \Rightarrow \text{Eqn.(5)}$$

Both two processes cost $\mathcal{O}(d)$ operations. Afterwards, we update the value of w with Eqn. (6) such that $\boldsymbol{w}^{(t)} = A\boldsymbol{x}^{(t)}$

Algorithm 1 Doubly Greedy Primal-Dual Coordinate method

```
Input: Training data $A \in \mathbb{R}^{n\times d}$, dual step size $\eta > 0$.
Initialize: $\boldsymbol{x}^{(0)} \leftarrow 0 \in \mathbb{R}^d$, $\boldsymbol{y}^{(0)} \leftarrow 0 \in \mathbb{R}^n$, $\boldsymbol{w}^{(0)} \equiv A\boldsymbol{x} = 0 \in \mathbb{R}^n$, $\boldsymbol{z}^{(0)} \equiv A^\top \boldsymbol{y} = 0 \in \mathbb{R}^d$
for $t = 1, 2, \cdots, T$ do
    Choose greedily the primal coordinate to update:
        $\bar{x}_k^{(t)} \leftarrow \arg\min_{\alpha} \{ \frac{1}{n} z_k^{(t-1)} \alpha + g_k(\alpha) \}, \ \ \forall k \in [d]$    (4)
        $j^{(t)} \leftarrow \arg\min_{k\in[d]} \{ \frac{1}{n} z_k^{(t-1)} (\bar{x}_k^{(t)} - x_k^{(t-1)}) + g_k(\bar{x}_k^{(t)}) - g_k(x_k^{(t-1)}) \}$    (5)
        $x_k^{(t)} \leftarrow \begin{cases} \bar{x}_k^{(t)} & \text{if } k = j^{(t)}, \\ x_k^{(t-1)} & \text{otherwise.} \end{cases}$
    Update $w$ to maintain the value of $A\boldsymbol{x}$:
        $\boldsymbol{w}^{(t)} \leftarrow \boldsymbol{w}^{(t-1)} + (x_j^{(t)} - x_j^{(t-1)}) A^j$    (6)
    Choose greedily the dual coordinate to update:
        $i^{(t)} \leftarrow \arg\max_{k\in[n]} |w_k^{(t-1)} - \frac{1}{n} (\phi_k^*)'(y_k^{(t-1)})|$    (7)
        $y_k^{(t)} \leftarrow \begin{cases} \arg\max_\beta \{ \frac{1}{n} w_k^{(t)} \beta - \phi_k^*(\beta) - \frac{1}{2\eta}(\beta - y_k^{(t-1)})^2 \} & \text{if } k = i^{(t)} \\ y_k^{(t-1)} & \text{otherwise.} \end{cases}$    (8)
    Update $z$ to maintain the value of $A^\top \boldsymbol{y}$
        $\boldsymbol{z}^{(t)} \leftarrow \boldsymbol{z}^{(t-1)} + (y_{i^{(t)}}^{(t)} - y_{i^{(t)}}^{(t-1)}) A_{i^{(t)}}$    (9)
end for
Output: $\boldsymbol{x}^{(T)}, \boldsymbol{y}^{(T)}$
```

in $\mathcal{O}(d)$ or $\mathcal{O}(\mathrm{nnz}(A^j))$ operations. This greedy choice of $j^{(t)}$ and aggressive update induces a sufficient primal progress, as shown in Lemma A.1.

Dual Updates. We note that the updates are not exactly symmetric in the primal $\boldsymbol{x}$ and dual $\boldsymbol{y}$ variables. The updates for the dual variables $\boldsymbol{y}$ do follow along similar lines as $\boldsymbol{x}$, except that we use the Gauss-Southwell rule to select variables, and introduce a step size η. This is motivated by our convergence analysis, which shows that each primal update step yields a large descent in the objective, while each dual update only ascends the dual objective modulo an error term. This required a subtle analysis to show that the error terms were canceled out in the end by the progress made in the primal updates. But to proceed with such an analysis required the use of a step size in the dual updates, to balance the progress made in the dual updates, and the error term it introduced. Note moreover, that we are using the Gauss-Southwell rule to choose the variable to optimize in the dual variables $\boldsymbol{y}$, while we simply use the coordinate that causes the most function descent in the primal variables x. This is because our choice of step size in the dual updates required computations that are shared with our current approach of selecting the optimal primal variable. This does incur more overhead when compared to the Gauss Southwell rule however, so that we simply use the latter for optimizing $\boldsymbol{y}$.

The most significant feature in our method is that we select and update one coordinate in both the primal and dual coordinates greedily. With a simple trick that maintains the value of $\boldsymbol{w} \equiv A\boldsymbol{x}$ and $\boldsymbol{z} \equiv A^\top \boldsymbol{y}$ (Lei et al., 2016), we are able to select and update primal and dual coordinates in $\mathcal{O}(n)$ and $\mathcal{O}(d)$ operations respectively. This happens when computing the value of $A\boldsymbol{x}$ and $A^\top \boldsymbol{y}$, which are the bottleneck in computing the gradient or updating the variables. An extension to choose and update a batch of primal and dual coordinate is straightforward. We provide further discussions on the designing of Algorithm 1 in Section 4.

In this paper, we have not incorporated an extrapolation/acceleration scheme to our algorithm. As noted earlier, in practice the condition number κ is usually comparable to n, thus adding an extrapolation term that reduces the conditioning from κ/n to $\sqrt{\kappa/n}$ is not necessarily materially advantageous in real applications. Meanwhile, an extrapolation step usually worsens the stability of the algorithm, and is not easily combined with incorporating greedy updates, which is crucial to the leveraging the primal or dual sparsity structure in this paper. We thus defer an accelerated extension of our algorithm incorporating extrapolation term to future work.

For Algorithm 1, each iteration can be seen to have a cost of $\mathcal{O}(n + d)$, while in Section 4 we show that the iteration complexity for our method is $\mathcal{O}((1 + \frac{\kappa}{n}) s \log(1/\epsilon))$ assuming that the primal variables are s-sparse. Therefore the overall time complexity for our algorithm is $\mathcal{O}\big((1 + \frac{\kappa}{n}) s(n + d) \log(1/\epsilon)\big)$, which is cheaper than the time complexity of even the accelerated SPDC algorithm $\mathcal{O}\big((1+\sqrt{\frac{\kappa}{n}})nd\log(1/\epsilon)\big)$ except for extremely ill conditioned cases.

3.1. A Practical Extension of DGPD

In real application settings, Algorithm 1 has some drawbacks. When data is sparse, we still require $\mathcal{O}(n)$ and $\mathcal{O}(d)$ operations to update primal and dual variables. Even when the data is dense, to find the greedy coordinate and to update it requires comparable time complexity, which suggests we should find some ways to eliminate overhead in practice.

To resolve these issues, we introduce the Doubly Greedy Primal-Dual Coordinate method with Active Sets in Algorithm 2. We make use of what we call *active sets*, that contains the newly selected coordinates as well as the current non-zero variables. We construct these active sets $\mathcal{A}_x$ and $\mathcal{A}_y$ for both primal and dual variables. Initially, they are set as empty sets. In each iteration, we recurrently select coordinates outside the active sets with the Gauss-Southwell rule, and add them to $\mathcal{A}_x$ and $\mathcal{A}_y$. We then optimize all the variables within the active sets. Once a primal/dual variable gets set to 0, we can drop it from the corresponding active sets. This practice keeps the active sets $\mathcal{A}_x$ and $\mathcal{A}_y$ as the support of primal and dual variables. Notice $g_k'(x_k)$ is 0 when x_k is zero, so that the variable selection step for primal variables can be simplified as stated in (10).

Now the time complexity per iteration becomes $|\mathcal{A}_x|n + |\mathcal{A}_y|d$. The sparsity in primal variables is encouraged by the choice of $\ell_1 + \ell_2$ regularization. Meanwhile, as shown by (Yen et al., 2016), a sparse set of primal variables usually induces a sparse set of dual variables. Therefore $|\mathcal{A}_x| \ll d$ and $|\mathcal{A}_y| \ll n$ in practice, and the cost per iteration is sub-linear to nd. We present further details in Section 3.2.

3.2. Efficient Implementation for Sparse Data Matrix

Suppose we are given a sparse data matrix A with number of non-zero elements of each column and each row bounded by nnz_y and nnz_x respectively, one can further reduce the cost for computing (10) and (12) from $\mathcal{O}(d|\mathcal{A}_y| + n|\mathcal{A}_x|)$ to $\mathcal{O}(\text{nnz}_x|\mathcal{A}_y| + \text{nnz}_y|\mathcal{A}_x|)$ by storing both $\{A_i\}_{i=1}^n$ and $\{A^j\}_{j=1}^d$ as sparse vectors and computing $A^\top \boldsymbol{y}$ and $A\boldsymbol{x}$ as

$$A^\top \boldsymbol{y} = \sum_{i\in\mathcal{A}_y} A_i^\top y_i, \quad A\boldsymbol{x} = \sum_{j\in\mathcal{A}_x} A^j x_j. \tag{14}$$

In our implementation, whenever the active sets $\mathcal{A}_y$, $\mathcal{A}_x$ are expanded, we further maintain a submatrix $[A]_\mathcal{A}$ which contains only rows in $\mathcal{A}_y$ and columns in $\mathcal{A}_x$, so the primal and dual updates (11), (13) only cost $\sum_{i\in\mathcal{A}_y}\text{nnz}([A_i]_{\mathcal{A}_x})$. This results in each update costing less than the search steps, and therefore, in practice, one can conduct multiple rounds of updates (11), (13) before conducting the search (10), (12), which in our experiment speeds up convergence significantly.

4. Convergence Analysis

In this section, we introduce the primal gap Δ_p and dual gap Δ_d and analyze the convergence rate in terms of their sum, which we call primal and dual sub-optimality $\Delta = \Delta_p + \Delta_d$.

Definition 4.1. *For the following convex-concave function $\mathcal{L}(\boldsymbol{x},\boldsymbol{y}) \overset{def}{=} g(\boldsymbol{x}) + \frac{1}{n}\boldsymbol{y}^\top A\boldsymbol{x} - \frac{1}{n}\sum_{i=1}^n \phi_i^*(y_i)$, with its primal form $P(\boldsymbol{x}) \overset{def}{=} \min_y \mathcal{L}(\boldsymbol{x},\boldsymbol{y})$, and dual form $D(\boldsymbol{y}) \overset{def}{=} \max_x \mathcal{L}(\boldsymbol{x},\boldsymbol{y})$, we define the primal gap at iteration t as*

$$\Delta_p^{(t)} \overset{def}{=} \mathcal{L}(\boldsymbol{x}^{(t+1)}, \boldsymbol{y}^{(t)}) - D(\boldsymbol{y}^{(t)})$$

, the dual gap at iteration t as

$$\Delta_d^{(t)} \overset{def}{=} D^* - D(\boldsymbol{y}^{(t)})$$

and sub-optimality as

$$\Delta^{(t)} \overset{def}{=} \Delta_p^{(t)} + \Delta_d^{(t)}.$$

Theorem 4.2. *Suppose in (1), g is μ-strongly convex ($\ell_1 + \ell_2$) regularization, and ϕ_i is $\frac{1}{\gamma}$-smooth. Let $R = \max_{i\in[n]}\|a_i\|_2$. Then DGPD achieves*

$$\Delta^{(t+1)} \le \frac{2n}{2n+\eta\gamma}\Delta^{(t)}, \tag{15}$$

if step size $\eta^{(t)}$ satisfies that

$$\eta^{(t)} \le \frac{2n^2\mu}{\|\boldsymbol{x}^{(t)} - \bar{\boldsymbol{x}}^{(t)}\|_0(5R^2 + n\gamma\mu)} \tag{16}$$

Suppose $\|\boldsymbol{x}^{(t)} - \bar{\boldsymbol{x}}^{(t)}\|_0 \le s$, if we choose step size $\eta = \frac{2n^2\mu}{(5R^2+n\gamma\mu)s}$, then it requires

$$\mathcal{O}(s(\frac{\kappa}{n}+1)\log\frac{1}{\epsilon})$$

iterations for achieving ϵ primal and dual sub-optimality.[1]

Proof sketch: The proof analysis is straightforward with the introduction of primal and dual sub-optimality Δ. We divide the proof into *primal-dual progress*, *primal progress*, and *dual progress*.

- Primal-Dual Progress (Lemma A.2).

$$\begin{aligned} & \Delta_d^{(t)} + \Delta_p^{(t)} - (\Delta_d^{(t-1)} + \Delta_p^{(t-1)}) \\ \le\ & \mathcal{L}(\boldsymbol{x}^{(t+1)}, \boldsymbol{y}^t) - \mathcal{L}(\boldsymbol{x}^t, \boldsymbol{y}^t) \\ & + \eta(\frac{1}{n}\langle A_{i^{(t)}}, \boldsymbol{x}^{(t)} - \bar{\boldsymbol{x}}^{(t)}\rangle)^2 \\ & - \eta(\frac{1}{n}\langle A_{i^{(t)}}, \bar{\boldsymbol{x}}^{(t)}\rangle - \frac{1}{n}(\phi_{i^{(t)}}^*)'(y_{i^{(t)}}^{(t)}))^2 \end{aligned} \tag{17}$$

[1] This result can be easily connected to traditional convergence analysis in primal or dual form. Notice $\Delta \le \epsilon$ is sufficient requirement that dual gap $\Delta_d = D^* - D(\boldsymbol{y}) \le \epsilon$, therefore the dual variable $\boldsymbol{y}^{(t)}$ converges to optimal $\boldsymbol{y}^*$ with the same convergence rate.

Algorithm 2 Doubly Greedy Primal-Dual Coordinate method with Active Sets

```
**Input:** Training data $A \in \mathbb{R}^{n\times d}$, dual step size $\eta > 0$.
**Initialize:** $\boldsymbol{x}^{(0)} \leftarrow 0 \in \mathbb{R}^d$, $\boldsymbol{y}^{(0)} \leftarrow 0 \in \mathbb{R}^n$, $\mathcal{A}_x^{(0)} \leftarrow \varnothing, \mathcal{A}_y^{(0)} \leftarrow \varnothing$
**for** $t \leftarrow 1, 2, \cdots, T$ **do**
    Update the active set $\mathcal{A}_x^{(t)}$ greedily based on the optimal primal variable $\bar{\boldsymbol{x}}^{(t-1)}$ and update $\boldsymbol{x}$ in its active set.
        $\bar{x}_k^{(t)} \leftarrow \arg\min_{\alpha} \{\frac{1}{n}\langle A^k, \boldsymbol{y}^{(t-1)}\rangle\alpha + g_k(\alpha)\}, \ \forall k \in [d]$
        $j^{(t)} \leftarrow \arg\max_{k\in[d]} |\bar{x}_k^{(t-1)}| \quad (10)$
        $\mathcal{A}_x^{(t)} \leftarrow \mathcal{A}_x^{(t-1)} \cup \{j^{(t)}\} \quad x_j^{(t)} \leftarrow \begin{cases} \bar{x}_j^{(t-1)}, & \text{if } j \in \mathcal{A}_x^{(t)} \\ x_j^{(t-1)}, & \text{if } j \notin \mathcal{A}_x^{(t)} \end{cases} \quad (11)$
    Update the active set $\mathcal{A}_y^{(t)}$ greedily based on the value of $\nabla_y \mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t-1)})$ and update $\boldsymbol{y}$ in its active set.
        $i^{(t)} \leftarrow \arg\max_{k\in[n]-\mathcal{A}_y^{(t-1)}} |\langle A_k, \boldsymbol{x}^{(t)}\rangle - \frac{1}{n}(\phi_k^*)'(y_k^{(t-1)})|. \quad (12)$
        $\mathcal{A}_y^{(t)} \leftarrow \mathcal{A}_y^{(t-1)} \cup \{i^{(t)}\}$
        $y_i^{(t)} \leftarrow \begin{cases} \arg\max_\beta \{\frac{1}{n}\langle A_i, \boldsymbol{x}^{(t)}\rangle\beta - \frac{1}{n}\phi_i^*(\beta) - \frac{1}{2\eta}(\beta - y_k^{(t-1)})\}, & \text{if } i \in \mathcal{A}_y^{(t)} \\ y_i^{(t-1)}, & \text{if } i \notin \mathcal{A}_y^{(t)} \end{cases} \quad (13)$
    Kick out 0 variables from active sets.
        $\mathcal{A}_y^{(t)} \leftarrow \mathcal{A}_y^{(t)} - \bigcup_{i, y_i^{(t)}=0} \{i\}, \quad \mathcal{A}_x^{(t)} \leftarrow \mathcal{A}_x^{(t)} - \bigcup_{j, x_j^{(t)}=0} \{j\}$
**end for**
**Output:** $\boldsymbol{x}^{(T)}, \boldsymbol{y}^{(T)}$
```

This lemma connects the descent in PD sub-optimality with primal progress and dual progress. The third term and the second terms respectively represent the potential dual progress if we used the optimal $\bar{\boldsymbol{x}}^{(t)}$, and the irrelevant part generated from the difference between $\bar{\boldsymbol{x}}^{(t)}$ and $\boldsymbol{x}^{(t)}$.

- Primal Progress (Lemma A.1).

$$\mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) - \mathcal{L}(\boldsymbol{x}^{(t+1)}, \boldsymbol{y}^{(t)}) \geq \frac{1}{\|\boldsymbol{x}^{(t)} - \bar{\boldsymbol{x}}^{(t)}\|_0 - 1}\Delta_p^{(t)} \tag{18}$$

This inequality simply demonstrates function loss from primal update is at least a ratio of primal gap.

- Dual Progress (Lemma A.3).

$$\begin{aligned} & (\frac{1}{n}\langle A_{i^{(t)}}, \boldsymbol{x}^{(t)} - \bar{\boldsymbol{x}}^{(t)}\rangle)^2 \\ & -(\frac{1}{n}\langle A_{i^{(t)}}, \bar{\boldsymbol{x}}^{(t)}\rangle - \frac{1}{n}(\phi_{i^{(t)}}^*)'(y_{i^{(t)}}^{(t)}))^2 \\ \leq \ & -\frac{\gamma}{2n}\Delta_d^{(t)} + \frac{5R^2}{2n^2}\|\boldsymbol{x}^{(t)} - \bar{\boldsymbol{x}}^{(t)}\|^2 \end{aligned} \tag{19}$$

Finally, we establish the relation between the dual progress in our algorithm with dual gap and difference between $\bar{\boldsymbol{x}}^{(t)}$ and $\boldsymbol{x}^{(t)}$. Now we can prove our main theorem 4.2.

For cleaner notation, write $a = \frac{\eta\gamma}{2n}$, $b = \frac{5\eta R^2}{2n^2}$. $\|\bar{\boldsymbol{x}}^{(t)} - \boldsymbol{x}^{(t)}\|_0 \leq s$. By combining (18) and (19) to (17), we get:

$$\begin{aligned} & \Delta_d^{(t)} - \Delta_d^{(t-1)} + \Delta_p^{(t)} - \Delta_p^{(t-1)} \\ \leq \ & \mathcal{L}(\boldsymbol{x}^{(t+1)}, \boldsymbol{y}^t) - \mathcal{L}(\boldsymbol{x}^t, \boldsymbol{y}^t) - a\Delta_d^{(t)} \\ & +\frac{\gamma}{2}b\|\boldsymbol{x}^{(t)} - \bar{\boldsymbol{x}}^{(t)}\|^2 \\ \leq \ & \mathcal{L}(\boldsymbol{x}^{(t+1)}, \boldsymbol{y}^t) - \mathcal{L}(\boldsymbol{x}^t, \boldsymbol{y}^t) - a\Delta_d^{(t)} \\ & +b\big(\mathcal{L}(\boldsymbol{x}^{(t)}, \boldsymbol{y}^{(t)}) - \mathcal{L}(\bar{\boldsymbol{x}}^{(t)}, \boldsymbol{y}^{(t)})\big) \\ = \ & (1-b)\big(\mathcal{L}(\boldsymbol{x}^{(t+1)}, \boldsymbol{y}^t) - \mathcal{L}(\boldsymbol{x}^t, \boldsymbol{y}^t)\big) - a\Delta_d^{(t)} \\ & +b\big(\mathcal{L}(\boldsymbol{x}^{(t+1)}, \boldsymbol{y}^t) - \mathcal{L}(\bar{\boldsymbol{x}}^{(t)}, \boldsymbol{y}^t)\big) \\ \leq \ & -\frac{1-b}{s-1}\Delta_p^{(t)} - a\Delta_d^{(t)} \\ & +b\big(\mathcal{L}(\boldsymbol{x}^{(t+1)}, \boldsymbol{y}^t) - \mathcal{L}(\bar{\boldsymbol{x}}^{(t)}, \boldsymbol{y}^t)\big) \\ = \ & -\big(\frac{1-b}{s-1} - b\big)\Delta_p^{(t)} - a\Delta_d^{(t)} \end{aligned}$$

Here the second inequality comes from strong convexity of $\mathcal{L}(\cdot, \boldsymbol{y}^{(t)})$. The fourth inequality comes from Lemma A.1.

Therefore when $a \leq \frac{1-b}{s-1} - b$, or sufficiently $a \leq (s(1 + 5\kappa/n))^{-1}$, we get $\Delta^{(t)} \leq \frac{1}{1+a}\Delta^{(t-1)}$. Since $a < 1$, $(a+1)^{-1/a} \leq 1/2$, therefore $\Delta^{(t)} \leq (1+a)^{-t}\Delta^{(0)} \leq$

$2^{-at}\Delta^{(0)}$. Therefore when $T \geq \mathcal{O}(s(1+\kappa/n)\log_2 \frac{\Delta^{(0)}}{\epsilon})$, $\Delta^{(T)} \leq \epsilon$.

4.1. Analysis on greedy methods for sparse iterates

In this section, we give a simple analysis of the greedy variable selection rule showing that when iterate and minimizer of a generic optimization problem are sparse, its convergence rate is faster than choosing random coordinates. We define the optimization problem in the space of $\mathbb{R}^n$:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x})$$

, where f is μ-strongly convex L-smooth:

$$|\nabla_i f(\boldsymbol{x}+\alpha\boldsymbol{e}_i) - \nabla_i f(\boldsymbol{x})| \leq L|\alpha|, \forall \boldsymbol{x} \in \mathbb{R}^n$$

Under this setting, a random coordinate descent method with step size $\frac{1}{L}$, achieves $\mathbb{E}[f(\boldsymbol{x}^+) - f^*] \leq (1 - \frac{\mu}{nL})\big(f(\boldsymbol{x}) - f^*\big)$, where $\boldsymbol{x}^+$ is the next iterate of $\boldsymbol{x}$.

Under the assumption that the current iterate $\boldsymbol{x}$ and the optimal $\boldsymbol{x}^*$ are both k-sparse, we thereby conduct greedy coordinate descent rule, i.e., $\boldsymbol{x}^+ = \boldsymbol{x} + \eta\boldsymbol{e}_{i^*}$, where η, i^* satisfies $f(\boldsymbol{x}+\eta\boldsymbol{e}_{i^*}) = \min_{i,\delta} f(\boldsymbol{x}+\delta\boldsymbol{e}_i)$. With L-Lipchitz continuity, we have:

$$\begin{aligned}
& f(\boldsymbol{x}+\eta\boldsymbol{e}_{i^*}) - f(\boldsymbol{x}) \\
\leq \quad & \min_{\delta,i} \big\{\langle\nabla f(\boldsymbol{x}), \delta\boldsymbol{e}_i\rangle + \frac{L}{2}\delta^2\big\} \\
= \quad & \min_{\delta,i} \big\{\langle\nabla f(\boldsymbol{x}), \delta\boldsymbol{e}_i\rangle + \frac{L}{2}\|\delta\boldsymbol{e}_i\|_1^2\big\} \\
= \quad & \min_{\Delta\boldsymbol{x}} \big\{\langle\nabla f(\boldsymbol{x}), \Delta\boldsymbol{x}\rangle + \frac{L}{2}\|\Delta\boldsymbol{x}\|_1^2\big\} \\
\leq \quad & \min_{\Delta\boldsymbol{x}} \big\{f(\boldsymbol{x}+\Delta\boldsymbol{x}) - f(\boldsymbol{x}) + \frac{L}{2}\|\Delta\boldsymbol{x}\|_1^2\big\} \\
\leq \quad & \min_{0\leq\delta\leq 1} \big\{f(x+\delta(x^*-x)) - f(\boldsymbol{x}) + \frac{L}{2}\delta^2\|\boldsymbol{x}^*-\boldsymbol{x}\|_1^2\big\} \\
\leq \quad & \min_{0\leq\delta\leq 1} \big\{\delta(f^*-f(\boldsymbol{x})) + \frac{L}{2}\delta^2\|\boldsymbol{x}^*-\boldsymbol{x}\|_1^2\big\}
\end{aligned}$$

The last two inequalities are obtained by constraining $\Delta\boldsymbol{x}$ to be of the form $\delta(\boldsymbol{x}^*-\boldsymbol{x})$ and by the convexity of f. For the k-sparse $\boldsymbol{x}$, and $\boldsymbol{x}^*$, $\boldsymbol{x}-\boldsymbol{x}^*$ is at most $2k$-sparse, and for any $2k$-sparse vector a, $\|a\|_1^2 \leq 2k\|a\|_2^2$. Hereby we obtain:

$$\begin{aligned}
& \min_{0\leq\delta\leq 1} \big\{\delta(f^*-f(\boldsymbol{x})) + \frac{L}{2}\delta^2\|\boldsymbol{x}^*-\boldsymbol{x}\|_1^2\big\} \\
\leq \quad & \min_{0\leq\delta\leq 1} \big\{\delta(f^*-f(\boldsymbol{x})) + Lk\delta^2\|\boldsymbol{x}^*-\boldsymbol{x}\|_2^2\big\} \\
\leq \quad & \min_{0\leq\delta\leq 1} \big\{\delta(f^*-f(\boldsymbol{x})) - \frac{2kL}{\mu}\delta^2(f^*-f(\boldsymbol{x}))\big\} \\
= \quad & \frac{\mu}{8kL}(f^*-f(\boldsymbol{x}))
\end{aligned}$$

Therefore $f(\boldsymbol{x}^+) - f^* \leq (1 - \frac{\mu}{8kL})(f(\boldsymbol{x}) - f^*)$, and when $k \ll n$, this convergence rate could be much better than randomized coordinate descent.

5. Experiment

In this section, we implement the Doubly-Greedy Primal-Dual Coordinate Descent algorithm with Active Sets, and compare its performance with other state-of-the-art methods for ℓ_1+ℓ_2-regularized Empirical Risk minimization, including Primal Randomized Coordinate Descent (PrimalRCD) (Richtárik & Takác, 2014), Dual Randomized Coordinate Descent (DualRCD, i.e., SDCA) (Shalev-Shwartz & Zhang, 2013b) and the Stochastic Primal-Dual Coordinate Method (SPDC) (Zhang & Xiao, 2014).

We conduct experiments on large-scale multi-class data sets with linear and non-linear feature mappings, as shown in Table 2. For Mnist and Aloi we use *Random Fourier (RF)* and *Random Binning (RB)* feature proposed in (Rahimi & Recht, 2007) to approximate effect of RBF Gaussian kernel and Laplacian Kernel respectively. The features generated by Random Fourier are dense, while Random Binning gives highly sparse data.

We give results for $\lambda \in \{0.1, 0.01\}$ and $\mu \in \{1, 0.1, 0.01\}$, where Figure 1 shows results for $\lambda = 0.1$, $\mu = 0.01$ and others can be found in Appendix B. In the above six figures, we compare the running time with objective function. While in the below figures, the x-axis is number of iterations. For the baseline methods, one iteration is one pass over all the variables, and for our method, it is several (5) passes over the active sets. From the figures, we can see that in all cases, DGPD has better performance than other methods. Notice for clear presentation purposes we use $\log$-scale for Mnist-RB-time, Aloi-RB-time and RCV-time, where our algorithm achieves improvements over others of orders of magnitude.

The result shows that, by exploiting sparsity in both the primal and dual, DGPD has much less cost per iteration and thus is considerably faster in terms of training time, while by maintaining an active set it does not sacrifice much in terms of convergence rate. Note since in practice we perform multiple updates after each search, the convergence rate (measured in outer iterations) can be sometimes even better than DualRCD.

6. Acknowledgements

I.D. acknowledges the support of NSF via CCF-1320746, IIS-1546452, and CCF-1564000. P.R. acknowledges the support of ARO via W911NF-12-1-0390 and NSF via IIS-1149803, IIS-1320894, IIS-1447574, and DMS-1264033, and NIH via R01 GM117594-01 as part of the Joint DMS/NIGMS Initiative to Support Research at the Interface of the Biological and Mathematical Sciences.

Table 2. Data statistics and number of non-zero primal & dual variables from DGPD (w/ $\lambda = 0.1$, $\mu = 0.01$).

Data set	#features	#nonzero/sample	#train samples	#test samples	#classes	#nz-primal	#nz-dual
Mnist-RF	10,000	10,000	58,000	2,000	10	1,730	2,000
Aloi-RF	10,000	10,000	90,000	8,000	1,000	891	1,428
Mnist-RB	1,572,556	1,000	58,000	2,000	10	1,733	1,208
Aloi-RB	636,910	200	90,000	8,000	1,000	1,032	782
RCV1-Regions	47,236	68.38	199,328	23,149	225	1,123	1,447
Sector	55,197	162.94	7,793	961	105	610	655

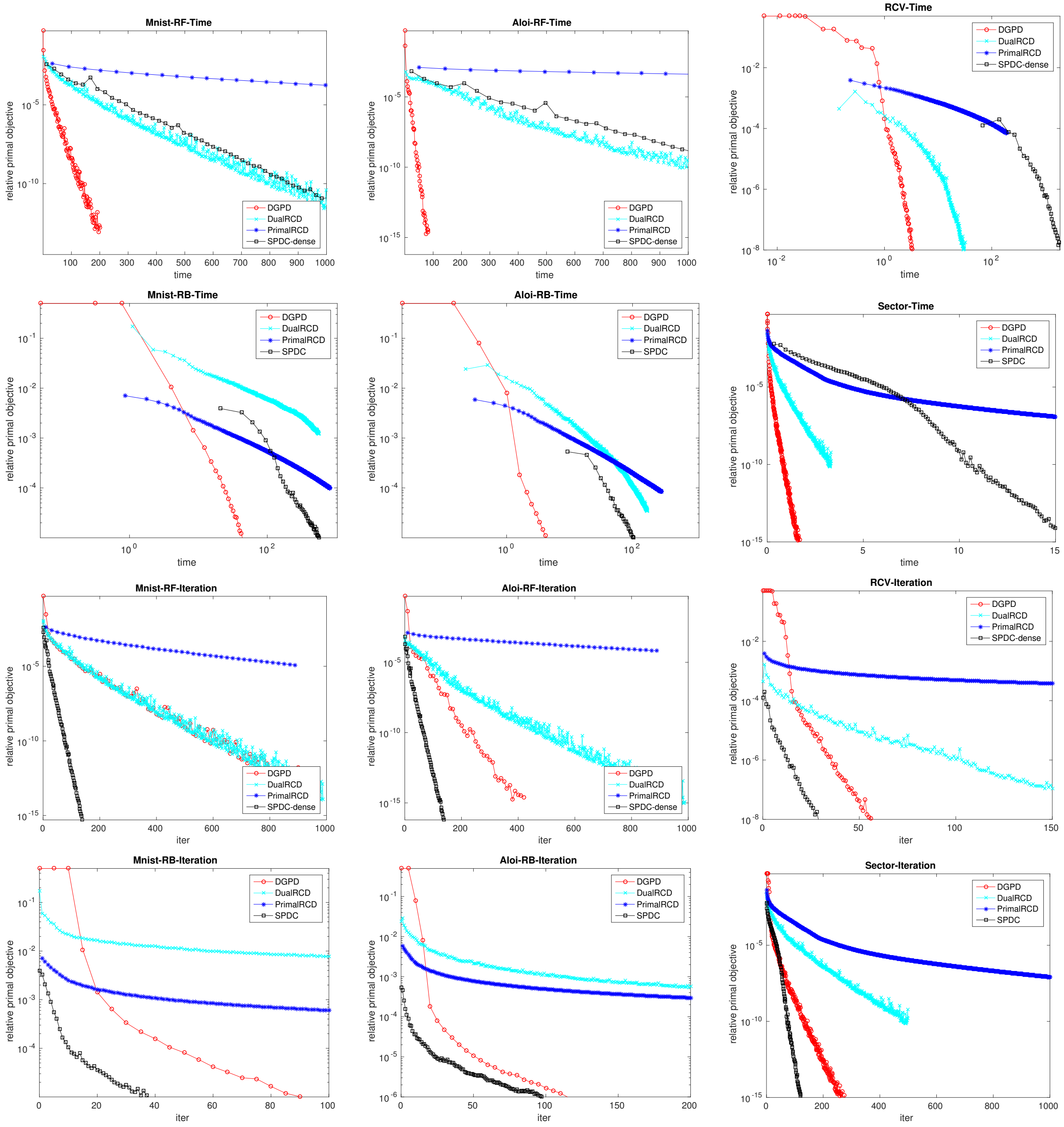

Figure 1. Relative Objective versus Time (the upper 2 rows) and versus # iterations (the lower 2 rows) for $\lambda = 0.1$, $\mu = 0.01$.

References

Bottou, Léon. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

Bousquet, Olivier and Elisseeff, André. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

Chang, Yin-Wen, Hsieh, Cho-Jui, Chang, Kai-Wei, Ringgaard, Michael, and Lin, Chih-Jen. Training and testing low-degree polynomial data mappings via linear svm. *The Journal of Machine Learning Research*, 11:1471–1490, 2010.

Chen, Jie, Wu, Lingfei, Audhkhasi, Kartik, Kingsbury, Brian, and Ramabhadrari, Bhuvana. Efficient one-vs-one kernel ridge regression for speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 2454–2458. IEEE, 2016.

Defazio, Aaron, Bach, Francis, and Lacoste-Julien, Simon. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pp. 1646–1654, 2014.

Dhillon, Inderjit S, Ravikumar, Pradeep K, and Tewari, Ambuj. Nearest neighbor based greedy coordinate descent. In *Advances in Neural Information Processing Systems*, pp. 2160–2168, 2011.

Hsieh, Cho-Jui, Chang, Kai-Wei, Lin, Chih-Jen, Keerthi, S Sathiya, and Sundararajan, Sellamanickam. A dual coordinate descent method for large-scale linear svm. In *Proceedings of the 25th international conference on Machine learning*, pp. 408–415. ACM, 2008.

Johnson, Rie and Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pp. 315–323, 2013.

Lei, Qi, Zhong, Kai, and Dhillon, Inderjit S. Coordinate-wise power method. In *Advances in Neural Information Processing Systems*, pp. 2056–2064, 2016.

Mairal, Julien. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

Nesterov, Y. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Science & Business Media, 2004.

Nesterov, Yu. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

Nutini, Julie, Schmidt, Mark, Laradji, Issam H, Friedlander, Michael, and Koepke, Hoyt. Coordinate descent converges faster with the gauss-southwell rule than random selection. *arXiv preprint arXiv:1506.00552*, 2015.

Qu, Zheng, Richtárik, Peter, and Zhang, Tong. Randomized dual coordinate ascent with arbitrary sampling. *arXiv preprint arXiv:1411.5873*, 2014.

Rahimi, Ali and Recht, Benjamin. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2007.

Richtárik, Peter and Takác, Martin. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2): 1–38, 2014.

Schmidt, Mark, Le Roux, Nicolas, and Bach, Francis. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, pp. 1–30, 2013.

Shalev-Shwartz, Shai and Zhang, Tong. Accelerated mini-batch stochastic dual coordinate ascent. In *Advances in Neural Information Processing Systems*, pp. 378–385, 2013a.

Shalev-Shwartz, Shai and Zhang, Tong. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14(1):567–599, 2013b.

Shalev-Shwartz, Shai and Zhang, Tong. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, 155(1-2):105–145, 2016.

Sonnenburg, Sören and Franc, Vojtech. Coffin: A computational framework for linear svms. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 999–1006, 2010.

Wu, Lingfei, Yen, Ian EH, Chen, Jie, and Yan, Rui. Revisiting random binning features: Fast convergence and strong parallelizability. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1265–1274. ACM, 2016.

Xiao, Lin and Zhang, Tong. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

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

Yen, Ian EH, Huang, Xiangru, Zhong, Kai, Ravikumar, Pradeep, and Dhillon, Inderjit S. Pd-sparse: A primal and dual sparse approach to extreme multiclass and multilabel classification. In *Proceedings of the 33nd International Conference on Machine Learning*, 2016.

Yen, Ian En-Hsu, Lin, Ting-Wei, Lin, Shou-De, Ravikumar, Pradeep K, and Dhillon, Inderjit S. Sparse random feature algorithm as coordinate descent in hilbert space. In *Advances in Neural Information Processing Systems*, pp. 2456–2464, 2014.

Yu, Adams Wei, Lin, Qihang, and Yang, Tianbao. Doubly stochastic primal-dual coordinate method for empirical risk minimization and bilinear saddle-point problem. *arXiv preprint arXiv:1508.03390*, 2015.

Zhang, Yuchen and Xiao, Lin. Stochastic primal-dual coordinate method for regularized empirical risk minimization. *arXiv preprint arXiv:1409.3257*, 2014.