Adaptive Stochastic Primal-Dual Coordinate Descent for Separable Saddle Point Problems

Zhanxing Zhu(✉) and Amos J. Storkey

Institute of Adaptive Neural Computation, School of Informatics,
The University of Edinburgh, Edinburgh EH8 9AB, UK
{zhanxing.zhu,a.storkey}@ed.ac.uk

Abstract. We consider a generic convex-concave saddle point problem with a *separable* structure, a form that covers a wide-ranged machine learning applications. Under this problem structure, we follow the framework of primal-dual updates for saddle point problems, and incorporate stochastic block coordinate descent with *adaptive* stepsizes into this framework. We theoretically show that our proposal of adaptive stepsizes potentially achieves a sharper linear convergence rate compared with the existing methods. Additionally, since we can select "mini-batch" of block coordinates to update, our method is also amenable to *parallel* processing for large-scale data. We apply the proposed method to regularized empirical risk minimization and show that it performs comparably or, more often, better than state-of-the-art methods on both synthetic and real-world data sets.

Keywords: Large-scale optimization · Parallel optimization · Stochastic coordinate descent · Convex-concave saddle point problems

1 Introduction

The generic convex-concave saddle point problem is written as

$$\min_{\mathbf{x}\in\mathbb{R}^d}\max_{\mathbf{y}\in\mathbb{R}^q}\left\{L(\mathbf{x},\mathbf{y})=g(\mathbf{x})+\langle\mathbf{x},\mathbf{K}\mathbf{y}\rangle-\phi^*(\mathbf{y})\right\}, \tag{1}$$

where $g(\mathbf{x})$ is a proper convex function, $\phi^*(\cdot)$ is the convex conjugate of a convex function $\phi(\cdot)$, and matrix $\mathbf{K}\in\mathbb{R}^{d\times q}$. Many machine learning tasks reduce to solving a problem of this form [3,6]. As a result, this saddle problem has been widely studied [1,2,4,5,14,16].

One important subclass of the general convex concave saddle point problem is where $g(\mathbf{x})$ or $\phi^*(\mathbf{y})$ exhibits an additive separable structure. We say $\phi^*(\mathbf{y})$ is *separable* when $\phi^*(\mathbf{y})=\frac{1}{n}\sum_{i=1}^n\phi_i^*(\mathbf{y}_i)$, with $\mathbf{y}_i\in\mathbb{R}^{q_i}$, and $\sum_{i=1}^n q_i=q$. Separability for $g(\cdot)$ is defined likewise. To keep the consistent notation for the machine learning applications discussed later, we introduce matrix $\mathbf{A}$ and let

A. Appice et al. (Eds.): ECML PKDD 2015, Part I, LNAI 9284, pp. 645–658, 2015.
DOI: 10.1007/978-3-319-23528-8_40

$\mathbf{K} = \frac{1}{n}\mathbf{A}$. Then we partition matrix $\mathbf{A}$ into n column blocks $\mathbf{A}_i \in \mathbb{R}^{d\times q_i}$, $i = 1, \ldots, n$, and $\mathbf{K}\mathbf{y} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{A}_i\mathbf{y}_i$, resulting in a problem of the form

$$\min_{\mathbf{x}\in\mathbb{R}^d}\max_{\mathbf{y}\in\mathbb{R}^q}\left\{L(\mathbf{x},\mathbf{y}) = g(\mathbf{x}) + \frac{1}{n}\sum_{i=1}^{n}\left(\langle\mathbf{x}, \mathbf{A}_i\mathbf{y}_i\rangle - \phi_i^*(\mathbf{y}_i)\right)\right\} \tag{2}$$

for $\phi^*(\cdot)$ separable. We call any problem of the form (1) where $g(\cdot)$ or $\phi^*(\cdot)$ has separable structure, a Separable Convex Concave Saddle Point (*Sep-CCSP*) problem. Eq. (2) gives the explicit form for when $\phi^*(\cdot)$ is separable.

In this work, we further assume that each $\phi_i^*(\mathbf{y}_i)$ is γ-strongly convex, and $g(\mathbf{x})$ is λ-strongly convex, i.e.,

$$\phi_i^*(\mathbf{y}_i') \geq \phi_i^*(\mathbf{y}_i) + \nabla\phi^*(\mathbf{y}_i)^T(\mathbf{y}_i' - \mathbf{y}_i) + \frac{\gamma}{2}\|\mathbf{y}_i' - \mathbf{y}_i\|_2^2, \quad \forall \mathbf{y}_i, \mathbf{y}_i' \in \mathbb{R}^{q_i}$$

$$g(\mathbf{x}') \geq g(\mathbf{x}) + \nabla g(\mathbf{x})^T(\mathbf{x}' - \mathbf{x}) + \frac{\lambda}{2}\|\mathbf{x}_i' - \mathbf{x}_i\|_2^2, \quad \forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^d,$$

where we use ∇ to denote both the gradient for smooth function and subgradient for non-smooth function. When the strong convexity cannot be satisfied, a small strongly convex perturbation can be added to make the problem satisfy the assumption [15].

One important instantiation of the Sep-CCSP problem in machine learning is the regularized empirical risk minimization (ERM, [3]) of linear predictors,

$$\min_{\mathbf{x}\in\mathbb{R}^d}\left\{J(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\phi_i(\mathbf{a}_i^T\mathbf{x}) + g(\mathbf{x})\right\}, \tag{3}$$

where $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^d$ are the feature vectors of n data samples, $\phi_i(\cdot)$ corresponds the convex loss function w.r.t. the linear predictor $\mathbf{a}_i^T\mathbf{x}$, and $g(\mathbf{x})$ is a convex regularization term. Many practical classification and regression models fall into this regularized ERM formulation, such as linear support vector machine (SVM), regularized logistic regression and ridge regression, see [3] for more details.

Reformulating the above regularized ERM by employing conjugate dual of the function $\phi_i(\cdot)$, i.e.

$$\phi_i^*(\mathbf{a}_i^T\mathbf{x}) = \max_{y_i\in\mathbb{R}}\langle\mathbf{x}, y_i\mathbf{a}_i\rangle - \phi_i^*(y_i), \tag{4}$$

leads directly to the following Sep-CCSP problem

$$\min_{\mathbf{x}\in\mathbb{R}^d}\max_{\mathbf{y}\in\mathbb{R}^n} g(\mathbf{x}) + \frac{1}{n}\sum_{i=1}^{n}\left(\langle\mathbf{x}, y_i\mathbf{a}_i\rangle - \phi_i^*(y_i)\right). \tag{5}$$

Comparing with the general form, we note that the matrix $\mathbf{A}_i$ in (2) is now a vector $\mathbf{a}_i$. For solving the general saddle point problem (1), many primal-dual algorithms can be applied, such as [1,2,4,5,16]. In addition, the saddle point problem we consider can also be formulated as a composite function minimization problem and then solved by Alternating Direction Method of Multipliers (ADMM) methods [9].

To handle the Sep-CCSP problem particularly for regularized ERM problem (5), Zhang and Xiao [15] proposed a stochastic primal-dual coordinate descent (SPDC) method. SPDC applies stochastic coordinate descent method [8,10,11] into the primal-dual framework, where in each iteration a random subset of dual coordinates are updated. This method inherits the efficiency of stochastic coordinate descent for solving large-scale problems. However, they use a conservative constant stepsize during the primal-dual updates, which leads to an unsatisfying convergence rate especially for unnormalized data.

In this work, we propose an *adaptive* stochastic primal-dual coordinate descent (*AdaSPDC*) method for solving the Sep-CCSP problem (2), which is a non-trivial extension of SPDC. By carefully exploiting the structure of individual subproblem, we propose an adaptive stepsize rule for both primal and dual updates according to the chosen subset of coordinate blocks in each iteration. Both theoretically and empirically, we show that AdaSPDC could yield a significantly better convergence performance than SPDC and other state-of-the-art methods.

The remaining structure of the paper is as follows. Section 2 summarizes the general primal-dual framework our method and SPDC are based on. Then we elaborate our method AdaSPDC in Section 3, where both the theoretical result and its comparison with SPDC are provided. In Section 4, we apply our method into regularized ERM tasks, and experiment with both synthetic and real-world datasets, and we show the superiority of AdaSPDC over other competitive methods empirically. Finally, Section 5 concludes the work.

2 Primal-dual Framework for Convex-Concave Saddle Point Problems

Chambolle and Pock [1] proposed a first-order primal-dual method for the CCSP problem (1). We refer this algorithm as PDCP. The update of PDCP in the $(t+1)$th iteration is as follows:

$$\mathbf{y}^{t+1} = \text{argmin}_{\mathbf{y}} \phi^*(\mathbf{y}) - \langle \overline{\mathbf{x}}^t, \mathbf{K}\mathbf{y} \rangle + \frac{1}{2\sigma}\|\mathbf{y} - \mathbf{y}^t\|_2^2 \tag{6}$$

$$\mathbf{x}^{t+1} = \text{argmin}_{\mathbf{x}} g(\mathbf{x}) + \langle \mathbf{x}, \mathbf{K}\mathbf{y}^{t+1} \rangle + \frac{1}{2\tau}\|\mathbf{x} - \mathbf{x}^t\|_2^2 \tag{7}$$

$$\overline{\mathbf{x}}^{t+1} = \mathbf{x}^{t+1} + \theta(\mathbf{x}^{t+1} - \mathbf{x}^t). \tag{8}$$

When the parameter configuration satisfies $\tau\sigma \leq 1/\|\mathbf{K}\|^2$ and $\theta = 1$, PDCP could achieve $O(1/T)$ convergence rate for general convex function $\phi^*(\cdot)$ and $g(\cdot)$, where T is total number of iterations. When $\phi^*(\cdot)$ and $g(\cdot)$ are both strongly convex, a linear convergence rate can be achieved by using a more scheduled stepsize. PDCP is a batch method and non-stochastic, i.e., it has to update all the dual coordinates in each iteration for Sep-CCSP problem, which will be computationally intensive for large-scale (high-dimensional) problems.

SPDC [15] can be viewed as a stochastic variant of the batch method PDCP for handling Sep-CCSP problem. However, SPDC uses a conservative constant stepsize for primal and dual updates. Both PDCP and SPDC do not consider the structure of matrix $\mathbf{K}$ and only apply constant stepsize for all coordinates of primal and dual variables. This might limit their convergence performance in reality.

Based on this observation, we exploit the structure of matrix $\mathbf{K}$ (i.e., $\frac{1}{n}\mathbf{A}$) and propose an adaptive stepsize rule for efficiently solving Sep-CCSP problem. A better linear convergence rate could be yielded when $\phi_i^*(\cdot)$ and $g(\cdot)$ are strongly convex. Our algorithm will be elaborated in the following section.

3 Adaptive Stochastic Primal-Dual Coordinate Descent

As a non-trivial extension of SPDC [15], our method AdaSPDC solves the Sep-CCSP problem (2) by using an adaptive parameter configuration. Concretely, we optimize $L(\mathbf{x}, \mathbf{y})$ by alternatively updating the dual and primal variables in a principled way. Thanks to the separable structure of $\phi(\mathbf{y})$, in each iteration we can randomly select m blocks of dual variables whose indices are denoted as S_t, and then we only update these selected blocks in the following way,

$$\mathbf{y}_i^{t+1} = \mathrm{argmin}_{\mathbf{y}_i} \left[\phi_i(\mathbf{y}_i) - \left\langle \overline{\mathbf{x}}^t, \mathbf{A}_i \mathbf{y}_i \right\rangle + \frac{1}{2\sigma_i} \|\mathbf{y}_i - \mathbf{y}_i^t\|_2^2 \right], \text{ if } i \in S_t. \tag{9}$$

For those coordinates in blocks not selected, $i \notin S_t$, we just keep $\mathbf{y}_i^{t+1} = \mathbf{y}_i^t$. By exploiting the structure of individual $\mathbf{A}_i$, we configure the stepsize parameter of the proximal term σ_i *adaptively*

$$\sigma_i = \frac{1}{2R_i} \sqrt{\frac{n\lambda}{m\gamma}}, \tag{10}$$

where $R_i = \|\mathbf{A}_i\|_2 = \sqrt{\mu_{\max}\left(\mathbf{A}_i^T \mathbf{A}_i\right)}$, with $\|\cdot\|_2$ is the spectral norm of a matrix and $\mu_{\max}(\cdot)$ to denote the maximum singular value of a matrix.

Our step size is different from the one used in SPDC [15], where R is a constant $R = \max\{\|\mathbf{a}_i\|_2 : i = 1, \ldots, n\}$ (since SPDC only considers ERM problem, the matrix $\mathbf{A}_i$ is a feature vector $\mathbf{a}_i$).

Remark. Intuitively, R_i in AdaSPDC can be understood as the coupling strength between the i-th dual variable block and primal variable, measured by the spectral norm of matrix $\mathbf{A}_i$. Smaller coupling strength allows us to use larger stepsize for the current dual variable block without caring too much about its influence on primal variable, and vice versa. Compared with SPDC, our proposal of an adaptive coupling strength for the chosen coordinate block directly results in larger step size, and thus helps to improve convergence speed.

In the stochastic dual update, we also use an intermediate variable $\overline{\mathbf{x}}^t$ as in PDCP algorithm, and we will describe its update later.

Since we assume $g(\mathbf{x})$ is not separable, we update the primal variable as a whole,

$$\mathbf{x}^{t+1} = \text{argmin}_{\mathbf{x}} \left[g(\mathbf{x}) + \left\langle \mathbf{x}, \mathbf{r}^t + \frac{1}{m} \sum_{j \in S_t} \mathbf{A}_j (\mathbf{y}_j^{t+1} - \mathbf{y}_j^t) \right\rangle + \frac{1}{2\tau^t} \|\mathbf{x} - \mathbf{x}^t\|_2^2 \right]. \tag{11}$$

The proximal parameter τ^t is also configured *adaptively*,

$$\tau^t = \frac{1}{2R_{\max}^t} \sqrt{\frac{m\gamma}{n\lambda}}, \tag{12}$$

where $R_{\max}^t = \max\{R_i | i \in S_t\}$, compared with constant R used in SPDC. To account for the incremental change after the latest dual update, an auxiliary variable $\mathbf{r}^t = \frac{1}{n}\sum_{i=1}^n \mathbf{A}_i \mathbf{y}_i^t$ is used and updated as follows

$$\mathbf{r}^{t+1} = \mathbf{r}^t + \frac{1}{n} \sum_{j \in S_t} \mathbf{A}_j \left(\mathbf{y}_j^{t+1} - \mathbf{y}_j^t\right). \tag{13}$$

Finally, we update the intermediate variable $\overline{\mathbf{x}}$, which implements an extrapolation step over the current $\mathbf{x}^{t+1}$ and can help to provide faster convergence rate [1,7].

$$\overline{\mathbf{x}}^{t+1} = \mathbf{x}^{t+1} + \theta^t (\mathbf{x}^{t+1} - \mathbf{x}^t), \tag{14}$$

where θ^t is configured adaptively as

$$\theta^t = 1 - \frac{1}{n/m + R_{\max}^t \sqrt{(n/m)/(\lambda\gamma)}}, \tag{15}$$

which is contrary to the constant θ used in SPDC.

The whole procedure for solving Sep-CCSP problem (2) using AdaSPDC is summarized in Algorithm 1. There are several notable characteristics of our algorithms.

- Compared with SPDC, our method uses adaptive step size to obtain faster convergence (will be shown in Theorem 1), while the whole algorithm does not bring any other extra computational complexity. As demonstrated in the experiment Section 4, in many cases, AdaSPDC provides significantly better performance than SPDC.
- Since, in each iteration, a number of block coordinates can be chosen and updated independently (with independent evaluation of individual step size), this directly enables parallel processing, and hence use on modern computing clusters. The ability to select an arbitrary number of blocks can help to make use of all the computation structure available as effectively as possible.

Algorithm 1. AdaSPDC for Separable Convex-Concave Saddle Point Problems

```
Input: number of blocks picked in each iteration $m$ and number of iterations $T$.
Initialize: $\mathbf{x}^0$, $\mathbf{y}^0$, $\overline{\mathbf{x}}^0 = \mathbf{x}^0$, $\mathbf{r}^0 = \frac{1}{n}\sum_{i=1}^n \mathbf{A}_i \mathbf{y}_i^0$
for $t = 0, 1, \ldots, T-1$ do
    Randomly pick $m$ dual coordinate blocks from $\{1, \ldots, n\}$ as indices set $S_t$, with
    the probability of each block being selected equal to $m/n$.
    According to the selected subset $S_t$, compute the adaptive parameter configura-
    tion of $\sigma_i$, $\tau^t$ and $\theta^t$ using Eq. (10), (12) and (15), respectively.
    for each selected block in parallel do
        Update the dual variable block using Eq.(9).
    end for
    Update primal variable using Eq.(11).
    Extrapolate primal variable block using Eq.(14).
    Update the auxiliary variable $\mathbf{r}$ using Eq.(13).
end for
```

3.1 Convergence Analysis

We characterise the convergence performance of our method in the following theorem.

Theorem 1. *Assume that each $\phi_i^*(\cdot)$ is γ-strongly convex, and $g(\cdot)$ is λ-strongly convex, and given the parameter configuration in Eq. (10), (12) and (15), then after T iterations in Algorithm 1, the algorithm achieves the following convergence performance*

$$\begin{aligned}&\left(\frac{1}{2\tau^T}+\lambda\right)\mathbb{E}\left[\|\mathbf{x}^T-\mathbf{x}^\star\|_2^2\right]+\mathbb{E}\left[\|\mathbf{y}^T-\mathbf{y}^\star\|_{\boldsymbol{\nu}}^2\right]\\ \le&\left(\prod_{t=1}^{T}\theta^t\right)\left(\left(\frac{1}{2\tau^T}+\lambda\right)\|\mathbf{x}^0-\mathbf{x}^\star\|_2^2+\|\mathbf{y}^0-\mathbf{y}^\star\|_{\boldsymbol{\nu}'}^2\right),\end{aligned} \tag{16}$$

where $(\mathbf{x}^\star, \mathbf{y}^\star)$ is the optimal saddle point, $\nu_i = \frac{1/(4\sigma_i)+\gamma}{m}$, $\nu_i' = \frac{1/(2\sigma_i)+\gamma}{m}$, and $\|\mathbf{y}^T-\mathbf{y}^\star\|_{\boldsymbol{\nu}}^2 = \sum_{i=1}^n \nu_i \|\mathbf{y}_i^T-\mathbf{y}_i^\star\|_2^2$.

Since the proof of the above is technical, we provide it in the full version of this paper [17].

In our proof, given the proposed parameter θ^t, the critical point for obtaining a sharper linear convergence rate than SPDC is that we configure τ^t and σ_i as Eq. (12) and (10) to guarantee the positive definiteness of the following matrix in the t-th iteration,

$$\mathbf{P} = \begin{bmatrix} \frac{m}{2\tau^t}\mathbf{I} & -\mathbf{A}_{S_t} \\ -\mathbf{A}_{S_t}^T & \frac{1}{2\mathrm{diag}(\boldsymbol{\sigma}_{S_t})} \end{bmatrix}, \tag{17}$$

where $\mathbf{A}_{S_t} = [\ldots, \mathbf{A}_i, \ldots] \in \mathbb{R}^{d\times mq_i}$ and $\mathrm{diag}(\boldsymbol{\sigma}_{S_t}) = \mathrm{diag}(\ldots, \sigma_i \mathbf{I}_{q_i}, \ldots)$ for $i \in S_t$. However, we found that the parameter configuration to guarantee the positive definiteness of $\mathbf{P}$ is not unique, and there exist other valid parameter configurations besides the proposed one in this work. We leave the further investigation on other potential parameter configurations as future work.

3.2 More Comparison with SDPC

Compared with SPDC [15], AdaSPDC follows the similar primal-dual framework. The crucial difference between them is that AdaSPDC proposes a larger stepsize for both dual and primal updates, see Eq. (10) and (12) compared with SPDC's parameter configuration given in Eq.(10) in [15], where SPDC applies a large constant $R = \max\{\|\mathbf{a}_i\|_2 : i = 1, \dots, n\}$ while AdaSPDC uses a more adaptive value of R_i and $R^t_{\max}$ for t-th iteration to account for the different coupling strength between the selected dual coordinate block and primal variable. This difference directly means that AdaSPDC can potentially obtain a significantly sharper linear convergence rate than SPDC, since the decay factor θ^t of AdaSPDC is smaller than θ in SPDC (Eq.(10) in [15]) , see Theorem 1 for AdaSPDC compared with SPDC (Theorem 1 in [15]). The empirical performance of the two algorithms will be demonstrated in the experimental Section 4.

To mitigate the problem that SPDC uses a large R, the authors of SPDC proposes to non-uniformly sample the the dual coordinate to update in each iteration according to the norm of the each $\mathbf{a}_i$. However, as we show later in the empirical experiments, this non-uniform sampling does not work very well for some datasets. By configuring the adaptive stepsize explicitly, our method AdaSPDC provides a better solution for unnormalized data compared with SPDC, see Section 4 for more empirical evidence.

Another difference is that SPDC only considers the regularized ERM task, i.e., only handling the case that each $\mathbf{A}_i$ is a feature vector $\mathbf{a}_i$, while AdaSPDC extends that $\mathbf{A}_i$ can be a matrix so that AdaSPDC can cover a wider range of applications than SPDC, i.e. in each iteration, a number of *block* coordinates could be selected while for SPDC only a number of coordinates are allowed.

4 Empirical Results

In this section, we appy AdaSPDC to several regularized empirical risk minimization problems. The experiments are conducted to compare our method AdaSPDC with other competitive stochastic optimization methods, including SDCA [13], SAG [12], SPDC with uniform sampling and non-uniform sampling [15]. In order to provide a fair comparison with these methods, in each iteration only one dual coordinate (or data instance) is chosen, i.e., we run all the methods sequentially. To obtain results that are independent of the practical implementation of the algorithm, we measure the algorithm performance in term of objective suboptimality w.r.t. the effective passes to the entire data set.

Each experiment is run 10 times and the average results are reported to show statistical consistency. We present all the experimental results we have done for each application.

4.1 Ridge Regression

We firstly apply our method AdaSPDC into a simple ridge regression problem with synthetic data. The data is generated in the same way as Zhang and Xiao [15];

$n = 1000$ i.i.d. training points $\{\mathbf{a}_i, b_i\}_{i=1}^n$ are generated in the following manner,

$$b = \mathbf{a}^T \mathbf{x}^\diamond + \epsilon, \quad \mathbf{a} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}), \quad \epsilon \sim \mathcal{N}(0, 1),$$

where $\mathbf{a} \in \mathbb{R}^d$ and $d = 1000$, and the elements of the vector $\mathbf{x}^\diamond$ are all ones. The covariance matrix $\mathbf{\Sigma}$ is set to be diagonal with $\Sigma_{jj} = j^{-2}$, for $j = 1, \ldots, d$. Then the ridge regression tries to solve the following optimization problem,

$$\min_{\mathbf{x} \in \mathbb{R}^d} \left\{ J(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} (\mathbf{a}_i^T \mathbf{x} - b_i)^2 + \frac{\lambda}{2} \|\mathbf{x}\|_2^2 \right\}. \tag{18}$$

The optimal solution of the above ridge regression can be found as

$$\mathbf{x}^\star = \left(\mathbf{A}\mathbf{A}^T + n\lambda \mathbf{I}_d \right)^{-1} \mathbf{A}\mathbf{b}.$$

By employing the conjugate dual of quadratic loss (crossref, Eq. (4)), we can reformulate the ridge regression as the following Sep-CCSP problem,

$$\min_{\mathbf{x} \in \mathbb{R}^d} \max_{\mathbf{y} \in \mathbb{R}^n} \frac{\lambda}{2} \|\mathbf{x}\|_2^2 + \frac{1}{n} \sum_{i=1}^{n} \left(\langle \mathbf{x}, y_i \mathbf{a}_i \rangle - \left(\frac{1}{2} y_i^2 + b_i y_i \right) \right). \tag{19}$$

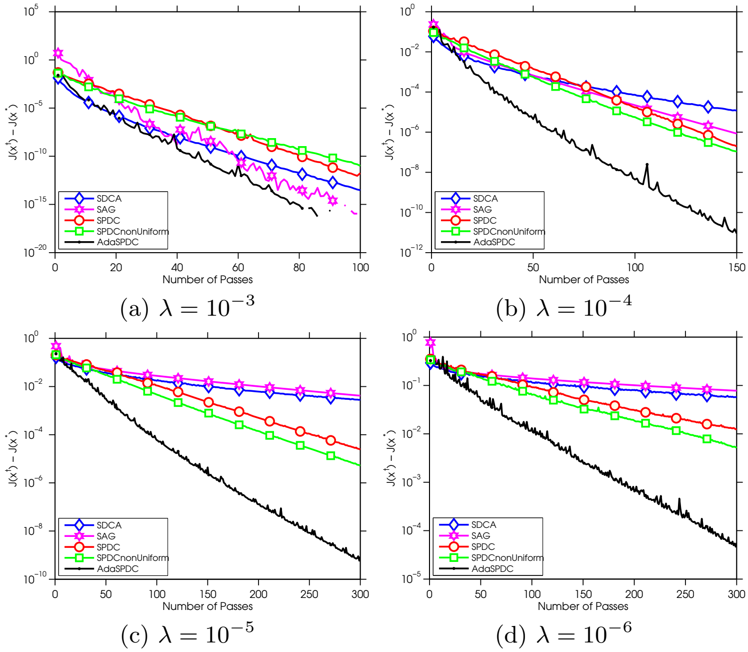
It is easy to figure out that $g(\mathbf{x}) = \lambda/2 \|\mathbf{x}\|_2^2$ is λ-strongly convex, and $\phi_i^*(y_i) = \frac{1}{2} y_i^2 + b_i y_i$ is 1-strongly convex.

Thus, for ridge regression, the dual update in Eq. (9) and primal update in Eq. (11) of AdaSPDC have closed form solutions as below,

$$\begin{aligned} y_i^{t+1} &= \frac{1}{1 + 1/\sigma_i} \left(\langle \overline{\mathbf{x}}^t, \mathbf{a}_i \rangle + b_i + \frac{1}{\sigma_i} y_i \right), \text{ if } i \in S_t \\ \mathbf{x}^{t+1} &= \frac{1}{\lambda + 1/\tau^t} \left(\frac{1}{\tau^t} \mathbf{x}^t - \left(\mathbf{r}^t + \frac{1}{m} \sum_{j \in S_t} \mathbf{a}_j (y_j^{t+1} - y_j^t) \right) \right) \end{aligned}$$

The algorithm performance is evaluated in term of objective suboptimality (measured by $J(\mathbf{x}^t) - J(\mathbf{x}^\star)$) w.r.t. number of effective passes to the entire datasets. Varying values of regularization parameter λ are experimented to demonstrate algorithm performance with different degree of ill-conditioning, $\lambda = \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$.

Fig. 1 shows algorithm performance with different degrees of regularization. It is easy to observe that AdaSPDC converges substantially faster than other compared methods, particularly for ill-conditioned problems. Compared with SPDC and its variant with non-uniform sampling, the usage of adaptive stepsize in AdaSPDC significantly improves convergence speed. For instance, in the case with $\lambda = 10^{-6}$, AdaSPDC achieves 100 times better suboptimality than both SPDC and its variant SPDC with non-uniform sampling after 300 passes.



(a) $\lambda = 10^{-3}$

(b) $\lambda = 10^{-4}$

(c) $\lambda = 10^{-5}$

(d) $\lambda = 10^{-6}$

Fig. 1. Ridge regression with synthetic data: comparison of convergence performance w.r.t. the number of passes. Problem size: $d = 1000, n = 1000$. We evaluate the convergence performance using objective suboptimality, $J(\mathbf{x}^t) - J(\mathbf{x}^\star)$.

4.2 Binary Classification on Real-world Datasets

We now compare the performance of our method AdaSPDC with other competitive methods on several real-world data sets. Our experiments focus on the freely-available benchmark data sets for binary classification, whose detailed information are listed in Table 1. The *w8a*, *covertype* and *url* data are obtained from the LIBSVM collection[1]. The *quantum* and *protein* data sets are obtained from KDD Cup 2004[2]. For all the datasets, each sample takes the form $(\mathbf{a}_i, b_i)$

Table 1. Benchmark datasets used in our experiments for binary classification.

Datasets	Number of samples	Number of features	Sparsity
w8a	49,749	300	3.9%
covertype	20,242	47,236	0.16%
url	2,396,130	3,231,961	0.0018%
quantum	50,000	78	43.44%
protein	145,751	74	99.21%

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

[2] http://osmot.cs.cornell.edu/kddcup/datasets.html

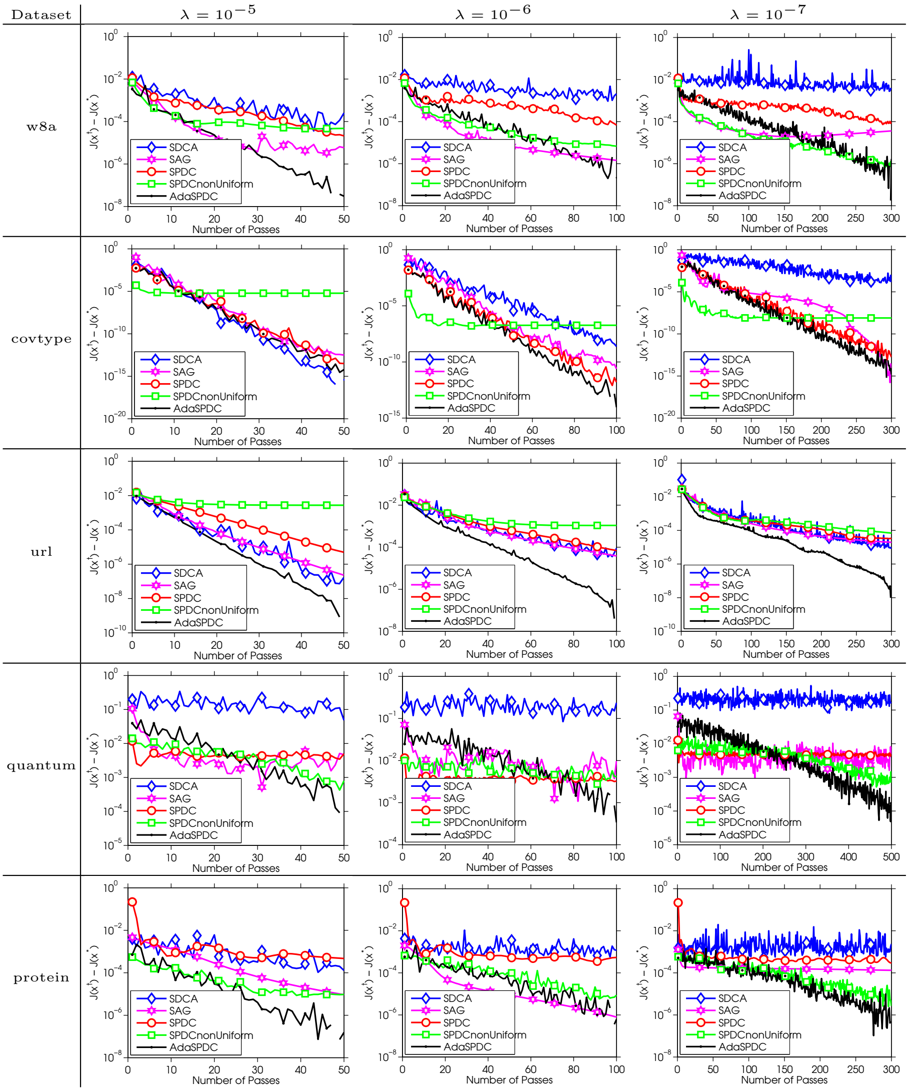


Fig. 2. Comparison of algorithm performance with smooth Hinge loss.

with $\mathbf{a}_i$ is the feature vector and b_i is the binary label -1 or 1. We add a bias term to the feature vector for all the datasets. We aim to minimize the regularized empirical risk with following form

$$J(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} \phi_i(\mathbf{a}_i^T \mathbf{x}) + \frac{\lambda}{2} \|\mathbf{x}\|_2^2 \tag{20}$$

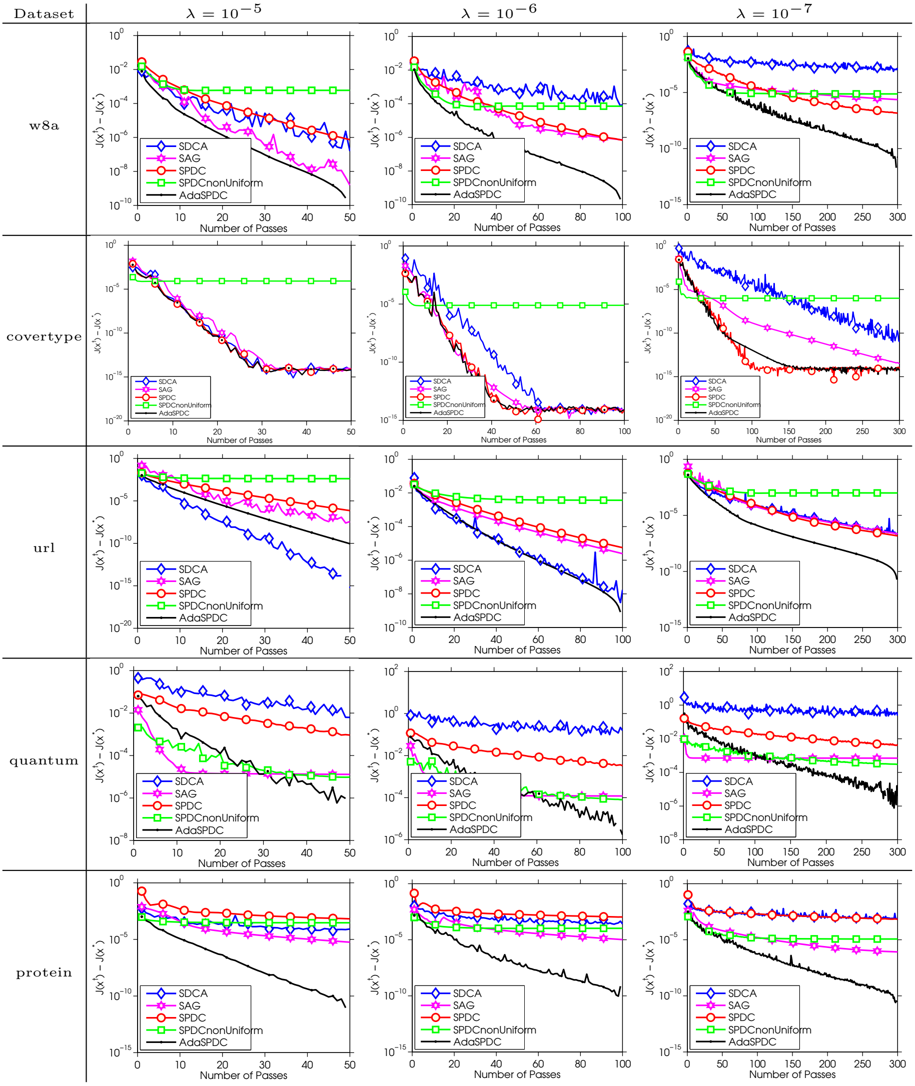


Fig. 3. Comparison of algorithm performance with Logistic loss.

To provide a more comprehensive comparison between these methods, we experiment with two different loss function $\phi_i(\cdot)$, smooth Hinge loss [13] and logistic loss, described in the following.

Smooth Hinge loss (with smoothing parameter $\gamma = 1$.)

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

And its conjugate dual is

$$\phi_i^*(y_i) = b_i y_i + \frac{1}{2} y_i^2, \text{ with } b_i y_i \in [-1, 0].$$

We can observe that $\phi_i^*(y_i)$ is γ-strongly convex with $\gamma = 1$. The dual update of AdaSPDC for smooth Hinge loss is nearly the same with ridge regression except the necessity of projection into the interval $b_i y_i \in [-1, 0]$.

Logistic loss

$$\phi_i(z) = \log\left(1 + \exp(-b_i z)\right),$$

whose conjugate dual has the form

$$\phi_i^*(y_i) = -b_i y_i \log(-b_i y_i) + (1 + b_i y_i) \log(1 + b_i y_i) \text{ with } b_i y_i \in [-1, 0].$$

It is also easy to obtain that $\phi_i^*(y_i)$ is γ-strongly convex with $\gamma = 4$. Note that for logistic loss, the dual update in Eq. (9) does not have a closed form solution, and we can start from some initial solution and further apply several steps of Newton's update to obtain a more accurate solution.

During the experiments, we observe that the performance of SAG is very sensitive to the stepsize choice. To obtain best results of SAG, we try different choices of stepsize in the interval $[1/16L, 1/L]$ and report the best result for each dataset, where L is Lipschitz constant of $\phi_i(\mathbf{a}_i^T \mathbf{x})$, $1/16L$ is the theoretical stepsize choice for SAG and $1/L$ is the suggested empirical choice [12]. For smooth Hinge loss, $L = \max_i\{\|\mathbf{a}_i\|_2, i = 1, \ldots, n\}$, and for logistic loss, $L = \frac{1}{4}\max_i\{\|\mathbf{a}_i\|_2, i = 1, \ldots, n\}$.

Fig. 2 and Fig. 3 depict the algorithm performance on the different methods with smooth Hinge loss and logistics loss, respectively. We compare all these methods with different values of $\lambda = \{10^{-5}, 10^{-6}, 10^{-7}\}$. Generally, our method AdaSPDC performs consistently better or at least comparably with other methods, and performs especially well for the tasks with small regularized parameter λ. For some datasets, such as covertype and quantum, SPDC with non-uniform sampling decreases the objective faster than other methods in early epochs, however, cannot achieve comparable results with other methods in later epochs, which might be caused by its conservative stepsize.

5 Conclusion and Future Work

In this work, we propose Adaptive Stochastic Primal-Dual Coordinate Descent (AdaSPDC) for separable saddle point problems. As a non-trivial extension of a recent work SPDC [15], AdaSPDC uses an adaptive step size choices for both primal and dual updates in each iteration. The design of the step size for our method AdaSPDC explicitly and adaptively models the coupling strength between chosen block coordinates and primal variable through the spectral norm of each $\mathbf{A}_i$. We theoretically characterise that AdaSPDC holds a sharper linear convergence rate than SDPC. Additionally, we demonstrate the superiority of the proposed AdaSPDC method on ERM problems through extensive experiments on both synthetic and real-world data sets.

An immediate further research direction is to investigate other valid parameter configurations for the extrapolation parameter θ, and the primal and dual step sizes τ and σ both theoretically and empirically. In addition, discovering the potential theoretical connections with other stochastic optimization methods will also be enlightening.

Acknowledgments. Z. Zhu is supported by China Scholarship Council/University of Edinburgh Joint Scholarship. The authors would like to thank Jinli Hu for insightful discussion on the proof of Theorem 1.

References

1. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision **40**(1), 120–145 (2011)
2. Esser, E., Zhang, X., Chan, T.: A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. SIAM Journal on Imaging Sciences **3**(4), 1015–1046 (2010)
3. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning, vol. 2. Springer (2009)
4. He, B., Yuan, X.: Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. SIAM Journal on Imaging Sciences **5**(1), 119–149 (2012)
5. He, Y., Monteiro, R.D.: An accelerated hpe-type algorithm for a class of composite convex-concave saddle-point problems. Optimization-online preprint (2014)
6. Jacob, L., Obozinski, G., Vert, J.P.: Group lasso with overlap and graph lasso. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 433–440. ACM (2009)
7. Nesterov, Y.: Introductory lectures on convex optimization: A basic course, vol. 87. Springer (2004)
8. Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM Journal on Optimization **22**(2), 341–362 (2012)
9. Ouyang, Y., Chen, Y., Lan, G., Pasiliao Jr., E.: An accelerated linearized alternating direction method of multipliers. SIAM Journal on Imaging Sciences **8**(1), 644–681 (2015)
10. Richtárik, P., Takáč, M.: Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Mathematical Programming **144**(1–2), 1–38 (2014)
11. Richtárik, P., Takáč, M.: Parallel coordinate descent methods for big data optimization. Mathematical Programming, 1–52 (2012)
12. Schmidt, M., Roux, N.L., Bach, F.: Minimizing finite sums with the stochastic average gradient. arXiv preprint arXiv:1309.2388 (2013)
13. Shalev-Shwartz, S., Zhang, T.: Stochastic dual coordinate ascent methods for regularized loss. The Journal of Machine Learning Research **14**(1), 567–599 (2013)

14. Tseng, P.: On accelerated proximal gradient methods for convex-concave optimization. submitted to SIAM Journal on Optimization (2008)
15. Zhang, Y., Xiao, L.: Stochastic primal-dual coordinate method for regularized empirical risk minimization. In: International Conference of Machine Learning (2015)
16. Zhu, M., Chan, T.: An efficient primal-dual hybrid gradient algorithm for total variation image restoration. UCLA CAM Report, pp. 08–34 (2008)
17. Zhu, Z., Storkey, A.J.: Adaptive stochastic primal-dual coordinate descent for separable saddle point problems. arXiv preprint arXiv:1506.04093 (2015)