Stochastic Dual Coordinate Descent

Guillaume Desforges & Michaël Karpe & Matthieu Roux

June 14th 2018

Introduction

In machine learning, the process of fitting a model to the data requires to solve an optimization problem. The difficulty resides in the fact that this optimization quickly becomes very complex when dealing with real problems. The Stochastic Gradient Descent (SGD) is a very popular algorithm to solve those problems because it has good convergence guaranties. Yet, the SGD does not have a good stopping criteria, and its solutions are often not accurate enough.

The Stochastic Dual Coordinate Ascent (SDCA) tries to solves the optimization problem by solving its dual problem. Instead of optimizing the weights, we optimize a dual variable from which we can compute the weights and thus solve the former. This method can give good results for specific problems : for instance, solving the dual problem of the SVM has proven to be effective and to give interesting results, with a linear convergence in some cases.

In this report, we compile the key theoretical points necessary to have a global understanding of the SDCA.

First we introduce the SDCA and its principles. We then present the machine learning problem our report focuses on. Then we study computational performances of the method by trying to apply SDCA on concret problems. Finally we conclude on SDCA strengths and weaknesses.

Note *We especially added experimentations since the presentation of our poster.*

1 Purpose of the report: a new SGD-like method

1.1 Difference between SGD and SDCA

A simple approach for solving Support Vector Machine learning is Stochastic gradient Descent (SGD). SGD finds an ϵ_P-sub-optimal solution in time $O(1/(\lambda\epsilon_P))$. This runtime does not depend on n and therefore is favorable when n is very large. However, the SGD approach has several disadvantages:

1. it does not have a clear stopping criterion
2. it tends to be too aggressive at the beginning of the optimization process, especially when λ is very small
3. while SGD reaches a moderate accuracy quite fast, its convergence becomes rather slow when we are interested in more accurate solutions

Therefore, an alternative approach is Dual Coordinate Ascent (DCA), which solves the dual problem instead of the primal problem.

1.2 General SDCA procedure

Let $x_1, \dots, x_n \in \mathbb{R}^d$, $\phi_1, \dots, \phi_n$ scalar convex functions, $\lambda > 0$ regularization parameter.

Let us focus on the following optimization problem:

$$\min_{w \in \mathbb{R}^d} P(w) = \left[\frac{1}{n} \sum_{i=1}^{n} \phi_i(w^\top x_i) + \frac{\lambda}{2} \|w\|^2 \right] \tag{1}$$

with solution $w^* = \arg\min_{w \in \mathbb{R}^d} P(w)$.

We say that a solution w is ϵ_P-sub-optimal if $P(w) - P(w^*) \leq \epsilon_P$.

We analyze here the required runtime to find an ϵ_P-sub-optimal solution using SDCA.

Let $\phi_i^* : \mathbb{R} \to \mathbb{R}$ be the convex conjugate of ϕ_i : $\phi_i^*(u) = \max_z (zu - \phi_i(z))$.

The dual problem of (1) is defined as follows:

$$\max_{\alpha \in \mathbb{R}^n} D(\alpha) = \left[\frac{1}{n} \sum_{i=1}^{n} -\phi_i^*(-\alpha_i) - \frac{\lambda}{2} \left\| \frac{1}{\lambda n} \sum_{i=1}^{n} \alpha_i x_i \right\|^2 \right] \tag{2}$$

with solution $\alpha^* = \arg\max_{a \in \mathbb{R}^n} D(\alpha)$.

We define $w(\alpha) = \frac{1}{\lambda n} \sum_{i=1}^{n} \alpha_i x_i$. Thanks to classic optimization results, we then have :

$$w(\alpha^*) = w^* \tag{3}$$

$$P(w^*) = D(\alpha^*) \tag{4}$$

We define the duality gap as $P(w(\alpha)) - D(\alpha)$. The duality gap reaches 0 when α is optimal.

The SDCA algorithm is described further.

1.3 Focus on the logistic regression

In order to fully grasp the method behind the first paper, let's take an example with the logistic regression. We will consider logistic regression only for binary classification.

We use the following usual notations : $X \in \mathbf{X} = \mathbb{R}^p$ the random variable for the description space, and $Y \in \mathbf{Y} = \{-1, 1\}$ the random variable for the label.

We recall that the model is the following :

$$\frac{\mathbb{P}(y = 1 | X = x)}{\mathbb{P}(y = -1 | X = x)} = w^T x, \quad w \in \mathbb{R}^p \tag{5}$$

We want to find w such that it maximizes the likelihood, or log-likelihood, with a term of regularization:

$$\min_w C \sum_i \log\left(1 + e^{-y_i w^T x_i}\right) + \frac{1}{2} w^T w \tag{6}$$

In order to get the dual problem, we rewrite it with an artificial constraint $z_i = e^{-y_i w^T x_i}$, and we have the following lagrangian :

$$\mathcal{L}(w, z, \alpha) = \sum_i (C \log(1 + z_i) + \alpha_i z_i) - \sum_i \alpha_i e^{-y_i w^T x_i} + \frac{1}{2} w^T w \tag{7}$$

We will note $w^* = \sum_i \alpha_i y_i x_i$ and z^* the variables solution of the optimization problem

$$\min_{w,z} \mathcal{L}(w, z, \alpha) = \mathcal{L}(w^*, z^*, \alpha) = \psi(\alpha) \tag{8}$$

In fact, it leads to the following dual problem :

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i \in I} (-\alpha_i \log(\alpha_i) - (C - \alpha_i) \log(C - \alpha_i)) - \frac{1}{2} \alpha^T Q \alpha \\ s.t. \quad & I = \{i,\ 0 < \alpha_i < C\} \\ & 0 \leq \alpha_i \leq C \end{aligned} \tag{9}$$

Now we got the dual problem, we need to solve a maximization problem. To do so, we will use in this paper the coordinate ascent method, which consist in optimizing the objective function coordinate by coordinate (or with groups of coordinates). The SDCA algorithm is described in the next subsection.

1.4 SDCA algorithm

Algorithm 1 Procedure SCDA

procedure SCDA($\alpha^{(0)}, \phi, T_0, T$)
  $w^{(0)} \leftarrow w(\alpha^{(0)})$
  for $t = 1, \ldots, T$ **do**
    Randomly pick i
    $\Delta\alpha_i \leftarrow \arg\max -\phi_i^*(-(\alpha_i^{(t-1)} + \Delta\alpha_i)) - \frac{\lambda n}{2} \left\| w^{(t-1)} + (\lambda n)^{-1} \Delta\alpha_i x_i \right\|^2$ (*)
    $\alpha^{(t)} \leftarrow \alpha^{(t-1)} + \Delta\alpha_i e_i$
    $w^{(t)} \leftarrow w^{(t-1)} + (\lambda n)^{-1} \Delta\alpha_i x_i$
  if Averaging option **then**
    return $\overline{w} = \frac{1}{T-T_0} \sum_{i=T_0+1}^{T} w^{(t-1)}$
  if Random option **then**
    return $\overline{w} = w^{(t)}$ for a random $t \in [|T_0 + 1, T|]$

1.5 Computation of closed forms

In the studied articles, SDCA is computed either for L-Lipschitz loss functions or for $(1/\gamma)$-smooth loss functions. We recall that a function $\phi_i : \mathbb{R} \to \mathbb{R}$ is L-Lipschitz if $\forall a, b \in \mathbb{R}$, $|\phi_i(a) - \phi_i(b)| \leq L\,|a - b|$, and that a function $\phi_i : \mathbb{R} \to \mathbb{R}$ is $(1/\gamma)$-smooth if it is differentiable and its derivative is $(1/\gamma)$-Lipschitz. Moreover, if ϕ_i is $(1/\gamma)$-smooth, then ϕ_i^* is γ-strongly convex. The different loss functions used are described in the table below. For experimentation, we mainly focused on log loss and square loss.

Some loss functions used in the report are described in the table in appendix A.

1.6 Algorithm termination

For the sake of simplicity, the studied articles consider the following assumptions: $\forall i, \|x_i\| \leq 1, \forall (i,a), \phi_i(a) \geq 0$ and $\forall i, \phi_i(0) \leq 1$. Under these assumptions, we have the following theorem:

Theorem Consider Procedure SDCA with $\alpha^{(0)} = 0$. Assume that $\forall i, \phi_i$ is L-Lipschitz (resp. $(1/\gamma)$-smooth). To obtain an expected duality gap of $\mathbb{E}[P(\overline{w}) - D(\overline{\alpha})] \leq \epsilon_P$, it suffices to have a total number of iterations of

$$T \geq n + \max\left(0, \left\lceil n \log\left(\frac{\lambda n}{2L^2}\right)\right\rceil\right) + \frac{20L^2}{\lambda \epsilon_P} \quad \left(\text{resp. } T > \left(n + \frac{1}{\lambda\gamma}\right)\log\left[\frac{1}{(T-T_0)\epsilon_P}\left(n + \frac{1}{\lambda\gamma}\right)\right]\right)$$

2 Experiments

2.1 Implementation

The experiments in this report were done with our own implementation, available on GitHub :

`https://github.com/GuillaumeDesforges/enpc-malap-project-sdca`

We implemented :

- `Estimator` objects that can fit, predict and score themselves : logistic loss and square loss
- `Optimizer` objects used for fitting : SGD and SDCA
- projections : polynomial and gaussian
- some data utilities

2.2 Description of the chosen data sets

We used our implementation on :

- *Arrhythmia* : `https://archive.ics.uci.edu/ml/datasets/Arrhythmia`
- *Adults* : `https://archive.ics.uci.edu/ml/datasets/adult`

While the Arrhythmia data set has 452 instances, which is quite low, it has 279 features, which is quite high. On the other hand, the Adult data set has 48842 instances but only 14 features.

The Arrhythmia data set will help us check the properties of SDCA when there are many features. The Adult data set will help us compare the SGD and SDCA when there are many instances.

2.3 Use of closed forms and numerical issues

In this report, we used the closed form presented above. The closed form for the logistic regression gave us numerous numerical issues.

On some cases, we can end up with catastrophic cancelations due to either the log or the exp.

A solution that is proposed by another study is to optimize a subproblem with a modified Newton algorithm for each iteration, and thus avoid catastrophic cancelations.

2.4 Choice of algorithm termination option

Because of the stochastic behavior of the algorithm, the output is very sensitive to the iteration at which it stops. Indeed, coefficients vary suddenly, and the convergence is not really monotonous : at some point, it is uncertain whether the loss improves or not.

There are essentially two ways of taking this into account. The first method is to stop at a random step, which actually yields good results. The second method consists in averaging the last $\alpha^{(t)}$ obtained by the algorithm, making sure that the local variations of α are corrected.

Considering this analysis, and as we did not achieve to check the theorem in Section 2.6., we decided to choose the average output option and to set $T_0 = T/2$, as suggested in the studied articles.

2.5 Choice of hyperparameters

The SGD has two hyperparameters `c` and `eps` while the SDCA has only one hyperparameter `c`.

In order to compare the algorithm, we chose to select the best hyperparameters for each optimizer and for each data set.

On every data set, for each hyperparameter, We computed the accuracy after a given number of epochs for a range of values, and plotted them.

The figures are in appendix B.

We selected the hyper parameters values :

Data set	SGD `c`	SGD `eps`	SDCA `c`
Arrhythmia	10^3	10^{-5}	10^{-1}
Adults	10^4	5.10^{-6}	5.10^{-2}

Table 1: Hyper parameter values used

2.6 Stopping time

With such data sets and hyper parameters, we compute the expected stopping time for a dual gap lower than 10^{-3}.

Data set	expected stopping time
Arrhythmia	401549
Adults	629840

Table 2: Expected stopping time

2.7 Comparison between SGD and SDCA on used data sets

We fit a logistic regression model on the datasets with the hyper parameters detailed above.

On each dataset, we used 85% of the data for trainign and 15% of the data for testing.

The figures are in appendix C.

In our experiment, the SDCA did not perform that well.

We can see that after a consequent number of iterations, the accuracy of the estimator trained with SDCA stops to move, while the accuracy of the one trained with SGD can keep moving, and reaches better accuracy levels.

In terms of performance, we had the SGD beating the SDCA. On the other hand, the convergence of the SDCA is much clearer.

We can also note that the expected stopping time is way above what we get in practice. This is not that surprising as the theoretical stopping time that we computed were only sufficiant conditions. Thus, it is clear that the condition is met in our experiments.

Conclusion

In this report, we summarized most of what is needed to understand the SDCA : its goal, its theorical framework and its algorithm.

While our implementation of the SDCA for logistic regression seems to work, it did not yield better performance than SGD. On the other hand, the SGD can keep fluctuating when the SDCA really converges. Depending on the problem, it can be a real advantage.

Other tracks need to be investigated in order to improve the performance of the SDCA.

References

This report is based on two main studies.

Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization (S. Shalev-Shwartz and T. Zhang, 2013) from `http://www.jmlr.org/papers/volume14/shalev-shwartz13a/shalev-shwartz13a.pdf` was our main interest. This paper compiles many theoretical results on the SDCA and gives a clear algorithm.

Dual Coordinate Descent Methods for Logistic Regression and Maximum Entropy Models (H.-F. Yu, F.-L. Huang, C.-J. Lin, 2011) from `https://www.csie.ntu.edu.tw/~cjlin/papers/maxent_dual.pdf` gives interesting insight for the logistic regression case, with a modified Newton method for each iteration step instead of the approximation of the closed form, which helps against the numerical issues.

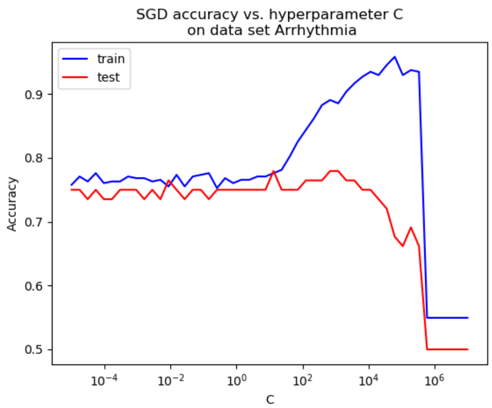
A Losses used

Squared loss: $$\phi_i(a) = (a - y_i)^2$$ $$\phi_i^*(-a) = -ay_i + a^2/4$$ $$\Delta\alpha_i = \frac{y_i - x_i^\top w^{(t-1)} - 0.5\alpha_i^{(t-1)}}{0.5 + \|x_i\|^2/(\lambda n))}$$
Absolute deviation loss: $$\phi_i(a) = \lvert a - y_i\rvert$$ $$\phi_i^*(-a) = -ay_i,\ a \in [-1, 1]$$ $$\Delta\alpha_i = \max\left(1, \min\left(1, \frac{y_i - x_i^\top w^{(t-1)}}{\|x_i\|^2/(\lambda n)} + \alpha_i^{(t-1)}\right)\right) - \alpha_i^{(t-1)}$$
Log loss: $$\phi_i(a) = \log(1 + \exp(-y_i a))$$ $$\phi_i^*(-a) = -ay_i \log(ay_i) + (1 - ay_i)\log(1 - ay_i)$$ $$\Delta\alpha_i = \frac{(1 + \exp(x_i^\top w^{(t-1)} y_i))^{-1} y_i - \alpha_i^{(t-1)}}{\max(1, 0.25 + \|x_i\|^2/(\lambda n))}$$
(γ-smoothed) Hinge loss: $$\phi_i(a) = \max\{0, 1 - y_i a\}$$ $$\phi_i^*(-a) = -ay_i + \gamma a^2/2,\ ay_i \in [0, 1]$$ $$\Delta\alpha_i = y_i \max\left(0, \min\left(1, \frac{1 - x_i^\top w^{(t-1)} y_i - \gamma\alpha_i^{(t-1)} y_i}{\|x_i\|^2/(\lambda n) + \gamma} + \alpha_i^{(t-1)} y_i\right)\right) - \alpha_i^{(t-1)}$$

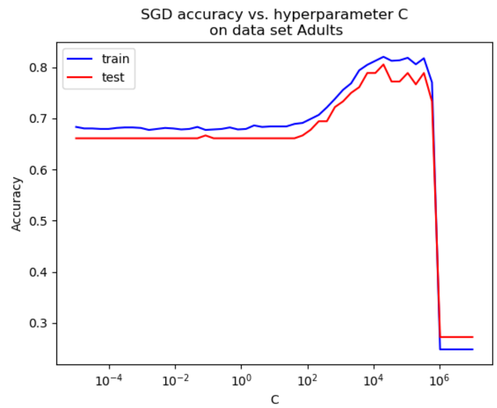
Table 3: Used loss functions, convex conjugates and closed form of solutions of problem (*).

$\Delta\alpha_i$ is the notation we use to represent the increment to add to α_i (one coordinate, at a given iteration) to maximize the objective function with respect to that coordinate.

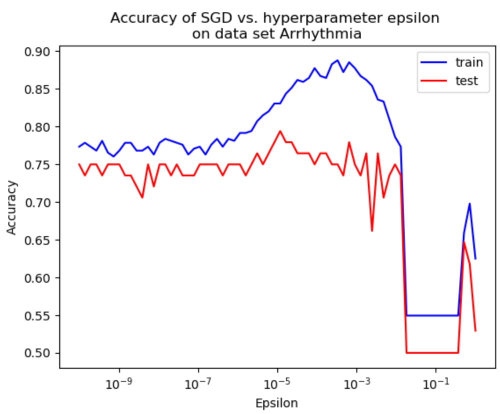
B Hyper parameters validation



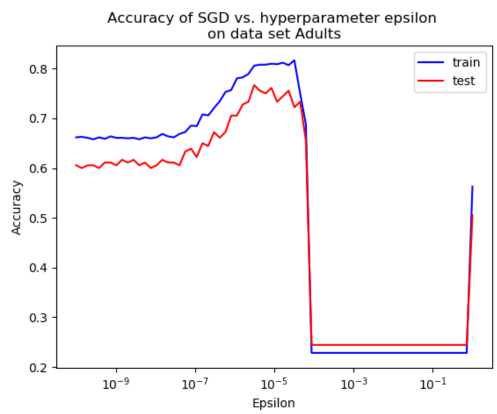
(a) `c` for the SGD for the data set Arrhythmia



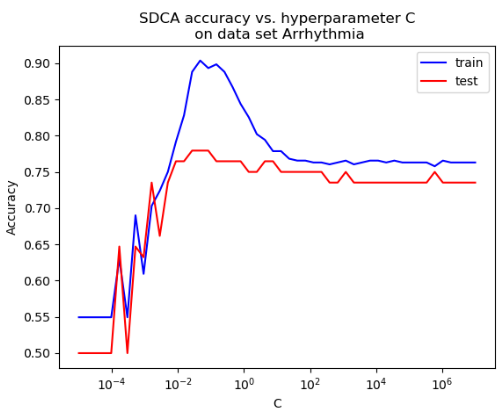
(b) `c` for the SGDfor the data set Adults



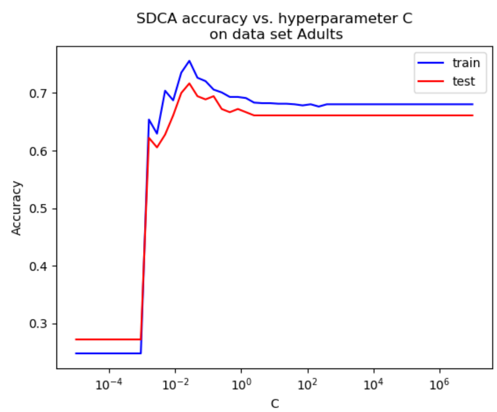
(c) `eps` for the SGD for the data set Arrhythmia



(d) `eps` for the SGD for the data set Adults



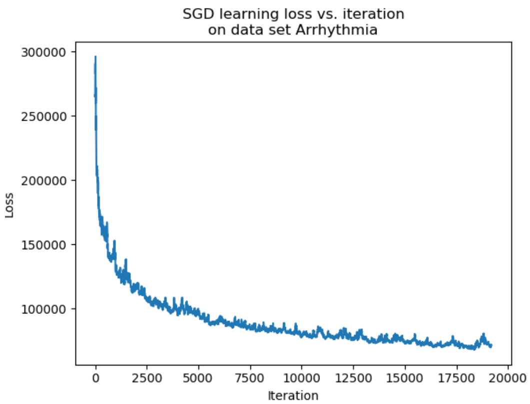
(e) `c` for the SDCA for the data set Arrhythmia



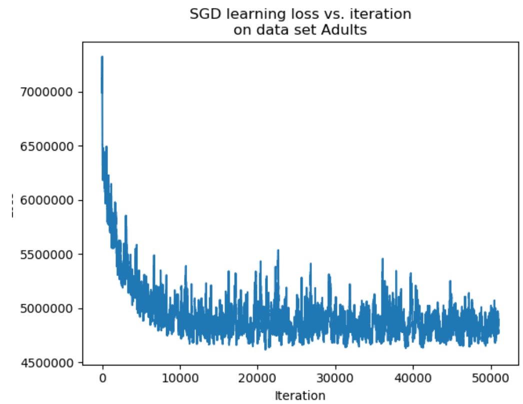
(f) `c` for the SDCA for the data set Adults

Figure 1: Selection of the hyperparameters

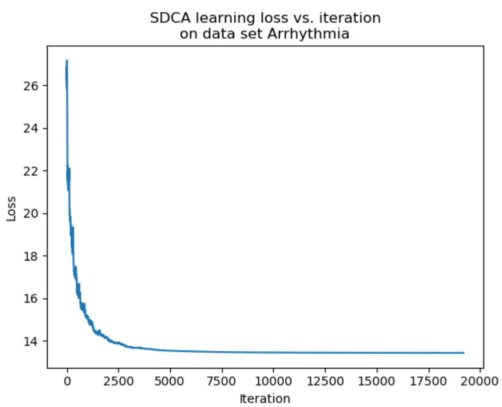
C Experiment results



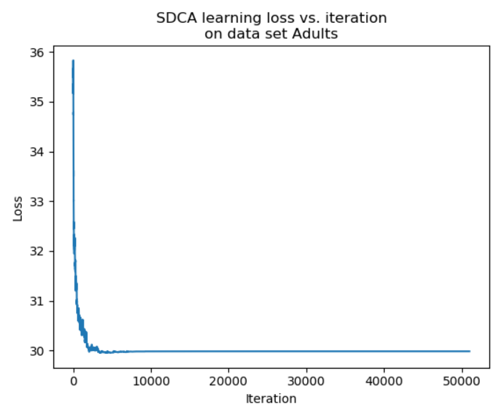
(a) Arrhythima data set with SGD



(b) Adults data set with SGD

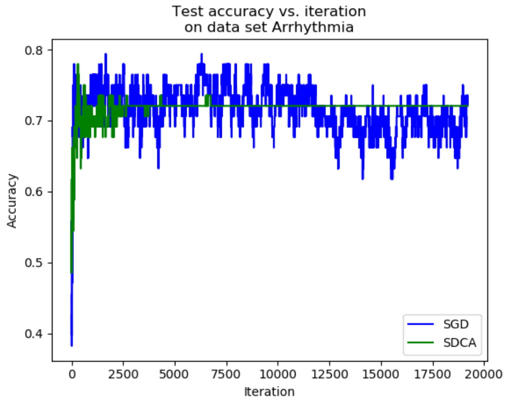


(c) Arrhythima data set with SDCA

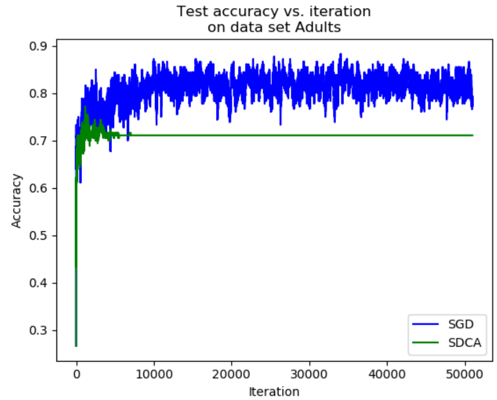


(d) Adults data set with SDCA

Figure 2: Evolution of the loss during the learning



(a) Arrhythima data set



(b) Adults data set

Figure 3: Evolution of the accuracy during the learning