

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 solve 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 $x_1, \dots, x_n \in \mathbb{R}^d$, ϕ_1, \dots, ϕ_n scalar convex functions, $\lambda > 0$ regularization parameter.

Let us focus on the following optimization problem:

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

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 $\phi_i^* : \mathbb{R} \rightarrow \mathbb{R}$ 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:

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

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

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

$$w(\alpha^*) = w^* \quad (3)$$

$$P(w^*) = D(\alpha^*) \quad (4)$$

$$\forall (w, \alpha), P(w) = D(\alpha) \quad (5)$$

We define the duality gap as $P(w(\alpha)) - P(w^*)$.

The SDCA algorithm is described further. T_0 can be chosen between 1 to T , and is generally chosen equal to $T/2$. However, in practice, these parameters are not required as the duality gap is used to terminate the algorithm.

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 :

$$\frac{\mathbb{P}(y = 1 | X = x)}{\mathbb{P}(y = -1 | X = x)} = w^T x, \quad w \in \mathbb{R}^p \quad (6)$$

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

$$\min_w C \sum_i \log(1 + e^{-y_i w^T x_i}) + \frac{1}{2} w^T w \quad (7)$$

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 :

$$\mathcal{L}(w, z, \alpha) = \sum_i (C \log(1 + z_i) + \alpha_i z_i) - \sum_i \alpha_i e^{-y_i w^T x_i} + \frac{1}{2} w^T w \quad (8)$$

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

$$\min_{w, z} \mathcal{L}(w, z, \alpha) = \mathcal{L}(w^*, z^*, \alpha) = \psi(\alpha) \quad (9)$$

In fact, it leads to the following dual problem :

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i \in I} (-\alpha_i \log(\alpha_i) - (C - \alpha_i) \log(C - \alpha_i)) - \frac{1}{2} \alpha^T Q \alpha \\ \text{s.t.} \quad & I = \{i, 0 < \alpha_i < C\} \\ & 0 \leq \alpha_i \leq C \end{aligned} \quad (10)$$

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  $t = 1, \dots, T$  do
    Randomly pick  $i$ 
     $\Delta \alpha_i \leftarrow \arg \max_{\alpha} -\phi_i^*(-(\alpha_i^{(t-1)} + \Delta \alpha_i)) - \frac{\lambda n}{2} \|w^{(t-1)} + (\lambda n)^{-1} \Delta \alpha_i x_i\|^2$  (*)
     $\alpha^{(t)} \leftarrow \alpha^{(t-1)} + \Delta \alpha_i e_i$ 
     $w^{(t)} \leftarrow w^{(t-1)} + (\lambda n)^{-1} \Delta \alpha_i x_i$ 
  if Averaging option then
    return  $\bar{w} = \frac{1}{T-T_0} \sum_{i=T_0+1}^T w^{(i-1)}$ 
  if Random option then
    return  $\bar{w} = w^{(t)}$  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 $\phi_i : \mathbb{R} \rightarrow \mathbb{R}$ is L -Lipschitz if $\forall a, b \in \mathbb{R}, |\phi_i(a) - \phi_i(b)| \leq L |a - b|$, and that a function $\phi_i : \mathbb{R} \rightarrow \mathbb{R}$ 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 $\mathbb{E}[P(\bar{w}) - D(\bar{\alpha})] \leq \epsilon_P$, 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 datasets

We used our implementation on :

- Two gaussians to check that our implementation works
- *Arrhythmia* : <https://archive.ics.uci.edu/ml/datasets/Arrhythmia>
- *Adults* : <https://archive.ics.uci.edu/ml/datasets/adult>

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

The Arrhythmia dataset will help us check the properties of SDCA when there are many features. The Adult dataset 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.

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 ϵ 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.

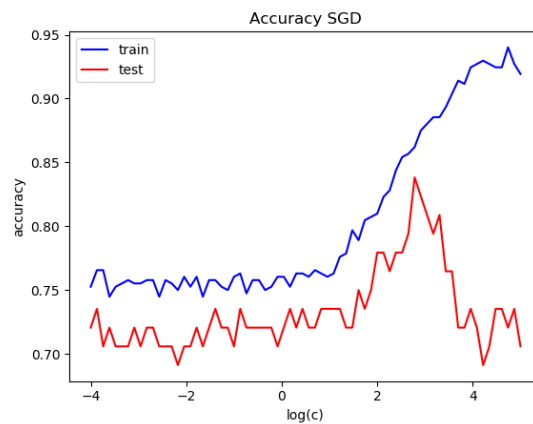


Figure 1: Selection of the hyperparameter c for the SGD for the dataset Arrhythmia

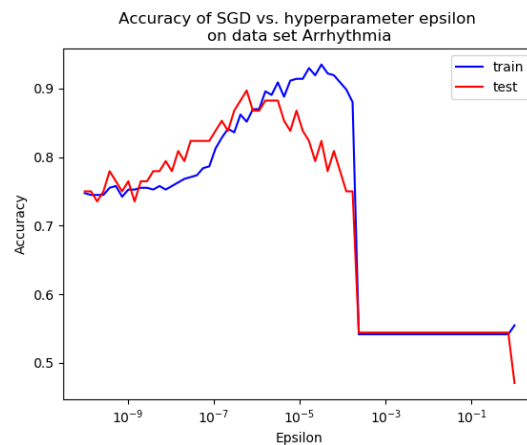


Figure 2: Selection of the hyperparameter ϵ for the SGD for the dataset Arrhythmia

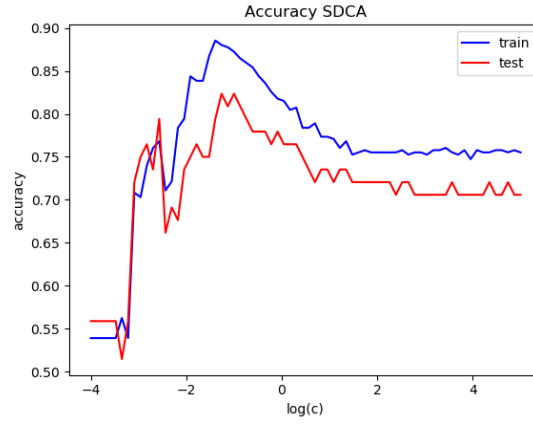


Figure 3: Selection of the hyperparameter c for the SDCA for the dataset Arrhythmia

We selected the hyper parameters values :

Data set	SGD c	SGD ϵ	SDCA c
Arrhythmia	1	1	1
Adults	1	1	1

Table 1: Hyper parameter values used

2.6 Comparison between SGD and SDCA on used datasets

Lorem ipsum.

Conclusion

References

A Losses used

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

Table 2: 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.