

SDCA [1] vs. Pegasos [2] for linear SVM fitting

Applications to blood cells classification

Dimitri Bouche, Cyril Verluise

May 8, 2018

1 Introduction

The aim of this project is to compare two optimization algorithms on the particular problem of fitting a linear SVM :

- Stochastic dual coordinate ascent (SDCA) [1] which is a stochastic version of a Dual coordinate ascent (DCA).
- Primal estimated subgradient solver for SVM (Pegasos) [2] which corresponds to a classic stochastic (sub)gradient descent (SGD) with a given choice of step-size.

As a reminder, the linear (regularized) SVM problem is the following :

$$\min_{w \in \mathbb{R}^d} P(w). \quad (1)$$

Where :

$$P(w) = \left[\frac{1}{n} \sum_{i=1}^n \phi_i(w^T x_i) + \frac{\lambda}{2} \|w\|^2 \right].$$

d being the number of features, n the number of data points and ϕ_i the hinge loss :

$$\phi_i(w^T x_i) = \max(0, 1 - y_i w^T x_i).$$

With $y_i \in \{-1, 1\}$ being x_i 's label.

Regarding theoretical guarantees, the Pegasos algorithm is stated to be faster by [2] yielding an ϵ suboptimal result in $\mathcal{O}(\frac{1}{\lambda \epsilon})$ (independent from the size of dataset) whereas the SDCA algorithm is said to yield such result in $\mathcal{O}(n + \frac{1}{\lambda \epsilon})$ [1], although it is argued that the latter can reach more precise results.

We will start by a short presentation of the two procedures and will then apply them both to the same problem of image classification (bloodcells classification) in order to see to what extent those theoretical guarantees apply in practise.

2 SDCA

We are here bound to paraphrase [2], so we will only state the updates formula that are of interest to us and perform the computations only when the closed form formula are not given (for instance for the SGD initialization).

2.1 SDCA-perm

We focus here on the dual of problem (1) :

$$\max_{\alpha \in \mathbb{R}^n} D(\alpha). \quad (2)$$

Where :

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

With ϕ_i^* defined as :

$$\begin{aligned} \phi_i^*(-a) &= -ay_i \text{ if } ay_i \in [0, 1] \\ \phi_i^*(-a) &= +\infty \text{ if } ay_i \notin [0, 1] \end{aligned}$$

Solving problem (2) is equivalent to solving problem (1), since any solution to (2) can be transformed into a solution to (1) using the following function [2] :

$$w(\alpha) = \frac{1}{\lambda n} \sum_{i=1}^n \alpha_i x_i$$

We implemented the SDCA-perm version, which runs in epochs instead of employing complete randomization. We also add a stopping criterion on the duality gap $P(w(\alpha)) - D(\alpha)$ as advised by the authors.

However, we do not apply the "Random option" (returning a randomly chosen value of α among the second half iterations) nor the "Average option" (returning the average value of α over the second half iterations) since it works very well in practice without. The pseudo code for our implementation is the following :

```

Data:  $\alpha^{(0)}, k_{max}, \epsilon$ 
Set  $w^{(0)} = w(\alpha^{(0)})$ ;
Set  $g = P(w^{(0)}) - D(\alpha^{(0)})$ ;
while  $g > \epsilon$  and  $k < k_{max}$  do
    Draw  $\{i_1, \dots, i_n\}$  random permutation of  $\{1, \dots, n\}$ ;
    for  $j = 1$  to  $n$  do
         $i = i_j$ ;
         $t \leftarrow t + 1$ ;
         $\Delta_i = \Delta_i(\alpha_i^{(t-1)}, w^{(t-1)})$ ;
         $\alpha^{(t)} \leftarrow \alpha^{(t-1)} + \Delta_i e_i$ ;
         $w^{(t)} \leftarrow w^{(t-1)} + \frac{1}{\lambda n} \Delta_i x_i$ ;
    end
     $k \leftarrow k + 1$ ;
end

```

Algorithm 1: SDCA Perm

With e_i the vector with 1 in the i -th position and 0s elsewhere, and Δ_i the coordinate update chosen to decrease the dual objective as given in [2]:

$$\Delta_i(\alpha_i^{(t-1)}, w^{(t-1)}) = y_i \max \left(0, \min \left(1, \frac{(\lambda n)(1 - x_i^T w^{(t-1)}) y_i}{\|x_i\|^2} + \alpha_i^{(t-1)} y_i \right) \right) - \alpha_i^{(t-1)}.$$

2.2 SGD initialization

References

- [1] Shai Shalev-Schwartz and Tong Zhang. "Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization". In: *Journal of machine learning research* (2013).
- [2] Shai Shalev-Schwartz et al. "Pegasos : Primal Estimated Gradient Solver for SVM." In: *Springer - Journal of mathematical programming* (2011).