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 :

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

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 theorethical guarantees, the Pegasos algorithm is stated to be faster by [2] yielding an ϵ suboptimal result in $\mathcal{O}(\frac{1}{\lambda\epsilon})$ (independant 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) :

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

Where :

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

With $\phi_i^\star$ defined as :

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

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 $\{i_1, ..., i_n\}$ random permutation of $\{1, ..., n\}$;
    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

It is advised in [2] to implement a different algorithm based on a modification of stochastic subgradient descent (SGD) just for the first epoch and then to switch back to SDCA. This is useful because SDCA tends to result in updates that are too small in the first epoch in comparison to SGD.

The pseudo-code for this first modified epoch is given by :

```
Set $w^{(0)} = 0$;
for $t = 1$ to $n$ do
    Find $\alpha_t$ to maximize $-\phi_t^\star(-\alpha_t) - \frac{\lambda t}{2}||w^{(t-1)} + (\lambda t)^{-1}\alpha_t x_t||^2$;
    $w^{(t)} = \frac{1}{\lambda t}\sum_{i=1}^{t} \alpha_i x_i$;
end
```

Algorithm 2: SDCA Perm

Since the closed form for the step of maximization is not given in [2], we proceed to the computations :

At each step $t \in \{1, ..., n\}$ of the epoch, we wish to find α_t that maximizes :

$$-\phi_t^\star(-\alpha_t) - \frac{\lambda t}{2}||w^{(t-1)} + (\lambda t)^{-1}\alpha_t x_t||^2$$

Let us remark first of all that we must take α_t to be such that $\alpha_t y_t \in [0, 1]$ since if this is not the case, $-\phi_t^\star(-\alpha_t) = -\infty$.

Now supposing that $\alpha_t y_t \in [0, 1]$, developping the previous expression yields :

$$\alpha_t y_t - \frac{\lambda t}{2}(||w^{(t-1)}||^2 + 2\frac{\alpha_t}{\lambda t}\langle w^{(t-1)}, x_t\rangle + \frac{\alpha_t^2}{\lambda^2 t^2}||x_t||^2)$$

This is a second order polynomial in α_t. With a negative coefficient on the second order term. Thus this is concave. Setting the derivative to 0 w.r.t α_t yields :

$$y_t - \langle w^{(t-1)}, x_t\rangle - \frac{\alpha_t}{\lambda t}||x_t||^2 = 0$$

This gives us an optimal α_t : $\alpha_t^\star$ defined by :

$$\alpha_t^\star = \frac{\lambda t}{||x_t||^2}(y_t - x_t^T w^{(t-1)})$$

Let us now ensure that we threshold this value to make sure that $\alpha_t y_t \in [0, 1]$: The optimal value is thus $\tilde{\alpha_t}$:

$$\begin{aligned}
\tilde{\alpha_t} &= \alpha_t^\star \; if \; \alpha_t^\star y_t \in [0, 1] \\
\tilde{\alpha_t} &= \min(0, y_t) \; if \; \alpha_t^\star < \min(0, y_t) \\
\tilde{\alpha_t} &= \max(0, y_t) \; if \; \alpha_t^\star > \max(0, y_t)
\end{aligned}$$

References

[1] Shai Shalev-Swhartz and Tong Zhang. “Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization”. In: *Journal of machine learning research* (2013).

[2] Shai Shalev-Swhartz et al. “Pegasos : Primal Estimated Gradient Solver for SVM.” In: *Springer - Journal of mathematical programming* (2011).