Regularized Stochastic Learning with Dual Averaging Methods

An Application to Linear SVM

Maxime Jumelle

Context

During this note, we are interested in evaluating efficiency of the Dual Averaging Methods in case of Stochastic Learning with regularization.

- First, we present the algorithm in its general form and associated properties and discuss intents of studied research article.
- Secondly, we propose a special case with mixed l_1 and l_2-regularization.
- Finally, we apply this special case on MNIST dataset with l_2-regularized hinge loss and compare sparsity patterns, as well as regret bounds and convergence results with others instances of online convex optimization algorithms.

Linear SVM

During this presentation, we will only focus on linear SVM with hinge loss and l_2 regularization, which loss function can be written as :

$$\frac{\lambda}{n}\sum_{i=1}^{n}(1 - b_i w^\top a_i)_+ + \frac{1}{2}\|w\|_2^2$$

with a fixed $\lambda > 0$.

Regularized Dual Algorithm

Regularization

Let f be a convex loss function and Ψ a closed convex function, referred as "regularization function". Then regularized stochastic learning aim at solving the following problem.

$$\min_w L(w; Z) = \min_w \{\mathbb{E}_Z[f(w; Z)] + \Psi(w)\}$$

Popular examples or regularization functions are :

- l_1-regularization with $\Psi(w) = \rho\|w\|_1$ with $\rho \geq 0$.
- l_2-regularization with $\Psi(w) = \rho\|w\|_2^2$.
- Mixed regularization with $\Psi(w) = \frac{1}{2}\|w\|_2^2 + \rho\|w\|_1$.

Dual Averaging

The key idea behind this algorithm is to minimize three terms at each iteration :

$$w_{t+1} = \text{argmin} \left\{ \langle \bar{g}_t, w \rangle + \Psi(w) + \frac{\beta_t}{t} h(w) \right\}$$

with h an auxiliary strongly convex function and $(\beta_t)_{t \geq 1}$ a nonnegative and nondecreasing sequence. This auxiliary function is supposed to ensure a more aggressive truncation thresholds than in the usual setting of only a single l_1 or l_2 regularization term, then significantly improving sparsity selection.

Regularized Dual Algorithm (RDA)

Algorithm 1: Regularized Dual Algorithm (RDA) method

Input: h auxiliary function and $(\beta_t)_{t\geq 1}$ nonnegative and nondecreasing sequence.

Set $w_1 = \text{argmin}_w h(w)$ and $g_0 = 0$. **for** $t = 1, \ldots, T$ **do**

Compute subgradient $\partial f_t(x_t)$ for a given f_t

Updates dual subgradient

$$\bar{g}_t = \frac{t-1}{t}\bar{g}_{t-1} + \frac{1}{t}g_t$$

Compute next weight

$$w_{t+1} = \text{argmin}\left\{\langle \bar{g}_t, w\rangle + \Psi(w) + \frac{\beta_t}{t}h(w)\right\}$$

end

return w_{T+1}

Special case : l_1-regularization

Soft l_1-regularization

This previous algorithm provides general settings for regularized dual averaging method with any regularization and auxiliary functions. In particular, let us choose

$$\Psi(w) = \sigma\|w\|_1 \qquad h(w) = \frac{1}{2}\|w\|_2^2 + \rho\|w\|_1$$

Recall minimization of loss function from the beginning, we can define a specific case of RAD method with such regularization and auxiliary functions as

$$f(w; a, b) = \frac{\lambda}{n}\sum_{i=1}^{n}(1 - b_i w^\top a_i)_+ + \frac{1}{2}\|w\|_2^2$$

then we can express RDA update as

$$w_{t+1} = \text{argmin}_w \left\{ \frac{\lambda}{n}\sum_{i=1}^{n}(1 - b_i w^\top a_i)_+ + \sigma\|w\|_1 + \frac{\beta_t}{t}\left(\frac{1}{2}\|w\|_2^2 + \rho\|w\|_1\right)\right\}$$

Soft l_1-regularization

This *soft* l_1-regularization enables us to compute a closed-form solution for every w_t :

$$w_{t+1}^{(i)} = \begin{cases} 0 & \text{if } |\bar{g}_t^{(i)}| \leq \lambda \\ -\frac{\sqrt{t}}{\gamma}\left(\bar{g}_t^{(i)} - \sigma \ \text{sgn}(\bar{g}_t^{(i)})\right) & \text{otherwise} \end{cases}$$

Enhanced l_1-RDA

Algorithm 2: Enhanced l_1-RDA method

Input: $\gamma > 0$ and $\rho \geq 0$.

Set $w_1 = 0$ and $g_0 = 0$. **for** $t = 1, \ldots, T$ **do**

Compute subgradient $\partial f_t(x_t)$ for a given f_t

Updates dual subgradient

$$\bar{g}_t = \frac{t-1}{t}\bar{g}_{t-1} + \frac{1}{t}g_t$$

Let $\lambda_t^{\text{RDA}} = \sigma + \gamma\rho\sqrt{t}$ and compute next weight

$$w_{t+1}^{(i)} = \begin{cases} 0 & \text{if } |\bar{g}_t^{(i)}| \leq \lambda \\ -\dfrac{\sqrt{t}}{\gamma}\left(\bar{g}_t^{(i)} - \sigma\ \text{sgn}(\bar{g}_t^{(i)})\right) & \text{otherwise} \end{cases}$$

end

return w_{T+1}

l_1-regularization with linear SVM

In our case, we considered a slightly different setting, as we are focusing on linear SVM so that $\frac{1}{2}\|w\|_2^2$ is already set. However, we can still play on this term as we define

$$\Psi(w) = \rho\|w\|_1 + \frac{\sigma}{2}\|w\|_2^2$$

to enable sparse model. In this case, it can be shown that

$$w_{t+1}^{(i)} = \begin{cases} 0 & \text{if } |\bar{g}_t^{(i)}| \leq \rho \\ -\frac{1}{\sigma}\left(\bar{g}_t^{(i)} - \rho\ \text{sgn}(\bar{g}_t^{(i)})\right) & \text{otherwise} \end{cases}$$

Setting $\sigma = 1$ and we are back in the linear SVM setup.

Mixed l_1/l_2-RDA

Algorithm 3: Mixed l_1/l_2-RDA

Input: $\rho \geq 0$.

Set $w_1 = 0$ and $g_0 = 0$. **for** $t = 1, \ldots, T$ **do**

Compute subgradient $\partial f_t(x_t)$ for a given f_t

Updates dual subgradient

$$\bar{g}_t = \frac{t-1}{t}\bar{g}_{t-1} + \frac{1}{t}g_t$$

Compute next weight

$$w_{t+1}^{(i)} = \begin{cases} 0 & \text{if } |\bar{g}_t^{(i)}| \leq \rho \\ -\frac{1}{\sigma}\left(\bar{g}_t^{(i)} - \rho \operatorname{sgn}(\bar{g}_t^{(i)})\right) & \text{otherwise} \end{cases}$$

end

return w_{T+1}

Experiments

Settings

We use MNIST dataset (LeCun et al. 1999) only on 0 et 1 digits. Our goal is to build a classifier in a 784-dimensional space able to correctly identify whether an input digit is likely to be a 0 (positive class) or a 1 (negative class).

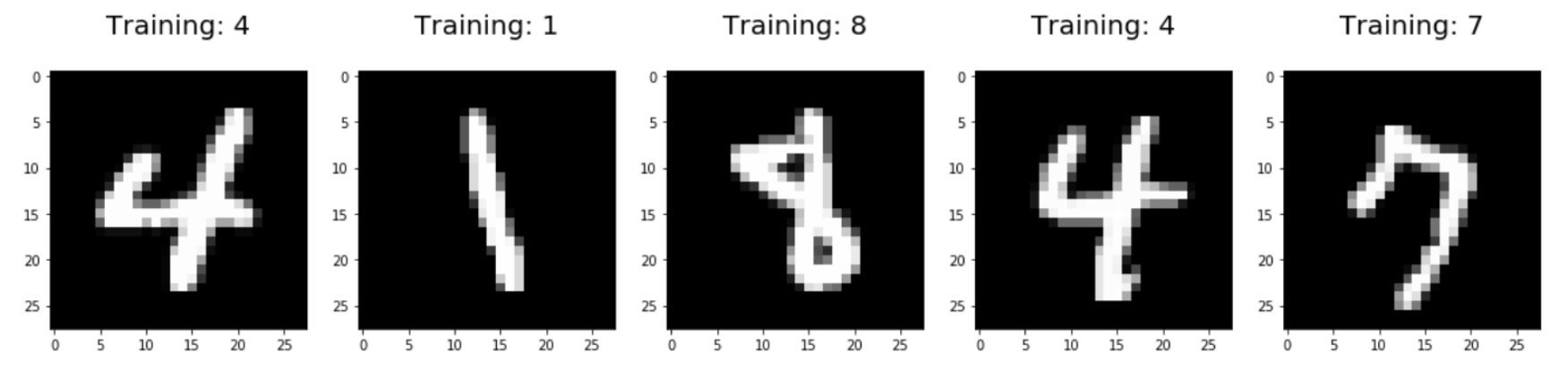


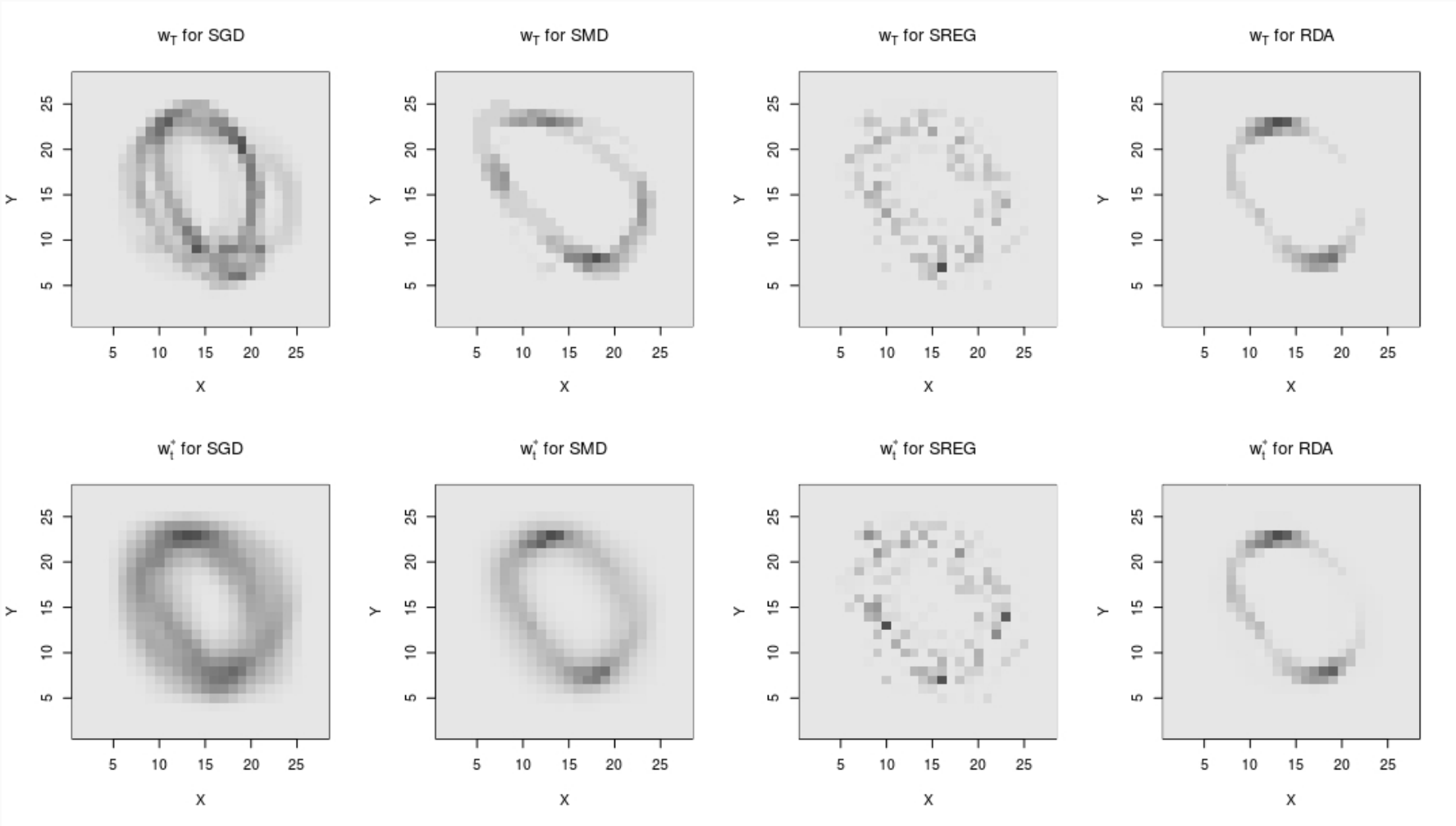
Figure 1: Example of a few digits in the dataset.

Settings

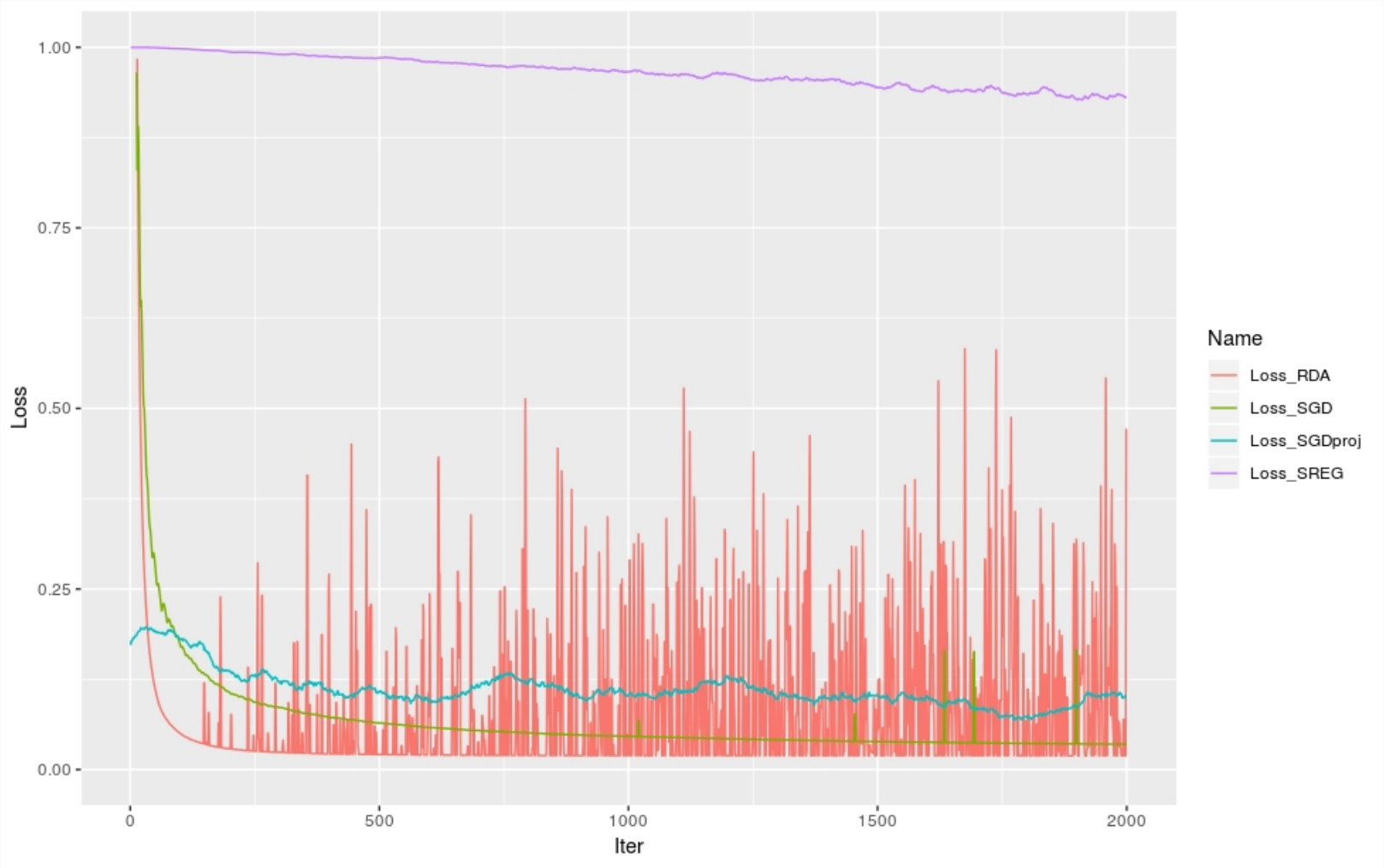
In addition with RDA algorithm, we will also train our model with other popular ones :

- Stochastic Gradient Descent
- Stochastic Mirror Descent
- Stochastic Randomized Exponentiated Gradients

Sparsity patterns for $T = 2000, \lambda = 1$ and $\rho = 0.25$



Convergence



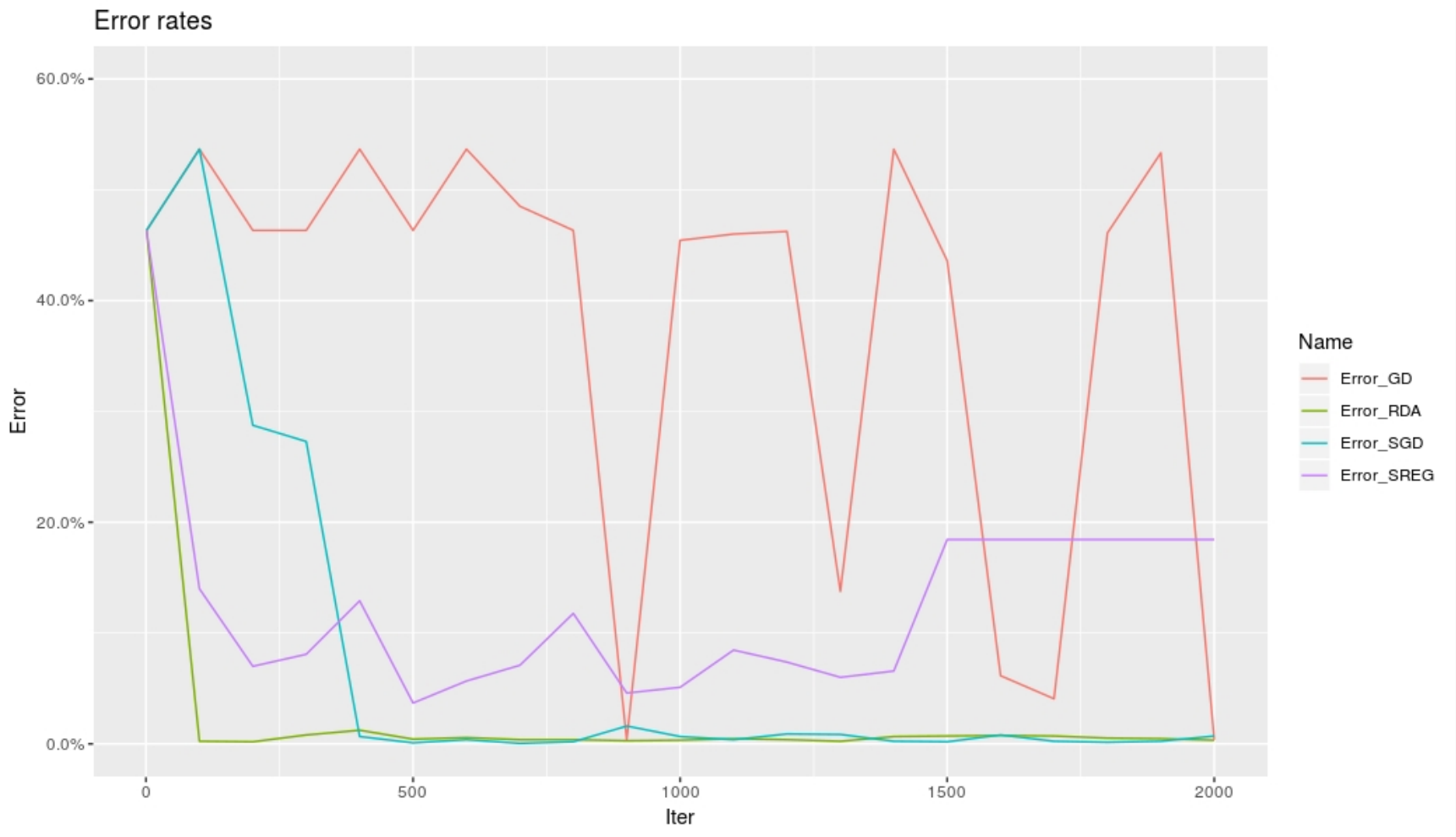
Error rates

Since we have trained our classifier on a train set, we would like to see any over-fitting process (although it is more unlikely for a linear model rather than a non-parametric one). We then use a test set $(\tilde{\mathbf{a}}, \tilde{\mathbf{b}}) = (\tilde{a}_i, \tilde{b}_i)_{1 \leq i \leq m}$ with no observation from train set and compute, at different iterations t for each algorithm with prediction function $\hat{g}$, the error rate :

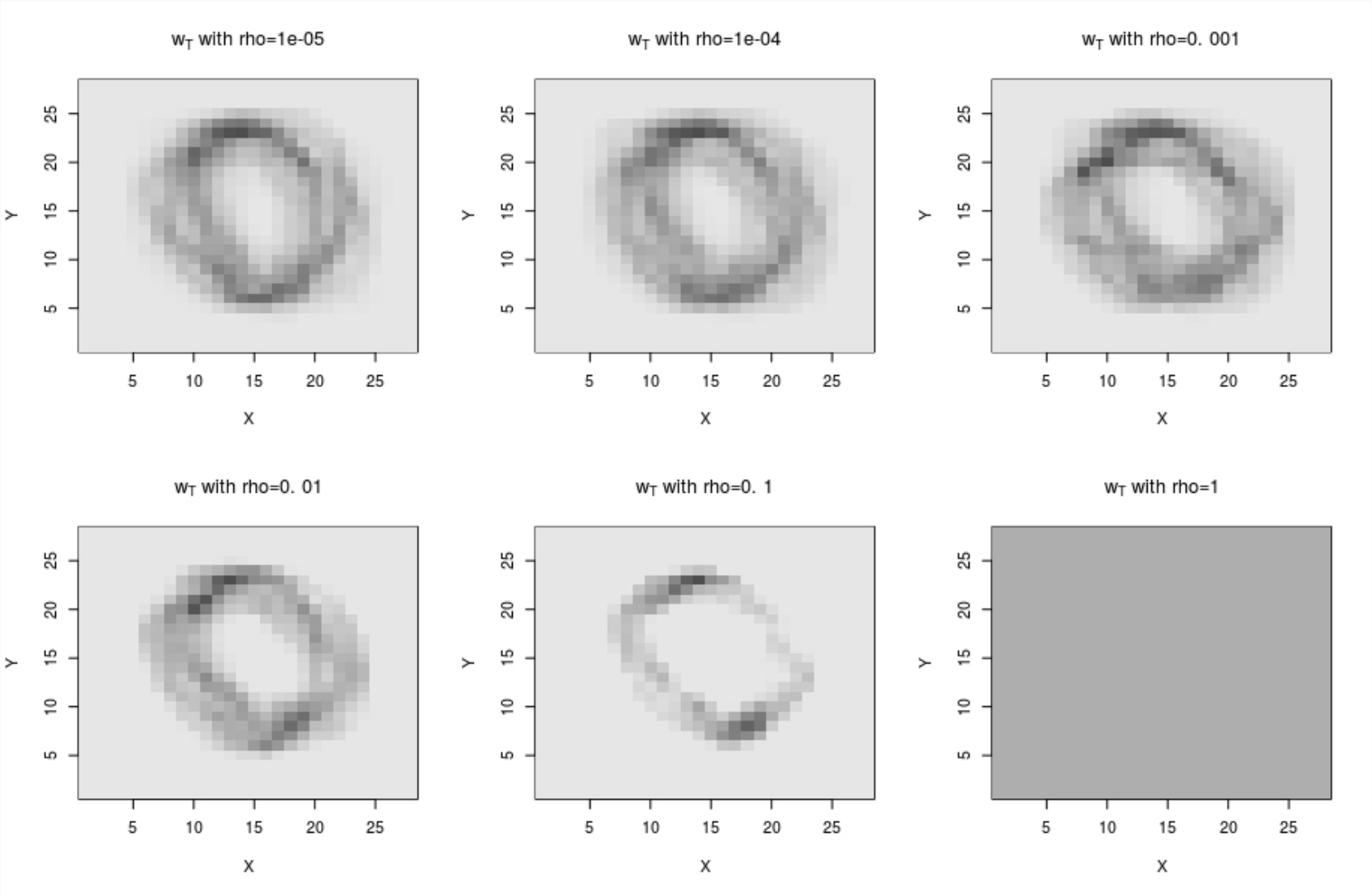
$$\frac{1}{m} \sum_{i=1}^{m} \mathbf{1}_{\{\hat{g}(\tilde{a}_i) = \tilde{b}_i\}}$$

Where $g(x) = \mathbf{1}_{\{\hat{w}^\top x > 0\}}$, with $\hat{w}$ learned parameters (or weights).

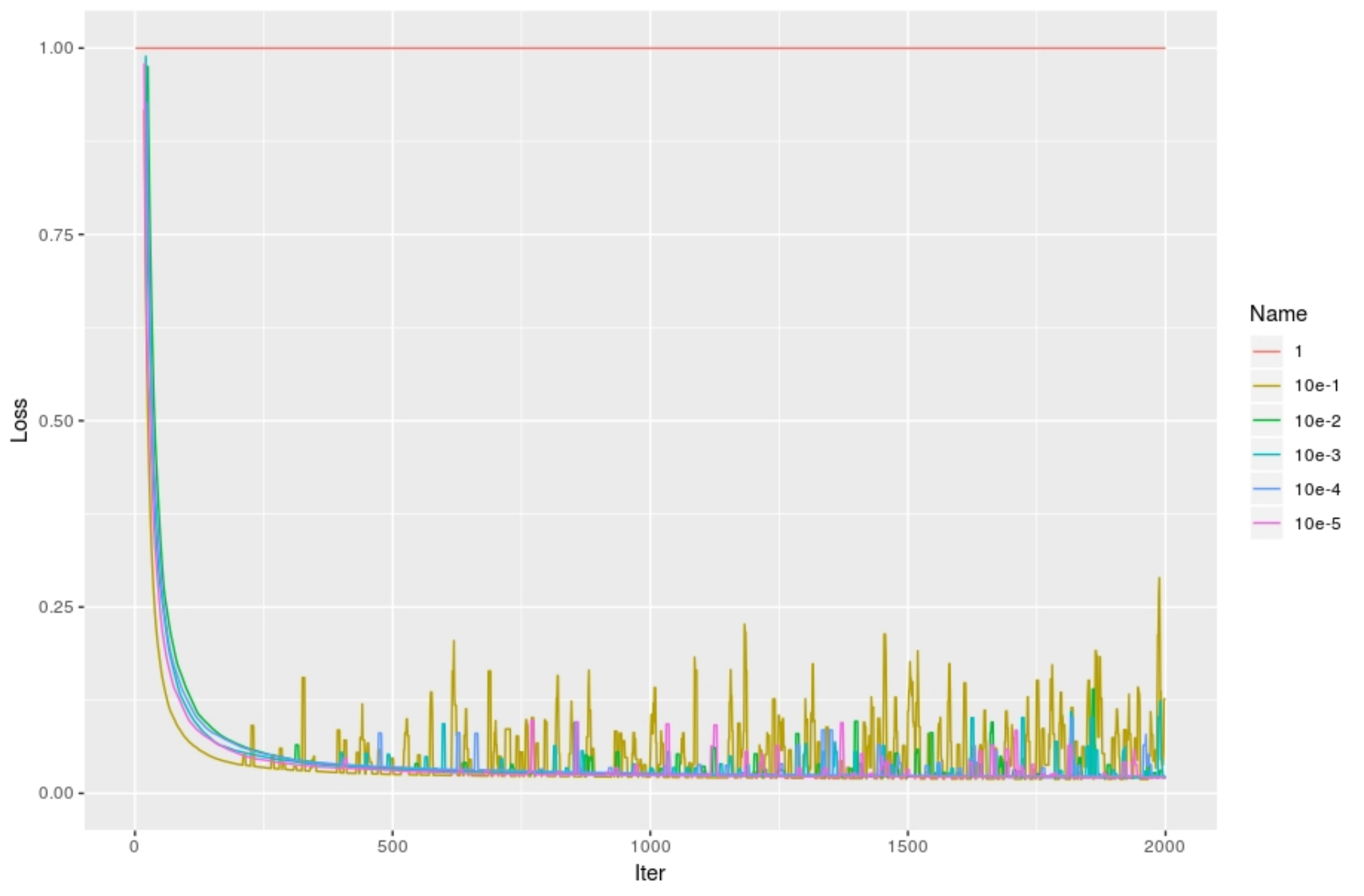
Error rates


Error rates
60.0%
40.0%
20.0%
0.0%
Error
0
500
1000
1500
2000
Iter
Name
Error_GD
Error_RDA
Error_SGD
Error_SREG

RDA with $\rho \in \{10^{-6}, \ldots, 1\}$



Convergence of RDA with $\rho \in \{10^{-6}, \ldots, 1\}$



Going further and improvements

There are still many possibilities since Nesterov's establishment of primal averaging (2009). In this situation, we could :

- Extends current experiment to all 10 digits, as well as studying other models rather than linear SVM.
- Fine-tuning of hyper-parameters with cross-validation techniques.
- Using other regularization functions and potentially discover closed-form solutions.