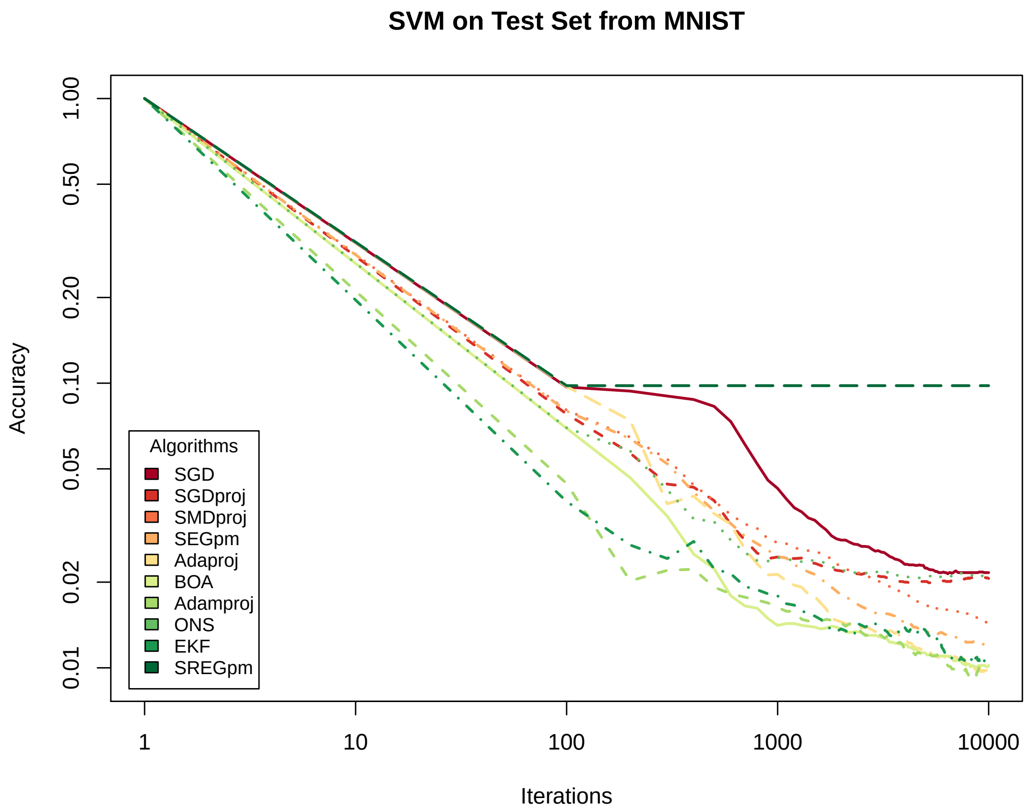
Online Convex Optimization

Olivier Wintenberger



2020–2021

Contents

Part I

Preliminaries

These lectures notes are reproducing most of the 6 first Chapters of Hazan's book

Introduction to Online Convex Optimization

`https://sites.google.com/view/intro-oco/`

Online Convex Optimization is the study of recursive algorithm and their theoretical guarantees called regret bounds. Due to the effectiveness of some algorithms of this vast class for training deep neural networks there is an excellent recent literature. Besides Hazan (2019), there is also the early Shalev-Shwartz et al. (2011) and the very recent Orabona (2019).

All the illustrations of these notes are maid on the MNIST handwritten digit dataset from

`http://yann.lecun.com/exdb/mnist/`

tuned into a classification problem recognizing the digit 0. The performances of linear SVM, trained on 60000 digits with different algorithms, are compared in terms of their accuracy on the test set of 10000 digits. The seed is fixed the same for all stochastic algorithms and all the experiments are ran on the R language from CRAN.

Chapter 1

Basic concepts in Convex Optimization (CO)

In this chapter we fix some notation form the usual CO problem, cf *Convex Optimization*, Boyd et al. (2004), the more recent introductory notes Bubeck (2014) and *Remise à niveau. Calcul différentiel et optimisation*, the lectures of the course of Claire Boyer and Maxime Sangnier.

1.1 Basic definition and setup

We are interested in algorithms for solving the CO problem. The key notion is convexity, convexity of a set $\mathcal{K}$

$$\alpha x + (1-\alpha)y \in \mathcal{K}, \qquad x, y \in \mathcal{K}, \quad \alpha \in (0,1)\,,$$

and convexity of a function $f : \mathcal{K} \mapsto \mathbb{R}$

$$f(\alpha x + (1-\alpha)y) \le \alpha f(x) + (1-\alpha)f(y)\,.$$

Examples of convex functions and convex sets.

In all the sequel $\mathcal{K}$ will be assumed closed and bounded with diameter D

$$\|x - y\| \le D\,, \qquad x, y \in \mathcal{K}\,.$$

Here the norm $\|\cdot\| = \|\cdot\|_2$ is the Euclidean one over $\mathbb{R}^d$, $d \ge 1$. On a closed and bounded domain a convex function adits a minimum.

Definition 1 (CO problem)**.** *A CO problem $(f, \mathcal{K})$ is to approximate the minimum of f over $\mathcal{K}$*

$$\min_{x\in\mathcal{K}} f(x)\,,$$

or, alternatively, to approximate one of the minimizers

$$x^* \in \arg\min_{x\in\mathcal{K}} f(x) = \{x \in \mathcal{K};\ f(x) = \min_{x\in\mathcal{K}} f(x)\}\,.$$

Another way of defining a CO problem is via its (sub-)gradient. A sub-gradient is a vector $\nabla f(x) \in \mathbb{R}^d$ satisfying the relation

$$f(y) \ge f(x) + \nabla f(x)^T (y - x)\,. \tag{1.1}$$

We assume that the sub-gradient is unique $\nabla f(x)$ at any point and $x \in \mathcal{K}$ and we call it the gradient. If the minimizer is in the interior of $\mathcal{K}$ then

$$x^* \in \arg\min_{x\in\mathcal{K}} f(x) \cap \overset{\circ}{\mathcal{K}} \Longleftrightarrow \nabla f(x^*) = 0\,.$$

In generality they might be problem on the boundary of $\mathcal{K}$.

Theorem (simple KKT)**.** *For any $y \in \mathcal{K}$ we have*

$$\nabla f(x^*)^T (y - x^*) \geq 0\,.$$

We denote $\Pi_{\mathcal{K}}(y) = \arg\min_{x\in\mathcal{K}} \|y - x\|$ the (convex) projection of y on $\mathcal{K}$. It is a CO problem that may be itself complicated Grünewälder (2017)! It has an explicit solution $x/\|x\|$ if $\mathcal{K} = B_2(1)$ the unitary euclidian ball and $x \notin \mathcal{K}$.

Theorem (Pythagorean)**.** *For any $z \in \mathcal{K}$ we have $\|y - z\| \geq \|\Pi_{\mathcal{K}}(y) - z\|$.*

Proof. We have he CO problem $\Pi_{\mathcal{K}}(y) = \arg\min_{x\in\mathcal{K}} f(x)$ with $f(x) = \|x - y\|^2$ thus

$$\begin{aligned}
\|y - z\|^2 - \|\Pi_{\mathcal{K}}(y) - z\|^2 &= \|y\|^2 - \|\Pi_{\mathcal{K}}(y)\|^2 + 2(\Pi_{\mathcal{K}}(y) - y)^T z \\
&= \|y\|^2 - \|\Pi_{\mathcal{K}}(y)\|^2 + \nabla f(\Pi_{\mathcal{K}}(y))z \\
&\geq \|y\|^2 - \|\Pi_{\mathcal{K}}(y)\|^2 + \nabla f(\Pi_{\mathcal{K}}(y))\Pi_{\mathcal{K}}(y) \\
&\geq |y\|^2 - \|\Pi_{\mathcal{K}}(y)\|^2 + 2(\Pi_{\mathcal{K}}(y) - y)^T \Pi_{\mathcal{K}}(y) \\
&\geq |y\|^2 + \|\Pi_{\mathcal{K}}(y)\|^2 - 2y^T \Pi_{\mathcal{K}}(y) \\
&\geq |y - \Pi_{\mathcal{K}}(y)\|^2 \geq 0\,.
\end{aligned}$$

We used simple KKT to derive the inequality. □

Note that the above theorem extends easily to any weighted norms

$$\|x\|_W^2 = x^T W x\,, \qquad W \succ 0\,.$$

We assume the sub-gradients are bounded, there exists some G so that $\nabla f(x) \leq G$ over $\mathcal{K}$. Then f is Lipschitz (and continuous) $|f(y) - f(x)| \leq G\|y - x\|$, $x, y \in \mathcal{K}$.

We may assume more regularity on the function f, namely

- f is α-strongly convex if

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{\alpha}{2}\|y - x\|^2\,, \qquad x, y \in \mathcal{K}\,,$$

- f is β-smooth if

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{\beta}{2}\|y - x\|^2$$

Note that β-smooth follows from f is convex and ∇f is β-Lipschitz.

Remark. *If the function f is twice differentiable we denote $\nabla^2 f(x)$ the Hessian $d \times d$ matrix at the point $x \in \mathcal{K}$ and*

- *f is α strongly convex iff $\nabla^2 f(x) \succeq \alpha I_d$ ($A \succeq 0$ meaning that A is a symmetric semi-definite positive matrix),*
- *f is β smooth iff $\nabla^2 f(x) \preceq \beta I_d$.*

When f is α-strongly convex and β-smooth f is γ-well-conditioned with $\gamma = \alpha/\beta \leq 1$. Example $f(x) = \|x\|^2$ is $\gamma = 1$ well-conditioned as $\alpha = \beta = 2$.

1.2 Gradient Descent algorithm (GD)

In view of (1.1), minimizing f from a given point $x \in \mathcal{K}$ is approximated by the CO problem on the *surrogate loss*, ie a simple approximation of the function $f(y) \approx f(x)+\nabla f(x)^T(y-x)$ in y. It is a linear function and one takes the step y from x in the opposite of the direction $x - \eta\nabla f(x)$ of the gradient so that $\nabla f(x)^T(y-x) < 0$. The role of η is to control the step-size, balancing between the gain in the surrogate CO problem (large η) and the quality of the approximation of the surrogate loss (small η).

Algorithm 1: Gradient Descent

Parameters: Epoch T, step-sizes (η_t).
Initialization: Initial point $x_1 \in \mathcal{K}$.
For each iteration $t = 1, \dots, T$:
Iteration: Update

$$\begin{aligned} y_{t+1} &= x_t - \eta_t \nabla f(x_t), \\ x_{t+1} &= \Pi_{\mathcal{K}}(y_{t+1})\,. \end{aligned}$$

Return x_{T+1}

Let $h_t = f(x_t) - f(x^*)$, we have the rates

- γ-well-conditioned, $\eta_t = 1/\beta$, $h_T = O(e^{-\gamma T})$,
- β-smooth, $\eta_t = \beta/T$, $h_T = O(\beta/T)$,
- α-strongly convex, $\eta_t = 1/(\alpha T)$, $h_T = O(1/(\alpha T))$,
- convex, $\eta_t = 1/\sqrt{T}$, $h_T = O(1/\sqrt{T})$.

The last two rates are optimal but the two first ones can be accelerated. Step sizes η_t depend on the regularity of the CO problem and the epoch T. The more regular is f the easier the CO problem $(f, \mathcal{K})$.

Definition 2. *Regularizing the CO problem $(f, \mathcal{K})$ consists in adjoining a regularization function R strongly convex on $\mathcal{K}$ and twice continuously differentiable so that $(f + R, \mathcal{K})$ becomes an easier CO problem.*

Consider the regularized problem $g(x) = f(x) + \alpha/2\|x - x_1\|^2$ where f is convex. Then g is α- strongly convex so that the CO problem gets easier and the error of the GD problem $h_T^g = g(x_T) - g(x^*)$ smaller. However the minimizer of the CO problem changes and we denote it x_g^*. We still have

$$\begin{aligned} f(x_T) - f(x^*) &= g(x_T) - g(x^*) + \alpha/2(\|x^* - x_1\|^2 - \|x_T - x_1\|^2) \\ &\le g(x_T) - g(x_g^*) + \alpha/2D^2 \qquad\qquad (g(x_g^*) \le g(x^*)) \\ &\le h_t^g + \alpha/2D^2\,. \end{aligned}$$

Examples: If f convex, g is α strongly convex, $h_T^g = O(1/(\alpha T))$ and then fix $\alpha = 1/\sqrt{T}$ or if f β-smooth, g is γ well conditioned, $\gamma = \alpha/(\beta + \alpha)$, $h_T = O(e^{-\gamma T})$ and then fix $\alpha = \beta \log T/T$.

It is also possible to smooth the loss function thanks to randomization.

Definition 3. *Randomization is the introduction of a sample scheme in an optimization problem.*

Consider $\widehat{f}_\delta(x) = \mathbb{E}_{U\sim\mathcal{U}(B_1)}[f(x+\delta U)]$ where $\mathcal{U}(B_1)$ is the uniform distribution on the unit Euclidean ball. We have

Proposition. *The randomized version $\widehat{f}_\delta$ is dG/δ-smooth and a δG uniform approximation of f:*

$$|\widehat{f}_\delta(x) - f(x)| \le \delta G, \qquad x \in \mathcal{K}.$$

We then have

$$\begin{aligned} f(x_T) - f(x^*) &\le \widehat{f}_\delta(x_T) - \widehat{f}_\delta(x^*) + 2\delta G \\ &\le \widehat{f}_\delta(x_T) - \widehat{f}_\delta(x^*_{\widehat{f}_\delta}) + 2\delta G \qquad\qquad (\widehat{f}_\delta(x^*_{\widehat{f}_\delta}) \le \widehat{f}_\delta(x^*)) \\ &\le h_t^{\widehat{f}_\delta} + 2\delta G\,. \end{aligned}$$

Example: If f is α strongly convex then $\widehat{f}_\delta$ is γ well conditioned, $\gamma = \alpha\delta/(dG)$, and then fix $\delta = dg \log T/(\alpha T)$.

1.3 Applications

1.3.1 Unconstrained CO problem

Consider the supervised classification problem of 2 classes $\{+1, 1\}$ and one observes labels b_i, $1 \le i \le n$, $b_i \in \{+1, 1\}$ together with explanatory variables $a_i \in \mathbb{R}^d$.

Examples: - Natural Language Processing (NLP) for spam classification: a_i encodes the list of words in an email, d is the number of words in the language, $a_{i,j} = 1$ if the j word appears in the ith mail, $= 0$ else.

- MNIST: Handwritten digit database $n = 60000$ from a_i is a $28x28$ grayscale image, $d = 784$ and one can consider two classes, 0 vs other digits ($b_i = 0$ if the digit is 0, else -1).

Definition 4. *Linear Support Vector Machine (SVM) are classifiers of the form sign$(x^T a_i)$ an hyperplane $x \in \mathbb{R}^d$.*

One wants to find the minimizer x of the accuracy

$$\mathbb{P}(\text{sign}(x^T a) \ne b) \approx x \mapsto \frac{1}{n'} \sum_{i=n+1}^{n+n'} 1\!\!1_{\text{sign}(x^T a_i) \ne b_i}$$

over a test set $(a_i, b_i)_{n+1 \le i \le n+n'}$.

Because of the lack of convexity of the 0/1 loss, the hard-margin problem of optimizing

$$\frac{1}{n} \sum_{i=1}^{n} 1\!\!1_{\text{sign}(x^T a_i) \ne b_i}$$

is non-polynomial. A common way of bypassing the issue is to relax the optimization problem to turn it to a CO problem.

Definition 5. *The hinge loss*

$$\ell_{a,b}(x) = hinge(bx^T a) = \max(0, 1 - bx^T a)$$

is a convex version of the $0-1$ *loss* $1\!\!1_{sign(x^T a) \ne b} = 1\!\!1_{bx^T a < 0}$.

Remark that the hinge loss is a convex function but not strongly convex with potentially multiple minimizers. We consider instead the regularized CO problem called the soft-margin problem

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} \text{hinge}(bx^T a_i) + \frac{\lambda}{2}\|x\|^2 .$$

It is a strongly convex CO.

1.3.2 ℓ^1-ball constrained CO as dual of the LASSO

Consider $f(x) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(x)$ to be minimized on the trained sample $(\ell_1, \ldots, \ell_n)$ assumed to be iid convex functions over $\mathbb{R}^d$. The aim is to minimize the unobserved risk $\mathbb{E}[\ell(x)]$. One can face generalization issues such as overfitting in high dimension.

Definition 6. *The information criteria (AIC, BIC) are penalized f of the form*

$$f(x) + \lambda\|x\|_0 = f(x) + \theta \sum_{i=1}^{d} \mathbb{1}_{x_i \neq 0}\,, \qquad \theta > 0.$$

Due to the lack of convexity, it is a non-polynomial optimization problem. It is relaxed using the convex ℓ^1-norm instead of $\|\cdot\|_0$.

Definition 7. *The LASSO are penalized f of the form*

$$f(x) + \theta\|x\|_1\,, \qquad \theta > 0.$$

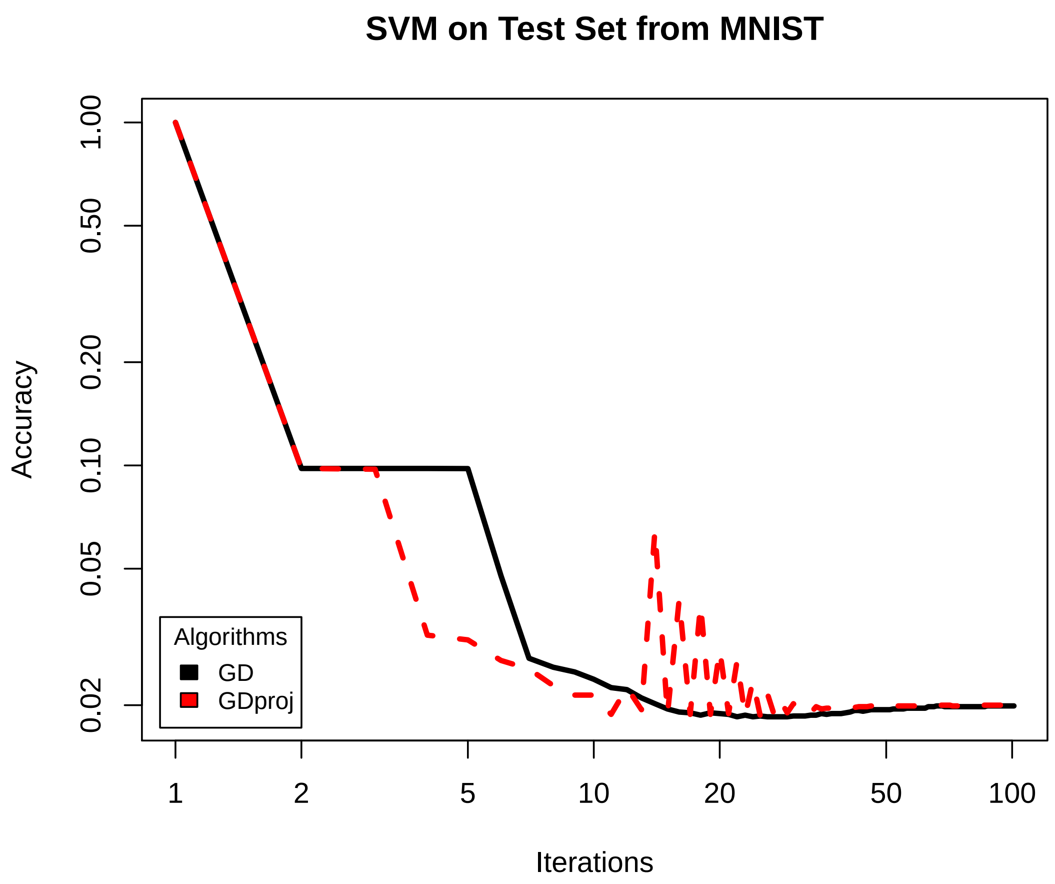
LASSO is an unconstrained CO problem. Using a Langrage dual argument one can turn it into a constrained CO problem.

Proposition. *If x^* is the minimizer of LASSO it is also the minimizer of*

$$\min_{\|x\|_1 \leq \tau} f(x)\,, \tag{1.2}$$

$\tau = x^*$ *. Moreover if x^* is a minimizer of then there exists θ such that it minimizes LASSO.*

Implementation of (projected) GD on MNIST with $\eta_t = 1/(\lambda t)$, regularization parameter $\lambda = 1/3$ and projection on $B_1(100) = \{x \in \mathbb{R}^d;\ \|x\|_1 \leq 100\}$, each iteration costs $O(nd + P)$ as it requires n gradients of dimension d and the projection on an ℓ^1-ball of complexity P.



We notice that the accuracy are better for the projected version with a faster convergence rate but then their accuracies are deteriorating. It is due to overfitting and motivates early stopping methods.

1.3.3 Explicit projection on $B_1(z)$

We consider the CO problem $\Pi_{B_1(z)}(x) = \min_{y\in B_1(z)} \|y-x\|$. One cannot use GD since then a projection step is required... Luckily there exists an explicit solution. Consider the simpler projection on the simplex $\Pi_{\Lambda)}(x)$ where $\Lambda = \{w \in \mathbb{R}^d_+;\ \sum_{i=1}^d w_i = 1\}$. We have the Lagrangian function

$$\mathcal{L}(w,\theta,\zeta) = \frac{1}{2}\|w-x\|^2 + \theta\Big(\sum_{i=1}^d w_i - 1\Big) - \sum_{i=1}^d \zeta_i w_i$$

with parameters $w \in \mathbb{R}^d$, $\theta \in \mathbb{R}$ and $\zeta \in \mathbb{R}^d_+$. We compute its gradient

$$\nabla\mathcal{L}(w,\theta,\zeta) = \begin{pmatrix} w - x + \theta\,\mathbb{I} - \zeta \\ \sum_{i=1}^d w_i - 1 \\ -w \end{pmatrix}, \qquad \mathbb{I} = (1,\dots,1)^T\,.$$

Thus KKT provides

$$\begin{cases} w^* = x - \theta^*\,\mathbb{I} + \zeta^*\,, \\ \sum_{i=1}^d w_i^* = 1 \\ w_i^* = 0 \text{ or } w_i^* > 0 \text{ and } \zeta_i^* = 0\,. \end{cases}$$

To sum up we obtain the soft-thresholding $w_i^* = \mathrm{SoftThreshold}(x_i, \theta^*) = \max(x_i - \theta^*, 0)$. We obtain

Algorithm 2: Projection On the Simplex Π_Λ

Input: $x \in \mathbb{R}^d$.
If $x \in \Lambda$
Then Return x.
Else
Sort $x_{(1)} \geq \cdots\cdots x_{(d)}$
Find $d_0 = \max\Big\{1 \leq i \leq d;\, x_{(i)} - \frac{1}{i}(\sum_{j=1}^{i} x_{(j)} - 1\Big)\Big\}$
Define $\theta^* = \frac{1}{d_0}(\sum_{j=1}^{d_0} x_{(j)} - 1\Big)$
Return $w^* = \mathrm{SoftThreshold}(x, \theta^*)$.

The projection over the ℓ^1-ball follows easily form the one on the simplex.

Algorithm 3: Projection On ℓ^1-ball $\Pi_{B_1(z)}$

Input: $x \in \mathbb{R}^d$.
If $x \in B_1(z)$
Then Return x.
Else
$w^* = \Pi_\Lambda(|x|/z)$
Return $y = \mathrm{sign}(x) w^*$.

Computational cost is $P = O(d \log(d))$ on averaged, as Quicksort.

Chapter 2

Online Gradient Descent for Online Convex Optimization (OCO)

We extend the previous CO setting to the the OCO problem and analyses the online gradient descent.

2.1 The setting

We consider now a recursive setting. At each iteration t of the algorithm, the algorithm predicts x_t and then the loss function f_t is revealed, potentially varying through time. Then the algorithm incurs the loss $f_t(x_t)$ and its aim is to minimize its regret at any horizon T

$$\mathrm{Regret}_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x)\,,$$

its cumulative losses relative to the best strategy frozen through time.

Definition 8. *The* **full adversarial** *setting corresponds to f_t chosen by an adversary as the worst possible loss function given the past predictions $x_t, x_{t-1}, \ldots$.*

Example 1 (Rock, Paper, Scissor)**.** *Consider the game with the following cost table where 0 denotes a draw, 1 denotes that the row player wins, and 1 denotes a column player victory:*

Nature \ *Algorithm*	*Rock*	*Paper*	*Scissor*
Rock	*0*	*-1*	*1*
Paper	*1*	*0*	*-1*
Scissor	*-1*	*1*	*0*

$= A$

where A is a 3×3 cost matrix. We consider then $\mathcal{K} = \{Rock, Paper, Scissor\}$, it is discrete (not convex). One randomizes the strategy (mixed strategy) by considering $x \in \Lambda$, the simplex $\{x \in \mathbb{R}^3_+; x_1 + x_2 + x_3 = 1\}$ so that the strategy is $\mathbb{P}(Rock) = x_1 \ldots$ Consider first the full adversarial setting; the algorithm have a randomized strategy x_t and plays Rock, Paper and Scissor $i_t = (0, 1, 2)$ according x_t. Then the adversary chooses the worst move according to x_t (and not the move that she cannot predict). We obtain $f_t(x_t) = \max_{y \in \Lambda} y^T A x_t$. It is *a convex function of x_t since*

$$\begin{aligned}\max_{y\in\Lambda} y^T A(\alpha x_1 + (1-\alpha)x_2) &= \max_{y\in\Lambda}(\alpha y^T Ax_1 + (1-\alpha)y^T Ax_2)\\ &\le \alpha \max_{y\in\Lambda} y^T Ax_1 + (1-\alpha)\max_{y\in\Lambda} y^T Ax_2 \,.\end{aligned}$$

It is a full adversarial OCO called the zero-sum game.

Of course OCO also embeds much more gentle adversarial settings:

Proposition. *If $f_t = f$ is constant then, back to CO and the accuracy of the averaging $\overline{x}_T = \frac{1}{T}\sum_{t=1}^T x_t$ satisfies*

$$h_t^f = f(\overline{x}_T) - f(x^*) \le \frac{1}{T}\sum_{t=1}^T f(x_t) - f(x^*) = \frac{Regret_T}{T}\,.$$

Definition 9. *Stochastic OCO is if (f_t) is an independent random function sequence with constant mean called the risk $R = \mathbb{E}[f_1]$.*

Proposition. *The accuracy of the averaging on the risk $R = \mathbb{E}[f_1]$ satisfies, on average,*

$$\mathbb{E}[h_T^R] \le \mathbb{E}\Big[\frac{Regret_T}{T}\Big]\,.$$

Proof. We have

$$\begin{aligned}\mathbb{E}[h_T^R] = \mathbb{E}[R(\overline{x}_T) - R(x^*)] &\le \mathbb{E}\Big[\frac{1}{T}\sum_{t=1}^T R(x_t) - R(x^*)\Big]\\ &\le \mathbb{E}\Big[\frac{1}{T}\sum_{t=1}^T \mathbb{E}[f_t](x_t) - \mathbb{E}[f_t](x^*)\Big]\\ &\le \mathbb{E}\Big[\frac{1}{T}\sum_{t=1}^T \mathbb{E}[f_t(x_t) - f_t(x^*) \mid \mathcal{F}_{t-1}]\Big]\\ &\le \mathbb{E}\Big[\frac{\mathrm{Regret}_T}{T}\Big]\,,\end{aligned}$$

where one has to introduce the natural filtration $\mathcal{F}_t = \sigma(f_t, f_{t-1}, \dots, f_1)$ and uses the fact that x_t is $\mathcal{F}_{t-1}$-measurable. □

The aim of OCO is to designed algorithms with the best possible regret and at least sub-linear regrets

$$Regret_T = o(T)\,.$$

2.2 Failure of Follow The Leader (FTL)

We call FTL the strategy from CO: at each t predicts

$$x_t = x_{t-1}^* \in \arg\min_{x\in\mathcal{K}} \sum_{k=1}^{t-1} f_k(x)\,.$$

This strategy fails in the OCO setting. Consider $\mathcal{K} = [-1, 1]$, $f_1(x) = x/2$ and

$$f_k(x) = \begin{cases} -x & \text{if } k \text{ is even} \\ x & \text{else.} \end{cases}$$

Thus

$$\sum_{k=1}^{t-1} f_k(x) = \begin{cases} -x/2 & \text{if } t \text{ is odd} \\ x/2 & \text{else.} \end{cases}$$

so that FTL predicts $x_t = -1$ if t is odd and $= 1$ else and occurs $f_t(x_t) = 1/2$. Thus the regret is

$$Regret_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x) = T/2 + 1/2\,.$$

2.3 Online Gradient Descent (OGD)

This online version of GD has been introduced by Zinkevich (2003).

Algorithm 4: Online Gradient Descent

Parameters: Step-sizes (η_t).
Initialization: Initial prediction $x_1 \in \mathcal{K}$.
For each recursion $t \geq 1$:
Predict: x_t
Incur: $f_t(x_t)$
Observe: $\nabla f_t(x_t)$
Recursion: Update

$$\begin{aligned} y_{t+1} &= x_t - \eta_t \nabla f_t(x_t), \\ x_{t+1} &= \Pi_{\mathcal{K}}(y_{t+1})\,. \end{aligned}$$

OGD succeeds where FTL fails:

Theorem. *OGD with $\eta_t = \frac{D}{G\sqrt{t}}$ satisfies*

$$Regret_T \leq \frac{3}{2} GD\sqrt{T}$$

Proof. We start with the gradient trick $f_t(x_t) - f_t(x^*) \leq \nabla f_t(x_t)(x_t - x^*)$, $t \geq 1$. Thus we will estimate the linear regret

$$\sum_{t=1}^{T} \nabla f_t(x_t)(x_t - x^*)$$

Using the updates, we have

$$\begin{aligned} \|x_{t+1} - x^*\|^2 &\leq \|\Pi_{\mathcal{K}}(x_t - \eta_t \nabla f_t(x_t)) - x^*\|^2 \\ &\leq \|x_t - \eta_t \nabla f_t(x_t) - x^*\|^2 \\ &\leq \|x_t - x^*\|^2 + \eta_t^2 \|\nabla f_t(x_t)\|^2 - 2\eta_t \nabla f_t(x_t)^T (x_t - x^*)\,. \end{aligned}$$

We get, whatever is x^*,

$$2\eta_t \nabla f_t(x_t)^T (x_t - x^*) \leq \|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2 + \eta_t^2 G^2\,. \tag{2.1}$$

One deduces that ($1/\eta_0 = 0$ by convention and $\|x_{t+1} - x^*\|^2 \geq 0$)

$$\begin{aligned}
2\sum_{t=1}^{T} \nabla f_t(x_t)^T(x_t - x^*) &\leq \sum_{t=1}^{T} \frac{\|x_t - x^*\|^2 - \|x_{t+1} - x^*\|^2}{\eta_t} + G^2 \sum_{t=1}^{T} \eta_t \\
&\leq \sum_{t=1}^{T} \|x_t - x^*\|^2 \Big(\frac{1}{\eta_t} - \frac{1}{\eta_{t+1}}\Big) + G^2 \sum_{t=1}^{T} \eta_t \\
&\leq D^2 \sum_{t=1}^{T} \Big(\frac{1}{\eta_t} - \frac{1}{\eta_{t+1}}\Big) + G^2 \sum_{t=1}^{T} \eta_t \\
&\leq \frac{D^2}{\eta_T} + G^2 \sum_{t=1}^{T} \eta_t \\
&\leq 3DG\sqrt{T}\,.
\end{aligned}$$

We use that $\sum_{t=1}^{T} \eta_t \leq 2\sqrt{T}$. □

Remark. *Note that if η is constant then a similar argument yields the upper bound*

$$\frac{1}{2}\Big(\frac{\|x_1 - x^*\|^2}{\eta} + \eta \sum_{t=1}^{T} \|\nabla f_t(x_t)\|^2\Big)$$

which is minimized for

$$\eta = \frac{\|x_1 - x^*\|}{\sqrt{\sum_{t=1}^{T} \|\nabla f_t(x_t)\|^2}}$$

and get the optimal regret bound

$$D\sqrt{\sum_{t=1}^{T} \|\nabla f_t(x_t)\|^2}\,.$$

However this bound is not achievable since the learning rate η is tuned knowing the gradients $\nabla f_t(x_t)$, $1 \leq y \leq T$, which are not observed.

Exercise 1. *Compute the regret in the OCO where FTL fails. Interpret.*

Moreover in favorable cases one can accelerate OGD:

Definition 10 (Strongly convex OCO)**.** *We consider the OCO problem over $\mathcal{K}$ and we assume the existence of $\alpha > 0$ so that the f_t are α-strongly convex.*

OGD satisfies an optimal regret bound $O(\log T)$ by modifying the step-sizes (learning rates) accordingly:

Theorem. *Assume the strongly convex OCO problem, then OGD with step sizes $\eta_t = 1/(\alpha t)$ satisfies*

$$Regret_t \leq \frac{G^2}{2\alpha}(1 + \log T)\,.$$

Proof. We write

$$Regret_T = \sum_{t=1}^{T} f_t(x_t) - f_t(x_T^*)$$

so that by α-strong convexity we get the improved gradient trick

$$2(f_t(x_t) - f_t(x_T^*)) \leq 2\nabla f_t(x_t)^T(x_t - x_t^*) - \alpha\|x_t - x_T^*\|^2 .$$

Using (2.1), namely

$$2\eta_t \nabla f_t(x_t)^T(x_t - x_T^*) \leq \|x_t - x_T^*\|^2 - \|x_{t+1} - x_T^*\|^2 + \eta_t^2 G^2$$

we get

$$\begin{aligned} 2(f_t(x_t) - f_t(x_T^*)) &\leq \frac{\|x_t - x_T^*\|^2 - \|x_{t+1} - x_T^*\|^2}{\eta_t} - \alpha\|x_t - x_T^*\|^2 + \eta_t G^2 \\ &\leq \alpha(t-1)\|x_t - x_T^*\|^2 - \alpha t\|x_{t+1} - x_T^*\|^2 + \frac{G^2}{\alpha t} . \end{aligned}$$

Thus one gets a telescoping argument when bounding the regret

$$\begin{aligned} 2Regret_T &= \sum_{t=1}^{T} 2(f_t(x_t) - f_t(x_T^*)) \\ &\leq \sum_{t=1}^{T} \alpha(t-1)\|x_t - x_T^*\|^2 - \alpha t\|x_{t+1} - x_T^*\|^2 + \frac{G^2}{\alpha t} \\ &\leq -\alpha T\|x_{T+1} - x_T^*\|^2 + \frac{G^2}{\alpha}(1 + \log T) \end{aligned}$$

since $\sum_{t=1}^{T} t^{-1} \leq 1 + \log T$. The desired result follows. □

Note that the regret bounds for OGD in the convex and strongly convex OCO problem are optimal.

2.4 Applications

2.4.1 Stochastic Gradient Descent (SGD)

Consider the CO problem $(f, \mathcal{K})$. Instead of using ∇f we use a noisy version of the gradient $\widehat{\nabla f}$ so that $\mathbb{E}[\widehat{\nabla f}(x)] = \nabla f(x)$ and $\mathbb{E}[\|\widehat{\nabla f}(x)\|^2] \leq G^2$, independent of anything else. The approximation $\widehat{\nabla f}$ is unbiased with bounded variance and the setting is called Stochastic Optimisation (SO).

Example 2. *Consider the CO problem $(f, \mathcal{K})$ where $f(x) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(x)$ as in the SVM classification. Then each step of a GD costs $O(nd)$ since it requires the query of n gradients $\nabla \ell_i$. Instead sample randomly uniformly $I \in \{1, \ldots, n\}$ and use $\widehat{\nabla f} = \nabla \ell_I$. We have*

$$\mathbb{E}[\widehat{\nabla f}(x)] = \sum_{i=1}^{n} \nabla \ell_i(x)\mathbb{P}(i = n) = \frac{1}{n}\sum_{i=1}^{n} \nabla \ell_i(x) = \nabla f(x)$$

and

$$\mathbb{E}[\|\widehat{\nabla f}(x)\|^2] = \sum_{i=1}^{n} \|\nabla \ell_i(x)\|^2\mathbb{P}(i = n) = \frac{1}{n}\sum_{i=1}^{n} \|\nabla \ell_i(x)\|^2 \leq G^2$$

as soon as $\|\nabla \ell_i(x)\| \leq G$. Each step of a Stochastic GD (SGD) on $\nabla \ell_I$ costs $O(d)$.

Remark that by Jensen's inequality we have $\|\nabla f\|^2 \leq \mathbb{E}[\|\widehat{\nabla f}(x)\|^2]$ in the example above.

Proposition. *Any SO problem reduces to a stochastic OCO problem by considering* $\nabla f_t(x_t) = \widehat{\nabla f}(x_t)$.

Proof. A SO problem requires that the approximations $\widehat{\nabla f}$ are all unbiased and independent and the optimizer introduces the randomness and chooses the distribution of $\widehat{\nabla f}$. In the stochastic OCO problem it is the nature which is random and chooses the distribution of f_t with mean $R = \mathbb{E}[f_1]$ and independent. Forgetting that the distribution is chosen by the optimizer, an algorithm robust to any choice of distribution will have good accuracy on the risk R in the stochastic OCO problem It implies the same accuracy bound on the deterministic function $f = R$ whatever the optimizer chooses as $\widehat{\nabla f}$. □

Based on this equivalence and on Proposition 2.1, SGD is a stochastic gradient descent together with an averaging step.

Algorithm 5: Stochastic Gradient Descent, Robbins and Monro (1951)

Parameters: Epoch T, step-sizes (η_t).
Initialization: Initial point $x_1 \in \mathcal{K}$.
For each iteration $t = 1, \dots, T$:
Iteration: Sample: $\widehat{\nabla f}(x_t)$
Update

$$\begin{aligned} y_{t+1} =& x_t - \eta_t \widehat{\nabla f}(x_t), \\ x_{t+1} =& \Pi_{\mathcal{K}}(y_{t+1}) \,. \end{aligned}$$

Return: $\overline{x}_{T+1} = \dfrac{1}{T+1} \displaystyle\sum_{t=1}^{T+1} x_t$

The SGD is an iterative algorithm that can be studied via the stochastic OCO setting.

Theorem. *SGD algorithm applied to* $(f, \mathcal{K})$ *with* $\eta_t = D/(G\sqrt{t})$ *have an accuracy satisfying, on average,*

$$\mathbb{E}[h_T] \leq \frac{3}{2} \frac{GD}{\sqrt{T}} \,.$$

SGD algorithm applied to α*-strongly convex* $(f, \mathcal{K})$ *with* $\eta_t = 1/(\alpha t)$ *have an accuracy satisfying, on average,*

$$\mathbb{E}[h_T] \leq \frac{G^2}{2\alpha} \frac{1 + \log T}{T} \,.$$

The original CO setting is deterministic, the randomness comes from the sample of the approximations $\widehat{\nabla f}$ at each iteration. The expectation holds on this randomness.

The results on SGD follows from the regret bounds for OCO together with the online to batch conversion provided in Proposition 2.1.

The bounds are optimal, up to a log term. We gain in term of complexity; assume we are interested in an average accuracy of order $\epsilon > 0$ in the strongly convex CO problem $(f, \mathcal{K})$. GD and SGD would require both $T = O(\epsilon^{-1})$ iterations. However the cost of each iteration is $O(nd)$ and $O(d)$ for GD and SGD, respectively, ending to a total cost of $O(nd\epsilon^{-1})$ and $O(d\epsilon^{-1})$, respectively. When n is large SGD is much more efficient on average!

2.4.2 Soft margin problem for linear SVM

Recall the soft margin problem which is a strongly convex CO problem with

$$f = \frac{1}{n}\sum_{i=1}^{n} \ell_{a_i,b_i}(x) + \frac{\lambda}{2}\|x\|^2$$

where $\ell_{a,b}(x) = \max(0, 1 - bx^T a)$.

Sampling I uniformly over $\{1, \dots, n\}$ one gets the approximation

$$\widehat{\nabla f}(x) = \nabla \ell_{a_I,b_I}(x) + \lambda x$$

which is unbiased. Since the learning rate is tuned as $1/(\lambda t)$ we get the SGD for solving the soft margin problem

---

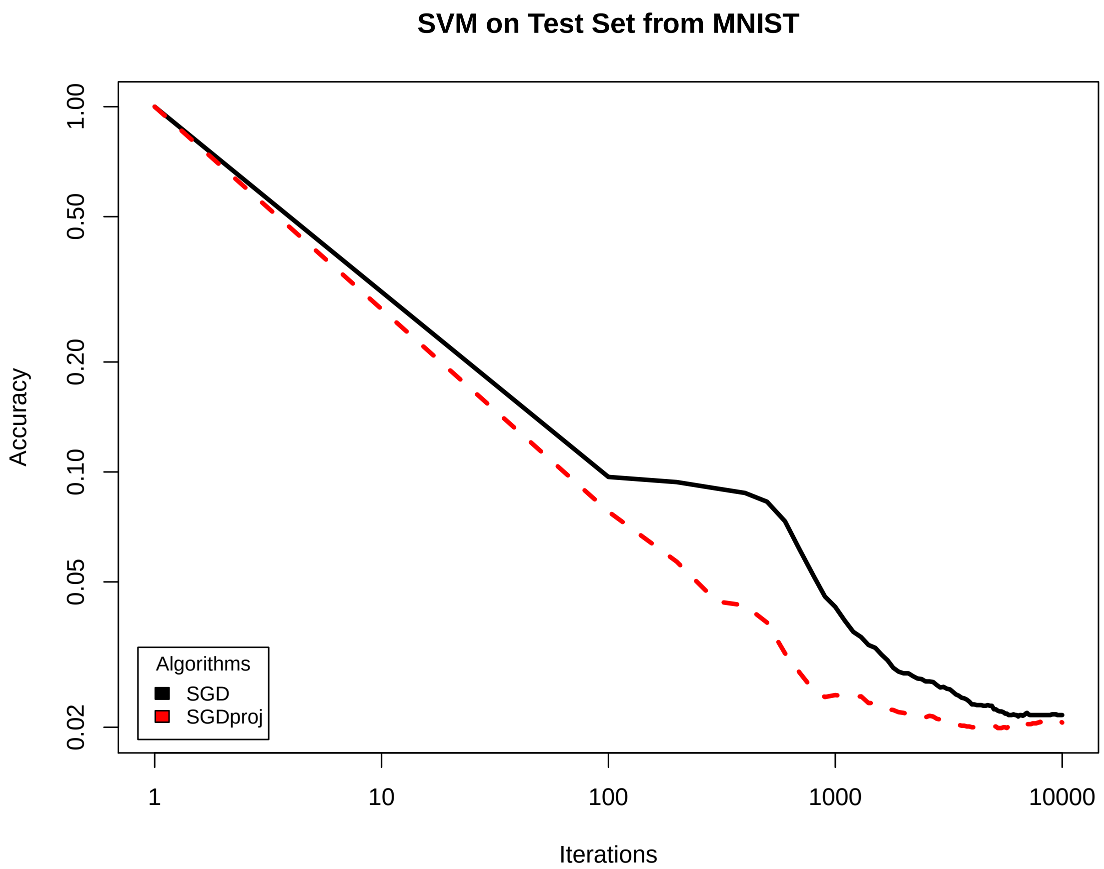
Algorithm 6: SGD for linear SVM.

Parameters: Epoch T, radius $z > 0$, regularization parameter $\lambda > 0$
Initialization: Initial point $x_1 = 0$.
Sample uniformly iid: $(I_t)_{1\le t\le T}$ from $\{1 \le i \le n\}$
For each iteration $t = 1, \dots, T$:
Iteration: Update

$$\begin{aligned} y_{t+1} &= (1 - 1/t)x_t - \frac{\nabla \ell_{a_{I_t},b_{I_t}}(x_t)}{\lambda t}, \\ x_{t+1} &= \Pi_{B_1(z)}(y_{t+1})\,. \end{aligned}$$

Return: $\overline{x}_{T+1} = \dfrac{1}{T+1}\displaystyle\sum_{t=1}^{T+1} x_t$

The accuracy of the algorithm is on average $O(1/T)$ neglecting log terms. Implementation of (projected) SGD on MNIST with regularization parameter $\lambda = 1/3$ and projection on $B_1(100) = \{x \in \mathbb{R}^d;\ \|x\|_1 \le 100\}$, each iteration costs $O(d)$ and the relative speed $1/1000$ compared to GD. The


SVM on Test Set from MNIST
Accuracy
1.00
0.50
0.20
0.10
0.05
0.02
Algorithms
SGD
SGDproj
1
10
100
1000
10000
Iterations

Chapter 3

Online Regularization

3.1 Online regularization

We develop a general strategy for designing efficient OCO algorithms. The basic idea is to regularize FTL online so that it does not change too abruptly. OGD fails in this class of Regularized FTL OCO algorithms.

Let R be a strongly convex regularization function twice continuously differentiable. Instead of the instance of FTL

$$x_t^* = \arg\min_{x \in \mathcal{K}} \sum_{k=1}^{t} f_k(x)$$

we design x_{t+1} such as

$$x_{t+1} = \arg\min_{x \in \mathcal{K}} \sum_{k=1}^{t} \nabla f_k(x_k)^T x + \frac{1}{\eta} R(x)\,.$$

We did this in two steps; the first one is to change the cumulative loss up to $t-1$ with a surrogate loss thanks to the gradient trick:

$$\sum_{k=1}^{t} f_k(x) - \sum_{k=1}^{t} f_k(x^*) \le \sum_{k=1}^{t} \nabla f_k(x)^T (x - x^*)\,.$$

and replacing the unobserved $\sum_{k=1}^{t} \nabla f_k(x)$ with approximations $\sum_{k=1}^{t} \nabla f_k(x_k)$. The obtained surrogate loss is linear

$$\sum_{k=1}^{t} \nabla f_k(x_k)^T (x - x^*)\,.$$

The second step consists in regularizing this simple linear (convex) loss

$$\sum_{k=1}^{t} \nabla f_k(x_k)^T (x - x^*) + \frac{1}{\eta} R(x)\,.$$

Doing so, we aim at obtaining an explicit formula for RFTL x_{t+1} (use of a simple surrogate loss) and a more stable (regularized) version of FTL. We obtain

Algorithm 7: Regularized Follow The Leader, Shalev-Shwartz and Singer (2007)

Parameters: Regularization function R, step-size $\eta > 0$.
Initialization: Initial prediction $x_1 \in \mathcal{K}$.
For each recursion $t \geq 1$:
Predict: x_t
Incur: $f_t(x_t)$
Observe: $\nabla f_t(x_t)$
Recursion: Update

$$x_{t+1} = \arg\min_{x \in \mathcal{K}} \sum_{k=1}^{t} \nabla f_k(x_k)^T x + \frac{1}{\eta} R(x)\,.$$

RFTL is a class of OCO algorithms. One has to specify R to specify the properties of the specific RFTL.

3.2 Online Mirror Descent

OMD is an alternative way of defining RFTL in a more explicit way. For that we use the convex duality defined as

Definition 11. *Let g be a function defined on the convex set $\mathcal{K}$ then its convex conjugate g^* is defined on the dual space $\mathcal{K}^*$ as*

$$g^*(x^*) = \max_{x \in \mathcal{K}} \{x^t x^* - g(x)\}\,, \qquad x^* \in \mathcal{K}^*\,.$$

Exercise 2. *Prove that g^* is convex, $g^*(x^*) + g(y) \geq y^T x^*$ (the Fenchel-Young inequlity) and when g is continuously differentiable*

$$\nabla g^*(x^*) = \arg\max_{x \in \mathcal{K}} \{x^T x^* - g(x)\}\,.$$

Compute the conjugate of $g(x) = |x|^p/p$ for any $1 < p < \infty$.

It is very natural to introduce the duality since a control of a scalar product provides a regret bound from the gradient trick

$$\sum_{t=1}^{T} \nabla f_t(x_t)^T x \leq R^*\Big(\sum_{t=1}^{T} \nabla f_t(x_t)\Big) + R(x)\,.$$

the OMD algorithm is designed for obtaining a good bound over $R^*\Big(\sum_{t=1}^{T} \nabla f_t(x_t)\Big)$. Note that when g is convex then implicitly $x^* = \nabla g(x)$, $x \in \mathcal{K}$. OMD is an Online Gradient Descent in the convex "dual" space of $\mathcal{K}$ through the regularization function R defined as $\mathcal{K}^* = \{\nabla R(x)^T; x \in \mathcal{K}\}$. The projection back to the primal space $\mathcal{K}$ is driven by the Bergman divergence of R rather than by the usual euclidian norm.

Definition 12. *The Bergman divergence associated to the regularization function R is defined as*

$$B_R(y||x) = R(y) - R(x) - \nabla R(x)^T (y - x)\,.$$

The Bergman divergence shares some similarities with weighted euclidian norms

$$\|x\|_W^2 = x^T W x\,, \qquad W \succ 0\,.$$

Exercise 3. *Show that $B_R(x||y) \geq 0$ and $B_r(x||y) = 0$ iff $x = y$.*

Show that if R is twice continuously differentiable then

$$B_R(x||y) = \frac{1}{2}\|x-y\|_z^2$$

where $\|\cdot\|_z$ is some local norm

$$\|x-y\|_z^2 = (x-y)^T \nabla^2 R(z)(x-y)$$

for $z \in \mathcal{K}$ on the segment $[x,y]$.

Thus OMD will be specified through the properties of the Bergman divergence of R

Algorithm 8: Online Mirror Descent (lazy version), Hazan and Kale (2010)

Parameters: Regularization function R, step-size $\eta > 0$.
Initialization: Initial prediction $x_1 = \arg\min_{x\in\mathcal{K}} B_R(x||y_1)$ with $y_1 \in \mathbb{R}^d$ such that $\nabla R(y_1) = 0$.
For each recursion $t \geq 1$:
Predict: x_t
Incur: $f_t(x_t)$
Observe: $\nabla f_t(x_t)$
Recursion: Update

$$\begin{aligned}\nabla R(y_{t+1}) &= \nabla R(y_t) - \eta \nabla f_t(x_t)\,,\\ x_{t+1} &= \arg\min_{x\in\mathcal{K}} B_R(x||y_{t+1})\,.\end{aligned}$$

Theorem. *The OMD is equivalent to RFTL.*

Proof. We prove that

$$\arg\min_{x\in\mathcal{K}} B_R(x||y_t) = \arg\min_{x\in\mathcal{K}} \sum_{k=1}^{t} \nabla f_k(x_k)^T x + \frac{1}{\eta} R(x)\,.$$

Observe first that by recursion

$$\nabla R(y_t) = \nabla R(y_{t-1}) - \eta \nabla f_{t-1}(x_{t-1}) = -\eta \sum_{k=1}^{t-1} \nabla f_k(x_k)\,,$$

thus y_t satisfies

$$y_t = \arg\min_{y\in\mathbb{R}^d} \sum_{k=1}^{t-1} \eta \nabla f_k(x_k)^T y + R(y)$$

since it satisfies the first order condition $\nabla R(y_t) = -\eta \sum_{k=1}^{t-1} \nabla f_k(x_k)$. Hence

$$\begin{aligned} B_R(x||y_t) &= R(x) - R(y_t) - \nabla R(y_t)^T(x - y_t) \\ &= R(x) - R(y_t) + \eta \sum_{k=1}^{t-1} \nabla f_k(x_k)^T(x - y_t) \\ &= \eta \sum_{k=1}^{t-1} \nabla f_k(x_k)^T x + R(x) - \underbrace{R(y_t) - \eta \sum_{k=1}^{t-1} \nabla f_k(x_k)^T y_t}_{\text{independent of } x} \end{aligned}$$

The desired results follows. □

Denoting $\theta_t = \nabla R(y_t)$ and using the relation

$$\nabla R^*(x) = \arg\max_{y \in \mathcal{K}} \{y^T x - R(y)\} .$$

we have the equivalent formulation of OMD: We update

$$\theta_{t+1} = \theta_t - \eta \nabla f_t(x_t) , \qquad x_{t+1} = \nabla R^*(\theta_{t+1}) ,$$

from the initialization $\theta_1 = 0$ and $x_1 = \arg\max_{y\in\mathcal{K}}\{y^T\theta_{t+1} - R(y)\} = \nabla R^*(\theta_0)$. From that simple formulation and using similar argument than for proving the OGD regret bound, we get

Theorem. *OMD and RFTL satisfy the regret bound, of any $u \in \mathcal{K}$,*

$$\sum_{t=1}^T f_t(x_t) - \sum_{t=1}^T f_t(u) \le \frac{\eta}{2} \sum_{t=1}^T \|\nabla f_t(x_t)\|_t^{*2} + \frac{R(u) - R(x_1)}{\eta} ,$$

where $\|\cdot\|_t^{*2} = \|\cdot\|^2_{\nabla^2 R^*(z_t^*)}$ for R^* the convex conjugate of R and z_t^* some point in $\mathcal{K}^*$.

Proof. We use the gradient trick

$$\sum_{t=1}^T f_t(x_t) - \sum_{t=1}^T f_t(u) \le \sum_{t=1}^T \nabla f_t(x_t)^T (x_t - u) .$$

We introduce the mirror analysis by introducing the dual θ_t:

$$\begin{aligned} \sum_{t=1}^T \nabla f_t(x_t)^T(x_t - u) &= -\frac{1}{\eta} \sum_{t=1}^T (\theta_{t+1} - \theta_t)^T x_t + \frac{1}{\eta} \theta_{T+1}^T u \\ &= -\frac{1}{\eta} \sum_{t=1}^T (\theta_{t+1} - \theta_t)^T \nabla R^*(\theta_t) + \frac{R(u) + R^*(\theta_{t+1})}{\eta} \end{aligned}$$

as $x_t = \nabla R^*(\theta_t)$ and applying Young's inequality. By definition of the Bergman divergence

$$\begin{aligned} \sum_{t=1}^T \nabla f_t(x_t)^T(x_t - u) &= \frac{1}{\eta} \sum_{t=1}^T (R^*(\theta_t) - R^*(\theta_{t+1}) + B_{R^*}(\theta_{t+1}||\theta_t)) + \frac{R(u) + R^*(\theta_{t+1})}{\eta} \\ &= \frac{1}{\eta} \sum_{t=1}^T B_{R^*}(\theta_{t+1}||\theta_t) + \frac{R(u) + R^*(\theta_1)}{\eta} . \end{aligned}$$

One recognizes $B_{R^*}(\theta_{t+1}||\theta_t) = \|\theta_{t+1} - \theta_t\|_t^{*2} = \eta^2 \|\nabla f_t(x_t)\|_t^{*2}$ for z_t^* on the segment $[\theta_t, \theta_{t+1}]$ and $R^*(\theta_1) = \max_{y\in\mathcal{K}} y^T\theta_1 - R(y) = \max_{y\in\mathcal{K}} -R(y) = -R(x_1)$. The desired result follows. □

3.3 Specific OMD

3.3.1 Quadratic Regularization

Online Mirror Descent is usually thought as RFLT associated to quadratic R. Consider $R(x) = \frac{1}{2}\|x - x_1\|^2$ for an arbitrary $x_1 \in \mathcal{K}$ and $\eta > 0$. Then $\nabla R(y_1) = (y_1 - x_1) = 0$ iff $y_1 = x_1$. Moreover

$$\begin{aligned} B_R(x\|y) &= \frac{1}{2}\|x - x_1\|^2 - \frac{1}{2}\|y - x_1\|^2 - (y - x_1)^T(x - y) \\ &= \frac{1}{2}\|x - y + y - x_1\|^2 - \frac{1}{2}\|y - x_1\|^2 - (y - x_1)^T(x - y) \\ &= \frac{1}{2}\|x - y\|^2 \end{aligned}$$

so that

$$x_{t+1} = \arg\min_{x \in \mathcal{K}} B_R(x\|y_{t+1}) = \Pi_{\mathcal{K}}(y_{t+1})\,.$$

We have $\nabla R(y_{t+1}) = (y_{t+1} - x_1) = \theta_{t+1} = -\eta \sum_{k=1}^{t} \nabla f_t(x_t)$ so that

$$y_{t+1} = y_t - \eta \nabla f_t(x_t), \qquad y_1 = x_1\,.$$

Thus OMD for quadratic regularization function is an unconstrained OGD then projected on $\mathcal{K}$ at each iteration.

Algorithm 9: Online Mirror Descent (for quadratic R)

Parameters: step-size $\eta > 0$.
Initialization: Initial prediction $x_1 = y_1 \in \mathcal{K}$.
For each recursion $t \geq 1$:
Predict: x_t
Incur: $f_t(x_t)$
Observe: $\nabla f_t(x_t)$
Recursion: Update

$$\begin{aligned} y_{t+1} &= y_t - \eta \nabla f_t(x_t)\,, \\ x_{t+1} &= \Pi_{\mathcal{K}}(y_{t+1})\,. \end{aligned}$$

Exercise 4. *Show that OMD is equivalent to OGD with a smaller step size $\eta'_t < \eta$ only when $\mathcal{K} = B_2(z)$, $z > 0$.*

Remark. *We say that the OMD is lazy, meaning that it moves less than the OGD for the same learning rates. An agile version of the general OMD consists in replacing the recursion step $\nabla R(y_{t+1}) = \nabla R(y_t) - \eta \nabla f_t(x_t)$ with $\nabla R(y_{t+1}) = \nabla R(x_t) - \eta \nabla f_t(x_t)$.*

Since $R^*(x^*) = \frac{1}{2}(\|x^* + x_1\|^2 - \|x_1\|^2) \leq \frac{1}{2}\|x^*\|^2$ for any $x^* \in \mathcal{K}^*$ so that $x^* = \nabla R(x) = x - x_1$ for some $x \in \mathcal{K}$, we get the regret bound

$$\begin{aligned} \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(u) &\leq \frac{\eta}{4} \sum_{t=1}^{T} \|\nabla f_t(x_t)\|^2 + \frac{\|u - x_1\|^2}{2\eta} \\ &\leq \frac{1}{2}\Big(\eta T/2G^2 + \frac{D^2}{\eta}\Big) \\ &\leq GD\sqrt{T/2} \end{aligned}$$

choosing $\eta = D/(G\sqrt{T/2})$.

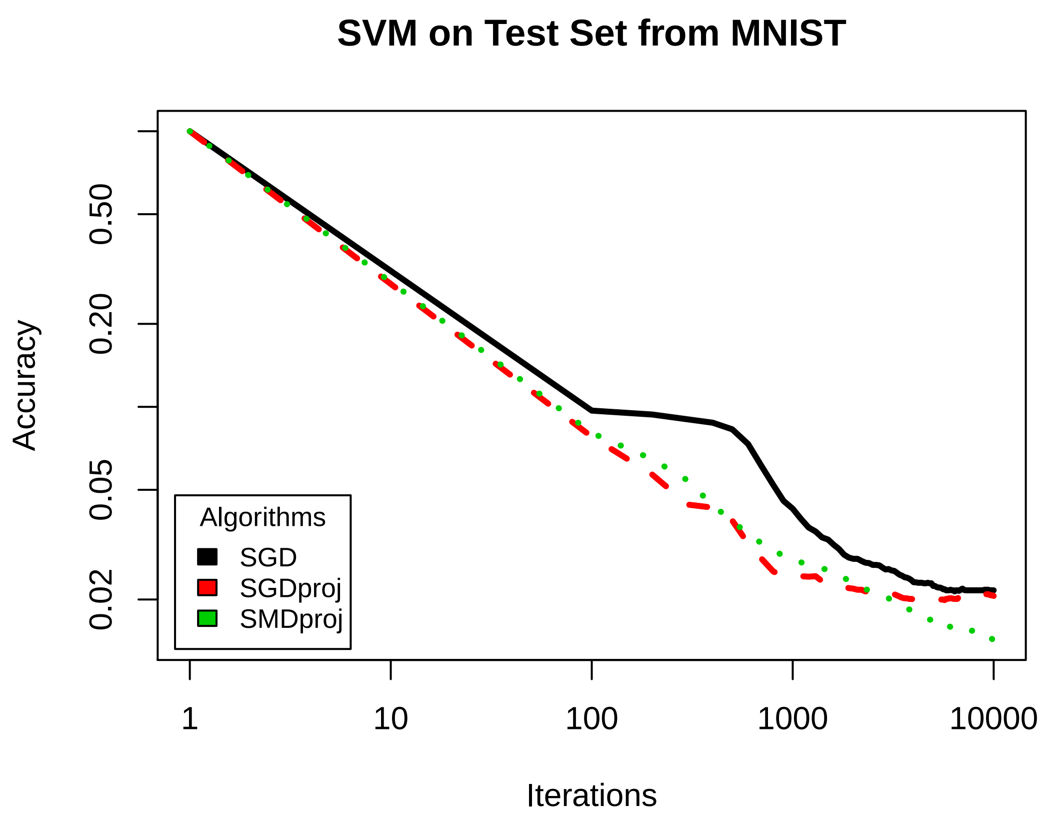
Algorithm 10: SMD for linear SVM

Parameters: Epoch T, radius $z > 0$
Initialization: Initial point $x_1 = y_1 = 0$.
Sample uniformly iid: $(I_t)_{1 \leq t \leq T}$ from $\{1 \leq i \leq n\}$
For each iteration $t = 1, \ldots, T$:
Iteration: Update

$$\begin{aligned}
\eta_t &= 1/\sqrt{t}\,, \\
y_{t+1} &= y_t - \eta_t \nabla \ell_{a_{I_t}, b_{I_t}}(x_t), \\
x_{t+1} &= \Pi_{B_1(z)}(y_{t+1})\,.
\end{aligned}$$

Return: $\overline{x}_{T+1} = \dfrac{1}{T+1} \sum_{t=1}^{T+1} x_t$

One implements the stochastic version of OMD on MNIST. Note that the regularization parameter is not required since the OMD presented here solves any convex problem and not only the strongly convex ones. Despite slower theoretical rates of convergences the accuracies are very similar to regularized SGD. The better convergences at the end of the experiments are due to the raise of the regularization that deviates regularized SGD from its objective.



3.3.2 Randomized strategies, expert advice

Recall the randomized strategy from the Rock, Paper and Scissor game. More generally, consider the setting of d experts with losses $\ell_{t,i}$, $1 \le i \le d$.

Definition 13. *The Expert Advice is the assignment of confidents weights $x_{t,i}$ to each experts $1 \le i \le d$ in order to get the best randomized strategy that picks randomly an expert i_t with probability $x_{t,i}$ so that the averaged regret*

$$\mathbb{E}[Regret_T] = \sum_{t=1}^T \mathbb{E}_{x_t}[\ell_t] - \min_{1\le i\le d} \sum_{t=1}^T \ell_{t,i}$$

is as small as possible.

Since $f_t(x) = \mathbb{E}_{x_t}[\ell_t] = \sum_{i=1}^d x_{t,i}\ell_{t,i} = x_t^T \ell_t$ it is an OCO problem on the linear loss over the simplex λ.

Exercise 5. *Check that* $\min_{x\in\Lambda} x^T \sum_{t=1}^T \ell_t = \min_{1\le i\le d} \sum_{t=1}^T \ell_{t,i}$.

Let $R(x) = x^T \log(x) = \sum_{i=1}^d x_i \log(x_i)$ be the negative entropy function. We consider it as a regularization function over Λ since

$$\nabla R(x) = 1 + \log(x)\,, \qquad \nabla^2 R(x) = \mathrm{Diag}(1/x^2) \succeq I_d$$

even if it is not well defined on the boundary of the simplex when $x_i = 0$ for some $1 \le i \le d$. We have $\Lambda^* \subset (-\infty, 1]^d$ and in order to express OMD, we compute

$$\nabla R^*(y) = \arg\max_{x\in\Lambda}\{y^T x - R(x)\}\,.$$

We compute the Lagrangian function

$$\mathcal{L}(x,\theta,\zeta) = y - \nabla R(x) + \theta\Big(\sum_{i=1}^d x_i - 1\Big) - \sum_{i=1}^d \zeta_i x_i$$

with parameters $x \in \mathbb{R}^d$, $\theta \in \mathbb{R}$ and $\zeta \in \mathbb{R}_+^d$. We compute its gradient

$$\nabla\mathcal{L}(x,\theta,\zeta) = \begin{pmatrix} y - \log(x) + (\theta-1)\,\mathbb{I} - \zeta \\ \sum_{i=1}^d x_i - 1 \\ -x \end{pmatrix}\,, \qquad \mathbb{I} = (1,\ldots,1)^T\,.$$

From KKT, we get the conditions

$$\begin{cases} x & = \exp(y + (\theta-1)\,\mathbb{I} - \zeta) \\ \sum_{i=1}^d x_i & = 1 \\ x_i\zeta_i & = 0\,, \qquad 1 \le i \le d\,. \end{cases}$$

From the first condition we see that $x_i > 0$ so that ζ must be null. We get

$$x = \frac{\exp(y)}{\sum_{i=1}^d \exp(y_i)}\,.$$

We obtain the mixed strategy of expert advice called Exponentiated Weighting Algorithm.

Algorithm 11: EWA, Littlestone and Warmuth (1994)

Parameters: step-size $\eta > 0$.
Initialization: Initial prediction $x = (1/d)\,\mathbb{I}$ and $\theta_1 = 0$.
For each recursion $t \geq 1$:
Pick an expert prediction randomly: $i_t \sim x_t$
Incur the average loss: $\mathbb{E}_{x_t}[\ell_t]$
Observe: $\ell_t \in \mathbb{R}^d$
Recursion: Update

$$\theta_{t+1} = \theta_t - \eta \ell_t\,,$$
$$x_{t+1} = \frac{\exp(\theta_{t+1})}{\sum_{i=1}^d \exp(\theta_{t+1,i})}\,.$$

Remark. *We have complete information since we observe all the losses $\ell_t \in \mathbb{R}^d$ despite we trusted only one expert $i_t \sim x_t$.*

Since $\nabla R^*(\theta) = \exp(\theta)/(\sum_{i=1}^d \exp(\theta_i)) = x$ we get $\nabla^2 R^*(y) = Diag(x) - xx^T$ so that

$$\|\nabla f_t(x_t)\|_t^{*2} = \frac{1}{2}\|\ell_t\|^2_{\nabla R^*(\tilde y)}$$

for some $\tilde\theta$ in the segment $[\theta_t, \theta_{t+1}]$ so that denoting $\tilde x = \exp(\tilde\theta)/(\sum_{i=1}^d \exp(\tilde\theta_i))$ the corresponding weights, we obtain

$$\|\nabla f_t(w_t)\|_t^{*2} = \frac{1}{2}\Big(\sum_{i=1}^d \tilde x_i \ell_{t,i}^2 - \Big(\sum_{i=1}^d \tilde x_i \ell_{t,i}\Big)^2\Big) \leq \frac{1}{2}G_\infty^2$$

where $|\ell_{t,i}| \leq G_\infty$ for all $t \geq 1$, $1 \leq i \leq d$. We obtain

$$\mathbb{E}[Regret_T] = Regret_T(f) \leq \frac{\eta T G^2}{4} + \frac{\log d}{\eta} \leq G_\infty \sqrt{T \log d}\,,$$

choosing $\eta = \frac{2}{G_\infty}\sqrt{\log d/T}$.

Remark. *The dependence on the dimension in $\sqrt{\log d}$ is optimal and is due to the use of the duality and the ℓ^1-ball. Indeed $G_\infty = \max_{1\leq i\leq d}|\ell_{t,i}| = \sup_{x\in\Lambda}\|\nabla f_t(x)\|_\infty = G_{R^*}$ and*

$$\log d = \sup_{x,x'\in\Lambda} R(x) - R(x') = D_R^2$$

the "diameter" of Λ for the regularization function. Thus the regret bound is of order $G_{R^*}D_R\sqrt{T}$. Here the dependence in the dimension of $G_{R^*}D_R$ is optimal in $\sqrt{\log d}$. Such a dependence in the dimension is not achievable when constraining to the ℓ^2 ball whatever is the choice of R, there the dependence in $GD \approx \sqrt{d}$ in the regret analysis of OGD is optimal without additional constrained.

Example 3. *Back to Rock, Paper and Scissor we get an averaged regret bound which is sub-linear in the complete adversarial setting*

$$\sum_{t=1}^T \max_{y\in\Lambda} y^T A x_t - \max_{1\leq i\leq 3}\sum_{t=1}^T \max_{y\in\Lambda} y^T A_i \leq \sqrt{T\log 3}\,,$$

where A_i represent the i-th column of the cost matrix. Since any deterministic strategy incur a loss of 1 *at each round in the complete adversarial setting, we get*

$$\sum_{t=1}^T \max_{y\in\Lambda} y^T A x_t \le T + \sqrt{T\log 3}\,,$$

It is a useless bound and shows the limit of the regret analysis which is relative to a fixed strategy that can be bad in a complete adversarial setting. Regret analysis to a non fixed objective are also developed under the name tracking the best expert Herbster and Warmuth (1998).

A powerful consequence of this analysis is the combination of EWA with the gradient trick for the OCO problem on $\mathcal{K} = \Lambda$. For any algorithm we have the gradient trick

$$Regret_T \le \sum_{t=1}^T \nabla f_t(x_t)(x_t - x^*) \le \sum_{t=1}^T (\nabla f_t(x_t)^T x_t - \nabla f_t(x_t)^T x^*)\,.$$

One interprets the upper bound as the regret of a mixed strategy of d experts with *linearized* losses $\ell_t = \nabla f_t(x_t)$. We obtain

---

Algorithm 12: Hedge, Littlestone and Warmuth (1994)

Parameters: step-size $\eta > 0$.
Initialization: Initial prediction $x = (1/d)\,\mathbb{1}$ and $\theta_1 = 0$..
For each recursion $t \ge 1$:
Predict: x_t
Incur: $f_t(x_t)$
Observe: $\nabla f_t(x_t) \in \mathbb{R}^d$
Recursion: Update

$$\begin{aligned}\theta_{t+1} &= \theta_t - \eta\nabla f_t(x_t)\,,\\ x_{t+1} &= \frac{\exp(\theta_{t+1})}{\sum_{i=1}^d \exp(\theta_{t+1,i})}\,.\end{aligned}$$

We immediately get the optimal regret bound

$$Regret_T = \sum_{t=1}^T f_t(x_t) - \min_{x\in\Lambda}\sum_{t=1}^T f_t(x) = \mathbb{E}[Regret_t(\nabla f)] = G_\infty\sqrt{\log d/T}\,.$$

Note that thanks to the gradient trick the regret bound is now relative to the best fixed strategy of the simplex.

Example 4. *Back to Rock, Paper and Scissor we get an averaged regret bound which is sub-linear in the complete adversarial setting*

$$\sum_{t=1}^T \max_{y\in\Lambda} y^T A x_t - \max_{x\in\Lambda}\sum_{t=1}^T \max_{y\in\Lambda} y^T A x \le \sqrt{T\log 3}\,,$$

where A_i represent the i-th column of the cost matrix. Since the mixed strategy $(1/3, 1/3, 1/3)$ *incurs the loss* 0 *at each round in the complete adversarial setting, we get*

$$\sum_{t=1}^T \max_{y\in\Lambda} y^T A x_t \le \sqrt{T\log 3}\,.$$

It is a very useful bound to prove the von Neumann minimax theorem in zero-sum games.

An extension to the OCO over $\mathcal{K} = B_1(z)$ the ℓ^1-ball of radius $z > 0$ is achieved using $2d$-experts. Indeed the gradient trick still holds and denoting $x_i = z(w_i - w_{i+d})$ the linearized loss

$$\nabla f_t(x_t)^T x_t = \sum_{i=1}^d \nabla f_t(x_t)_i x_{t,i} = z\Big(\sum_{i=1}^d \nabla f_t(x_t)_i w_{t,i} - \sum_{i=1}^d \nabla f_t(x_t)_i w_{t,i+d}\Big) = z \pm\widehat{\nabla f_t}(x_t)^T w_t$$

where $w_t \in \Lambda_{2d}$ and

$$\pm\widehat{\nabla f_t}(x_t) = (\nabla f_t(x_t)_1, \ldots, \nabla f_t(x_t)_d, -\nabla f_t(x_t)_1, \ldots, -\nabla f_t(x_t)_d) \in \mathbb{R}^{2d}.$$

Note that it is aways possible since then $x_{i+}/z = \max(x_i, 0)/z = w_i - \min(w_i, w_{i+d})$ and $x_{i-}/z = \max(-x_i, 0) = w_{i+d} - \min(w_i, w_{i+d})$. Then, parametrizing by $\lambda_i = \min(w_i, w_{i+d}) \geq 0$ we get

$$w_i = x_{i+}/z + \lambda_i\,, \quad w_{i+d} = x_{i-}/z + \lambda_i\,, \qquad 0 \leq 2z\|\lambda\|_1 = z - \|x\|_1 \leq z\,,$$

which admits at least one solution (but potentially several if $\|x\|_1 < z$). Thus we obtain a reduction of the OCO problem on $B_1(z)$ to the OCO problem on $w \in \Lambda_{2d}$ and we get

Algorithm 13: Exponentiated Gradient +/-, Kivinen and Warmuth (1997)

Parameters: step-size $\eta > 0$, radius $z > 0$.
Initialization: Initial prediction $x = 0$ weights $w = 1/(2d)\,\mathbb{I}$ and $\theta_1 = 0 \in \mathbb{R}^{2d}$.
For each recursion $t \geq 1$:
Predict: x_t
Incur: $f_t(x_t)$
Observe: $\nabla f_t(x_t) \in \mathbb{R}^d$
Recursion: Update

$$\begin{aligned}
\theta_{t+1,i} &= \theta_{t,i} - \eta \nabla f_t(x_t)_i\,, \qquad 1 \leq i \leq d\,,\\
\theta_{t+1,i} &= \theta_{t,i} + \eta \nabla f_t(x_t)_i\,, \qquad d+1 \leq i \leq 2d\,,\\
w_{t+1} &= \frac{\exp(\theta_{t+1})}{\sum_{i=1}^{2d} \exp(\theta_{t+1,i})}\,,\\
x_{t+1,i} &= z(w_{t+1,i} - w_{t+1,i+d})\,, \qquad 1 \leq i \leq d\,.
\end{aligned}$$

We immediately get the optimal regret bound

$$Regret_T = G_\infty z \sqrt{\log(2d)/T}\,.$$

One implements the stochastic version of EG+/- on MNIST and improved the performances of SMD (the radius of the ℓ^1-ball is still $z = 100$).

Algorithm 14: SEG+/- for linear SVM

Parameters: Epoch T, radius $z > 0$.

Initialization: Initial point $x_1 = 0$, weights $w = 1/(2d)\,\mathbb{I}$ and $\theta_1 = 0 \in \mathbb{R}^{2d}$.

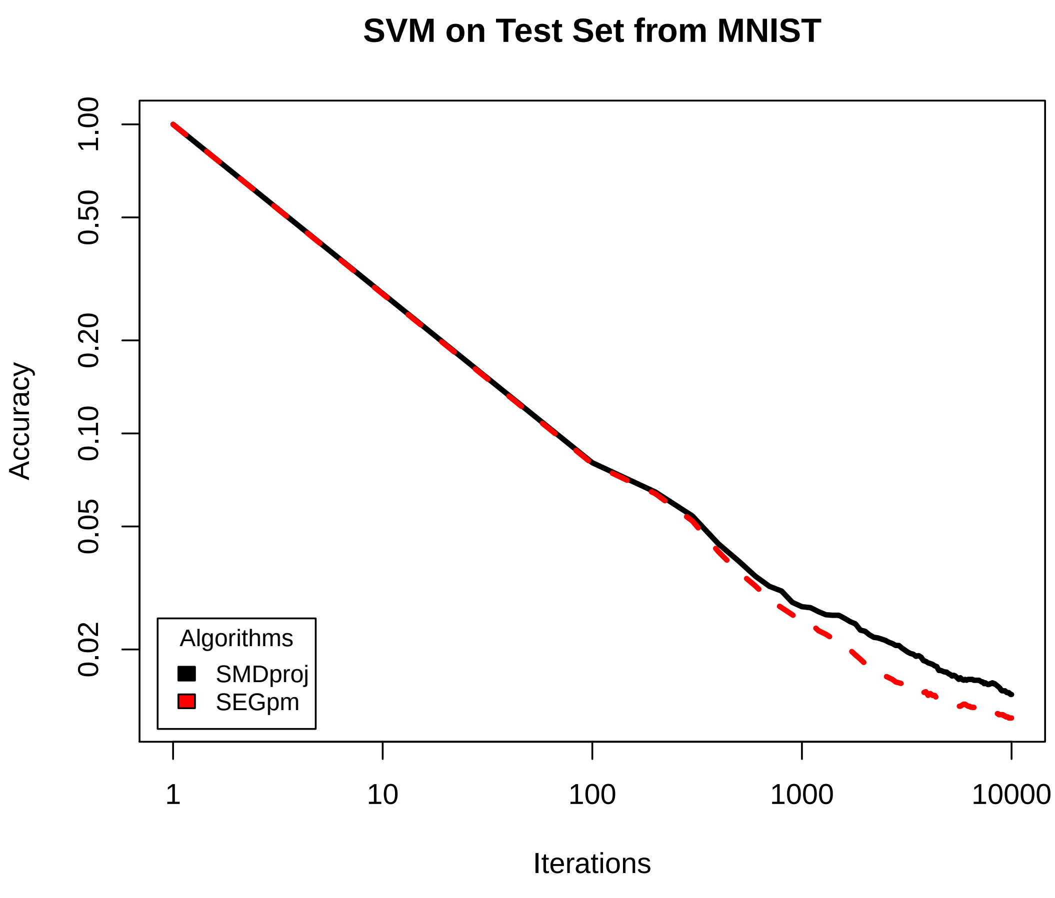
Sample uniformly iid: $(I_t)_{1 \le t \le T}$ from $\{1 \le i \le n\}$

For each iteration $t = 1, \ldots, T$:

Iteration: Update

$$\begin{aligned}
\eta_t &= \sqrt{1/t} \\
\theta_{t+1} &= \theta_t - \eta_t \pm \nabla \ell_{a_{I_t}, b_{I_t}}(x_t)\,, \qquad 1 \le i \le d\,, \\
w_{t+1} &= \frac{\exp(\theta_{t+1})}{\sum_{i=1}^{2d} \exp(\theta_{t+1,i})}\,, \\
x_{t+1,i} &= z(w_{t+1,i} - w_{t+1,i+d})\,, \qquad 1 \le i \le d\,.
\end{aligned}$$

Return: $\bar{x}_{T+1} = \dfrac{1}{T+1} \displaystyle\sum_{t=1}^{T+1} x_t$



3.3.3 AdaGrad

We recall the regret bound for the general OMD or RFTL

$$Regret_T(u) \le \frac{\eta}{2}\sum_{t=1}^T \|\nabla f_t(x_t)\|_t^{*2} + \frac{R(u)-R(x_1)}{\eta}$$

which is equal to, if η is optimized by an oracle,

$$Regret_T(u) \le \sqrt{2\sum_{t=1}^T \|\nabla f_t(x_t)\|_t^{*2}(R(u)-R(x_1))}\,.$$

As we saw this bound heavily depends on the choice of the regularization function R. The best choice of R heavily depends on the properties of the gradients $\nabla f_t(x_t)$ of the losses of the algorithm itself. AdaGrad will learn how to choose the best regularization function.

We restrict R to the class of weighted quadratic regularization functions $R \in \mathcal{H}$ satisfying

$$\forall x \in \mathcal{K}, \nabla^2 R(x) = D = \mathrm{Diag}(s)\,, \qquad s \in (0,\infty)^d,\ \|s\| \le 1,.$$

Remark.

- $R(x) = \frac{1}{2\sqrt{d}}\|x\|^2$ *such that* $\nabla^2 R(x) = \frac{1}{\sqrt{d}} I_d$ *and* $R \in \mathcal{H}$,
- $R(x) = x^T \log(x)$ *such that* $\nabla^2 R(x) = Diag(1/x)$ *is not in* $\mathcal{H}$.

We first determine what could be the best possible regret bound. We compute the second derivative of the convex conjugate $\nabla^2 R^*(x^*) = D^{-1}$ for $D \succ 0$ and

$$\begin{aligned} 2\min_{R\in H}\sum_{t=1}^T \|\nabla f_t(x_t)\|_t^{*2} &= \min_{D=\mathrm{Diag}(s)}\sum_{t=1}^T \|\nabla f_t(x_t)\|^2_{D^{-1}} \\ &= \min_{s\in\mathbb{R}_+^d,\|s\|_1\le 1}\sum_{t=1}^T\sum_{i=1}^d \nabla f_t(x_t)_i^2 s_i^{-1} \end{aligned}$$

Applying Cauchy-Schwartz, we have

$$\sum_{i=1}^d \Big(\sqrt{\sum_{t=1}^T \nabla f_t(x_t)^2/s}\Big)_i^2 \sum_{i=1}^d \sqrt{s}_i^2 \ge \Big(\sum_{i=1}^d \sqrt{\sum_{t=1}^T \nabla f_t(x_t)_i^2}\Big)^2$$

so that

$$\min_{s\in\mathbb{R}_+^d,\|s\|_1\le 1}\sum_{t=1}^T\sum_{i=1}^d \nabla f_t(x_t)_i^2 s_i^{-1} \ge \Big(\sum_{i=1}^d \sqrt{\sum_{t=1}^T \nabla f_t(x_t)_i^2}\Big)^2.$$

Note that this minimizer is achieved by $\|s^*\| = 1$ for

$$s_i^* = \frac{\sum_{t=1}^T \nabla f_t(x_t)_i^2}{\Big(\sum_{i=1}^d \sqrt{\sum_{t=1}^T \nabla f_t(x_t)_i^2}\Big)^2}\,.$$

Thus the best possible regret in this class of regularization function is

$$Regret_T(u) \le \sum_{i=1}^d \sqrt{(R(u)-R(x_1))\sum_{t=1}^T \nabla f_t(x_t)_i^2}\,.$$

However such learning rate is not known before step T. AdaGrad solves this problem by considering a **multiple adaptive learning rates** approach; each coordinate will have its own gradient step close to the optimal s_i.

Algorithm 15: AdaGrad (diagonal version), Duchi et al. (2011)

Parameters: step-size $\eta > 0$.
Initialization: Initial prediction $y_1 = x_1 \in \mathcal{K}$, initial multiple learning rates $S_0 = 0$ (or $= \delta\, \mathbb{I}$ small).
Incur the average loss: $f_t(x_t)$
Observe: $\nabla f_t(x_t) \in \mathbb{R}^d$
Recursion: Update

$$\begin{aligned}
S_t &= S_{t-1} + \nabla f_t(x_t)^2 \\
D_t &= \mathrm{Diag}(\sqrt{S_t}) \\
y_{t+1} &= x_t - \eta D_t^{-1} \nabla f_t(x_t)\,, \\
x_{t+1} &= \arg\min_{x \in \mathcal{K}} \|x - y_{t+1}\|^2_{D_t}\,, \qquad 1 \le i \le d\,.
\end{aligned}$$

We notice that AdaGrad is an agile (x_t in place of y_t) OMD algorithm with adaptive regularization functions $R_t(x) = \frac{1}{2}\|x - x_1\|^2_{D_t}$ since then

$$\nabla R_t(y_{t+1}) = \nabla R_t(x_t) - \eta \nabla f_t(x_t)\,, \qquad B_{R_t}(x\|y) = \frac{1}{2}\|x - y\|^2_{D_t}\,.$$

Theorem. *For $\eta = D_\infty/\sqrt{2}$ with $D_\infty = \max_{x,y\in\mathcal{K}} \|x - y\|_\infty$ AdaGrad get the regret bound*

$$\mathit{Regret}_T \le D_\infty \sum_{i=1}^d \sqrt{2 \sum_{t=1}^T \nabla f_t(x_t)_i^2}\,.$$

Since the regularization functions are depending on t, one has to adapt the simple analysis of OGD above.

Proof. We start from the recursive relation $y_{t+1} - u = y_t - u - \eta D_t^{-1} \nabla f_t(x_t)$ that we rewrite as $D_t(y_{t+1} - u) = D_t(y_t - u) - \eta \nabla f_t(x_t)$ so that multiplying both relations we get

$$\begin{aligned}
\|y_{t+1} - u\|^2_{D_t} &= (y_{t+1} - u)^T D_t (y_{t+1} - u) \\
&= \|x_t - u\|^2_{D_t} - 2\eta \nabla f_t(x_t)^T (x_t - u) + \eta^2 \|\nabla f_t(x_t)\|^2_{D_t^{-1}}\,.
\end{aligned}$$

By the pythagorean Theorem we also have $\|x_{t+1} - u\|^2_{D_t} \le \|y_{t+1} - u\|^2_{D_t}$ so that

$$2\eta \nabla f_t(x_t)^T (x_t - u) \le \|x_t - u\|^2_{D_t} - \|x_{t+1} - u\|^2_{D_t} + \eta^2 \|\nabla f_t(x_t)\|^2_{D_t^{-1}}\,.$$

Then we get

$$\begin{aligned}
2\sum_{t=1}^T \nabla f_t(x_t)^T (x_t - u) \le& \frac{1}{\eta} \sum_{t=1}^T \Big(\|x_t - u\|^2_{D_t} - \|x_{t+1} - u\|^2_{D_t} + \eta^2 \|\nabla f_t(x_t)\|^2_{D_t^{-1}} \Big) \\
\le& \frac{1}{\eta} \sum_{t=1}^T \Big(\|x_t - u\|^2_{D_t} - \|x_t - u\|^2_{D_{t-1}} \Big) \\
&+ \eta \sum_{t=1}^T \|\nabla f_t(x_t)\|^2_{D_t^{-1}}\,,
\end{aligned}$$

with the convention $D_0 = \mathrm{Diag}(S_0)$.

For the first term we use the telescoping sum

$$\sum_{t=1}^T (x_t - u)^T (D_t - D_{t-1})(x_t - u) \le D_\infty^2 \sum_{i=1}^d \sum_{t=1}^T (\sqrt{S_t} - \sqrt{S_{t-1}})_i \le D_\infty^2 \sum_{i=1}^d \sqrt{S_{T,i}}\,.$$

For the last term, we get

$$\begin{aligned}\sum_{t=1}^T \|\nabla f_t(x_t)\|^2_{D_t^{-1}} &= \sum_{t=1}^T \sum_{i=1}^d (\nabla f_t(x_t)^2/\sqrt{S_t})_i \\ &\le \sum_{i=1}^d \sum_{t=1}^T ((S_t - S_{t-1})/\sqrt{S_t})_i\,.\end{aligned}$$

By a comparison with an integral, we get

$$\sum_{t=1}^T ((S_{t,i} - S_{t-1,i})/\sqrt{S_{t,i}} \le \int_0^{S_{T,i}} \frac{dx}{\sqrt{x}} = 2\sqrt{S_{T,i}}\,.$$

Finally we obtain the regret bound

$$\frac{1}{2}\Big(\frac{D_\infty^2}{\eta} + 2\eta\Big) \sum_{i=1}^d \sqrt{S_{T,i}}$$

which yields the desired result as $\eta = D_\infty/\sqrt{2}$. □

Remark. *Denoting $D_i = \max_{x,y\in\mathcal{K}} |(x-y)_i|$ we immediately improve the regret bound to*

$$Regret_T \le \sum_{i=1}^d D_i \sqrt{2\sum_{t=1}^T \nabla f_t(x_t)_i^2} \le \sqrt{\sum_{i=1}^d D_i^2} \sqrt{2\sum_{t=1}^T \nabla f_t(x_t)_i^2}$$

by Cauchy-Schwartz inequality. Since for hyper-rectangles $\mathcal{K}$ we have $\sum_{i=1}^d D_i^2 = D^2 = \max_{x,y\in\mathcal{K}} \|x-y\|^2$, if the gradients were the same for OGD and Adagrad then the best possible regret for OGD is always larger than the one for Adagrad.

In order to implement AdaGrad to the linear SVM one has to explicit the projection step.

Algorithm 16: Adagrad for linear SVM

Parameters: Epoch T, radius $z > 0$.
Initialization: Initial point $x_1 = y_1 = 0$ and $S_0 = 0$ (or $= \delta\mathbb{1}$ small).
Sample uniformly iid: $(I_t)_{1\le t\le T}$ from $\{1 \le i \le n\}$
For each iteration $t = 1, \dots, T$:
Iteration: Update

$$\begin{aligned}S_t &= S_{t-1} + \nabla\ell_{a_{I_t},b_{I_t}}(x_t)^2 \\ D_t &= \mathrm{Diag}(\sqrt{S_t}) \\ y_{t+1} &= x_t - D_t^{-1}\nabla\ell_{a_{I_t},b_{I_t}}(x_t)\,, \\ x_{t+1} &= \arg\min_{x\in B_1(z)} \|x - y_{t+1}\|^2_{D_t}\,, \qquad 1 \le i \le d\,.\end{aligned}$$

Return: $\overline{x}_{T+1} = \dfrac{1}{T+1}\displaystyle\sum_{t=1}^{T+1} x_t$

For that one has to adapt the euclidian projection on the simplex to weighted norms. One has to solve the CO

$$\arg\min_{w\in B_1(z)} \sum_{i=1}^{d} \|w-x\|_D^2 .$$

We have the Lagrangian function

$$\mathcal{L}(x,\theta,\zeta) = \frac{1}{2}(w-x)^T D(w-x) + \theta\Big(\sum_{i=1}^{d} w_i - 1\Big) - \sum_{i=1}^{d} \zeta_i w_i$$

with parameters $w \in \mathbb{R}^d$, $\theta \in \mathbb{R}$ and $\zeta \in \mathbb{R}_+^d$. We compute its gradient

$$\nabla\mathcal{L}(w,\theta,\zeta) = \begin{pmatrix} D(w-x) + \theta\,\mathbb{I} - \zeta \\ \sum_{i=1}^{d} w_i - 1 \\ -w \end{pmatrix}, \qquad \mathbb{I} = (1,\ldots,1)^T .$$

Thus KKT provides

$$\begin{cases} w^* = x - D^{-1}(\theta^*\,\mathbb{I} + \zeta^*)\,, \\ \sum_{i=1}^{d} w_i^* = 1 \\ w_i^* = 0 \text{ or } w_i^* > 0 \text{ and } \zeta_i^* = 0\,. \end{cases}$$

To sum up we obtain the weighted soft-thresholding

$$w_i^* = \max(x - D^{-1}\theta^*\,\mathbb{I}, 0) = D^{-1}\mathrm{SoftThreshold}(Dx, \theta^*).$$

Thus denoting $\|w^*\|_0 = d_0$ we get the relation

$$1 = \sum_{j=1}^{d_0} w_{(j)}^* = \sum_{j=1}^{d_0} D^{-1}\mathrm{SoftThreshold}(Dx,\theta^*) = \sum_{j=1}^{d_0} x_{(j)} - \sum_{j=1}^{d_0} D_{(j)}^{-1}\theta^*$$

where $D_{(j)}$ is the diagonal element of D with the same ordering so that necessarily

$$\theta^* = \frac{1}{\sum_{j=1}^{d_0} D_{(j)}^{-1}}\Big(\sum_{j=1}^{d_0} x_{(j)} - 1\Big).$$

We obtain

Algorithm 17: Projection on the simplex with weighted norm $\|\cdot\|_D$

Input: $x \in \mathbb{R}^d$ and D diagonal.
If $x \in \Lambda$
Then Return x.
Else
Sort $(Dx)_{(1)} \ge \cdots \ge (Dx)_{(d)}$
Find $d_0 = \max\Big\{1 \le i \le d;\ (Dx)_{(i)} - \frac{1}{\sum_{j=1}^{i} D_{(j)}^{-1}}(\sum_{j=1}^{i} x_{(j)} - 1\Big)\Big\}$
Define $\theta^* = \frac{1}{\sum_{j=1}^{d_0} D_{(j)}^{-1}}\Big(\sum_{j=1}^{d_0} x_{(j)} - 1\Big)$
Return $w^* = D^{-1}\mathrm{SoftThreshold}(Dx,\theta^*)$.

Recall that for the hinge loss we have, for any instance (a,b) of the training set $(a_t,b_t)_{1\le t\le d}$

$$\nabla\ell_{a,b}(x_t) = \begin{cases} 0 & \text{if } bx^Ta > 1\,, \\ -ba & \text{else}\,. \end{cases}$$

Assume that the design is sparse, i.e. we have

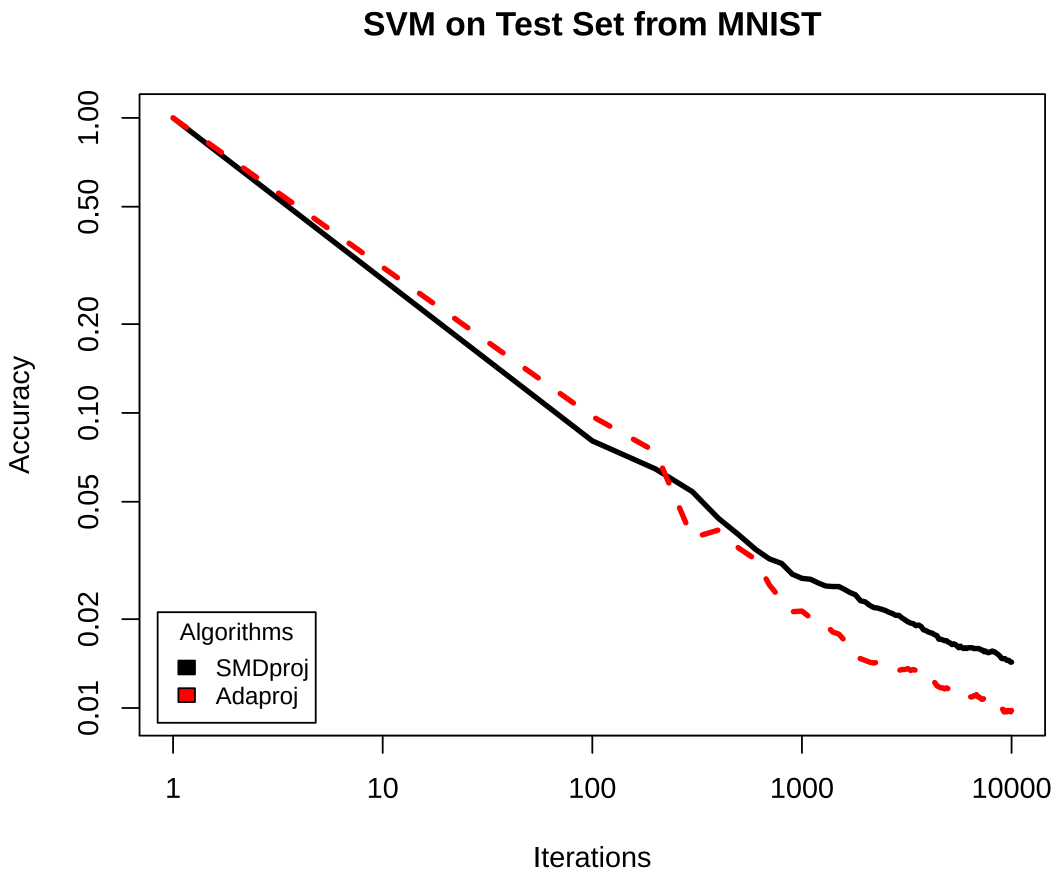
$$\mathbb{P}(a_i \neq 0) = \min\{1, ci^{-\alpha}\}$$

for some $\alpha \in (2, \infty)$ and any $1 \leq i \leq d$. Then the regret of Adagrad is low in expectation

$$\begin{aligned}
\mathbb{E}[Regret_t] &\leq 2D_\infty \sum_{i=1}^{d} \mathbb{E}\Big[\sqrt{\sum_{t=1}^{T} \nabla f_t(x_t)_i^2}\Big] \\
&\leq 2D_\infty \sum_{i=1}^{d} \sqrt{\sum_{t=1}^{T} \mathbb{E}\Big[\nabla f_t(x_t)_i^2\Big]} \\
&\leq 2D_\infty G_\infty \sqrt{T} \sum_{i=1}^{d} \sqrt{\mathbb{P}(a_i \neq 0)} \\
&\leq 2D_\infty G_\infty \sqrt{cT} \sum_{i=1}^{d} i^{-\alpha/2} \\
&\leq 2D_\infty \sqrt{cT}(1 + \log d)\,,
\end{aligned}$$

where $\|[\nabla f_t(x_t)\|_\infty \leq G_\infty$ for any $1 \leq t \leq T$. Taking advantage of the sparsity thanks to the adaptivity of AdaGrad one turns a $G \approx \sqrt{d}$ regret bound of OGD into a $\log d$ one.

Implemented on MNIST, Adagrad clearly takes advantage of the sparsity in the pixels of the the handwritten digits in a better way than the projection of the ℓ^1-ball. It is due to the fact that Adagrad learns the sparsity via the gradients whereas the radius of the ℓ^1-ball (or equivalently the regularization parameter in the dual LASSO problem) is fixed a priori (here arbitrarily to $z = 100$).



3.3.4 BOA

BOA is a multiple learning rate version of EG. The idea is to combine the adaptivity of the simplicity of the gradients as in Adagrad together with the use of the geometry of the convex set $\mathcal{K} = \Lambda$ via the negative entropy regularization function. Note that it is necessary to add a quadratic compensation to the gradient in the exponential weights in order to get theoretical guarantees.

Implemented on MNIST the rate seems to be faster than for Adaproj. It s due to the introduction of the quadratic compensation that can be seen as an estimation of the noise level. Then BOA seeks at achieving a good bias-variance tradeoff in stochastic environment. Note that an even better bias-variance tradeoff will be achieved by Adam thanks to momentum.

Algorithm 18: SBOA+/- for linear SVM, Wintenberger (2017)

Parameters: Epoch T, radius $z > 0$.

Initialization: Initial point $x_1 = 0$, weights $w = 1/(2d)\,\mathbb{1}$ and $\eta_0 = \mathbb{1} \in \mathbb{R}^{2d}$.

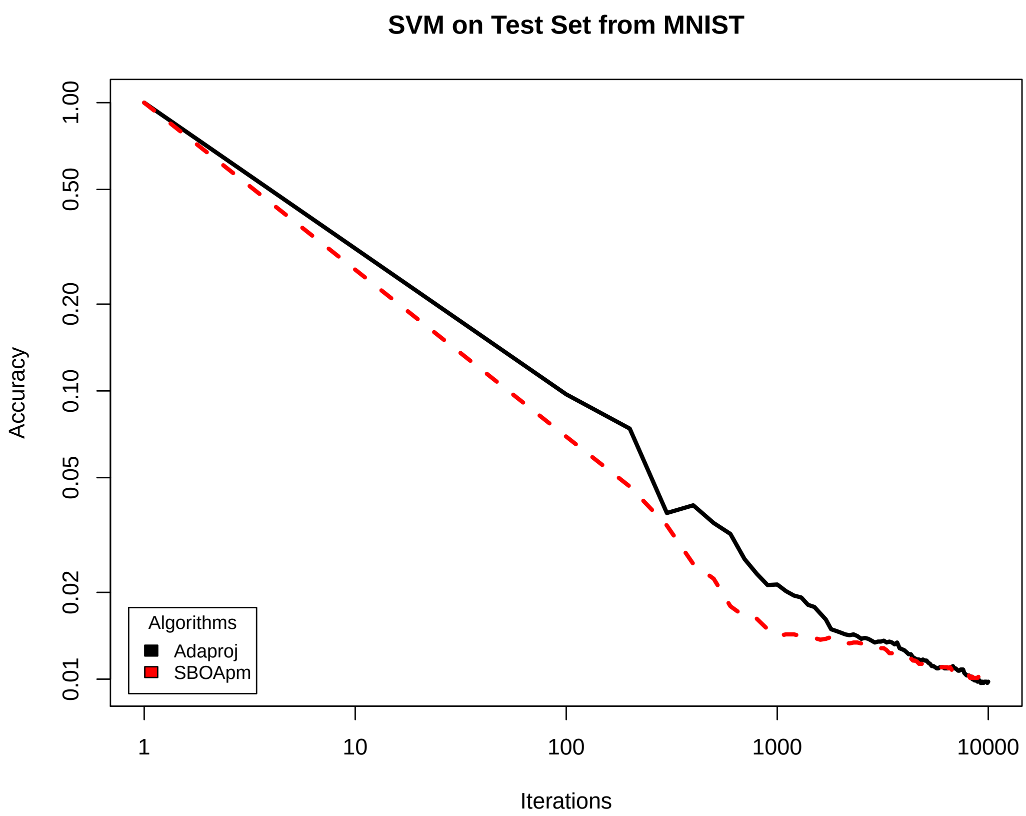
Sample uniformly iid: $(I_t)_{1\le t\le T}$ from $\{1 \le i \le n\}$

For each iteration $t = 1, \dots, T$:

Iteration: Update

$$\begin{aligned}
\overline{\nabla\ell}_t &= w_t^T \pm \nabla\ell_{a_{I_t},b_{I_t}}(x_t)\\
\theta_{t+1} &= \theta_t - \pm\nabla\ell_{a_{I_t},b_{I_t}}(x_t) - \eta_{t-1}(\pm\nabla\ell_{a_{I_t},b_{I_t}}(x_t) - \overline{\nabla\ell}_t)^2\,,\\
\eta_t &= \sqrt{\eta_{t-1}^2/(1+\eta_{t-1}^2(\pm\nabla\ell_{a_{I_t},b_{I_t}}(x_t) - \overline{\nabla\ell}_t)^2)}\\
w_{t+1} &= \frac{\eta_t \exp(\eta_t\theta_{t+1})}{\sum_{i=1}^{2d}\eta_{t,i}\exp(\eta_t\theta_{t+1,i})}\,,\\
x_{t+1,i} &= z(w_{t+1,i} - w_{t+1,i+d})\,, \qquad 1 \le i \le d\,.
\end{aligned}$$

Return: $\overline{x}_{T+1} = \dfrac{1}{T+1}\displaystyle\sum_{t=1}^{T+1} x_t$



Chapter 4

Accelerated OCO algorithms

4.1 Momentum

A variant of AdaGrad with **momentum** is the popular Adam, Kingma and Ba (2014). Momentum has been introduced first by Nesterov (1983) as an acceleration scheme in the CO method. Initially the momentum step was applied to the iterate x_t of the algorithm. It can also accelerate OCO algorithms in practice in the stochastic OCO setting. A natural way of introducing momentum in SGD methods is directly on the successive gradients.

Recall that SGD is based on an unbiased noisy version of the CO problem $(f, \mathcal{K})$ denoted $\widehat{\nabla f}$. In such a setting $\widehat{\nabla f}(x_t)$ is an unbiased estimator of $\nabla f(x_t)$ since $\widehat{\nabla f}$ is a noisy version of ∇f with mean ∇f. One defines a better estimator of this mean by averaging. However the objective $\nabla f(x_t)$ is evolving through time t.

The momentum estimator m_t is an iterative way of approximating $\nabla f(x_t)$ called Exponential Moving Average, namely

$$m_t = \beta m_{t-1} + (1-\beta)\widehat{\nabla f}(x_t) \qquad \Longleftrightarrow \qquad m_t = (1-\beta)\sum_{j=0}^{t-1} \beta^j \widehat{\nabla f}(x_{t-j})\,.$$

Note that since $(1-\beta)\sum_{j=0}^{t-1}\beta^j = 1-\beta^t \neq 1$ on should debiased the momentum m_t by dividing it by $1-\beta^t$. If successive gradients are pointeing to different direction (as they are noisy) then the erratic directions could averaged and canceled. However if β is too large then past gradients are taken into account in the momentum which might introduce a bias in the estimation of the last gradient $\nabla f(x_t)$.

Indeed the variation of each coordinate $m_{t,i}$

$$v_{t,i} = (1-\beta)\sum_{j=0}^{t-1} \beta^j (\widehat{\nabla f}(x_{t-j})_i - (1-\beta)\sum_{j=0}^{t-1} \beta^j \widehat{\nabla f}(x_{t-j})_i)^2$$

satisfies the recursive relation

$$v_{t,i} = (1-\beta)(v_{t-1,i} + \beta(\widehat{\nabla f}(x_{t-1})_i - m_{t-1,i})^2)$$

so that it is comparable to $(1-\beta)(\widehat{\nabla f}(x_t)_i - m_{t-1,i})^2$ in a stationary regime when $v_{t,i} \approx v_{t-1,i}$. The variation of noisy $\widehat{\nabla f}$ might then be reduced from a factor $(1-\beta)$ thanks to momentum.

The choice of β is tricky since the larger β the smaller the variations of m_t but the longer the memory of the momentum. If β is well chosen, a momentum on the stochastic gradients might increase the accuracy of the estimation of the true gradient by reducing the variance and thus accelerating the convergence without deteriorating the stability of the algorithm.

Remark. *There exists many other ways of accelerating SGD by improving the estimate of the gradient $\nabla f(x_t)$.*

One way is to use a streaming minibatch which can be seen as moving averages, i.e. considering

$$m_t = \frac{1}{k}\sum_{i=1}^{k} \widehat{\nabla f}(x_{t-k+1})$$

in the recursion instead of the instantaneous noisy gradient $\widehat{\nabla f}(x_t)$. A large k decreases the variation of the mini-batch of order k^{-1} but may increase the bias.

A usual mini-batch scheme

$$m_t = \frac{1}{k}\sum_{i=1}^{k} \widehat{\nabla f}_k(x_{t-1})$$

decreases the variance without deteriorating the bias. However it increase the complexity of each gradient step by a factor k.

The novelty to Adam is to apply a momentum to the squares of the gradient as well. The motivation comes from the multiple learning rates of Adagrad

$$\frac{1}{\sqrt{t}} \frac{1}{\sqrt{\frac{1}{t}\sum_{k=1}^{t} \widehat{\nabla f}_t(x_t)_i^2}}\,, \qquad 1 \leq i \leq d\,,$$

and to interpret it as the multiplication of the learning rate $\dfrac{1}{\sqrt{t}}$ together with the inverse of an estimator of the noise level

$$\sqrt{\frac{1}{t}\sum_{k=1}^{t} \widehat{\nabla f}_t(x_t)_i^2} \approx \sqrt{\mathbb{E}[\widehat{\nabla f}_t(x_t)_i^2]}\,.$$

In this interpretation the noise level of the approximation $\widehat{\nabla f}$ is measured according to a moment of order 2. The same reasoning as before shows that a momentum might improve the estimation of the second order moments of the noisy gradients for a well chosen coefficient β.

Algorithm 19: Adam for linear SVM, Kingma and Ba (2014)

Parameters: Epoch T, radius $z > 0$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$.
Initialization: Initial point $x_1 = y_1 = 0$ and $S_0 = 0$ (or $= \delta \mathbb{I}$ small).
Sample uniformly iid: $(I_t)_{1\le t\le T}$ from $\{1 \le i \le n\}$
For each iteration $t = 1, \dots, T$:
Iteration: Update

$$\begin{aligned}
\eta_t &= 1/\sqrt{t}\,,\\
m_t &= \beta_1 m_{t-1} + (1-\beta_1)\nabla \ell_{a_{I_t},b_{I_t}}(x_t)\\
S_t &= \beta_2 S_{t-1} + (1-\beta_2)\nabla \ell_{a_{I_t},b_{I_t}}(x_t)^2\\
D_t &= \mathrm{Diag}(\sqrt{S_t})\\
y_{t+1} &= x_t - \eta_t D_t^{-1} m_t\,,\\
x_{t+1} &= \arg\min_{x\in B_1(z)} \|x - y_{t+1}\|_{D_t}^2\,, \qquad 1 \le i \le d\,.
\end{aligned}$$

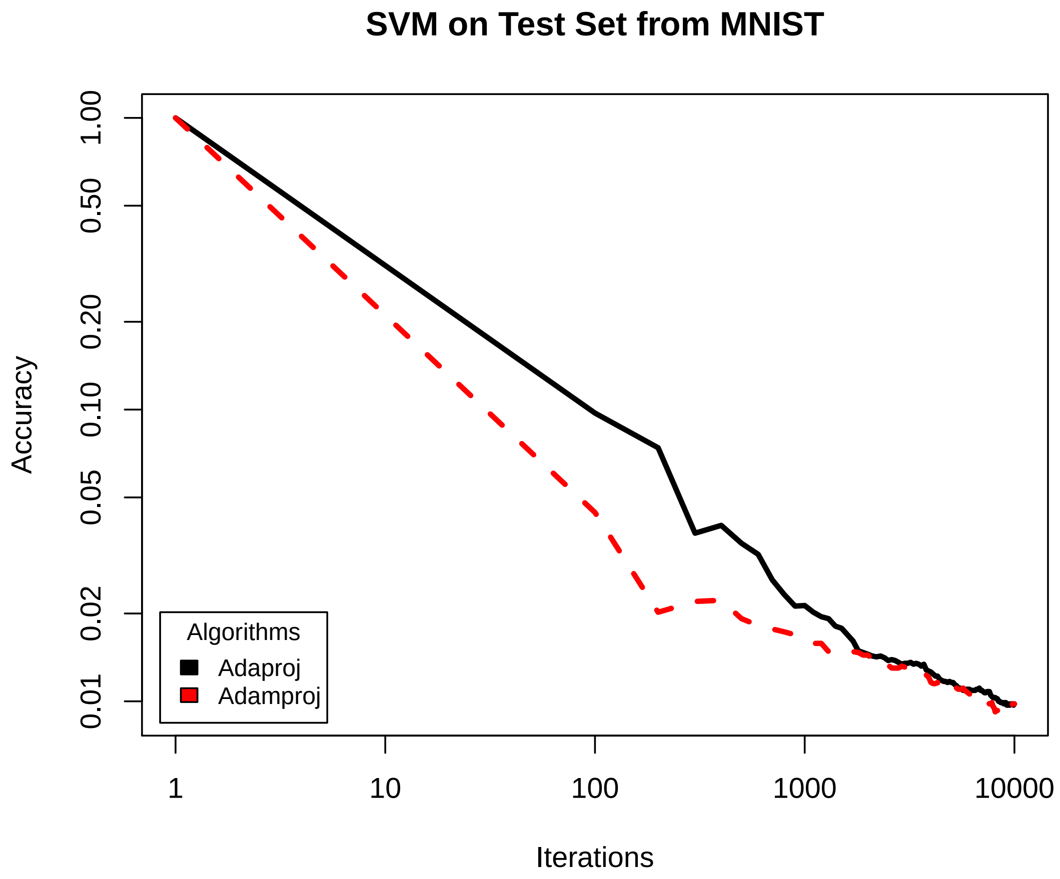
Return: $\overline{x}_{T+1} = \dfrac{1}{T+1}\displaystyle\sum_{t=1}^{T+1} x_t$

In practice one chooses $\beta_2 >>> \beta_1$ as the noise level of the gradient directions are thought as more stable than the direction of the gradient. The best bias-variance trade-off is then achieved for $\beta_2 \approx 1$, taking into account a large number of past squared gradients.

Remark. *Acceleration of algorithms might be introduced carefully in order to control the noise stability. Momentum on gradients does not accelerate nor deteriorate OCO algorithms in theory. Together with momentum on the squared gradients as in Adam it accelerates it in stochastic OCO settings.*

In an adversarial setting with $\mathcal{K} = [-1, 1]$ and $\nabla \ell_{a_{I_t},b_{I_t}}(x_t) = C > 2$ for $t = 3k - 1$, $k \ge 1$, and $\nabla \ell_{a_{I_t},b_{I_t}}(x_t) = -1$ otherwise Adam does not converge since $\eta_t D_t^{-1}$ is not a decreasing sequence. Worst Adam converges to 1 whereas $x^* = -1$ for $\beta_2 = (1 + C^2)^{-1}$. Thus β_2 should be taken very large but as soon as $\sqrt{\beta_2} > \beta_1$ the regret of Adam is linear, see Reddi et al. (2019).

However Adam is very efficient in practice when applied to MNIST dataset. Moreover it works also extremely well in deep learning training, beyond the convex loss function setting (flat high-dimensional problems) where the objective is not necessarily to converge. It explains the success of this algorithm and its variants in deep learning.



4.2 Online Newton Step (ONS)

Despite excellent practical acceleration observed in practice in Adam, it is impossible to accelerate OCO algorithms without extra assumptions on the loss function. We have already seen that for α strongly convex loss functions then SGD with learning rates $\eta_t = 1/(\alpha t)$ achieves a logarithmic regret.

4.2.1 Exp-concave functions

In many practical situation the strong convexity assumption is too strong.

Example 5. *Consider the linear SVM setting in MNIST with pixels $a \in \mathbb{R}^d$ and label $b \in \{-1, 1\}$ together with the square loss as a relaxation of the 0/1 loss*

$$\widetilde{\ell}_{a,b}(x) = (b - x^T a)^2 \,.$$

Not that despite the square function $y \to y^2$ is 2-strongly convex, it is not always the case of $\widetilde{\ell}$. Indeed one computes

$$\nabla \widetilde{\ell}_{a,b}(x) = 2(b - x^T a)a \qquad \text{and} \qquad \nabla^2 \widetilde{\ell}_{a,b}(x) = 2aa^T \,,$$

that is convex iff

$$2aa^T \succeq \alpha I_d.$$

We get a contradiction since aa^T is a rank one matrix that cannot be invertible (except for $d = 1$). Thus $\widetilde{\ell}_{a,b}$ is not strongly convex.

We have to introduce the notion of exp-concavity

Definition 14. *A convex function $f : \mathcal{K} \mapsto \mathbb{R}$ is exp-concave on iff the function $g(x) = \exp(-\mu f(x))$ is concave.*

We have the following property

Lemma. *A twice differentiable function $f : \mathcal{K} \mapsto \mathbb{R}$ is μ-exp-concave iff*

$$\nabla^2 f(x) \succeq \mu \nabla f(x) \nabla f(x)^T , \qquad x \in \mathcal{K} .$$

Proof. We have $g(x)$ twice differentiable that is concave iff $\nabla^2 g(x) \preceq 0$, $x \in \mathcal{K}$. We compute

$$\begin{aligned} \nabla g(x) &= -\mu \nabla f(x) \exp(-\mu f(x)) \\ \nabla^2 g(x) &= (\mu^2 \nabla f(x) \nabla f(x)^T - \mu \nabla^2 f(x)) \exp(-\mu f(x)) \end{aligned}$$

and $\nabla^2 g(x) \preceq 0$ iff

$$\begin{aligned} \mu^2 \nabla f(x) \nabla f(x)^T - \mu \nabla^2 f(x) &\preceq 0 \\ \mu \nabla f(x) \nabla f(x)^T - \mu \nabla^2 f(x) &\preceq \nabla^2 f(x) \end{aligned}$$

Notice that the rank one matrix $\nabla f(x) \nabla f(x)^T \succeq 0$ by construction so that $\nabla^2 f(x) \succeq 0$ and f is convex. □

Exercise 6. *A α-strongly convex G-Lipschitz function is α/G^2-exp-concave.*

However there are many examples of exp-concave functions that are not strongly convex.

Example 6. *In Example 5 we have*

$$\nabla \widetilde{\ell}_{a,b}(x) \nabla \widetilde{\ell}_{a,b}(x)^T = 4(b - x^T a)^2 aa^T \preceq \frac{2}{\mu} aa^T$$

iff $\max_{x \in \mathcal{K}} 2(b - x^T a)^2 \le \frac{1}{\mu}$. In particular μ is proportional to the amplitude of the square loss and is independent of the dimension of the OCO problem.

We would need the following stronger property of exp-concavity, valid in the usual bounded setting.

Proposition. *Let $f : \mathcal{K} \mapsto \mathbb{R}$ be μ-exp-concave, D be the diameter of $\mathcal{K}$ and $\max_{x \in \mathcal{K}} \|\nabla f(x)\| \le G$ for some $G > 0$ as usual. Then*

$$f(y) \ge f(x) + \nabla f(x)^T (y - x)| + \frac{\gamma}{2} (y - x)^T \nabla f(x) \nabla f(x)^T (y - x) , \qquad x, y \in \mathcal{K} ,$$

with $\gamma \le \frac{1}{2} \min(\frac{1}{4GD}, \mu)$.

Proof. As $2\gamma \le \mu$ then $h(x) = \exp(-2\gamma f(x))$ is a concave function and

$$h(y) \le h(x) + \nabla h(x)^T (y - x) .$$

We compute $\nabla h(x) = -2\gamma \nabla f(x) \exp(-2\gamma f(x))$ so that by plugging in

$$\begin{aligned}\exp(-2\gamma f(y)) &\le \exp(-2\gamma f(x)) - \exp(-2\gamma f(x)) 2\gamma f(x)^T (y-x) \\ &\le \exp(-2\gamma f(x))(1 - 2\gamma f(x)^T (y-x)) \,.\end{aligned}$$

Thus we get

$$f(y) \ge f(x) - \frac{1}{2\gamma} \log(1 - 2\gamma \nabla f(x)^T (y-x)) \,.$$

Using the boundedness $|2\gamma \nabla f(x)^T (y-x)| \le 2\gamma G D \le 1/4$ and that

$$-\log(1-z) \ge z + \frac{1}{4} z^2 \qquad |z| \le 1/4 \,,$$

we obtain

$$f(y) \ge f(x) + \nabla f(x)^T (y-x) + \frac{1}{8\gamma} (2\gamma \nabla f(x)^T (y-x))^2$$

and the desired result follows. $\square$

4.2.2 Online Newton Step (ONS) algorithm

The ONS is an OCO adaptation of the Newton-Raphson step from CO problems

$$x_{t+1} = x_t - \eta H_t^{-1} \nabla f(x_t)$$

where $H_t = \nabla^2 f(x_t)$ and $\eta > 0$.

In the OCO setting, one can advantageously replace H_t by an approximation as a function of the gradients only, namely $\frac{1}{t} \sum_{k=1}^t \nabla f_k(x_k) \nabla f_k(x_k)^T$, under weaker assumption, namely exp-concavity. We obtain the ONS algorithm

Algorithm 20: Online Newton Step, Hazan and Kale (2011)

Initialization: $\gamma > 0$ and $\epsilon > 0$.
Initialization: Initial prediction $x_1 \in \mathcal{K}$ and $A_0 = \epsilon I_d$.
Predict: x_t
Incur: $f_t(x_t)$
Observe: $\nabla f_t(x_t) \in \mathbb{R}^d$
Recursion: Update

$$\begin{aligned} A_t &= A_{t-1} + \nabla f_t(x_t) \nabla f(x_t)^T \\ y_{t+1} &= x_t - \frac{1}{\gamma} A_t^{-1} \nabla f_t(x_t) \,, \\ x_{t+1} &= \arg\min_{x \in \mathcal{K}} \|x - y_{t+1}\|_{A_t}^2 \,, \qquad 1 \le i \le d \,. \end{aligned}$$

Remark. *There exists some resemblance with Adagrad in the sense that it can be seen as an agile OMD with adaptive regularization function $R_t(x) = \frac{1}{2}\|x - x_1\|_{A_t}$. A major difference is that the diagonal of A_t in the regularization function R_t are equal to the square of the weights in Adagrad, namely D_t^2, of the form*

$$\frac{1}{t} \frac{1}{t^{-1} \sum_{k=1}^t \nabla f_k(x_k)_i^2} \,.$$

Theorem. *ONS with f_t μ-exp-concave and $\gamma = \frac{1}{2}\min\{\frac{1}{4GD}, \mu\}$ and $\epsilon = 1/(\gamma D)^2$ achieves a regret*

$$\mathit{Regret}_T \le 2\Big(\frac{1}{\mu} + 4GD\Big) d \log T\,, \qquad T \ge 3\,.$$

Proof. We improve the gradient trick using Proposition 4.2.1

$$\sum_{t=1}^T f(x_t) - f(x^*) \le \sum_{t=1}^T \nabla f_t(x_t)^T (x_t - x^*) - \frac{\gamma}{2}\sum_{t=1}^T \|x_t - x^*\|^2_{\nabla f_t(x_t)\nabla f_t(x_t)^T}\,.$$

Using the recursion and the Pythagorean theorem (still valid) we get

$$\begin{aligned}
\|x_{t+1} - x^*\|^2_{A_t} &\le \|y_{t+1} - x^*\|^2_{A_t} \\
&\le (y_{t+1} - x^*)^T A_t (y_{t+1} - x^*)^T \\
&\le \|x_t - x^*\|^2_{A_t} - \frac{1}{\gamma^2}\|\nabla f_t(x_t)\|^2_{A_t^{-1}} - \frac{2}{\gamma}\nabla f_t(x_t)^T (x_t - x^*)\,.
\end{aligned}$$

Thus we get

$$\begin{aligned}
\sum_{t=1}^T \nabla f_t(x_t)^T (x_t - x^*) \le& \frac{\gamma}{2}\sum_{t=1}^T (\|x_t - x^*\|^2_{A_t} - \|x_{t+1} - x^*\|^2_{A_t}) + \frac{2}{\gamma}\sum_{t=1}^T \|\nabla f_t(x_t)\|^2_{A_t^{-1}} \\
\le& \frac{\gamma}{2}\Big(\sum_{t=2}^T (\|x_t - x^*\|^2_{A_t} - \|x_t - x^*\|^2_{A_{t-1}}) + \|x_1 - x^*\|^2_{A_1}\Big) \\
&+ \frac{1}{2\gamma}\sum_{t=1}^T \|\nabla f_t(x_t)\|^2_{A_t^{-1}} \\
\le& \frac{\gamma}{2}\Big(\sum_{t=2}^T \|x_t - x^*\|^2_{\nabla f_t(x_t)\nabla f_t(x_t)^T} + \|x_1 - x^*\|^2_{A_1}\Big) \\
&+ \frac{1}{2\gamma}\sum_{t=1}^T \|\nabla f_t(x_t)\|^2_{A_t^{-1}}
\end{aligned}$$

.

We immediately derive

$$\mathit{Regret}_T \le \frac{\gamma}{2}\|x_1 - x^*\|^2_{A_1 - \nabla f_1(x_1)\nabla f_1(x_1)^T} + \frac{1}{2\gamma}\sum_{t=1}^T \|\nabla f_t(x_t)\|^2_{A_t^{-1}}\,.$$

The first term in the upper bound is equal to $\epsilon\|x_1 - x^*\|^2 \le 1/\gamma^2$. We upper-bound the second term such as

$$\begin{aligned}
\sum_{t=1}^T \|\nabla f_t(x_t)\|^2_{A_t^{-1}} &= \sum_{t=1}^T \mathrm{Tr}(A_t^{-1}\nabla f_t(x_t)\nabla f_t(x_t)^T) \\
&\le \sum_{t=1}^T \mathrm{Tr}(A_t^{-1}(A_t - A_{t-1})) \\
&\le \sum_{t=1}^T \log(|A_t|/|A_{t-1}|) \\
&\le \log(|A_T|/|A_0|)\,.
\end{aligned}$$

Since $A_T = \sum_{t=1}^T \nabla f_t(x_t)\nabla f_t(x_t)^T + \epsilon I_d$ then $|A_T| \le (TG^2+\epsilon)^d$ and

$$|A_T|/|A_0| \le (1+TG^2/\epsilon)^d \le (1+TG^2\gamma^2D^2)^d \le (1+T/8)^d \le T^d$$

for any $T \ge 2$. We get the bound

$$Regret_T \le \frac{1}{2\gamma}(1+d\log T) \le (4GD+1/\mu)(1+d\log T) \le 2(4GD+1/\mu)d\log T$$

since $1/\gamma = 2\max(4GD, 1/\mu) \le 2(4GD+1/\mu)$ and $T \ge 3$. □

Each recursion of the ONS would require to invert a large $\times d$ matrix A_t. Actually one should avoid such inversion by considering the Sherman-Morrisson formula which provides the recursion on A_t^{-1}

$$A_t^{-1} = (A_{t-1} + \nabla f_t(x_t)\nabla f(x_t)^T)^{-1} = A_{t-1}^{-1} - \frac{A_{t-1}^{-1}\nabla f_t(x_t)\nabla f_t(x_t)^T A_{t-1}}{1+\nabla f_t(x_t)^T A_{t-1}\nabla f_t(x_t)}\,.$$

Thus the recursion on A_t should be accompanied with one on A_t^{-1}. Moreover the projection $\arg\min_{x\in\mathcal{K}} \|x - y_{t+1}\|^2_{A_t}$ is not explicit for A_t non diagonal (up to my knowledge). One could approximate it with $\arg\min_{x\in\mathcal{K}} \|x-y_{t+1}\|^2_{\mathrm{Diag}(A_t)}$ where $\mathrm{Diag}(A_t)$ is the diagonal matrix extracted from A_t. Then the total cost of one recursion is $O(d^2)$.

One can implement the ONS on MNIST

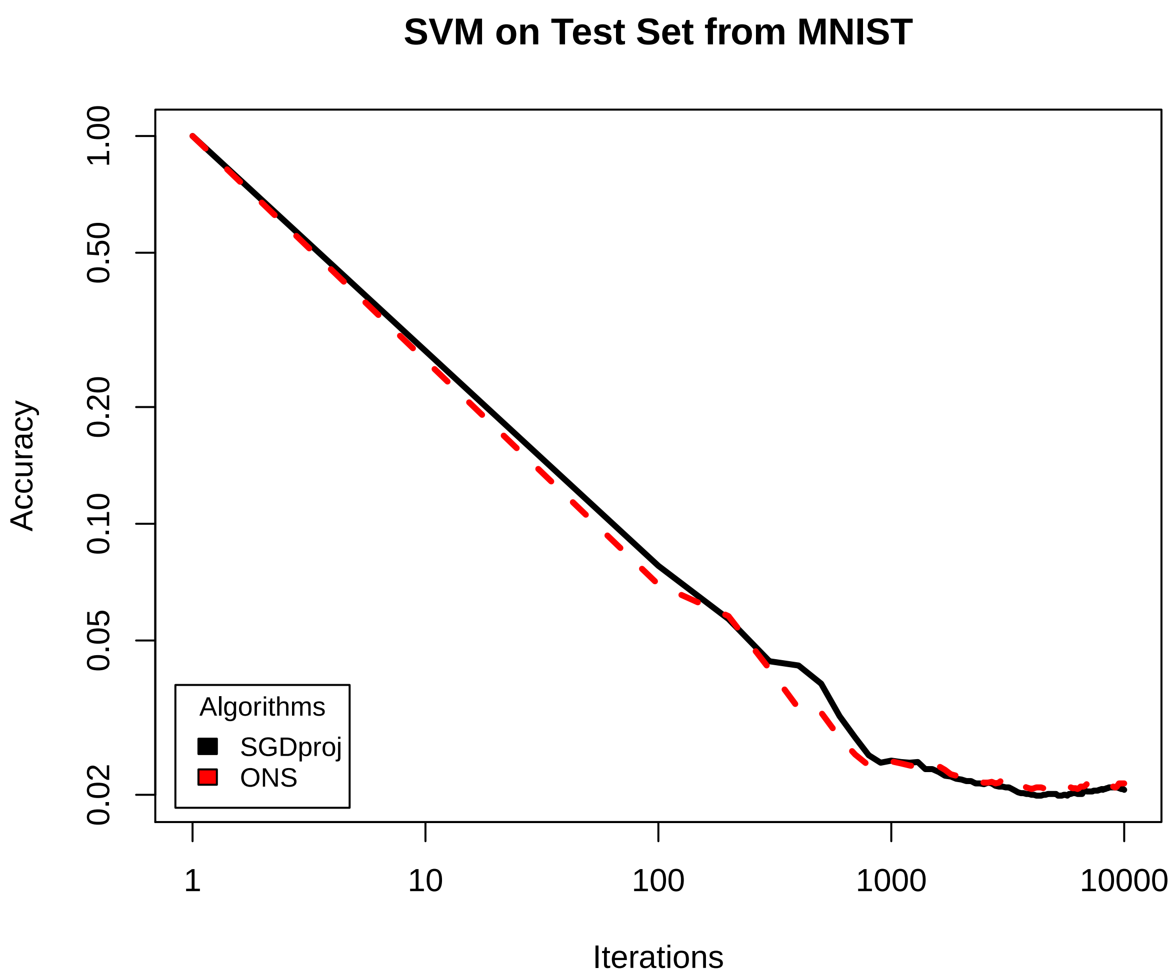
Algorithm 21: ONS for linear SVM

Parameters: Epoch T, radius $z>0$, regularization parameter $\lambda>0$ and $\gamma>0$.
Initialization: Initial point $x_1 = y_1 = 0$, $A_0 = 1/\gamma^2 I_d$ and $A_0^{-1} = \gamma^2 I_d$.
Sample uniformly iid: $(I_t)_{1\le t\le T}$ from $\{1 \le i \le n\}$
For each iteration $t = 1,\dots,T$:
Iteration: Update

$$\begin{aligned}
\nabla_t &= \nabla \ell_{a_{I_t},b_{I_t}}(x_t) + \lambda x_t\\
A_t &= A_{t-1} + \nabla_t\nabla_t^T\\
A_t^{-1} &= A_{t-1}^{-1} - \frac{A_{t-1}^{-1}\nabla_t\nabla_t^T A_{t-1}^{-1}}{1+\nabla_t^T A_{t-1}^{-1}\nabla_t}\\
y_{t+1} &= x_t - \frac{1}{\gamma}A_t^{-1}\nabla_t\,,\\
x_{t+1} &= \arg\min_{x\in B_1(z)} \|x-y_{t+1}\|^2_{A_t}\,, \qquad 1 \le i \le d\,.
\end{aligned}$$

Return: $\overline{x}_{T+1} = \dfrac{1}{T+1}\sum_{t=1}^{T+1} x_t$

It slightly improves SGD but at the price of a recursion step at $O(d^2)$. Actually it is very complicated to calibrate and the choice of γ is tricky. The performances are very similar to the regularized SGD, both with regularization parameter taken as $\lambda = 1/3$. However the loss in speed is a relative factor of 20.



4.2.3 Natural gradient and EKF

For statistical settings such as binary classification (a_t, b_t) iid, ONS might be useful for statistical purposes. Indeed one updates a matrix A_t^{-1} such that

$$\frac{1}{T}\sum_{t=1}^{T} A_t^{-1} \approx \mathbb{E}[\nabla \ell_{(a,b)}(x^*)\nabla \ell_{(a,b)}(x^*)^T]^{-1}$$

for T sufficiently large so that $\nabla f_T(x_T) \approx \nabla f_T(x^*)$. One recognize the variance of the vector score associated to a model

$$(a_t, b_t) \sim c\exp(-\ell_{a,b}(x^*))\,,$$

where $c > 0$ is the normalizing constant $c = \int_{\mathbb{R}^d} \exp(-\ell_{a,1}(x^*))da + \int_{\mathbb{R}^d} \exp(-\ell_{a,-1}(x^*))da$. For regular models we can have the identity

$$\mathbb{E}[\nabla \ell_{(a,b)}(x^*)\nabla \ell_{(a,b)}(x^*)^T] = \mathbb{E}[\nabla^2 \ell_{(a,b)}(x^*)]\,.$$

In such a case $\overline{x}_{T+1}$ might be a good approximation of the maximum likelihood estimator and $\mathbb{E}[\nabla \ell_{(a,b)}(x^*)\nabla \ell_{(a,b)}(x^*)^T]^{-1}$ its associated asymptotic variance

$$\sqrt{T}(\overline{x}_{T+1} - x^*) \sim \mathcal{N}(0, \mathbb{E}[\nabla \ell_{(a,b)}(x^*)\nabla \ell_{(a,b)}(x^*)^T]^{-1})\,.$$

That ONS provides an estimator of the asymptotic variance $\frac{1}{T}\sum_{t=1}^T A_t^{-1}$ is very useful for instance for significancy testing.

The Online Natural Gradient (or Stochastic Newton) approach replaces $\nabla_t \nabla_t^T$ in the recursion on A_t^{-1} with the better (optimistic) approximation of the variance of the score

$$\mathbb{E}_{(a,b)\sim c\exp(-\ell_{a,b}(x_t))}\,[\nabla \ell_{(a,b)}(x_t)\nabla \ell_{(a,b)}(x_t)^T].$$

Such quantity is explicit in exponential family under its natural parametrization and then the Stochastic Newton algorithm coincides with the static Extended Kalman Filter. Thus we are forced to consider the logistic model associated with the loss function

$$\frac{(b+1)x^T a}{2} - \log(1 + e^{x^T a})\,.$$

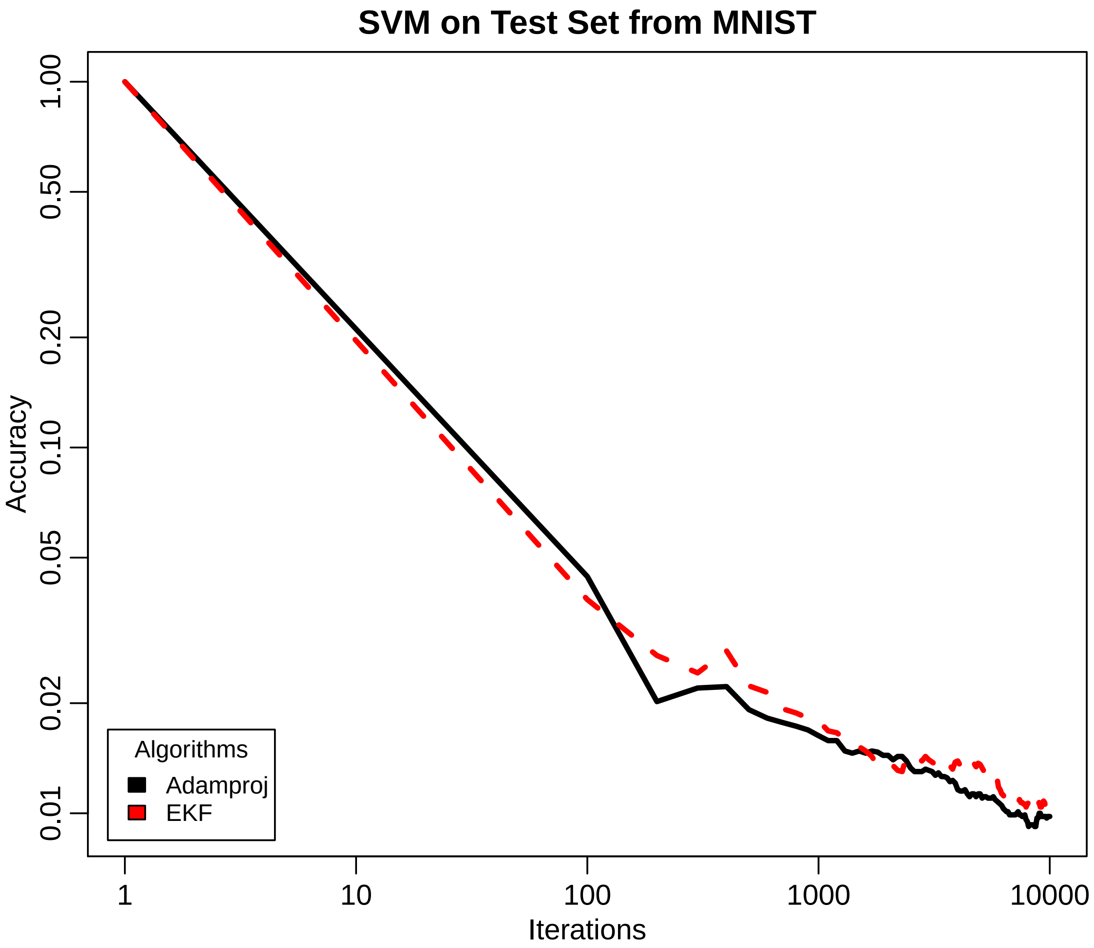
Algorithm 22: EKF for linear SVM, Fahrmeir (1992)

Parameters: Epoch T.
Initialization: Initial point $x_1 = 0$ and $P_0 = I_d$.
Sample uniformly iid: $(I_t)_{1\le t\le T}$ from $\{1 \le i \le n\}$
For each iteration $t = 1, \ldots, T$:
Iteration: Update

$$\begin{aligned}
\hat{p}_t &= \frac{e^{b_{I_t} x_t^T a_{I_t}}}{1 + e^{b_{I_t} x^T a_{I_t}}} \\
P_t &= P_{t-1} - \frac{P_{t-1} a_{I_t} a_{I_t}^T P_{t-1}}{1/(\hat{p}_t(1-\hat{p}_t)) + a_{I_t}^T P_{t-1} a_{I_t}} \\
x_{t+1} &= x_t + b_{I_t}(1-\hat{p}_t) P_t a_{I_t}\,.
\end{aligned}$$

Return: x_{T+1}

Note that the recursion depends heavily on the parameter $\hat{p}_t$ which is the sigmoid function applied to $b_{I_t} x_t^T a_{I_t}$ and corresponds to the probability of observing b_{I_t} from a_{I_t} in the logistic model driven by x_t. Thus one can also EKF as an explicit version of a Recursive Bayesian algorithm. In such approach there is no need of the projection nor the averaging step. The obtained accuracy is very close to the one of Adam and the relative loss of speed is only of a relative factor of 10. The gain of relative speed of 1/2 compared with ONS is due to the use of a unique automatically fine tune matrix P rather than the use of two more degenerate matrices A and A^{-1}. Optimal regret bounds have been derived for EKF in such stochastic setting in De Vilmarest and Wintenberger (2020).


SVM on Test Set from MNIST
Accuracy
1.00
0.50
0.20
0.10
0.05
0.02
0.01
Iterations
1
10
100
1000
10000
Algorithms
Adamproj
EKF

Chapter 5

Exploration

5.1 Bandit Convex Optimization

In extremely high dimension, even the inquiry of one gradient $\nabla f_t(x_t) \in \mathbb{R}^d$ might be too costly. A strategy from consists in inquiring only one direction of the gradient $\nabla f_t(x_t)_{i_t} \in \mathbb{R}$ at each step. Since the full instantaneous gradient $\nabla f_t(x_t)$ is not observed we are in the incomplete information setting.

We have a new setting, called the Multi-Armed Bandit (MAB) setting, where similarly as in the Expert Advice setting at each round the algorithm assigns confident weights $x_t \in \Lambda$, pick randomly an expert $i_t \sim x_t$ and incurs the averaged loss $\mathbb{E}_{x_t}[\ell_t]$. However the information of the algorithm is now limited to the loss ℓ_{t,i_t} but the regret is unchanged

$$\mathbb{E}[Regret_T] = \sum_{t=1}^{T} \mathbb{E}_{x_t}[\ell_t] - \min_{1\le i\le d} \sum_{t=1}^{T} \ell_{t,i}\,.$$

At each round the algorithm can either explore and pick a new expert that has been never played or exploit and pick an already chosen expert in order to learn its performances. This is an exploration-exploitation trade-off.

The same reduction to the OCO setting, combined with the gradient trick, as for EA applies here and one can equivalently minimizes

$$\mathbb{E}[Regret_T] = \mathbb{E}\Big[\sum_{t=1}^{T} f_t(x_t) - \min_{x\in\Lambda} \sum_{t=1}^{T} f_t(x)\Big]$$

by considering the linearized losses $f_t(x) = \nabla f_t(x_t)^T x$, $x \in \Lambda$. However the only information used by the algorithm is now $\nabla f_t(x_t)_{i_t}$, where $i_t = 1, \dots, d$ is the direction picked randomly by the algorithm. Thus an expectation is necessary even in this setting since the incomplete information is random.

A general reduction from this BCO setting to the OCO setting is to replace $\nabla f_t(x_t)$ by the basic unbiased estimator $\widehat{\nabla f_t(x_t)} = d\nabla f_t(x_t)_{i_t} e_{i_t}$ where $\{e_i\}$ are the element of the canonical basis and i_t are iid uniform over $1, \dots, d$. Indeed then

$$\mathbb{E}[\widehat{\nabla f_t(x_t)}] = \sum_{j=1}^{d} d\nabla f_t(x_t)_j e_j \mathbb{P}(i_t = j) = \sum_{j=1}^{d} \nabla f_t(x_t)_j e_j = \nabla f_t(x_t)\,.$$

Thus BCO algorithms are given by randomized OCO algorithms that explores randomly the space at each recursion. Since any norm on $\nabla f_t(x_t)$ would be multiplied by d on $\widehat{\nabla f_t(x_t)}$, the OCO regret bounds obtained above for OMD methods turn into a BCO regret bound of the form

$$\mathbb{E}[Regret_T] = \mathbb{E}\Big[O\Big(\frac{d^2 G_R^2 T}{\eta} + \eta D_R\Big)\Big] = O\Big(\frac{d^2 G_R^2 T}{\eta} + \eta D_R\Big),$$

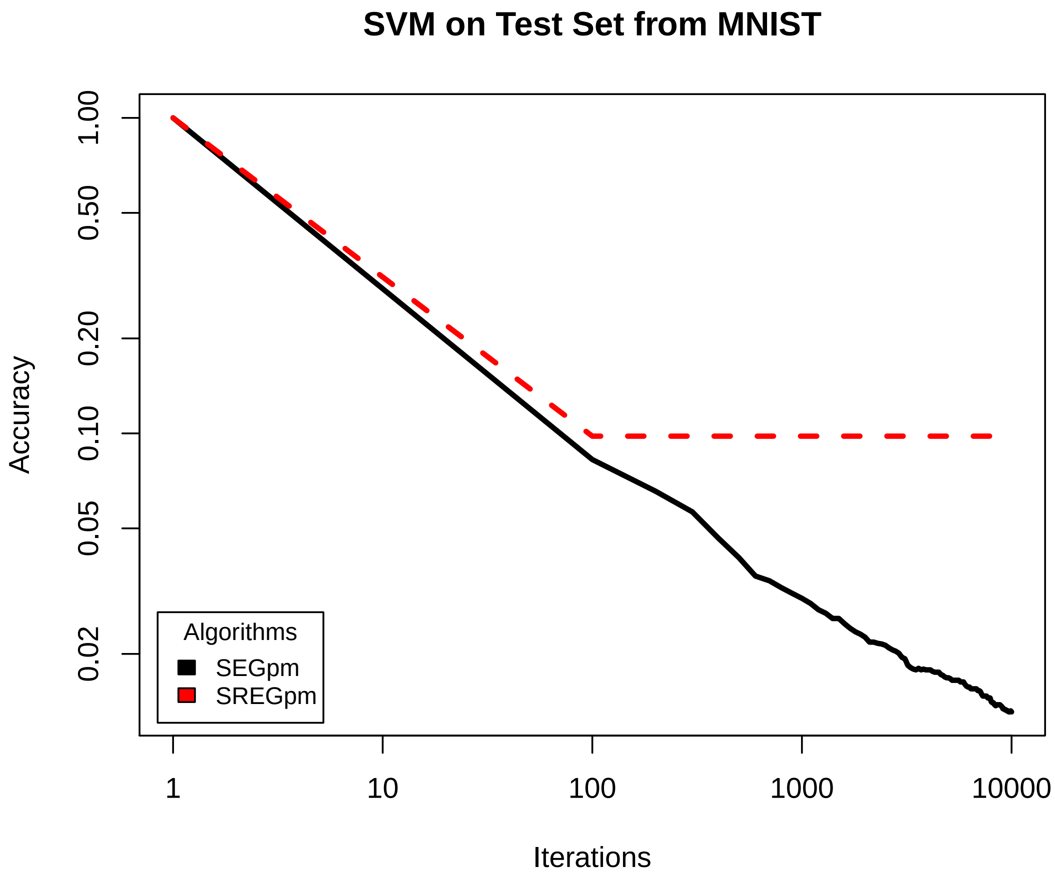
Optimizing in η, the randomized algorithm achieves a regret bound deteriorated from the OCO setting by a factor d. For instance, one gets the algorithm below called SREG

Algorithm 23: SREG+/- for linear SVM

Parameters: Epoch T, radius $z > 0$.
Initialization: Initial point $x_1 = 0$, weights $w = 1/(2d)\,\mathbb{1}$ and $\theta_1 = 0 \in \mathbb{R}^{2d}$.
Sample uniformly iid: $(I_t)_{1\le t\le T}$ from $\{1 \le i \le n\}$
For each iteration $t = 1, \dots, T$:
Pick a direction randomly: $i_t \in \{1, \dots d\}$ uniformly
Iteration: Update

$$\begin{aligned} \eta_t &= \sqrt{1/t} \\ w_{t+1,i_t} &= \exp(-\eta_t d \nabla f_t(x_t)_{i_t})\,, \\ w_{t+1,i_t+d} &= \exp(\eta_t d \nabla f_t(x_t)_{i_t}), \\ w_{t+1} &= \frac{w_t}{\sum_{i=1}^{2d} w_{t,i}} \\ x_{t+1,i} &= z(w_{t+1,i} - w_{t+1,i+d})\,, \qquad 1 \le i \le d\,. \end{aligned}$$

Return: $\overline{x}_{T+1} = \dfrac{1}{T+1}\displaystyle\sum_{t=1}^{T+1} x_t$



The algorithm is stuck at the accuracy 0.1, the accuracy of the random guess (as there is only 10% of label 1 in the dataset, digit 0). The algorithm explores at each round and does not achieve a good exploration-exploitation trade-off.

5.2 Exp3 algorithm

Back to the MAB setting, a solution for obtaining the trade-off exploration-exploitation has been provided by copying the EWA replacing ℓ_t in the exponential by some unbiased estimator. Considering $\ell_{t,i}$ for the direction which coincides with the chosen one $i = i_t$ and 0 elsewhere, we get

$$\mathbb{E}_{x_t}[\ell_{t,i}\,\mathbb{I}_{i=i_t}] = \sum_{j=1}^{d} \ell_{t,i}\,\mathbb{I}_{i=j}\mathbb{P}(i_t = j) = \ell_{t,i}x_{t,i}\,, \qquad 1 \le i \le d\,,$$

which is a biased ℓ_{t,i_t}. Indeed knowing i_t makes the strategy biased in favor of this expert rather than the other unobserved ones. In order to debiased it we consider instead

$$\widehat{\ell}_{t,i} = \frac{\ell_{t,i}}{x_{t,i}}\,\mathbb{I}_{i=i_t}\,.$$

Then we get an unbiased estimator of $\ell_{t,i}$, indeed

$$\mathbb{E}_{x_t}[\widehat{\ell}_{t,i} \mid i_t] = \sum_{j=1}^{d} \frac{\ell_{t,i}}{x_{t,i}}\,\mathbb{I}_{i=j}\mathbb{P}(i_t = j) = \sum_{j=1}^{d} \frac{\ell_{t,i}}{x_{t,i}}\,\mathbb{I}_{i=j}x_{t,j} = \ell_{t,i}\,.$$

We obtain the Exp3 (Exponential weights for Exploration and Exploitation) algorithm

Algorithm 24: Exp3 algorithm, Auer et al. (2002)

Parameters: step-size $\eta > 0$.
Initialization: Initial prediction $x = (1/d)\,\mathbb{I}$.
For each recursion $t \geq 1$:
Pick an expert prediction randomly: $i_t \sim x_t$
Incur the average loss: $\mathbb{E}_{x_t}[\ell_t]$
Observe: $\ell_{t,i_t} \in \mathbb{R}$
Recursion: Update

$$\widehat{\ell}_{t,i} = \frac{\ell_{t,i}}{x_{t,i}}\,\mathbb{I}_{i=i_t}\,, \qquad 1 \leq i \leq d$$

$$x_{t+1} = \frac{\exp(-\eta\widehat{\ell}_t)x_t}{\sum_{i=1}^d \exp(-\eta\widehat{\ell}_{t,i})x_{t,i}}\,.$$

We obtain the regret bound

Theorem. *In the MAB setting with $\|\ell_t\| \leq G$, for $\eta = G^{-1}\sqrt{2\log d/T}$ we obtain a regret bound for Exp3 as*

$$Regret_T \leq G\sqrt{2T\log d}\,.$$

Proof. We refine the previous analysis of EWA for random loss $\widehat{\ell}_t > 0$ and $\eta > 0$ in order to get the regret bound

$$Regret_T = \sum_{t=1}^T x_t^T\widehat{\ell}_t - \min_{1\leq i\leq d}\sum_{t=1}^T \widehat{\ell}_{t,i} \leq \frac{\eta}{2}\sum_{t=1}^T x_t^T\widehat{\ell}_t^2 + \frac{\log d}{\eta}\,.$$

We notice that in the OMD setting we had a regret bounded with terms of the form

$$\|\eta\nabla f_t(x_t)\|_t^{*2} = B_{R^*}(\theta_t - \eta\nabla f_t(x_t), \theta_t)\,,$$

for $R(x) = x^T\log(x)$ and $\theta_t = \nabla R(y_t)$. Then we make the important remark that the update of Exp3 (and also EWA) can be written in the agile version as in Algorithm 24, namely

$$\nabla R(y_{t+1}) = \nabla R(x_t) - \eta\nabla f_t(x_t) \text{ and } x_{t+1} = \arg\min_{x\in\mathcal{K}} B_R(x||y_{t+1})\,.$$

Thus one can replace θ_t with $\nabla R(x_t)$. As in our setting we have $\nabla f_t(x_t) = \widehat{\ell}_{t,i}$, we get

$$\begin{aligned} B_{R^*}(\theta_t - \eta\widehat{\ell}_{t,i}, \theta_t) =& B_{R^*}(\nabla R(x_t) - \eta\widehat{\ell}_{t,i}, \nabla R(x_t)) \\ =& R^*(\nabla R(x_t) - \eta\widehat{\ell}_{t,i}) - R^*(\nabla R(x_t)) \\ &+ \eta\nabla R^*(\nabla R(x_t))^T\widehat{\ell}_{t,i}. \end{aligned}$$

Since $R^*(x^*) = y^{*T}x^* - R(y^*)$ with $\nabla R(y^*) = x^*$, we get

$$\begin{aligned} R^*(\nabla R(x_t)) &= x_t^T\nabla R(x_t) - R(x_t) = x_t^T\,\mathbb{I}, \\ R^*(\nabla R(x_t) - \eta\widehat{\ell}_{t,i}) &= (x_te^{-\eta\widehat{\ell}_{t,i}})^T(\nabla R(x_t) - \eta\widehat{\ell}_{t,i}) - R(x_te^{-\eta\widehat{\ell}_{t,i}}) \\ &= (x_te^{-\eta\widehat{\ell}_{t,i}})^T\,\mathbb{I}. \end{aligned}$$

Finally, using the relation $\nabla R^*(\nabla R(x_t)) = x_t$ we get

$$B_{R^*}(\theta_t - \eta\widehat{\ell}_{t,i}, \theta_t) = x_t^T(e^{-\eta\widehat{\ell}_{t,i}} - 1 + \eta\widehat{\ell}_t)\,.$$

Since $\exp(-x) - x - 1 \le x^2/2$ for any $x > 0$ we get the desired improved regret bound. Note that the regret bound is still depending on the randomness of i_t via $\widehat{\ell}_t$. We take its expectation

$$\mathbb{E}[Regret_T] \le \mathbb{E}\Big[\frac{\eta}{2}\sum_{t=1}^{T} x_t^T \mathbb{E}_{x_t}[\widehat{\ell}_t^2] + \frac{\log d}{\eta}\Big]\,.$$

Thus we have to upper bound

$$\mathbb{E}_{x_t}[\widehat{\ell}_{t,i}^2] = \sum_{j=1}^{d}\Big(\frac{\ell_{t,i}}{x_{t,i}}\Big)^2 \mathbb{1}_{i=j}\mathbb{P}(i_t = j) = \frac{\ell_{t,i}^2}{x_{t,i}}$$

and we obtain

$$\mathbb{E}[Regret] \le \frac{\eta}{2}\sum_{t=1}^{T}\sum_{i=1}^{d}\ell_{t,i}^2 + \frac{\log d}{\eta} \le \frac{\eta}{2}TG^2 + \frac{\log d}{\eta}\,,$$

and the desired result follows. □

The regret bound is less accurate than in the complete information setting with a relative loss of G that can be as large as $\sqrt{d}$. It is still optimal up to log terms in this incomplete information setting since one has to explore the space. Exp3 can be improved as the variance of the unbiased estimator $\widehat{\nabla f_t(x_t)}$ is not controlled since

$$\mathbb{E}_{x_t}[\widehat{\nabla f_t(x_t)}^2] = \mathbb{E}_{x_t}\Big[\frac{\ell_{t,i}^2}{x_{t,i}^2}\mathbb{1}_{i=i_t}\Big] = \sum_{j=1}^{d}\frac{\ell_{t,i}^2}{x_{t,i}^2}\mathbb{1}_{i=j}\mathbb{P}(i_t = j) = \frac{\ell_{t,i}^2}{x_{t,i}}$$

can be as large as possible since the confident weights $x_{t,i}$ can be as small as possible. This issue has raised many improvements of Exp3, see Bubeck and Cesa-Bianchi (2012).

On the contrary to the EWA algorithm, the recursion is not invariant by a shift of the loss. Thus the adaptation of Exp3 to the OCO setting is more complicated. One would like to use the gradient trick and the linearized loss $\nabla f_t(x_t)^T x$ but then the loss is not positive.

Bibliography

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *arXiv preprint arXiv:1405.4980*, 2014.

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.

Joseph De Vilmarest and Olivier Wintenberger. Stochastic online optimization using kalman recursion. *arXiv preprint arXiv:2002.03636*, 2020.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Ludwig Fahrmeir. Posterior mode estimation by extended kalman filtering for multivariate dynamic generalized linear models. *Journal of the American Statistical Association*, 87 (418):501–509, 1992.

Steffen Grünewälder. Compact convex projections. *The Journal of Machine Learning Research*, 18(1):8089–8131, 2017.

Elad Hazan. Introduction to online convex optimization. *arXiv preprint arXiv:1909.05207*, 2019.

Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. *Machine learning*, 80(2-3):165–188, 2010.

Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 421–436, 2011.

Mark Herbster and Manfred K Warmuth. Tracking the best expert. *Machine learning*, 32 (2):151–178, 1998.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Jyrki Kivinen and Manfred K Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *information and computation*, 132(1):1–63, 1997.

Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

Francesco Orabona. A modern introduction to online learning. *arXiv preprint arXiv:1912.13213*, 2019.

Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Shai Shalev-Shwartz and Yoram Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69(2-3):115–142, 2007.

Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2):107–194, 2011.

Olivier Wintenberger. Optimal learning with bernstein online aggregation. *Machine Learning*, 106(1):119–141, 2017.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.