Stochastic optimization

Master 2 Optimization

Olivier Fercoq

November 16, 2021

Contents

Chapter 1

Introduction

1.1 Expected risk minimization

Let us consider a mesurable function

$$\begin{aligned} f : R^d \times \Xi &\to \mathbb{R} \\ (x,t) &\mapsto f(x,t) \end{aligned}$$

and a random variable ξ on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with values in Ξ. We suppose that $\mathbb{E}(|f(x,\xi)|) < +\infty$.
In the lecture, we are interested in the numerical resolution of the optimization problem

$$\min_{x \in \mathbb{R}^d} \mathbb{E}(f(x,\xi)) \tag{1.1.1}$$

The challenge is that the law of ξ is not supposed to be known and we cannot compute $\mathbb{E}$, or its computation is expensive. Instead, the law is revealed through (ξ_k), a sequence of i.i.d. samples of ξ.
Note that the assumption of i.i.d. samples (ξ_k) means in particular that this sequence does not depend on the optimization variable x.
This kind of optimization problems is ubiquitus when solving a machine learning problem. Let us illustrate this by the example of logistic regression

Exercise 1.1 (Maximum likelihood estimator for logistic regression)**.**
We consider a classification problem defined by observations $(x_i, y_i)_{1 \le i \le n}$ where for all i, $x_i \in \mathbb{R}^p$ and $y_i \in \{-1, 1\}$. We propose the following linear model for the generation of the data. Each observation is supposed to be independent and there exists a vector $w \in \mathbb{R}^p$ and $w_0 \in \mathbb{R}$ such that for all i, (y_i, x_i) is a realization of the random variable (Y, X) whose law $\mathcal{D}$ satisfies

$$\mathbb{P}_{w,w_0}(Y = 1|X) = \frac{\exp(X^\top w + w_0)}{1 + \exp(X^\top w + w_0)} .$$

1. Show that $\forall i \in \{1, \dots, n\}$, $\mathbb{P}(Y_i = y_i|x_i) = \dfrac{1}{1 + \exp(-y_i(x_i^\top w + w_0))}$.

2. Show that the maximum likelihood estimator is

$$(\hat{w}, \hat{w}_0) = \arg\min_{w,w_0} \sum_{i=1}^n \log(1 + \exp(-y_i(x_i^\top w + w_0)))$$

3. Denote $f(w, w_0) = \sum_{i=1}^n \log(1 + \exp(-y_i(x_i^\top w + w_0)))$. Compute $\nabla f(w, w_0)$.

In the exercise, we have $\xi_i = (x_i, y_i)$. Since, we have n observations, it is possible to evaluate the objective function. However, when n is large, say millions or billions, this can be a tedious task.

1.2 Gradient descent

The gradient descent method is the most basic minimization method for a differentiable function f. It requires access to the full function: it is thus not well adapted to our problem template. However, it will be the basis for the development of specialized algorithms. It consists in a sequence $(x_k)_{k\in\mathbb{N}}$ of points in $\mathbb{R}^n$ defined by induction from $x_0 \in \mathbb{R}^n$ by

Algorithm 1: Gradient descent

$$x_{k+1} = x_k - \gamma_k \nabla f(x_k)$$

where for all k, γ_k is a positive coefficient.

Theorem 1.1. *Let f be a convex differentiable function that has a minimizer x^* and whose gradient is L-Lipschitz continuous. The gradient method with constant step size $\gamma_k = \frac{1}{L}$ satisfies*

$$f(x_k) - f(x^*) \leq \frac{L\|x_0 - x^*\|^2}{2k}$$

If moreover f is μ-strongly convex, then

$$\begin{aligned} f(x_k) - f(x^*) &\leq \Big(1 - \frac{\mu}{L}\Big)^k (f(x_0) - f(x^*) + \tfrac{L}{2}\|x_0 - x^*\|^2) \\ \|x_k - x^*\|^2 &\leq \Big(1 - \frac{\mu}{L}\Big)^k (\tfrac{2}{L}(f(x_0) - f(x^*)) + \|x_0 - x^*\|^2) \end{aligned}$$

Proof. We will prove a more general results in the rest of the lecture. □

1.3 How to compute gradient?

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function. In order to run the algorithm, we would like to compute its gradient. By definition, $\nabla f(x)$ is the unique vector of $\mathbb{R}^n$ such that

$$f(x + h) = f(x) + \langle \nabla f(x), h\rangle + o(h) \ .$$

There are several ways to compute a gradient. All should give the same result.

1.3.1 Using partial derivatives

We know that the gradient is the vector of all the partial derivatives. Hence, we can compute $\frac{\partial f}{\partial x_i}(x)$ for all i and reconstruct the vector.
Example. Let us consider the function $f(x) = \|Ax - b\|^2$ where $A \in \mathbb{R}^{m\times n}$. We can write

$$f(x) = \sum_{j=1}^m \Big(\sum_{i=1}^n A_{j,i}x_i - b_j\Big)^2$$

and so

$$\frac{\partial f}{\partial x_k}(x) = 2\sum_{j=1}^{m} A_{j,k}\Big(\sum_{i=1}^{n} A_{j,i}x_i - b_j\Big) \ .$$

We recognise the components of the vector

$$\nabla f(x) = 2A^\top(Ax-b) \ .$$

1.3.2 Using the definition

We compute $f(x+h)$ and try isolating $f(x)$, a term linear in h and a negligible term.
Example. We consider $f(x) = \|Ax-b\|^2$.

$$\begin{aligned} f(x+h) = \|A(x+h)-b\|^2 &= \|Ax-b\|^2 + 2\langle Ax-b, Ah\rangle + \|Ah\|^2 \\ &= f(x) + 2\langle A^\top(Ax-b), h\rangle + o(h) \end{aligned}$$

thus, $\nabla f(x) = 2A^\top(Ax-b)$.

1.3.3 Using the chain rule

Let $g : \mathbb{R}^n \to \mathbb{R}^m$ and $f : \mathbb{R}^m \to \mathbb{R}^p$. The chain rule states that the Jacobian matrix of the function $f \circ g$ at x is given by

$$J_{f\circ g}(x) = J_f(g(x)) \times J_g(x) \ .$$

We recall that

$$J_g(x) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1}(x) & \dots & \frac{\partial g_1}{\partial x_n}(x) \\ \vdots & & \vdots \\ \frac{\partial g_m}{\partial x_1}(x) & \dots & \frac{\partial g_m}{\partial x_n}(x) \end{bmatrix}$$

is the unique linear map such that

$$g(x+h) = g(x) + J_g(x)h + o(h) \ .$$

The chain rule allows us to combine simple functions in order to obtain complex functions. In is at the basis of automatic differentiation and the resolution of neural network models.
When $f : \mathbb{R}^m \to \mathbb{R}$ and $g(x) = Ax$ where A is a $m \times n$ matrix, the formula simplifies as

$$\nabla(f \circ A)(x) = A^\top \nabla f(Ax) \ .$$

Example. We consider $f(x) = \|Ax-b\|^2$.
Let us remark that $f(x) = h(Ax)$ where $h(y) = \|y-b\|^2$.
Since $h(y+h) = \|y+h-b\|^2 = \|y-b\|^2 + 2\langle y-b, h\rangle + \|h\|^2$, we know that $\nabla h(y) = 2(y-b)$.
Using the chain rule, we get $\nabla f(x) = \nabla(h \circ A)(x) = A^\top \nabla h(Ax) = 2A^\top(Ax-b)$.

Algorithm 2: Subgradient method

```
select $g_k \in \partial f(x_k)$
$x_{k+1} = x_k - \gamma_k g_k$
```

1.4 Subgradient method

When the function we want to minimize is not differentiable but is still convex we can use subgradients instead of gradients. In return, we shall set smaller, diminishing step-sizes to ensure that the algorithm continues to converge. We obtain the algorithm
where for all k, γ_k is a positive coefficient.

Theorem 1.2. *Let f be a convex function that has a minimizer x^* and γ_k be a sequence such that $\frac{\sum_{l=0}^k \gamma_l^2}{\sum_{l=0}^k \gamma_l} \to 0$ when $k \to +\infty$. Then the subgradient method satisfies*

$$f(x_k) - f(x^*) \to 0$$

Proof. We will prove a more general results in the rest of the lecture. □

1.5 Implementation project

Each student will be assigned a problem and an algorithm to implement on this problem. We will use the database MNIST `http://yann.lecun.com/exdb/mnist/` where the goal is to classify digits between 0 and 9.
Two models, hence two functions to minimize will be considered :

- Multinomial logisitic regression with squared 2-norm regularization. Denote $y_{i,j} = 1$ if the image i represents digit j and 0 otherwise and consider a positive real number α. The objective function is the convex function

$$F(w, w_0) = \frac{1}{n}\sum_{i=1}^n \log\Big(\sum_{j=0}^9 \exp\big(\sum_{k=1}^d x_{i,k}w_{k,j} + w_{0,j}\big)\Big) - \sum_{j=0}^9 y_{i,j}\big(\sum_{k=1}^d x_{i,k}w_{k,j} + w_{0,j}\big) + \frac{\alpha}{2}\|w\|_2^2$$

  Here d is the number of pixel in the image $x_{i,:}$ and n is the number of images is the training data set. In this case, you should derive the formula for the stochastic gradients and use this formula in the algorithm.

- A multilayer perceptron neural network with 2 dense layers, rectified linear unit activation functions, a softmax output and categorical cross entropy loss function. In that more complex case, the stochastic gradients will be computed using the automatic differentiation tool tensorflow with its keras API.

The work to do is as follows.

1. Each student chooses a model and an algorithm to determine its parameters.
2. By keeping part of the training set into a validation set, find a good value for the hyperparameters of the model (for instance, the number α, the number of neurons in each layer).

Let us explain the procedure for the multinomial logistic regression case and forget about w_0 for simplification. Denote α the hyperparameter and A the set of its possible values. Let $F_\alpha(w)$ be the loss defined by the statistical model for the parameter w and hyperparameter α, that is

$$F_\alpha(w) = \frac{1}{n_{\text{train}}} \sum_{i=1}^{n_{\text{train}}} \log\Big(\sum_{j=0}^{9} \exp\big(\sum_{k=1}^{d} x_{i,k}^{\text{train}} w_{k,j}\big)\Big) - \sum_{j=0}^{9} y_{i,j}^{\text{train}}\big(\sum_{k=1}^{d} x_{i,k}^{\text{train}} w_{k,j}\big) + \frac{\alpha}{2}\|w\|_2^2$$

Denote $w^\alpha = \arg\min_w F_\alpha(w)$.

We also define the accuracy as $L(w) = \frac{1}{n_{\text{valid}}} \sum_{j=1}^{n_{\text{valid}}} \ell((x_i^{\text{valid}})^\top w, y_i^{\text{valid}})$ where $\ell((x_i^{\text{valid}})^\top w, y_i^{\text{valid}}) = 1$ if the largest value of the 10-dimensional vector $(x_i^{\text{valid}})^\top w$ is for the good digit and 0 otherwise.

The optimization problem we are trying to solve in this question is a bilevel optimization problem.

$$\begin{aligned} &\min_{\alpha \in A} L(w^\alpha) \\ &w^\alpha \in \arg\min_w F_\alpha(w) \end{aligned}$$

We will then solve a sequence of optimization problems indexed by α and choose the best α.

3. We will compare models and optimization algorithms in terms of classification performance, quality of the local optima returned by each method, the speed of convergence.

Chapter 2

Stochastic gradient

2.1 Algorithm

We want to solve the problem

$$\min_{x\in\mathbb{R}^d} \mathbb{E}[f(x,\xi)]$$

Given a sequence of step sizes γ_k, the algorithm reads

Algorithm 3: Stochastic gradient

$$x_{k+1} = x_k - \gamma_k \nabla f(x_k, \xi_{k+1})$$

where $\nabla f(x_k, \xi_{k+1})$ is the gradient of $(x \mapsto f(x, \xi_{k+1}))$ at x_k.

Remark 2.1. If $(x \mapsto f(x, \xi_{k+1}))$ is not differentiable, one can use a subgradient of the function instead of its gradient.

Example 2.1 (Empirical Risk Minimization). In this context, we are given N data points, each of which is associated with a loss function f_i, $1 \leq i \leq N$. A typical model in machine learning consists in minimizing the empirical risk given by

$$\min_x \frac{1}{N}\sum_{i=1}^{N} f_i(x)$$

This corresponds to Problem (2.3.1) with $\xi = I \sim U(\{1,\dots,N\})$. The expectation is computable but N may be so large that this takes a long time. Running stochastic gradient on this problem leads to an algorithm with very low complexity per iteration, which is often used in practice:

$$\begin{cases} \text{Generate } I_{k+1} \sim U(\{1,\dots,N\}) \\ x_{k+1} = x_k - \gamma_k \nabla f_{I_{k+1}}(x_k) \end{cases}$$

Example 2.2 (Least Mean Squares). We are given a random variable $\xi = (X, Y)$ where $X \in \mathbb{R}^n$ and $Y \in \mathbb{R}$. Least Mean Squares (LMS) is a regression problem in expectation

$$\min_{w\in\mathbb{R}^n} \frac{1}{2}\mathbb{E}[(Y - X^\top w)^2]$$

Show that stochastic gradient on this problems writes

$$w_{k+1} = w_k - \gamma_k (X_{k+1}^\top w_k - Y_{k+1}) X_{k+1}$$

2.2 Convergence

We denote $F(x) = \mathbb{E}(f(x,\xi))$ and by $\mathbb{E}_k$ the expectation knowing $(\xi_1, \dots, \xi_k)$. Note that x_k is measurable with respect to $(\xi_1, \dots, \xi_k)$.

2.2.1 Nonconvex objective

Theorem 2.1. *Suppose that:*

- *$(x \mapsto f(x,\xi))$ is differentiable for all ξ with a L-Lipschitz gradient,*
- *there exists $C > 0$ such that $\mathbb{E}(\|\nabla f(x,\xi)\|^2) \leq C$ for all x,*
- *the sequence γ_k is deterministic.*

The iterates of the stochastic gradient algorithm $x_{k+1} = x_k - \gamma_k \nabla f(x_k, \xi_{k+1})$ satisfy the convergence guarantee

$$\mathbb{E}\Big[\min_{0\leq l\leq k} \|\nabla F(x_l)\|^2\Big] \leq \frac{2(F(x_0) - \inf F) + CL\sum_{l=0}^k \gamma_l^2}{2\sum_{l=0}^k \gamma_l} .$$

Proof. By Taylor-Lagrange inequality,

$$\begin{aligned} F(x_{k+1}) &\leq F(x_k) + \langle \nabla F(x_k), x_{k+1} - x_k\rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2 \\ &\leq F(x_k) - \gamma_k \langle \nabla F(x_k), \nabla f(x_k, \xi_{k+1})\rangle + \frac{L\gamma_k^2}{2}\|\nabla f(x_k, \xi_{k+1})\|^2 \end{aligned}$$

where we use the fact that $x_{k+1} - x_k = -\gamma_k \nabla f(x_k, \xi_{k+1})$. We apply the conditional expectation $\mathbb{E}_k$:

$$\begin{aligned} \mathbb{E}_k[F(x_{k+1})] &\leq F(x_k) - \gamma_k\|\nabla F(x_k)\|^2 + \frac{L}{2}\gamma_k^2 \mathbb{E}_k[\|\nabla f(x_k, \xi_{k+1})\|^2] \\ \gamma_k\|\nabla F(x_k)\|^2 &\leq F(x_k) - \mathbb{E}_k[F(x_{k+1})] + \frac{L}{2}\gamma_k^2 C \end{aligned}$$

We then apply total expectation and sum for l between 0 and k

$$\mathbb{E}\Big[\sum_{l=0}^k \gamma_l \|\nabla F(x_l)\|^2\Big] \leq F(x_0) - \mathbb{E}[F(x_{k+1})] + \frac{L}{2}\sum_{l=0}^k \gamma_l^2 C$$

The result follows by remarking that $\|\nabla F(x_l)\|^2 \geq \min_{0\leq l'\leq k} \|\nabla F(x_l')\|^2$ for all l and $\mathbb{E}[F(x_{k+1})] \geq \inf F$. □

2.2.2 Convex objective

Theorem 2.2. *Suppose that:*

- *$(x \mapsto f(x,\xi))$ is convex and differentiable for all ξ,*
- *there exists $C > 0$ such that $\mathbb{E}(\|\nabla f(x,\xi)\|^2) \leq C$ for all x*
- *there exists $x^* \in \arg\min F$,*
- *the sequence γ_k is deterministic.*

The iterates of the stochastic gradient algorithm $x_{k+1} = x_k - \gamma_k \nabla f(x_k, \xi_{k+1})$ satisfy the convergence guarantee

$$\mathbb{E}\Big[F(\bar{x}_k^\gamma) - F(x^*)\Big] \leq \frac{\mathbb{E}[\|x_0 - x^*\|^2] + C\sum_{l=0}^k \gamma_l^2}{2\sum_{l=0}^k \gamma_l}$$

where $\bar{x}_k^\gamma = \dfrac{\sum_{l=0}^k \gamma_l x_l}{\sum_{j=0}^k \gamma_j}$ is a convex combination of all previous iterates.

Proof. Let us first remark that $\mathbb{E}[\nabla f(x,\xi)] \in \partial F(x)$ for all x. Indeed,

$$\begin{aligned}
&f(y,\xi) \geq f(x,\xi) + \langle \nabla f(x,\xi), y - x\rangle \\
&F(y) = \mathbb{E}[f(y,\xi)] \geq \mathbb{E}[f(x,\xi)] + \mathbb{E}[\langle \nabla f(x,\xi), y-x\rangle] = F(x) + \langle \mathbb{E}[\nabla f(x,\xi)], y - x\rangle\ .
\end{aligned}$$

Now, we apply $\mathbb{E}_k$ the expectation knowing $(\xi_1, \ldots, \xi_k)$.

$$\begin{aligned}
\mathbb{E}_k[\|x_{k+1} - x^*\|] &= \mathbb{E}_k[\|x_k - x^*\| + 2\langle x_{k+1} - x_k, x_k - x^*\rangle + \|x_{k+1} - x_k\|^2] \\
&= \|x_k - x^*\| - 2\gamma_k\langle \mathbb{E}_k[\nabla f(x_k, \xi_{k+1})], x_k - x^*\rangle + \gamma_k^2 \mathbb{E}_k[\|\nabla f(x_k, \xi_{k+1})\|^2] \\
&\leq \|x_k - x^*\| + 2\gamma_k\langle \mathbb{E}_k[\nabla f(x_k, \xi_{k+1})], x^* - x_k\rangle + \gamma_k^2 C \\
&\leq \|x_k - x^*\| + 2\gamma_k(F(x^*) - F(x_k)) + \gamma_k^2 C\ .
\end{aligned}$$

We reorganise and apply total expectation:

$$\mathbb{E}[\gamma_k(F(x_k) - F(x^*))] \leq -\frac{1}{2}\mathbb{E}[\|x_{k+1} - x^*\|^2] + \frac{1}{2}\mathbb{E}[\|x_k - x^*\|^2] + \frac{\gamma_k^2 C}{2}$$

We sum for l between 0 and k:

$$\mathbb{E}[\sum_{l=0}^k \gamma_l(F(x_l) - F(x^*))] \leq -\frac{1}{2}\mathbb{E}[\|x_{k+1} - x^*\|^2] + \frac{1}{2}\mathbb{E}[\|x_0 - x^*\|^2] + \sum_{l=0}^k \frac{\gamma_l^2 C}{2}$$

The result follows by convexity of F:

$$\mathbb{E}\Big[F(\bar{x}_l^\gamma) - F(x^*)\Big] \leq \frac{1}{\sum_{j=0}^k \gamma_j}\mathbb{E}[\sum_{l=0}^k \gamma_l(F(x_l) - F(x^*))] \leq \frac{\mathbb{E}[\|x_0 - x^*\|^2] + C\sum_{l=0}^k \gamma_l^2}{2\sum_{j=0}^k \gamma_j} \qquad \square$$

2.2.3 Step size sequence

We know that $\mathbb{E}\Big[F(\bar{x}_l^\gamma) - F(x^*)\Big] \leq \frac{\mathbb{E}[\|x_0 - x^*\|^2] + C\sum_{l=0}^k \gamma_l^2}{2\sum_{l=0}^k \gamma_l}$. A natural question is: which sequence (γ_k) should we take?
We would like $\sum_{j=1}^k \gamma_j \to +\infty$ and $\frac{\sum_{l=1}^k \gamma_l^2}{\sum_{j=1}^k \gamma_j} \to 0$. Such a sequence can be for instance taken as $\gamma_k = \frac{\gamma_0}{(k+1)^\alpha}$ with $0 < \alpha < 1$. Then,

$$\sum_{j=0}^k \gamma_j = \sum_{j=0}^k \frac{\gamma_0}{(j+1)^\alpha} \geq \sum_{j=0}^k \int_{j+1}^{j+2} \frac{\gamma_0}{t^\alpha} dt = \int_1^{k+2} \frac{\gamma_0}{t^\alpha} dt = \frac{\gamma_0}{1-\alpha}\Big[t^{1-\alpha}\Big]_1^{k+2} = \frac{\gamma_0}{1-\alpha}\Big((k+2)^{1-\alpha} - 1\Big)$$

$$\sum_{j=0}^k \gamma_j^2 = \sum_{j=0}^k \frac{\gamma_0^2}{(j+1)^{2\alpha}} \leq \gamma_0^2 + \sum_{j=1}^k \int_j^{j+1} \frac{\gamma_0^2}{t^{2\alpha}} dt = \gamma_0^2 + \int_1^{k+1} \frac{\gamma_0^2}{t^{2\alpha}} dt = \begin{cases} \gamma_0^2(1 + \ln(k+1)) & \text{if } \alpha = 1/2 \\ \gamma_0^2(1 + \frac{(k+1)^{1-2\alpha} - 1}{1-2\alpha}) & \text{if } \alpha \neq 1/2 \end{cases}$$

We obtain the following cases:

	$\frac{1}{\sum_{j=0}^{k} \gamma_j}$	$\frac{\sum_{l=1}^{k} \gamma_l^2}{\sum_{j=1}^{k} \gamma_j}$
$0 < \alpha < 1/2$	$O\Big(\frac{1}{k^{1-\alpha}}\Big)$	$O\Big(\frac{1}{k^{\alpha}}\Big)$
$\alpha = 1/2$	$O\Big(\frac{1}{k^{1/2}}\Big)$	$O\Big(\frac{\ln(k)}{k^{1/2}}\Big)$
$1/2 < \alpha < 1$	$O\Big(\frac{1}{k^{1-\alpha}}\Big)$	$O\Big(\frac{1}{k^{1-\alpha}}\Big)$

The best rate is obtained with $\alpha = 1/2$, that is $\gamma_k = \frac{\gamma_0}{\sqrt{k+1}}$. With this choice, we have

$$\mathbb{E}[F(\bar{x}_k^{\gamma}) - F(x^*)] \in O\Big(\frac{\ln(k)}{\sqrt{k}}\Big).$$

Remark 2.2. If we know the number of iterations K we are going to perform, we can set a constant step size $\gamma_k = \frac{a}{\sqrt{K}}$ and obtain a guarantee $\mathbb{E}[F(\bar{x}_K^{\gamma}) - F(x^*)] \in O\Big(\frac{1}{\sqrt{K}}\Big)$

2.2.4 Strongly convex objective

When F is μ-strongly convex, we can show that a step size decreasing as $\gamma_k = \frac{a}{\mu(k+b)}$ gives an improved rate $\mathbb{E}[F(x_k) - F(x^*)] \in O\Big(\frac{1}{k}\Big)$.

Theorem 2.3. *Suppose that:*

- $(x \mapsto f(x, \xi))$ *is convex and differentiable for all* ξ*,*
- F *is* μ*-strongly convex and its gradient is* L*-Lipschitz,*
- *there exists* $C > 0$ *such that* $\mathbb{E}(\|\nabla f(x, \xi)\|^2) \leq C$ *for all* x
- *there exists* $x^* \in \arg\min F$*,*
- *the sequence* γ_k *is deterministic and satisfies* $\gamma_k = \frac{a}{\mu(k+b)}$ *for a given* $a > 0.5$*,* $b > 0$*.*

The iterates of the stochastic gradient algorithm $x_{k+1} = x_k - \gamma_k \nabla f(x_k, \xi_{k+1})$ *satisfy the convergence guarantee*

$$\mathbb{E}[\|x_k - x^*\|^2] \leq \frac{\frac{a^2 C}{(2a-1)\mu^2}}{k+b}$$
$$\mathbb{E}[F(x_k) - F(x^*)] \leq \frac{\frac{a^2 CL}{(4a-2)\mu^2}}{k+b}$$

Proof. Compared to the convex case, we replace the inequality $F(y) \geq F(x) + \langle \mathbb{E}[\nabla f(x, \xi)], y - x\rangle$ by the stronger one $F(y) \geq F(x) + \langle \mathbb{E}[\nabla f(x, \xi)], y - x\rangle + \frac{\mu}{2}\|y - x\|^2$. Hence

$$\begin{aligned} \mathbb{E}_k[\|x_{k+1} - x^*\|] &\leq \|x_k - x^*\| - 2\gamma_k \langle \mathbb{E}_k[\nabla f(x_k, \xi_{k+1})], x_k - x^*\rangle + \gamma_k^2 \mathbb{E}_k[\|\nabla f(x_k, \xi_{k+1})\|^2] \\ &\leq \big(1 - \mu\gamma_k\big)\|x_k - x^*\| + 2\gamma_k(F(x^*) - F(x_k)) + \gamma_k^2 C \ . \end{aligned} \tag{2.2.1}$$

We use $F(x^*) - F(x_k) \leq -\frac{\mu}{2}\|x_k - x^*\|^2$ to get

$$\mathbb{E}_k[\|x_{k+1} - x^*\|^2] \leq \big(1 - 2\mu\gamma_k\big)\|x_k - x^*\|^2 + \gamma_k^2 C\ .$$

We will now show by induction that for $\gamma_k = \frac{a}{\mu(k+b)}$, we have a convergence in $\mathbb{E}[\|x_k - x^*\|^2] \leq \frac{\Delta}{k+b}$.
We suppose that for a given iterate k, there exists Δ such that $\mathbb{E}[\|x_k - x^*\|^2] \leq \frac{\Delta}{k+b}$.

$$\begin{aligned}\mathbb{E}[\|x_{k+1} - x^*\|^2] &\leq \big(1 - 2\mu\gamma_k\big)\mathbb{E}[\|x_k - x^*\|^2] + \gamma_k^2 C \\ &\leq \big(1 - 2\frac{a}{k+b}\big)\frac{\Delta}{k+1} + \frac{a^2}{\mu^2(k+b)^2}C\end{aligned}$$

$$\begin{aligned}\big(1 - 2\frac{a}{k+b}\big)\frac{\Delta}{k+b} + \frac{a^2}{\mu^2(k+b)^2}C &\leq \frac{\Delta}{k+b+1} \\ \Leftrightarrow \big(1 - 2\frac{a}{k+b}\big)\Delta(k+b+1) + \frac{a^2(k+b+1)}{\mu^2(k+b)}C &\leq \Delta(k+b) \\ \Leftrightarrow \Delta - 2a\Delta\frac{k+b+1}{k+b} + \frac{a^2(k+b+1)}{\mu^2(k+b)}C &\leq 0 \\ \Leftrightarrow \Delta\frac{k+b}{k+b+1} - 2a\Delta + \frac{a^2}{\mu^2}C &\leq 0\end{aligned}$$

This holds true for all k as soon as $a > 0.5$ and $\Delta \leq \frac{a^2 C}{(2a-1)\mu^2}$.
What is left is to come back to function values. We can do it because $F(x) - F(x^*) \leq \nabla F(x^*), x - x^*\rangle + \frac{L}{2}\|x - x^*\|^2 = \frac{L}{2}\|x - x^*\|^2$. □

2.3 Proximal stochastic gradient

The previous theorem is nice but it requires in particular the objective to be at same time Lipschitz continuous, strongly convex and to have a Lipschitz gradient. Unfortunately, this never happens, which makes its usefulness questionable. Yet, if we replace "Lipschitz" by "locally Lipschitz", this issue disappears. The proof can be modified to manage a proximal term, potentially encounting for a projection onto a bounded domain. Also, we shall write the proof for this case with the bounded variance condition $\mathbb{E}(\|\nabla f(x,\xi) - \nabla F(x_k)\|^2) \leq C$, which is less restrictive than bounded stochastic gradients $\mathbb{E}(\|\nabla f(x,\xi)\|^2) \leq C$.
We consider the problem

$$\min_{x \in \mathcal{X}} \mathbb{E}(f(x,\xi)) + g(x) \tag{2.3.1}$$

where $f(\cdot,\xi)$ is differentiable for all ξ and g has a simple proximal operator ($\mathrm{prox}_g(x) = \arg\min_y g(y) + \frac{1}{2}\|x - y\|^2$ is easily computable). We shall denote $F(x) = \mathbb{E}(f(x,\xi))$.
Consider the proximal stochastic gradient algorithm

Algorithm 4: Proximal stochastic gradient algorithm

$$x_{k+1} = \mathrm{prox}_{\gamma_k g}\big(x_k - \gamma_k \nabla f(x_k, \xi_{k+1})\big)$$

Theorem 2.4. *Suppose that:*

- *$(x \mapsto f(x, \xi))$ is convex and differentiable for all ξ,*
- *g is a proper convex, l.s.c. function,*
- *F is μ-strongly convex and has a L-Lipschitz gradient,*
- *there exists $C > 0$ such that $\mathbb{E}(\|\nabla f(x, \xi) - \nabla F(x_k)\|^2) \leq C$ for all $x \in \operatorname{dom} g$*
- *there exists $x^* \in \arg\min F + g$,*
- *the sequence γ_k is deterministic, satisfies $\gamma_k = \frac{a}{\mu(k+b)}$ for given $a > 1$ and $b > 0$ such that $\gamma_0 = \frac{a}{\mu b} \leq \frac{1}{2L}$.*

The iterates of the proximal stochastic gradient algorithm $x_{k+1} = \operatorname{prox}_{\gamma_k g}\left(x_k - \gamma_k \nabla f(x_k, \xi_{k+1})\right)$ satisfy the convergence guarantee

$$\mathbb{E}[\|x_k - x^*\|^2] \leq \frac{\frac{4a^2 C}{(a-1)\mu^2}}{k+b}$$

Proof. By the properties of the proximal operator, if we denote $p = \operatorname{prox}_{\gamma g}(x)$, we know that for all y,

$$\gamma g(p) + \frac{1}{2}\|p - x\|^2 \leq \gamma g(y) + \frac{1}{2}\|y - x\|^2 - \frac{1}{2}\|y - p\|^2$$

Applying this to $p = x_{k+1}$, $x = x_k - \gamma_k \nabla f(x_k, \xi_{k+1})$ and $y = x^*$ yields

$$\gamma_k g(x_{k+1}) + \frac{1}{2}\|x_{k+1} - x_k + \gamma_k \nabla f(x_k, \xi_{k+1})\|^2 \leq \gamma_k g(x^*) + \frac{1}{2}\|x^* - x_k + \gamma_k \nabla f(x_k, \xi_{k+1})\|^2 - \frac{1}{2}\|x^* - x_{k+1}\|^2$$

$$\frac{1}{2}\|x^* - x_{k+1}\|^2 + \frac{1}{2}\|x_{k+1} - x_k + \gamma_k \nabla f(x_k, \xi_{k+1})\|^2 \leq \frac{1}{2}\|x^* - x_k\|^2 + \gamma_k \Big(g(x^*) - g(x_{k+1})$$
$$+ \langle \nabla f(x_k, \xi_{k+1}), x^* - x_k \rangle\Big) + \frac{\gamma_k^2}{2}\|\nabla f(x_k, \xi_{k+1})\|^2$$

$$\mathbb{E}_k[\|x^* - x_{k+1}\|^2] \leq (1 - \gamma_k \mu)\|x^* - x_k\|^2 + 2\gamma_k \Big(g(x^*) - \mathbb{E}_k[g(x_{k+1})] + F(x^*) - F(x_k)\Big) + \cancel{\gamma_k^2 \mathbb{E}_k[\|\nabla f(x_k, \xi_{k+1})\|^2}$$
$$- \mathbb{E}_k[\|x_{k+1} - x_k\|^2] - \cancel{\gamma_k^2 \mathbb{E}_k[\|\nabla f(x_k, \xi_{k+1})\|^2]} - 2\gamma_k \mathbb{E}_k[\langle \nabla f(x_k, \xi_{k+1}), x_{k+1} - x_k \rangle]$$

We combine this with Taylor-Lagrange inequality.

$$F(x_{k+1}) \leq F(x_k) + \langle \nabla F(x_k), x_{k+1} - x_k \rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2$$

$$-F(x_k) \leq -F(x_{k+1}) + \langle \nabla f(x_k, \xi_{k+1}), x_{k+1} - x_k \rangle + \langle \nabla F(x_k) - \nabla f(x_k, \xi_{k+1}), x_{k+1} - x_k \rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2$$

$$\begin{aligned}\mathbb{E}_k[\|x^* - x_{k+1}\|^2] \leq{}& (1 - \gamma_k \mu)\|x^* - x_k\|^2 + 2\gamma_k \Big(g(x^*) - \mathbb{E}_k[g(x_{k+1})] + F(x^*) - F(x_{k+1})\Big) \\ &+ (L\gamma_k - 1)\mathbb{E}_k[\|x_{k+1} - x_k\|^2] + 2\gamma_k \mathbb{E}_k[\langle \nabla F(x_k) - \nabla f(x_k, \xi_{k+1}), x_{k+1} - x_k \rangle] \\ \leq{}& (1 - \gamma_k \mu)\|x^* - x_k\|^2 + (L\gamma_k - 0.5)\mathbb{E}_k[\|x_{k+1} - x_k\|^2] + 4\gamma_k^2 \mathbb{E}_k[\|\nabla F(x_k) - \nabla f(x_k, \xi_{k+1})\|^2]\end{aligned}$$

where we used the inequality $2\langle a, b \rangle \leq \|a\|^2 + \|b\|^2$ for $a = 2\gamma_k(\nabla F(x_k) - \nabla f(x_k, \xi_{k+1}))$ and $b = x_{k+1} - x_k$. By assumption, $L\gamma_k \leq 0.5$ so that the term $\mathbb{E}_k[\|x_{k+1} - x_k\|^2]$ cancels out. We are left with a recursion similar to what we had without the proximal term:

$$\mathbb{E}[\|x^* - x_{k+1}\|^2] \leq (1 - \gamma_k \mu)\mathbb{E}[\|x^* - x_k\|^2] + 4\gamma_k^2 C$$

and we solve it similarly by induction. □

2.4 Comparison of the results depending on the assumption

We have shown in this chapter several convergence results for the stochastic gradient algorithm, depending on the assumptions we made. The following table summarizes them.

Problem	Convex	Lipschitz ∇F	Noise	Step-size	Rate
$\min\limits_{x} \mathbb{E}[f(x,\xi)]$	no	yes	$\mathbb{E}(\lVert\nabla f(x,\xi)\rVert^2) \leq C$	$\gamma_k = \frac{\gamma_0}{\sqrt{k+1}}$	$\mathbb{E}\Big[\min\limits_{l\leq k} \lVert\nabla F(x_l)\rVert^2\Big] \in O(\frac{\ln(k)}{\sqrt{k}})$
$\min\limits_{x} \mathbb{E}[f(x,\xi)]$	yes	no	$\mathbb{E}(\lVert\nabla f(x,\xi)\rVert^2) \leq C$	$\gamma_k = \frac{\gamma_0}{\sqrt{k+1}}$	$\mathbb{E}[F(\bar{x}_k^\gamma) - F(x^*)] \in O(\frac{\ln(k)}{\sqrt{k}})$
$\min\limits_{x} \mathbb{E}[f(x,\xi)] + g(x)$	μ-str. conv.	yes	$\mathbb{E}(\lVert\nabla f(x,\xi) - \nabla F(x)\rVert^2) \leq C$	$\gamma_k = \frac{a}{\mu(k+b)}$	$\mathbb{E}[\lVert x_k - x^*\rVert^2] \in O(\frac{1}{k})$

Chapter 3

Stochastic variance-reduced gradient

3.1 Motivation and algorithm

In this chapter, we shall concentrate on the minimization of an objective function which can be written as a finite sum:

$$\min_x \frac{1}{N} \sum_{i=1}^{N} f_i(x)$$

If the sum involves a large number of summands, it is worth considering a stochastic algorithm to solve the problem [Bot10]. Indeed, stochastic gradient descent (SGD) will perform n iterations for the cost of 1 iteration of gradient descent (GD). Suppose that $F(x) = \frac{1}{N}\sum_{i=1}^{N} f_i(x)$ is μ-strongly convex and that we are interested in solving the problem up to precision ϵ, then SGD will require a number of iterations of the order of $1/\epsilon$. In comparison GD requires $O(\ln(1/\epsilon)$ iterations. If $1/\epsilon < N\ln(1/\epsilon)$, then SGD is preferable.

Variance-reduced stochastic gradient methods try to go beyond this alternative and take profit of the advantages of both algorithms: a cheap per iteration cost together with a linear convergence rate on strongly convex functions. The idea is to compute stochastic gradients with a lower variance thanks to the concept of control variates. The control variate we use is a periodically computed full gradient. By carefully setting the period, we can mitigate the heavy cost of the computation of this gradient while improving a lot the quality of the stochastic gradients. Starting from a given point x_0, $w_0 = x_0$ and using a probability $p < 1$ of updating the control variate, Stochastic Variance Reduced Gradient (SVRG)[1] writes as follows:

Algorithm 5: Stochastic variance-reduced gradient

$$\begin{aligned}
&i_{k+1} \sim U(\{1, \dots, N\}) \\
&g_{k+1} = \nabla F(w_k) + \nabla f_{i_{k+1}}(x_k) - \nabla f_{i_{k+1}}(w_k) \\
&x_{k+1} = x_k - \gamma g_{k+1} \\
&w_{k+1} = \begin{cases} x_k & \text{with probability } p \\ w_k & \text{with probability } 1-p \end{cases}
\end{aligned}$$

[1] we shall use the version of [KHR20]

3.2 Convergence

We will need the following result on convex functions with a Lipschitz gradient.

Proposition 3.1. *Let f be a convex function with an L-Lipschitz gradient. For all x and y,*

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{1}{2L} \|\nabla f(x) - \nabla f(y)\|^2$$

Proof. Let us fix a vector y. Let $\phi : x \mapsto f(x) - \langle \nabla f(y), x - y \rangle$.
We can see that ϕ is convex and $\nabla \phi(x) = \nabla f(x) - \nabla f(y)$. Hence $\nabla \phi(y) = 0$ and $y \in \arg\min \phi$.
Thus, using Taylor Lagrange inequality

$$\begin{aligned} \phi(y) &\leq \phi(x - \frac{1}{L}\nabla\phi(x)) \leq \phi(x) + \langle \nabla\phi(x), -\frac{1}{L}\nabla\phi(x)\rangle + \frac{L}{2}\|\frac{1}{L}\nabla\phi(x)\|^2 \\ \phi(y) &\leq \phi(x) - \frac{1}{2L}\|\nabla\phi(x)\|^2 \end{aligned}$$

Reminding the definition of ϕ, we obtain $f(x) - \langle \nabla f(y), x - y\rangle \geq f(y) + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2$. □

Theorem 3.2. *Suppose that for all $i \in \{1, \ldots, N\}$, f_i is convex and differentiable, ∇f_i is L-Lipschitz and F is μ-strongly convex. Denote x^* the unique minimizer of F and suppose that $\gamma \leq \frac{1}{6L}$. The iterates of SVRG converge linearly as*

$$\mathbb{E}[\|x_k - x^*\|^2] \leq c^k \Delta_0$$

where $c = \max(1 - \gamma\mu, 1 - p/2)$ and $\Delta_0 = \|x_0 - x^*\|^2 + \frac{4\gamma^2}{pN}\sum_{i=1}^N \|\nabla f_i(x_0) - \nabla f_i(x^*)\|^2$. Moreover, the expected cost of an iteration is $2 + pN$ stochastic gradients.

Proof. Computing $\nabla F(w_k)$ requires N stochastic gradients but we do it only with probability p. Then we need to compute $\nabla f_{i_{k+1}}(x_k)$ and $\nabla f_{i_{k+1}}(w_k)$ at each iteration. This gives the cost in number of stochastic gradients per iteration.
We now proceed to the convergence rate. Note that $\mathbb{E}_k[g_{k+1}] = \nabla F(w_k) + \nabla F(x_k) - \nabla F(w_k) = \nabla F(x_k)$.

$$\begin{aligned} \|x_{k+1} - x^*\|^2 &= \|x_k - x^*\|^2 - 2\gamma\langle g_{k+1}, x_k - x^*\rangle + \gamma^2\|g_{k+1}\|^2 \\ \mathbb{E}_k[\|x_{k+1} - x^*\|^2] &= \|x_k - x^*\|^2 + 2\gamma\langle \nabla F(x_k), x^* - x_k\rangle + \gamma^2\mathbb{E}_k[\|g_{k+1}\|^2] \\ \mathbb{E}_k[\|x_{k+1} - x^*\|^2] &\leq (1 - \gamma\mu)\|x_k - x^*\|^2 + 2\gamma(F(x^*) - F(x_k)) + \gamma^2\mathbb{E}_k[\|g_{k+1}\|^2] \end{aligned}$$

We deal with the noise term $\|g_{k+1}\|^2$.

$$\begin{aligned} \mathbb{E}_k[\|g_{k+1}\|^2] &= \mathbb{E}_k[\|\nabla F(w_k) + \nabla f_{i_{k+1}}(x_k) - \nabla f_{i_{k+1}}(w_k)\|^2] \\ &= \mathbb{E}_k[\|\nabla f_{i_{k+1}}(x_k) - \nabla f_{i_{k+1}}(x^*) + \nabla F(w_k) + \nabla f_{i_{k+1}}(x^*) - \nabla f_{i_{k+1}}(w_k)\|^2] \quad \leq 2\mathbb{E}_k[\|\nabla f_{i_{k+1}}(x_k) - \nabla \\ &\leq 2\mathbb{E}_k[\|\nabla f_{i_{k+1}}(x_k) - \nabla f_{i_{k+1}}(x^*)\|^2] + 2\mathbb{E}_k[\|\nabla f_{i_{k+1}}(x^*) - \nabla f_{i_{k+1}}(w_k)\|^2] \end{aligned}$$

where we used $\mathbb{E}[X^2] = \mathbb{E}[X]^2 + \mathbb{E}[(X - \mathbb{E}[X])^2] \geq \mathbb{E}[(X - \mathbb{E}[X])^2]$.
Using the inequality $f_i(x) \geq f_i(x^*) + \langle \nabla f_i(x^*), x - x^*\rangle + \frac{1}{2L}\|\nabla f_i(x) - \nabla f_i(x^*)\|^2$, we can further bound

$$\mathbb{E}_k[\|\nabla f_{i_{k+1}}(x_k) - \nabla f_{i_{k+1}}(x^*)\|^2] \leq \frac{2L}{N}\sum_{i=1}^N f_i(x_k) - f_i(x^*) - \langle \nabla f_i(x^*), x_k - x^*\rangle = 2L(F(x_k) - F(x^*))$$

Let us denote

$$\mathcal{D}_k = \frac{4\gamma^2}{pN}\sum_{i=1}^{N}\|\nabla f_i(w_k) - \nabla f_i(x^*)\|^2 = \frac{4\gamma^2}{p}\mathbb{E}_k[\|\nabla f_{i_{k+1}}(w_k) - \nabla f_{i_{k+1}}(x^*)\|^2$$

so that

$$\mathbb{E}_k[\|x_{k+1} - x^*\|^2] \leq (1-\gamma\mu)\|x_k - x^*\|^2 + 2\gamma(F(x^*) - F(x_k)) + 4L\gamma^2(F(x_k) - F(x^*)) + \frac{p}{2}\mathcal{D}_k \ .$$

Using again the inequality $f_i(x) \geq f_i(x^*) + \langle \nabla f_i(x^*), x - x^* \rangle + \frac{1}{2L}\|\nabla f_i(x) - \nabla f_i(x^*)\|^2$, we obtain

$$\mathcal{D}_k \leq \frac{8L\gamma^2}{pN}\sum_{i=1}^{N} f_i(w_k) - f_i(x^*) - \langle \nabla f_i(x^*), w_k - x^* \rangle = \frac{8L\gamma^2}{p}\big(F(w_k) - F(x^*)\big)$$

We can remove w_k from the bound by using the update rule

$$\mathbb{E}_k[\mathcal{D}_{k+1}] = (1-p)\mathcal{D}_k + p\frac{4\gamma^2}{pN}\sum_{i=1}^{N}\|\nabla f_i(x_k) - \nabla f_i(x^*)\|^2 \leq (1-p)\mathcal{D}_k + 8L\gamma^2\big(F(x_k) - F(x^*)\big)$$

Combining the bounds on $\mathbb{E}_k[\|x_{k+1} - x^*\|^2]$ and $\mathbb{E}[\mathcal{D}_{k+1}]$ we get

$$\begin{aligned}\mathbb{E}_k[\|x_{k+1} - x^*\|^2 + \mathcal{D}_{k+1}] &\leq (1-\gamma\mu)\|x_k - x^*\|^2 + 2\gamma(F(x^*) - F(x_k)) + 4L\gamma^2(F(x_k) - F(x^*)) \\ &\quad + \frac{p}{2}\mathcal{D}_k + (1-p)\mathcal{D}_k + 8L\gamma^2\big(F(x_k) - F(x^*)\big) \\ &\leq (1-\gamma\mu)\|x_k - x^*\|^2 + (1 - \frac{p}{2})\mathcal{D}_k + \big(12L\gamma^2 - 2\gamma\big)\big(F(x_k) - f(x^*)\big)\end{aligned}$$

Since we choose $\gamma \leq \frac{1}{6L}$ we obtain a contraction in expectation with rate $\max(1-\gamma\mu, 1-p/2)$. □

Chapter 4

Adaptive step-sizes

A major issue with the stochastic gradient method is that the step-size sequence should be determined beforehand and has no reason to be adapted the problem at stake. The following proposition tries to answer this issue.

Proposition 4.1 ([SZL13]). *Consider $x \in \mathbb{R}^d$ and a function $f : \mathbb{R}^d \times \Xi \to \mathbb{R}$ such that for any ξ, $(x \mapsto f(x,\xi))$ is differentiable with an L-Lipschitz gradient. Denote $x^+(\gamma) = x - \gamma \nabla f(x,\xi)$, where $\gamma \in \mathbb{R}^d_+$ does not depend on ξ.*

$$\inf_{\gamma \in \mathbb{R}^d_+} \mathbb{E}[f(x^+(\gamma),\xi)] \leq \mathbb{E}[f(x^+(\gamma^*),\xi)] \leq f(x) - \frac{1}{2L}\sum_{j=1}^d \frac{(\mathbb{E}[\nabla_j f(x,\xi)]^4)}{\mathbb{E}[\nabla_j f(x,\xi)]^2}$$

where $\gamma^*_j = \frac{1}{L}\frac{(\mathbb{E}[\nabla_j f(x,\xi)])^2}{\mathbb{E}[\nabla_j f(x,\xi)^2]}$.

Proof. Denote $F(x) = \mathbb{E}[f(x,\xi)]$. By Taylor Lagrange inequality

$$F(x^+(\gamma)) \leq F(x) - \langle \nabla F(x), \gamma \nabla f(x,\xi)\rangle + \frac{L}{2}\|\gamma \nabla f(x,\xi)\|^2$$

$$\mathbb{E}[F(x^+(\gamma))] \leq F(x) - \sum_{j=1}^d \gamma_j (\nabla_j F(x))^2 + \frac{L}{2}\sum_{j=1}^d \gamma_j^2 \mathbb{E}[\nabla_j f(x,\xi)^2]$$

Minimizing the right hand side with respect to γ give the result. □

This proposition shows that if we knew the law of ξ, we could design step-sizes for the stochastic gradient method that would ensure a nice decrease in the objective function. Moreover, these step-sizes would be adaptive to the local behaviour of the function and decrease to 0 at the optimal rate. However, we cannot set step-sizes as required by Proposition 4.1 because the law of ξ is unknown.

In this chapter, we are going to study two algorithms that have adaptive step-sizes : Adagrad and Adam. In fact, they go even further than the previous proposition, they define step-sizes that depend on the whole history of stochastic gradients, ξ_{k+1} included. We shall denote ab for the element-wise product of two vectors a and b, $\frac{a}{b}$ for their element-wise division and $\|a\|_b^2 = \sum_{i=1}^d b_i a_i^2$.

4.1 Adagrad

Adagrad has been introduced in [DHS11]. The algorithm writes

Algorithm 6: Adagrad

$$
\begin{aligned}
v_{k+1} &= \sum_{s=0}^{k} \nabla f(x_s, \xi_{s+1})^2 \\
\gamma_{k+1} &= \frac{\alpha}{\sqrt{v_{k+1}}} \\
x_{k+1} &= x_k - \gamma_{k+1} \nabla f(x_k, \xi_{k+1})^2
\end{aligned}
$$

Theorem 4.2. *Suppose that*

- $f(\cdot, \xi)$ *is convex for all* ξ
- *There exists* $x^* \in \arg\min F$*, where* $F(x) = \mathbb{E}[f(x, \xi)]$
- *For all* $k \geq 0$*, for all* $i \in \{1, \ldots, d\}$*,* $|x_{k,i} - x_i^*| \leq D$
- *For all* x, ξ*, for all* $g \in \partial f(x, \xi)$*,* $\|g\| \leq G$

Then the iterates of Adagrad satisfy

$$\mathbb{E}[F(\bar{x}_K) - F(x^*)] \leq \frac{2dD^2 G}{\alpha\sqrt{K}} + \frac{2\alpha dG}{\sqrt{K}}$$

where $\bar{x}_K = \frac{1}{K}\sum_{k=0}^{K-1} x_k$*.*

Proof. Since the step-sizes are not deterministic any more, we need to pay more attention than before when applying conditional expectations. Yet, the proof will begin with similar arguments as in Theorem 2.2.

$$
\begin{aligned}
f(x_k, \xi_{k+1}) - f(x_*, \xi_{k+1}) &\leq \langle \nabla f(x_k, \xi_{k+1}), x_k - x_* \rangle \\
&\leq \frac{1}{2}\|x_k - x_*\|^2_{\gamma_{k+1}^{-1}} - \frac{1}{2}\|x_{k+1} - x_*\|^2_{\gamma_{k+1}^{-1}} + \frac{1}{2}\|\nabla f(x_k, \xi_{k+1})\|^2_{\gamma_{k+1}}
\end{aligned}
$$

We now sum for k between 0 and $K-1$

$$\sum_{k=0}^{K-1} f(x_k, \xi_{k+1}) - f(x_*, \xi_{k+1}) \leq \sum_{k=0}^{K-1} \Big(\frac{1}{2}\|x_k - x_*\|^2_{\gamma_{k+1}^{-1}} - \frac{1}{2}\|x_{k+1} - x_*\|^2_{\gamma_{k+1}^{-1}}\Big) + \sum_{k=0}^{K-1} \frac{1}{2}\|\nabla f(x_k, \xi_{k+1})\|^2_{\gamma_{k+1}} \tag{4.1.1}$$

Note that up to now, we have not applied any form of expectation. The difference of norms is nearly telescoping.

$$
\begin{aligned}
&\sum_{k=0}^{K-1} \Big(\|x_k - x_*\|^2_{\gamma_{k+1}^{-1}} - \|x_{k+1} - x_*\|^2_{\gamma_{k+1}^{-1}}\Big) = \|x_0 - x_*\|^2_{\gamma_1^{-1}} - \|x_K - x_*\|^2_{\gamma_K^{-1}} + \sum_{k=0}^{K-1} (\|x_k - x_*\|^2_{(\gamma_{k+1}^{-1} - \gamma_k^{-1})} \\
&\leq \sum_{i=1}^{d} \frac{D^2}{\gamma_{1,i}} + \sum_{k=0}^{K-1} D^2\Big(\frac{1}{\gamma_{k+1,i}} - \frac{1}{\gamma_{k,i}}\Big) \leq \sum_{i=1}^{d} \frac{D^2}{\gamma_{1,i}} + \frac{D^2}{\gamma_{K,i}}
\end{aligned}
\tag{4.1.2}
$$

where we used the fact that $\frac{1}{\gamma_{k+1}} - \frac{1}{\gamma_k} \geq 0$ and $\frac{1}{\gamma_0} = 0$. Moreover, $\gamma_{K,i} \geq \frac{\alpha}{G\sqrt{K}}$.

We now turn to the second part of (4.1.1): $\sum_{k=0}^{K-1}\|\nabla f(x_k,\xi_{k+1})\|^2_{\gamma_{k+1}}$. By definition of γ_k, if we denote $a_k^{(i)} = \nabla_i f(x_k,\xi_{k+1})^2 \geq 0$, then

$$\sum_{k=0}^{K-1}\|\nabla f(x_k,\xi_{k+1})\|^2_{\gamma_{k+1}} = \alpha\sum_{k=0}^{K-1}\sum_{i=1}^{d}\frac{a_k^{(i)}}{\sqrt{\sum_{s=0}^{K-1} a_s^{(i)}}}$$

Lemma 4.3. *Let (a_k) be a sequence of nonnegative numbers. Then*

$$\sum_{k=0}^{K-1}\frac{a_k}{\sqrt{\sum_{s=0}^{k} a_s}} \leq 2\sqrt{\sum_{s=0}^{K-1} a_s}$$

Proof. Denote $h_K = \sum_{k=0}^{K-1}\frac{a_k}{\sqrt{\sum_{s=0}^{k} a_s}}$. We will show the result by induction. Clearly, $h_0 = \sqrt{a_0} \leq 2\sqrt{a_0}$.

We now assume that $h_K \leq 2\sqrt{\sum_{s=0}^{K-1} a_s}$.

$$h_{K+1} = h_K + \frac{a_K}{\sqrt{\sum_{s=0}^{K} a_s}} \leq 2\sqrt{\sum_{s=0}^{K-1} a_s} + \frac{a_K}{\sqrt{\sum_{s=0}^{K} a_s}}$$

Now, since the square root is concave, we have

$$\sqrt{b-a} \leq \sqrt{b} - \frac{a}{2\sqrt{b}}$$

as long as $b-a \geq 0$ and $b > 0$. Hence,

$$h_{K+1} \leq 2\Big(\sqrt{\sum_{s=0}^{K} a_s} - \frac{a_K}{2\sum_{s=0}^{K} a_s}\Big) + \frac{a_K}{\sqrt{\sum_{s=0}^{K} a_s}} = 2\sqrt{\sum_{s=0}^{K} a_s}$$

Induction proceed, so the lemma is proved. □

We apply the lemma to our sequence of stochastic gradients to get:

$$\sum_{k=0}^{K-1}\|\nabla f(x_k,\xi_{k+1})\|^2_{\gamma_{k+1}} \leq 2\alpha\sum_{i=1}^{d}\sqrt{\sum_{k=0}^{K-1}(\nabla_i f(x_k,\xi_{k+1}))^2} \leq 2\alpha dG\sqrt{K}$$

We combine the inequality with (4.1.2) to get

$$\sum_{k=0}^{K-1} f(x_k,\xi_{k+1}) - f(x_*,\xi_{k+1}) \leq \frac{D^2G(1+\sqrt{K})}{\alpha} + 2\alpha dG\sqrt{K}$$

Remark that $F(x_k) = \mathbb{E}[f(x_k,\xi_{k+1})|\xi_1,\dots,\xi_k]$ so that $\mathbb{E}[F(x_k)] = \mathbb{E}[f(x_k,\xi_{k+1})]$. Hence, we apply expectation and use convexity of F:

$$\mathbb{E}[F(\bar{x}_K) - F(x^*)] \leq \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[f(x_k,\xi_{k+1}) - f(x_*,\xi_{k+1})] \leq \frac{D^2G(1+\sqrt{K})}{\alpha K} + \frac{2\alpha dG}{\sqrt{K}}$$

□

4.2 Adam

We are now going to see an algorithm that is often used for the resolution of neural network models: ADAM, which stands for stochastic gradient with adaptive moment estimation [KB14]. Its main ingredients are an adaptive estimation of the first and second moments of the stochastic gradient and coordinate-wise step-sizes. The idea is to design an exponential moving average of previous gradients and square gradients as an estimation of its moments. Finally, instead of just using the estimate of $\nabla F(x)$ to set the step-size, ADAM uses it directly as a means of reducing the variance of the stochastic gradient. The algorithm uses parameters $\alpha > 0$, $\beta_1 \in [0,1]$, $\beta_2 \in [0,1]$ and $\epsilon > 0$. It is initialized with a fixed x_0 and $m_0 = v_0 = 0$. It is given by

Algorithm 7: Adam

$$\begin{aligned}
m_{k+1} &= \beta_1 m_k + (1-\beta_1)\nabla f(x_k, \xi_{k+1}) \\
\hat{m}_{k+1} &= \frac{m_{k+1}}{1-\beta_1^{k+1}} \\
v_{k+1} &= \beta_2 v_k + (1-\beta_2)\nabla f(x_k, \xi_{k+1})^2 \\
\hat{v}_{k+1} &= \max\big(\hat{v}_k, \frac{v_{k+1}}{1-\beta_2^{k+1}}\big) \\
x_{k+1} &= x_k - \frac{\alpha_k}{\epsilon + \sqrt{\hat{v}_{k+1}}}\hat{m}_{k+1}
\end{aligned}$$

Theorem 4.4. *Suppose that*

- $f(\cdot, \xi)$ *is convex for all* ξ
- $\exists x^* \in \arg\min F,\ F(x) = \mathbb{E}[f(x,\xi)]$
- *For all* k*, for all* i*,* $|x_{k,i} - x_i^*| \le D$
- *For all* x, ξ*, for all* i*,* $|\nabla_i f(x,\xi)| \le G$
- $\alpha_k = \frac{\alpha_0}{k+1}$
- $\beta_1^2 < \beta_2$

Then the iterates of Adam satisfy

$$\mathbb{E}[F(\bar{x}_K) - F(x^*)] \le \frac{dD^2}{2(1-\beta_1)} \frac{\sqrt{1-\beta_2}G}{\alpha_0(\sqrt{K}+K)} + \frac{1+2\beta_1}{2(1-\beta_1)} \frac{\alpha_0\sqrt{1+\ln(K)}G}{\sqrt{1-\beta_2}\sqrt{1-\frac{\beta_1^2}{\beta_2}}\sqrt{K}} \in O(\frac{\ln(K)}{\sqrt{K}})$$

where $\bar{x}_K = \frac{1}{K}\sum_{k=0}^{K-1} x_k$.

We will prove this theorem in the form of an exercise.

Exercise 4.1. We will denote $\hat{\gamma}_{k+1} = \frac{\alpha_k}{(1-\beta_1^{k+1})(\epsilon+\sqrt{\hat{v}_{k+1}})}$ so that $x_{k+1} = x_k - \hat{\gamma}_{k+1} m_{k+1}$.

1. Show that
$$f(x_k, \xi_{k+1}) - f(x^*, \xi_{k+1}) \le \langle \nabla f(x_k, \xi_{k+1}), x_k - x^* \rangle$$

2. Using the relation $m_{k+1} = \beta_1 m_k + (1-\beta_1)\nabla f(x_k, \xi_{k+1})$, show that

$$\langle \nabla f(x_k, \xi_{k+1}), x_k - x^* \rangle = \langle m_{k+1}, x_k - x^* \rangle + \frac{\beta_1}{1-\beta_1}\Big(\langle m_{k+1}, x_{k+1} - x^* \rangle - \langle m_k, x_k - x^* \rangle \Big) + \frac{\beta_1}{1-\beta_1} \|m_{k+1}\|^2_{\hat\gamma_{k+1}}$$

3. Show that

$$\langle m_{k+1}, x_k - x^* \rangle = \frac{1}{2}\|x_k - x^*\|^2_{\hat\gamma^{-1}_{k+1}} - \frac{1}{2}\|x_{k+1} - x^*\|^2_{\hat\gamma^{-1}_{k+1}} + \frac{1}{2}\|m_{k+1}\|^2_{\hat\gamma_{k+1}}$$

4. Show that

$$\begin{aligned}\sum_{k=0}^{K-1} f(x_k, \xi_{k+1}) - f(x^*, \xi_{k+1}) \le \frac{\beta_1}{1-\beta_1}\Big(\langle m_K, x_K - x^* \rangle - \langle m_0, x_0 - x^* \rangle \Big) \\ + \sum_{k=0}^{K-1} \Big(\frac{1}{2}\|x_k - x^*\|^2_{\hat\gamma^{-1}_{k+1}} - \frac{1}{2}\|x_{k+1} - x^*\|^2_{\hat\gamma^{-1}_{k+1}} \Big) \\ + \Big(\frac{\beta_1}{1-\beta_1} + \frac{1}{2} \Big) \sum_{k=0}^{K-1} \|m_{k+1}\|^2_{\hat\gamma_{k+1}}\end{aligned}$$

5. Show that $(\hat\gamma_k)$ is a decreasing sequence and that $\hat\gamma_k \ge \frac{\alpha_0}{\sqrt{k}\sqrt{1-\beta_2 G}}$.

6. Show that

$$\sum_{k=0}^{K-1} \frac{1}{2}\|x_k - x^*\|^2_{\hat\gamma^{-1}_{k+1}} - \frac{1}{2}\|x_{k+1} - x^*\|^2_{\hat\gamma^{-1}_{k+1}} \le \frac{D^2}{2} \sum_{i=1}^{d} \Big(\frac{1}{\hat\gamma_{K,i}} + \frac{1}{\hat\gamma_{1,i}} - \frac{1}{\hat\gamma_{0,i}} \Big)$$

7. Show that

$$\langle m_K, x_K - x^* \rangle \le \frac{1}{2}\|m_K\|^2_{\hat\gamma_K} + \frac{D^2}{2}\sum_{i=1}^{d} \frac{1}{\hat\gamma_{K,i}} \le \frac{1}{2}\sum_{k=0}^{K-1} \|m_{k+1}\|^2_{\hat\gamma_{k+1}} + \frac{D^2}{2}\Big(\sum_{i=1}^{d} \frac{1}{\hat\gamma_{K,i}} + \frac{1}{\hat\gamma_{1,i}} \Big)$$

.

8. Denote $\gamma_{k+1} = \frac{\alpha_k}{(1-\beta_1)\sqrt{v_{k+1}}}$. Show that $\gamma_{k+1} \ge \hat\gamma_{k+1}$.

9. Let $x, y, z \in \mathbb{R}^d_+$ be nonnegative vectors and let p, q, r be positive real numbers such that $\frac{1}{p} + \frac{1}{q} + \frac{1}{r} = 1$. Show that $\sum_{j=1}^d x_j y_j z_j \le \|x\|_p \|y\|_q \|z\|_r$.

10. Denote $g_{k+1} = \nabla_i f(x_k, \xi_{k+1})$. In the following sequence of inequalities, tell the reason why each one is true

$$\begin{aligned}(m_{k,i})^2 \hat\gamma_{k,i} &\le (m_{k,i})^2 \gamma_{k,i} \\ &= \frac{\alpha_{k-1}}{(1-\beta_1)} \frac{\Big((1-\beta_1) \sum_{j=1}^k \beta_1^{k-j} g_j \Big)^2}{\sqrt{(1-\beta_2)\sum_{j=1}^k \beta_2^{k-j} g_j^2}} \\ &= \frac{\alpha_{k-1}(1-\beta_1)}{\sqrt{1-\beta_2}} \frac{\Big(\sum_{j=1}^k \big(\beta_2^{\frac{k-j}{4}} |g_j|^{\frac{1}{2}}\big) \big(\beta_1 \beta_2^{1/2}\big)^{\frac{k-j}{2}} \big(\beta_1^{k-j}|g_j|\big)^{\frac{1}{2}} \Big)^2}{\sqrt{\sum_{j=1}^k \beta_2^{k-j} g_j^2}} \\ &\le \frac{\alpha_{k-1}(1-\beta_1)}{\sqrt{1-\beta_2}} \Big(\sum_{j=1}^k \big(\tfrac{\beta_1^2}{\beta_2}\big)^{k-j} \Big)^{\frac{1}{2}} \Big(\sum_{j=1}^k \beta_1^{k-j} |g_j| \Big) \\ &\le \frac{\alpha_{k-1}(1-\beta_1)}{\sqrt{1-\beta_2}\sqrt{1-\frac{\beta_1^2}{\beta_2}}} \sum_{j=1}^k \beta_1^{k-j}|g_j|\end{aligned}$$

11. By remarking that $\sum_{k=j}^{K-1} \alpha_t \beta_1^{k-j} \leq \frac{\alpha_j}{1-\beta_1}$, show that

$$\sum_{k=0}^{K-1} (m_{k+1,i})^2 \hat{\gamma}_{k,i} \leq \frac{1}{\sqrt{1-\beta_2}\sqrt{1-\frac{\beta_1^2}{\beta_2}}} \sum_{k=0}^{K-1} \alpha_k |\nabla_i f(x_k, \xi_{k+1})|$$

12. Show that

$$\sum_{k=0}^{K-1} (m_{k+1,i})^2 \hat{\gamma}_{k,i} \leq \frac{\alpha_0 \sqrt{1+\ln(K)}}{\sqrt{1-\beta_2}\sqrt{1-\frac{\beta_1^2}{\beta_2}}} \sqrt{\sum_{k=0}^{K-1} (\nabla_i f(x_k, \xi_{k+1}))^2}$$

13. Conclude

Chapter 5

Coordinate descent

5.1 Exact coordinate descent

The idea of coordinate descent is to decompose a large optimisation problem into a sequence of one-dimensional optimisation problems. The algorithm was first described for the minimization of quadratic functions by Gauss and Seidel in [Sei74]. Coordinate descent methods have become unavoidable in machine learning because they are very efficient for key problems, namely Lasso, logistic regression and support vector machines. Moreover, the decomposition into small sub-problems means that only a small part of the data is processed at each iteration and this makes coordinate descent easily scalable to high dimensions.

We first decompose the space of optimisation variables X into blocks $X_1 \times \ldots \times X_n = X$. A classical choice when $X = \mathbb{R}^n$ is to choose $X_1 = \ldots = X_n = \mathbb{R}$. We will denote U_i the canonical injection from X_i to X, that is U_i is such that for all $h \in X_i$,

$$U_i h = (\underbrace{0, \ldots, 0}_{i-1 \text{ zeros}}, h^\top, \underbrace{0, \ldots, 0}_{n-i \text{ zeros}})^\top \in X.$$

For a function $f : X_1 \times \ldots \times X_n \to \mathbb{R}$, we define the following algorithm.

Algorithm 8: Exact coordinate descent

```
Start at $x_0 \in X$.
At iteration $k$, choose $l = (k \mod n) + 1$ (cyclic rule) and define $x_{k+1} \in X$ by
    $\begin{cases} x_{k+1}^{(i)} = \arg\min_{z \in X_l} f(x_k^{(1)}, \ldots, x_k^{(l-1)}, z, x_k^{(l+1)}, \ldots, x_k^{(n)}) & \text{if } i = l \\ x_{k+1}^{(i)} = x_k^{(i)} & \text{if } i \neq l \end{cases}$
```

Proposition 5.1 ([War63]). *If f is continuously differentiable and strictly convex and there exists $x_* = \arg\min_{x \in X} f(x)$, then the exact coordinate descent method (Alg. 8) converges to x_*.*

Example 5.1 (least squares). $f(x) = \frac{1}{2}\|Ax - b\|_2^2 = \frac{1}{2}\sum_{j=1}^m (a_j^\top x - b_j)^2$
At each iteration, we need to solve in z the 1D equation

$$\frac{\partial f}{\partial x^{(l)}}(x_k^{(1)}, \ldots, x_k^{(l-1)}, z, x_k^{(l+1)}, \ldots, x_k^{(n)}) = 0$$

For all $x \in \mathbb{R}^n$,

$$\frac{\partial f}{\partial x^{(l)}}(x) = a_l^\top (Ax - b) = a_l^\top a_l x^{(l)} + a_l^\top (\sum_{j \neq l} a_j x^{(j)}) - a_l^\top b$$

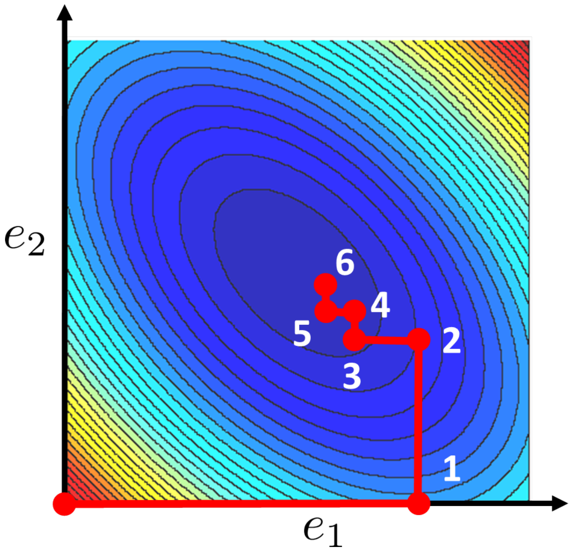

Figure 5.1: The successive iterates of the coordinate descent method on a 2D example. The function we are minimising is represented by its level sets: the bluer is the circle, the lower is the function values.

so we get

$$z^* = x_{k+1}^{(l)} = \frac{1}{\|a_l\|_2^2}\Big(- a_l^\top (\sum_{j\neq l} a_j x_k^{(j)}) + a_l^\top b\Big) = x_k^{(l)} - \frac{1}{\|a_l\|_2^2}\Big(a_l^\top (\sum_{j=1}^{n} a_j x_k^{(j)}) - a_l^\top b\Big)$$

Example 5.2 (non-differentiable function). $f(x^{(1)}, x^{(2)}) = |x^{(1)} - x^{(2)}| - \min(x^{(1)}, x^{(2)}) + \iota_{[0,1]^2}(x)$ f is convex but not differentiable. If we nevertheless try to run exact coordinate descent, the algorithm proceeds as $x_1^{(1)} = \arg\min_z f(z, x_0^{(2)}) = x_0^{(2)}$, $x_2^{(2)} = \arg\min_z f(x_1^{(1)}, z) = x_0^{(2)}$, and so on. Thus exact coordinate descent converges in two iterations to $(x_0^{(2)}, x_0^{(2)})$: the algorithm is stuck on the non-differentiability point on the line $\{x^{(1)} = x^{(2)}\}$ and does not reach the minimiser $(1, 1)$.

Example 5.3 (non-convex differentiable function).
$f(x^{(1)}, x^{(2)}, x^{(3)}) = -(x^{(1)}x^{(2)} + x^{(2)}x^{(3)} + x^{(3)}x^{(1)}) + \sum_{i=1}^{3} \max(0, |x^{(i)}| - 1)^2$
As shown by [Pow76], exact coordinate descent on this function started at the initial point $x^{(0)} = (-1 - \epsilon, 1 + \epsilon/2, -1 - \epsilon/4)$ has a limit cycle around the 6 corners of the cube that are not minimisers and avoids the 2 corners that are minimisers.
Exercise: Show Powell's result.

Example 5.4 (Adaboost). The Adaboost algorithm [CSS02] was designed to minimise the exponential loss given by

$$f(x) = \sum_{j=1}^{m} \mathbb{E}(-y_j h_j^\top x).$$

At each iteration, we select the variable l such that $l = \arg\max_i |\nabla_i f(x)|$ and we perform an exact coordinate descent step along this coordinate.
This variable selection rule is called the greedy or Gauss-Southwell rule. Like the cyclic rule, it leads to a converging algorithm but requires to compute the full gradient at each iteration. Greedy coordinate descent is interesting in the case of the exponential loss because the gradient of the function has a few very large coefficients and many negligible coefficients.

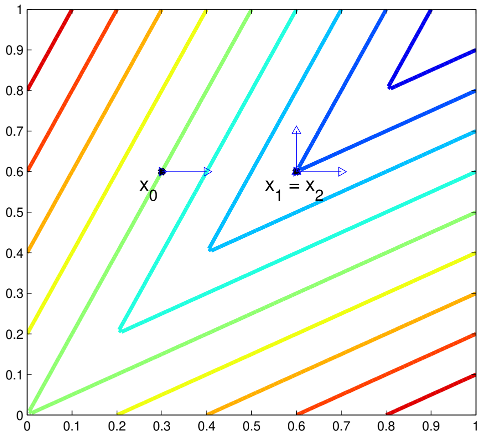

Figure 5.2: The function in Example 5.2

Exercise: Suppose that $y_j \in \{-1, 1\}$ and $h_{j,i} \in \{-1, 0, 1\}$ for all j, i. Give the explicit formulas of $\nabla_i f(x)$ and of the next update x_{k+1} knowing x_k.

5.2 Coordinate gradient descent

Solving a one-dimensional optimisation problems is generally easy and the solution can be approximated very well by algorithms like the bisection method. However, for the exact coordinate descent method, one needs to solve a huge number of one-dimensional problems and the expense quickly becomes prohibitive. Moreover, why should we solve to high accuracy the 1-dimensional problem and destroy this solution at the next iteration?
The idea of coordinate gradient descent is to perform one iteration of gradient descent in the 1-dimensional problem $\min_{z \in X_l} f(x_k^{(1)}, \dots, x_k^{(l-1)}, z, x_k^{(l+1)}, \dots, x_k^{(n)})$ instead of solving it completely. In general, this reduces drastically the cost of each iteration while keeping the same convergence behaviour.

Algorithm 9: Coordinate gradient descent

```
Start at $x_0$.
At iteration $k$, choose $i_{k+1} \in \{1, \dots, n\}$ and define $x_{k+1}$ by
    $\begin{cases} x_{k+1}^{(i)} = x_k^{(i)} - \gamma_i \nabla_i f(x_k) & \text{if } i = i_{k+1} \\ x_{k+1}^{(i)} = x_k^{(i)} & \text{if } i \neq i_{k+1} \end{cases}$
```

When choosing the cyclic or greedy rule, the algorithm does converge for any convex function f that has a Lipschitz-continuous gradient and such that $\arg\min_x f(x) \neq \emptyset$.
In fact we will assume that we actually know the *coordinate-wise* Lipschitz constants of the gradient of f, namely the Lipschitz constants of the functions

$$\begin{aligned} g_{i,x} : \; & X_i \to \mathbb{R} \\ & h \mapsto f(x + U_i h) = f(x^{(1)}, \dots, x^{(i-1)}, x^{(i)} + h, x^{(i+1)}, \dots, x^{(n)}) \end{aligned}$$

We will denote $L_i = L(\nabla g_{i,x})$ this Lipschitz constant. Written in terms of f, this means that

$$\forall x \in X, \forall i \in \{1,\dots,n\}, \forall h \in X_i, \quad \|\nabla f(x + U_i h) - \nabla f(x)\|_2 \leq L_i \|U_i h\|_2.$$

Lemma 5.2. *If f has a coordinate-wise Lipschitz gradient with constants $L_1, \dots, L_n$, then $\forall x \in X$, $\forall i \in \{1,\dots,n\}, \forall h \in X_i$,*

$$f(x + U_i h) \leq f(x) + \langle \nabla_i f(x), h \rangle + \frac{L_i}{2}\|h\|^2$$

Proof. This is Taylor's inequality applied to $g_{i,x}$. Note that we do not require the function to be twice differentiable. □

Proposition 5.3 ([BT13])**.** *Assume that f is convex, ∇f is Lipschitz continuous and* $\arg\min_{x\in X} f(x) \neq \emptyset$*. If i_{k+1} is chosen with the cyclic rule $i_{k+1} = (k \mod n) + 1$ and $\forall i$, $\gamma_i = \frac{1}{L_i}$, then the coordinate gradient descent method (Alg. 9) satisfies*

$$f(x_{k+1}) - f(x_*) \leq 4L_{\max}(1 + n^3 L_{\max}^2 / L_{\min}^2)\frac{R^2(x_0)}{k + 8/n}$$

where $R^2(x_0) = \max_{x,y\in X}\{\|x-y\| : f(y) \leq f(x) \leq f(x_0)\}$, $L_{\max} = \max_i L_i$ and $L_{\min} = \min_i L_i$.

The proof of this result is quite technical and in fact the bound is much more pessimistic than what is observed in practice (n^3 is very large if n is large). This is due to the fact that the cyclic rule behaves particularly bad on some extreme examples. To avoid such traps, it has been suggested to randomise the coordinate selection process.

Proposition 5.4 ([Nes12])**.** *Assume that f is convex, ∇f is Lipschitz continuous and* $\arg\min_{x\in X} f(x) \neq \emptyset$*. If i_{k+1} is randomly generated, independently of $i_1, \dots, i_k$ and $\forall i \in \{1,\dots,n\}$, $\mathbb{P}(i_{k+1} = i) = \frac{1}{n}$ and $\gamma_i = \frac{1}{L_i}$, then the coordinate gradient descent method (Alg. 9) satisfies for all $x_* \in \arg\min_x f(x)$*

$$\mathbb{E}[f(x_{k+1}) - f(x_*)] \leq \frac{n}{k+n}\Big((1 - \frac{1}{n})(f(x_0) - f(x_*)) + \frac{1}{2}\|x_* - x_0\|_L^2\Big)$$

where $\|x\|_L^2 = \sum_{i=1}^n L_i \|x^{(i)}\|_2^2$.

Proof. This is a particular case of the method developed in the next section. □

Comparison with gradient descent The iteration complexity of the gradient descent method is

$$f(x_{k+1}) - f(x_*) \leq \frac{L(\nabla f)}{2(k+1)}\|x_* - x_0\|_2^2$$

This means that to get an ϵ-solution (*i.e.* such that $f(x_k) - f(x_*) \leq \epsilon$), we need at most $\frac{L(\nabla f)}{2\epsilon}\|x_* - x_0\|_2^2$ iterations. What is most expensive in gradient descent is the evaluation of the gradient $\nabla f(x)$ with a cost C, so the total cost of the method is

$$C_{\text{grad}} = C\frac{L(\nabla f)}{2\epsilon}\|x_* - x_0\|_2^2$$

Neglecting the effect of randomisation, we usually have an ϵ-solution with coordinate descent in $\frac{n}{\epsilon}\big((1 - \frac{1}{n})(f(x_0) - f(x_*)) + \frac{1}{2}\|x_* - x_0\|_L^2\big)$ iterations. The cost of one iteration of coordinate descent is of the order of the cost of evaluation one partial derivative $\nabla_i f(x)$, with a cost c, so the total cost of the method is

$$C_{\text{cd}} = c\frac{n}{\epsilon}\big((1 - \frac{1}{n})(f(x_0) - f(x_*)) + \frac{1}{2}\|x_* - x_0\|_L^2\big)$$

How do these two quantities compare?
Let us consider the case where $f(x) = \frac{1}{2}\|Ax - b\|_2^2$.

- Computing $\nabla f(x) = A^\top(Ax-b)$ amounts to updating the residuals $r = Ax - b$ (one matrix vector product and a sum) and computing one matrix vector product. We thus have $C = O(\mathrm{nnz}(A))$.

- Computing $\nabla_i f(x) = e_i^\top A^\top(Ax-b)$ amounts to

  1. updating the residuals $r = Ax - b$: one scalar-vector product and a sum since we have $r_{k+1} = r_k + (x_{k+1}^{(i_{k+1})} - x_k^{(i_{k+1})})Ae_{i_{k+1}}$,
  2. computing one vector-vector product (the i^{th} column of A versus the residuals).

  Thus $c = O(\mathrm{nnz}(Ae_{i_{k+1}})) = O(\mathrm{nnz}(A)/n) = C/n$ if the columns of A are equally sparse.

- $f(x_0) - f(x_*) \leq \frac{L(\nabla f)}{2}\|x_0 - x_*\|_2^2$ and it may happen that $f(x_0) - f(x_*) \ll \frac{L(\nabla f)}{2}\|x_0 - x_*\|_2^2$

- $L(\nabla f) = \lambda_{\max}(A^\top A)$ and $L_i = a_i^\top a_i$ with $a_i = Ae_i$. We always have $L_i \leq L(\nabla f)$ and it may happen that $L_i = O(L(\nabla f)/n)$.

To conclude, in the quadratic case, $C_{\mathrm{cd}} \leq C_{\mathrm{grad}}$ and we may have $C_{\mathrm{cd}} = O(C_{\mathrm{grad}}/n)$.

5.3 Proximal coordinate descent

We are often interested in solving problems of the type

$$\min_{x \in X} F(x) = \min_{x \in X} f(x) + g(x) \tag{5.3.1}$$

where f and g are convex so that $F = f + g$ is convex, f has a Lipschitz continuous gradient and g may be nonsmooth but is separable. This means that for all $x \in X = X_1 \times \ldots X_n$,

$$g(x) = \sum_{i=1}^n g_i(x^{(i)}).$$

We can solve this kind of problems with the proximal coordinate descent method (Alg. 10, [Tse01]), which is also using the coordinate-wise Lipschitz constant.

Algorithm 10: Proximal coordinate descent

```
Start at $x_0 \in X$.
At iteration $k$, choose $i_{k+1} \in \{1, \dots, n\}$ and define $x_{k+1} \in X$ by
    $\begin{cases} x_{k+1}^{(i)} = \arg\min_{x \in X_i} g_i(x) + f(x_k) + \langle \nabla_i f(x_k), x - x_k^{(i)} \rangle + \frac{L_i}{2}\|x - x_k^{(i)}\|^2 & \text{if } i = i_{k+1} \\ x_{k+1}^{(i)} = x_k^{(i)} & \text{if } i \neq i_{k+1} \end{cases}$
```

For this algorithm to be practical, we need to be able to compute efficiently

$$\mathrm{prox}_{\gamma,g}(y) = \arg\min_{x \in X} g(x) + \frac{1}{2}\|x - y\|_{\gamma^{-1}}^2,$$

the proximal operator of g (remember that $\|x\|_{\gamma^{-1}}^2 = \sum_{i=1}^n \frac{1}{\gamma_i}\|x^{(i)}\|_2^2$).

Example 5.5 (Simple proximal operators).

- Indicator of a box: if $g(x) = \iota_{[a,b]}(x)$, then $\mathrm{prox}_{\gamma,g}(y) = \max(a, \min(x, b))$. This is the projection on $[a, b]$ (it does not depend on γ).
- Absolute value: if $g(x) = \lambda|x|$, then $\mathrm{prox}_{\gamma,g}(y) = \mathrm{sign}(y)\max(0, |y| - \gamma\lambda)$. This is the soft-thresholding operator.

We define

$$\bar{x}_{k+1} = \mathrm{prox}_{L^{-1},g}(x_k - L^{-1}\nabla f(x_k)) = \arg\min_{x\in X} g(x) + f(x_k) + \langle \nabla f(x_k), x - x_k\rangle + \frac{1}{2}\|x - x_k\|_L^2,$$

so that

$$x_{k+1}^{(i)} = \begin{cases} \bar{x}_{k+1}^{(i)} & \text{if } i = i_{k+1} \\ x_k^{(i)} & \text{if } i \neq i_{k+1} \end{cases}$$

Lemma 5.5. *For all $\gamma \in \mathbb{R}^n_{+*}$ and $x \in X$*

$$g(\bar{x}_{k+1})+\langle \nabla f(x_k), x_{k+1}-x_k\rangle+\frac{1}{2}\|x_{k+1}-x_k\|^2_{\gamma^{-1}} \leq g(x)+\langle \nabla f(x_k), x-x_k\rangle+\frac{1}{2}\|x-x_k\|^2_{\gamma^{-1}}-\frac{1}{2}\|x_{k+1}-x\|^2_{\gamma^{-1}}$$

Proof. The function $\psi : x \mapsto g(x) + \langle \nabla f(x_k), x - x_k\rangle + \frac{1}{2}\|x - x_k\|^2_{\gamma^{-1}}$ is strongly convex and its minimiser is $\bar{x}_{k+1}$. The inequality is just the strong convexity inequality of this function with respect to the norm $\|\cdot\|^2_{\gamma^{-1}}$ and applied at x and $\bar{x}_{k+1}$. □

Theorem 5.6 ([RT14])**.** *The proximal coordinate descent method (Alg. 10) with the random selection rule applied to Problem 5.3.1 satisfies for all $x_* \in \arg\min_x F(x)$*

$$\mathbb{E}[F(x_{k+1}) - F(x_*)] \leq \frac{n}{k+n}\Big((1 - \frac{1}{n})(F(x_0) - F(x_*)) + \frac{1}{2}\|x_* - x_0\|^2_L\Big)$$

where $\|x\|^2_L = \sum_{i=1}^n L_i\|x^{(i)}\|^2_2$.

Proof. By definition of the algorithm, $x_{k+1} - x_k = U_{i_{k+1}}(x_{k+1}^{(i_{k+1})} - x_k^{(i_{k+1})})$, so by Lemma 5.2,

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \langle \nabla_{i_{k+1}} f(x_k), x_{k+1}^{(i_{k+1})} - x_k^{(i_{k+1})}\rangle + \frac{L_{i_{k+1}}}{2}\|x_{k+1}^{(i_{k+1})} - x_k^{(i_{k+1})}\|^2 \\ &= f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k\rangle + \frac{1}{2}\|x_{k+1} - x_k\|^2_L \end{aligned} \tag{5.3.2}$$

Using the notation $\bar{x}_{k+1} = \arg\min_{x\in X} g(x) + f(x_k) + \langle \nabla f(x_k), x - x_k\rangle + \frac{1}{2}\|x - x_k\|^2_L$, we have

$$x_{k+1}^{(i)} = \begin{cases} \bar{x}_{k+1}^{(i)} & \text{if } i = i_{k+1} \\ x_k^{(i)} & \text{if } i \neq i_{k+1} \end{cases}$$

Using the conditional expectation knowing $\mathcal{F}_k = (i_1, \dots i_k)$, we get

$$\mathbb{E}[x_{k+1}^{(i)}|\mathcal{F}_k] = \mathbb{P}(i_{k+1} = i)\bar{x}_{k+1}^{(i)} + \mathbb{P}(i_{k+1} \neq i)x_k^{(i)} = \frac{1}{n}\bar{x}_{k+1}^{(i)} + (1 - \frac{1}{n})x_k^{(i)}$$

$$\begin{aligned} \mathbb{E}[\langle \nabla_i f(x_k), x_{k+1}^{(i)} - x_k^{(i)}\rangle|\mathcal{F}_k] &= \mathbb{P}(i_{k+1} = i)\langle \nabla_i f(x_k), \bar{x}_{k+1}^{(i)} - x_k^{(i)}\rangle + \mathbb{P}(i_{k+1} \neq i)\langle \nabla_i f(x_k), x_k^{(i)} - x_k^{(i)}\rangle \\ &= \frac{1}{n}\langle \nabla_i f(x_k), \bar{x}_{k+1}^{(i)} - x_k^{(i)}\rangle \end{aligned}$$

$$\mathbb{E}[\langle \nabla f(x_k), x_{k+1} - x_k\rangle|\mathcal{F}_k] = \sum_{i=1}^n \mathbb{E}[\langle \nabla_i f(x_k), x_{k+1}^{(i)} - x_k^{(i)}\rangle|\mathcal{F}_k] = \frac{1}{n}\langle \nabla f(x_k), \bar{x}_{k+1} - x_k\rangle \tag{5.3.3}$$

$$\mathbb{E}[\frac{1}{2}\|x_{k+1} - x_k\|^2_L|\mathcal{F}_k] = \sum_{i=1}^n \mathbb{E}[\frac{L_i}{2}\|x_{k+1}^{(i)} - x_k^{(i)}\|^2|\mathcal{F}_k] = \frac{1}{2n}\|\bar{x}_{k+1} - x_k\|^2_L \tag{5.3.4}$$

$$\mathbb{E}[g(x_{k+1}) - g(x_k)|\mathcal{F}_k] = \sum_{i=1}^n \mathbb{E}[g_i(x_{k+1}^{(i)}) - g_i(x_k^{(i)})|\mathcal{F}_k] = \frac{1}{n}(g(\bar{x}_{k+1}) - g(x_k)) \tag{5.3.5}$$

Combining (5.3.3), (5.3.4) and (5.3.5) with (5.3.2), we get

$$\begin{aligned}\mathbb{E}[g(x_{k+1}) + f(x_{k+1})|\mathcal{F}_k] &\leq \mathbb{E}[g(x_{k+1})|\mathcal{F}_k] + \mathbb{E}\Big[f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k\rangle + \frac{1}{2}\|x_{k+1} - x_k\|_L^2\Big|\mathcal{F}_k\Big] \\ &= (1 - \frac{1}{n})g(x_k) + \frac{1}{n}g(\bar{x}_{k+1}) + f(x_k) + \frac{1}{n}\langle \nabla f(x_k), \bar{x}_{k+1} - x_k\rangle + \frac{1}{2n}\|\bar{x}_{k+1} - x_k\|_L^2\end{aligned}$$

Using Lemma 5.5 with $x = x_k$, we get

$$\begin{aligned}\mathbb{E}[F(x_{k+1})|\mathcal{F}_k] = \mathbb{E}[g(x_{k+1}) + f(x_{k+1})|\mathcal{F}_k] &\leq g(x_k) + f(x_k) - \frac{1}{n}\|\bar{x}_{k+1} - x_k\|_2^2 \\ &\leq g(x_k) + f(x_k) = F(x_k) \qquad (5.3.6)\end{aligned}$$

Then, using Lemma 5.5 again, with $x = x_*$, we get

$$\begin{aligned}&\mathbb{E}[F(x_{k+1})|\mathcal{F}_k] = \mathbb{E}[g(x_{k+1}) + f(x_{k+1})|\mathcal{F}_k] \\ &\quad\leq (1 - \frac{1}{n})g(x_k) + f(x_k) + \frac{1}{n}g(x_*) + \frac{1}{n}\langle \nabla f(x_k), x_* - x_k\rangle + \frac{1}{2n}\|x_* - x_k\|_L^2 - \frac{1}{2n}\|x_* - \bar{x}_{k+1}\|_L^2\end{aligned}$$

We remark that

$$\mathbb{E}[\frac{1}{2}\|x_* - x_k\|_L^2 - \frac{1}{2}\|x_* - x_{k+1}\|_L^2|\mathcal{F}_k] = \frac{1}{2n}\|x_* - x_k\|_L^2 - \frac{1}{2n}\|x_* - \bar{x}_{k+1}\|_L^2,$$

so that

$$\begin{aligned}\mathbb{E}[F(x_{k+1})|\mathcal{F}_k] \leq (1 - \frac{1}{n})g(x_k) + f(x_k) + \frac{1}{n}g(x_*) + \frac{1}{n}\langle \nabla f(x_k), x_* - x_k\rangle \\ + \frac{1}{2}\|x_* - x_k\|_L^2 - \frac{1}{2}\mathbb{E}[\|x_* - x_{k+1}\|_L^2|\mathcal{F}_k].\end{aligned}$$

We use the convexity of f:

$$\mathbb{E}[F(x_{k+1})|\mathcal{F}_k] \leq (1 - \frac{1}{n})(g(x_k) + f(x_k)) + \frac{1}{n}(g(x_*) + f(x_*)) + \frac{1}{2}\|x_* - x_k\|_L^2 - \frac{1}{2}\mathbb{E}[\|x_* - x_{k+1}\|_L^2|\mathcal{F}_k].$$

We rearrange and we apply total expectation:

$$\mathbb{E}[F(x_{k+1}) - F(x_*) + \frac{1}{2}\|x_* - x_{k+1}\|_L^2] \leq \mathbb{E}[(1 - \frac{1}{n})(F(x_k) - F(x_*)) + \frac{1}{2}\|x_* - x_k\|_L^2].$$

Summing for k from 0 to K-1 yields

$$\begin{aligned}\mathbb{E}[F(x_{K+1}) - F(x_*)] + \frac{1}{2}\mathbb{E}[\|x_* - x_{K+1}\|_L^2] + \sum_{k=1}^{K-1} \mathbb{E}[\frac{1}{n}(F(x_k) - F(x_*)] \\ \leq (1 - \frac{1}{n})(F(x_0) - F(x_*)) + \frac{1}{2}\|x_* - x_0\|_L^2].\end{aligned}$$

Using (5.3.6) and the fact that $\mathbb{E}[\|x_* - x_{K+1}\|_L^2] \geq 0$,

$$(1 + \frac{k}{n})\mathbb{E}[F(x_{K+1}) - F(x_*)] \leq (1 - \frac{1}{n})(F(x_0) - F(x_*)) + \frac{1}{2}\|x_* - x_0\|_L^2]$$

We just need to divide by $\frac{n}{k+1}$ to conclude. □

Example 5.6 (Lasso). Proximal coordinate descent is widely used to solve the Lasso problem given by

$$\min_{x\in\mathbb{R}^p} \frac{1}{2}\|y - Zx\|_2^2 + \lambda\|x\|_1$$

Here, $f(x) = \frac{1}{2}\|y - Zx\|_2^2$ is differentiable while $g(x) = \lambda\|x\|_1$ is a non-differentiable function whose proximal operator is the soft-thresholding operator.

Example 5.7 (Multi-task Lasso). In the multi-task framework, the Lasso problem can be generalised as

$$\min_{x\in\mathbb{R}^{p\times q}} \frac{1}{2}\|Y - Zx\|_F^2 + \lambda\sum_{j=1}^{p}\|x_{j,:}\|_2.$$

Here, the optimisation variable is a $p \times q$ matrix. One can see that the nonsmooth part of the objective is $g(X) = \lambda\sum_{j=1}^{p}\|x_{j,:}\|_2$. This function is not separable when we consider the entries of x one by one but it is separable if we group these entries column-wise. Hence, we can consider block coordinate descent with p blocks of size q for the resolution of the multi-task Lasso problem.

Example 5.8 (ℓ_1/ℓ_2-regularised multinomial logistic regression). Logistic regression is famous for classification problems. One observes for each $i \in [n]$ a class label $c_i \in \{1,\dots,q\}$ and a vector of features $z_i \in \mathbb{R}^p$. This information can be recast into a matrix $Y \in \mathbb{R}^{n\times q}$ filled by 0's and 1's: $Y_{i,k} = \mathbf{1}_{\{c_i=k\}}$. A matrix $B \in \mathbb{R}^{p\times q}$ is formed by q vectors encoding the hyperplanes for the linear classification. The multinomial ℓ_1/ℓ_2 regularized regression reads:

$$\min_{B\in\mathbb{R}^{p\times q}} \sum_{i=1}^{n}\left(\sum_{k=1}^{q} -Y_{i,k}z_i^\top B_{:,k} + \log\left(\sum_{k=1}^{q}\mathbb{E}\left(z_i^\top B_{:,k}\right)\right)\right) + \lambda\sum_{j=1}^{p}\|B_{j,:}\|_2,$$

Like the multi-task Lasso problem, this problem can be solved with proximal coordinate descent as long as we consider blocks of variables corresponding to the columns of B rather than single variables.

Exercise:

1. Find the proximal operator of the non-smooth function $g(B) = \lambda\sum_{j=1}^{p}\|B_{j,:}\|_2$.

2. Give the expression of the partial derivatives of the smooth function

$$f(B) = \sum_{i=1}^{n}\left(\sum_{k=1}^{q} -Y_{i,k}z_i^\top B_{:,k} + \log\left(\sum_{k=1}^{q}\mathbb{E}\left(z_i^\top B_{:,k}\right)\right)\right)$$

3. Give an estimate of the p block-wise Lipschitz constants of ∇f.

4. Write the proximal coordinate descent method for ℓ_1/ℓ_2-regularised multinomial logistic regression.

5.4 Stochastic dual coordinate ascent for support vector machines

In this section, we focus on the linear Support Vector Machines (SVM) problem

$$\min_{w\in\mathbb{R}^p} C\sum_{i=1}^{n}\max(0, 1 - y_i z_i^\top w) + \frac{1}{2}\|w\|_2^2$$

where C is a positive real number, $y \in \mathbb{R}^n$ and $\forall i,\ z_i \in \mathbb{R}^p$. Note that we consider the formulation without intercept. The objective function contains a non-smooth and non-separable term so we cannot apply coordinate descent to it.
However, a dual formulation of the SVM problem is given by

$$\max_{\alpha \in \mathbb{R}^n} -\frac{1}{2}\sum_{j=1}^{p}\big(\sum_{i=1}^{n} Z_{i,j} y_i \alpha^{(i)}\big)^2 + \sum_{i=1}^{n}\alpha^{(i)} - \iota_{[0,C]^n}(\alpha).$$

The objective function of this problem does decompose into a differentiable concave function $f(\alpha) = -\frac{1}{2}\sum_{j=1}^{p}\big(\sum_{i=1}^{n} Z_{i,j}y_i\alpha^{(i)}\big)^2 + \sum_{i=1}^{n}\alpha^{(i)}$ and a nonsmooth concave and separable function $g(\alpha) = -\iota_{[0,C]^n}(\alpha)$. Stochastic Dual Coordinate Ascent (SDCA) is proximal coordinate ascent (the version of coordinate descent for concave functions) on this problem.

Exercise 5.1. Write an implement of SDCA. It may be useful to maintain "residuals" w_k defined by $w_k^{(j)} = \sum_{i=1}^{n} Z_{i,j}y_i\alpha_k^{(i)}$ for all $j \in \{1, \dots, p\}$.

Even if we are running the algorithm in the dual, we are interested in the primal problem. The following result shows that we can recover a good primal solution from the dual solution and gives theoretical guarantees for the convergence in the primal.

Theorem 5.7 ([SSZ13]). *Let us define a primal point $w_k = Z^\top \mathrm{Diag}(y)\,\alpha_k$, where $(\alpha_k)_{k\geq 0}$ is generated by SDCA. The duality gap satisfies for all $K \geq n$,*

$$\mathbb{E}\Big[\frac{1}{K}\sum_{k=K}^{2K-1} P(w_k) - D(\alpha_k)\Big] \leq \frac{n}{K+n}\Big((1-\frac{1}{n})(D(\alpha_*) - D(\alpha_0)) + \frac{1}{2}\|\alpha_* - \alpha_0\|_L^2\Big) + \frac{n}{2K}C^2\sum_{i=1}^{n} L_i$$

where the primal value is $P(w_k) = C\sum_{i=1}^{n}\max(0, 1 - y_i z_i^\top w_k) + \frac{1}{2}\|w_k\|_2^2$, the dual value is $D(\alpha_k) = -\frac{1}{2}\|Z^\top \mathrm{Diag}(y)\,\alpha_k\|_2^2 + \sum_{i=1}^{n}\alpha_k^{(i)} - \iota_{[0,C]^n}(\alpha_k)$ and $\forall i, L_i = y_i^2\|z_i\|^2$.

Proof. As SDCA solves the dual problem with coordinate ascent, by Theorem 5.6,

$$\mathbb{E}[D(\alpha_*) - D(\alpha_{k+1})] \leq \frac{n}{k+n}\Big((1-\frac{1}{n})(D(\alpha_*) - D(\alpha_0)) + \frac{1}{2}\|\alpha_* - \alpha_0\|_L^2\Big).$$

The goal of the theorem is to upper bound $\mathbb{E}[P(w_k) - D(\alpha_k)]$ by quantities involving $\mathbb{E}[D(\alpha_*) - D(\alpha_{k+1})]$. Note that by weak duality, $P(w_k) - D(\alpha_k) \geq D(\alpha_*) - D(\alpha_{k+1})$ but what we need is an inequality in the other way. For this, we will need to use the fact that $(\alpha_k)_{k\geq 0}$ is generated by the coordinate ascent method.
Using the feasibility of α_k and the definition of w_k, we can simplify $D(\alpha_{k+1})$ as

$$D(\alpha_{k+1}) = -\frac{1}{2}\|Z^\top \mathrm{Diag}\,(y)\,\alpha_{k+1}\|_2^2 + \sum_{i=1}^{n}\alpha_{k+1}^{(i)} - \iota_{[0,C]^n}(\alpha_{k+1}) = -\frac{1}{2}\|w_{k+1}\|^2 + e^\top\alpha_{k+1}$$

As $\alpha_{k+1} = \alpha_k + U_{i_{k+1}}(\bar\alpha_{k+1}^{(i_{k+1})} - \alpha_k^{(i_{k+1})})$, $w_{k+1} = w_k + z_{i_{k+1}}y_{i_{k+1}}(\bar\alpha_{k+1}^{(i_{k+1})} - \alpha_k^{(i_{k+1})})$ and

$$D(\alpha_{k+1}) = -\frac{1}{2}\|w_k + z_{i_{k+1}}y_{i_{k+1}}(\bar\alpha_{k+1}^{(i_{k+1})} - \alpha_k^{(i_{k+1})})\|^2 + e^\top\alpha_k + \bar\alpha_{k+1}^{(i_{k+1})} - \alpha_k^{(i_{k+1})}$$

To simplify notations, we will write here $i = i_{k+1}$. Note that

$$\begin{aligned}\bar\alpha_{k+1}^{(i)} &= \arg\max_{a\in[0,C]}(y_i z_i Z^\top \mathrm{Diag}\,(y)\,\alpha_k + 1)(a - \alpha_k^{(i)}) - \frac{\|y_i z_i\|^2}{2}(a - \alpha_k^{(i)})^2 \\ &= \arg\max_{a\in[0,C]} -\frac{1}{2}\|w_k + z_i y_i(a - \alpha_k^{(i)})\|^2 + a - \alpha_k^{(i)}.\end{aligned}$$

So let us consider $\phi : x \mapsto C \max(0, 1-x)$, $u \in -\partial\phi(y_i z_i^\top w_k) \subseteq [0, C]$ and $s \in [0,1]$.

$$\begin{aligned}
D(\alpha_{k+1}) &= \max_{a \in [0,C]} -\frac{1}{2}\|w_k + z_i y_i (a - \alpha_k^{(i)})\|^2 + e^\top \alpha_k + a - \alpha_k^{(i)} \\
&\geq -\frac{1}{2}\|w_k + z_i y_i ((su + (1-s)\alpha_k^{(i)}) - \alpha_k^{(i)})\|^2 + e^\top \alpha_k + (su + (1-s)\alpha_k^{(i)}) - \alpha_k^{(i)} \\
&\geq -\frac{1}{2}\|w_k + z_i y_i s(u - \alpha_k^{(i)})\|^2 + e^\top \alpha_k + s(u - \alpha_k^{(i)}) \\
&= -\frac{1}{2}\|w_k\|^2 - \frac{s^2}{2}\|z_i y_i\|_2^2 (u - \alpha_k^{(i)})^2 - s(u - \alpha_k^{(i)}) y_i z_i^\top w_k + e^\top \alpha_k + s(u - \alpha_k^{(i)}) \\
&= D(\alpha_k) - \frac{s^2}{2}\|z_i y_i\|_2^2 (u - \alpha_k^{(i)})^2 - s(u - \alpha_k^{(i)}) y_i z_i^\top w_k + s(u - \alpha_k^{(i)})
\end{aligned}$$

As $u \in -\partial\phi(y_i z_i^\top w_k) \subseteq [0, C]$ and $\phi^*(q) = q + \iota_{[-C,0]}(q)$, Fenchel-Young equality leads to: $\phi(y_i z_i^\top w_k) - u = -u y_i z_i^\top w_k$. Hence,

$$D(\alpha_{k+1}) \geq D(\alpha_k) - \frac{s^2}{2}\|z_i y_i\|_2^2 (u - \alpha_k^{(i)})^2 + s\phi(y_i z_i^\top w_k) + s\alpha_k^{(i)} y_i z_i^\top w_k - s\alpha_k^{(i)}$$

Applying conditional expectation, we get

$$\mathbb{E}[D(\alpha_{k+1}) | \mathcal{F}_k] \geq D(\alpha_k) - \frac{s^2}{2n}\sum_{i=1}^n \|z_i y_i\|_2^2 (u - \alpha_k^{(i)})^2 + \frac{s}{n}\sum_{i=1}^n \left(\phi(y_i z_i^\top w_k) + \alpha_k^{(i)} y_i z_i^\top w_k - \alpha_k^{(i)}\right)$$

Now,

$$\begin{aligned}
P(w_k) - D(\alpha_k) &= C\sum_{i=1}^n \max(0, 1 - y_i z_i^\top w_k) + \frac{1}{2}\|w_k\|_2^2 - (-\frac{1}{2}\|w_k\|^2 + e^\top \alpha_k) \\
&= \sum_{i=1}^n \phi(y_i z_i^\top w_k) + \alpha_k^{(i)} y_i z_i^\top w_k - \alpha_k^{(i)}
\end{aligned}$$

So that

$$\begin{aligned}
\mathbb{E}[D(\alpha_{k+1}) | \mathcal{F}_k] - D(\alpha_k) &\geq -\frac{s^2}{2n}\sum_{i=1}^n \|z_i y_i\|_2^2 (u - \alpha_k^{(i)})^2 + \frac{s}{n}(P(w_k) - D(\alpha_k)) \\
&\geq -\frac{s^2}{2n}\sum_{i=1}^n (\|z_i y_i\|_2^2) C^2 + \frac{s}{n}(P(w_k) - D(\alpha_k))
\end{aligned}$$

where the last inequality derives from $\alpha_k^{(i)} \in [0, C]$ and $u \in [0, C]$.
We apply total expectation and we sum for k from K_1 to $K-1$:

$$\begin{aligned}
\frac{s}{n}\sum_{k=K_1}^{K-1} \mathbb{E}[P(w_k) - D(\alpha_k)] &\leq \mathbb{E}[D(\alpha_K)] - \mathbb{E}[D(\alpha_{K_1})] + \frac{s^2}{2n}C^2 \sum_{i=1}^n (\|z_i y_i\|_2^2)(K - K_1) \\
&\leq \mathbb{E}[D(\alpha_*)] - \mathbb{E}[D(\alpha_{K_1})] + \frac{s^2}{2n}C^2 \sum_{i=1}^n (\|z_i y_i\|_2^2)(K - K_1) \\
&\leq \frac{c_0 n}{K_1 + n} + \frac{s^2}{2n}C^2 \sum_{i=1}^n (\|z_i y_i\|_2^2)(K - K_1)
\end{aligned}$$

where $c_0 = (1 - \frac{1}{n})(D(\alpha_*) - D(\alpha_0)) + \frac{1}{2}\|\alpha_* - \alpha_0\|_L^2$. Choosing $K = 2K_1$ and $s = \frac{n}{K_1}$, we obtain, for $K_1 \geq n$ (because we need $s \leq 1$)

$$\frac{1}{K_1}\sum_{k=K_1}^{2K_1-1}\mathbb{E}[P(w_k) - D(\alpha_k)] \leq \frac{c_0 n}{K_1 + n} + \frac{n}{2K_1}C^2\sum_{i=1}^{n}(\|z_i y_i\|_2^2)$$

□

Bibliography

[Bot10] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010. 14

[BT13] Amir Beck and Luba Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013. 26

[CSS02] Michael Collins, Robert E Schapire, and Yoram Singer. Logistic regression, adaboost and bregman distances. *Machine Learning*, 48(1-3):253–285, 2002. 24

[DHS11] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011. 17

[KB14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 20

[KHR20] Dmitry Kovalev, Samuel Horváth, and Peter Richtárik. Don't jump through hoops and remove those loops: Svrg and katyusha are better without the outer loop. In *Algorithmic Learning Theory*, pages 451–467. PMLR, 2020. 14

[Nes12] Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012. 26

[Pow76] MJD Powell. Some global convergence properties of a variable metric algorithm for minimization without exact line searches. *Nonlinear programming*, 9:53–72, 1976. 24

[RT14] Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014. 28

[Sei74] Ludwig Seidel. Ueber ein verfahren, die gleichungen, auf welche die methode der kleinsten quadrate führt, sowie lineäre gleichungen überhaupt, durch successive annäherung aufzulösen:(aus den abhandl. dk bayer. akademie ii. cl. xi. bd. iii. abth. *Verlag der k. Akademie*, 1874. 23

[SSZ13] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14(1):567–599, 2013. 31

[SZL13] Tom Schaul, Sixin Zhang, and Yann LeCun. No more pesky learning rates. In *International Conference on Machine Learning*, pages 343–351. PMLR, 2013. 17

[Tse01] Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001. 27

[War63] Jack Warga. Minimizing certain convex functions. *Journal of the Society for Industrial & Applied Mathematics*, 11(3):588–593, 1963. 23

[Wri15] Stephen J Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.