Exercise sheet 3: Exam 2023-2024 (adapted)

Optimization for Machine Learning, M2 MIAGE ID Apprentissage

Updated version: January 17, 2025

Dauphine | PSL
UNIVERSITÉ PARIS

Exercise 1: A nonconvex problem

Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ be a dataset with $y_i \in (0,1)$ for every i. Given the following loss function:

$$\ell(h, y) := \left(y - \frac{1}{1+\exp(-h)}\right)^2, \tag{1}$$

we consider the optimization problem corresponding to fitting a linear model to the data, given by

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}}\, \phi(\boldsymbol{w}) := \frac{1}{n}\sum_{i=1}^n \ell\left(\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w}, y_i\right) = \frac{1}{n}\sum_{i=1}^n \left(y_i - \frac{1}{1+\exp(-\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{w})}\right)^2. \tag{2}$$

The function ϕ is $\mathcal{C}^2$ and it is nonconvex.

a) Justify that 0 is a lower bound on the function ϕ. Is it necessarily its optimal value?

b) We wish to apply the gradient descent algorithm to (2).
   i) Write the iteration of this algorithm with an arbitrary stepsize.
   ii) Give two possible choices for the stepsize.
   iii) Under appropriate assumptions, what is the complexity of the algorithm on a problem such as (2)? What quantity does this result apply to?

c) Suppose that gradient descent returns a point with a zero gradient. Is it necessarily a minimum?

d) State the second-order necessary optimality conditions for problem (2). Is a point satisfying these conditions a minimum?

e) Suppose that we start gradient descent from a randomly selected initial point, and that we observe that the method converges towards a point satisfying the second-order necessary optimality conditions. What result seen in class does this illustrate?

Exercise 2: Convex matrix recovery

We consider a data matrix $\boldsymbol{X} \in \mathbb{R}^{d_1 \times d_2}$, and a subset $\mathcal{S} \subset \{1, \dots, d_1\} \times \{1, \dots, d_2\}$. The *matrix recovery* problem consists in finding the best approximation of $\boldsymbol{X}$ given some observed entries $\{\boldsymbol{X}_{ij} \mid (i,j) \in \mathcal{S}\}$. This amounts to solving the following optimization problem:

$$\underset{\boldsymbol{W} \in \mathbb{R}^{d_1 \times d_2}}{\text{minimize}} \frac{1}{2} \sum_{(i,j) \in \mathcal{S}} (\boldsymbol{W}_{ij} - \boldsymbol{X}_{ij})^2 \tag{3}$$

For any value of $\mathcal{S}$, the problem (3) can be reformulated as a vector optimization problem. Indeed, if we denote by $\boldsymbol{w} \in \mathbb{R}^d$ the concatenation of all columns of $\boldsymbol{W} \in \mathbb{R}^{d_1 \times d_2}$ (with $d = d_1 d_2$), problem (3) can be rewritten as

$$\underset{\boldsymbol{w} \in \mathbb{R}^d}{\text{minimize}} f(\boldsymbol{w}) := \frac{1}{2} \sum_{(i,j) \in \mathcal{S}} \left([\boldsymbol{w}]_{i+(j-1)d_1} - \boldsymbol{X}_{ij} \right)^2 . \tag{4}$$

The objective function of problem (4) is convex and $\mathcal{C}^1$

a) The objective function of problem (4) is convex and $\mathcal{C}^1$.
   i) How can we characterize a solution of problem (4) using the derivative of f?
   ii) Give an example of a $\mathcal{C}^1$, convex function that does not possess a minimum.

b) The standard complexity of gradient descent on a convex problem such as (4) is $\mathcal{O}(\epsilon^{-1})$. What quantity does this rate apply to?

c) What is the corresponding complexity for accelerated gradient? What is the algorithmic idea behind this method?

d) We consider the special case in which all entries of the matrix are observed, i.e. $\mathcal{S} = \{1, \dots, d_1\} \times \{1, \dots, d_2\}$.
   i) In that case, the objective function of (3) (or, equivalently, that of (4)) is strongly convex. What can be said about local minima of strongly convex functions?
   ii) Justify that the problem (3) has a unique global minimum in the context of this question. What is this minimum?
   iii) When all entries are observed, the objective function f is a strongly convex quadratic function. Name one algorithm other than accelerated gradient that possesses better complexity guarantees than gradient descent on this problem.

Exercise 4: Stochastic gradient

In this exercise, we consider a finite-sum minimization problem of the form :

$$\underset{\boldsymbol{w}\in\mathbb{R}^d}{\text{minimize}}\, f(\boldsymbol{w}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{w}), \tag{5}$$

where every function f_i is assumed to be $\mathcal{C}^1$ and depends solely on the ith element in a dataset $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$.

a) Why is the structure of problem (5) amenable to applying stochastic gradient techniques?

b) Write the stochastic gradient iteration with a decreasing step size proportional to $\frac{1}{k+1}$, with k being the iteration index.

c) What is the cost of a stochastic gradient iteration in terms of accesses to the dataset? How does this compare to the cost of a gradient descent iteration?

d) Suppose that we perform K iterations of stochastic gradient and K iterations of gradient descent where $K = n\,E$ for some integer $E \geq 1$. We wish to compare the performance of both algorithms.

   i) Justify that comparing the values of f obtained for the final iterates of both methods is not a good metric.

   ii) Propose a relevant metric for comparing both methods without performing more iterations.

e) We now assume that the various items in the dataset are distributed across r processors, with r being a value between 1 and n.

   i) Write the iteration of a batch stochastic gradient method with a constant batch size equal to n_b, and a constant step size.

   ii) What can be the computational advantage of setting $n_b = r$?

   iii) If $r \approx n$, however, what is a possible drawback of using $n_b = r$?

   iv) If $1 < r \ll n$, setting $n_b = r$ corresponds to doing mini-batching. Does that necessarily lead to a better performance than $n_b = 1$? Justify your answer.

f) We finally consider an iteration of the Adam variant on stochastic gradient. Explain how this iteration differs from the vanilla stochastic gradient iteration.

Solutions

Solutions for Exercise 1

a) The function ϕ is an average of nonnegative terms, thus $\phi(\boldsymbol{w}) \geq 0$ for any $\boldsymbol{w} \in \mathbb{R}^d$, and 0 is a lower bound on the function ϕ. In order to have $0 = \min_{\boldsymbol{w}\in\mathbb{R}^d} \phi(\boldsymbol{w})$, there must exist $\boldsymbol{w} \in \mathbb{R}^d$ such that $\phi(\boldsymbol{w}) = 0$. Since the existence of such a $\boldsymbol{w}$ depends on the problem data, 0 is not necessarily the optimal value.

b) i) $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha_k \nabla\phi(\boldsymbol{w}_k)$, where $\alpha_k > 0$.

ii) The stepsize sequence $\{\alpha_k\}$ can be chosen constant ($\alpha_k = \alpha > 0$ for all k) or decreasing ($\alpha_k \updownarrow 0$, e.g. $\alpha_k = \frac{1}{k+1}$). Another option is an adaptive stepsize choice, for instance using a line search.

iii) The complexity of the method on a nonconvex optimization problem such as (2) is $\mathcal{O}(\epsilon^{-2})$, where this bound applies to the number of iterations of gradient descent. More precisely, given $\epsilon > 0$, gradient descent computes a point such that $\|\nabla f(\boldsymbol{w}_K)\| \leq \epsilon$ (or, equivalently, $\min_{0\leq k\leq K-1} \|\nabla f(\boldsymbol{w}_k)\| \leq \epsilon$) when run for $K = \mathcal{O}(\epsilon^{-2})$ iterations.

c) Since the function is nonconvex, a point with zero gradient is not necessarily a minimum, and can either be a saddle point or a (local or global) maximum.

d) If $\bar{\boldsymbol{w}} \in \mathbb{R}^d$ is a local minimum of problem (2), then we have

$$\nabla\phi(\bar{\boldsymbol{w}}) = \mathbf{0} \quad \text{and} \quad \nabla^2\phi(\bar{\boldsymbol{w}}) \succeq \mathbf{0}.$$

e) A result seen in class states that gradient descent with random initialization converges almost surely towards a point satisfying the second-order necessary optimality conditions.

Solutions for Exercise 2

a) i) The set of solutions of problem (4) corresponds to the set of vectors $\boldsymbol{w} \in \mathbb{R}^d$ such that $\nabla f(\boldsymbol{w}) = \mathbf{0}$.

ii) A linear function $\boldsymbol{w} \mapsto \boldsymbol{a}^{\mathrm{T}}\boldsymbol{w}$ is a $\mathcal{C}^1$, convex function that does not have a minimum when $\boldsymbol{a} \neq \mathbf{0}$.

b) The complexity of gradient in the convex case corresponds to a bound on the number of iterations necessary to satisfy $f(\boldsymbol{w}_k) - \min_{\boldsymbol{w}\in\mathbb{R}^d} f(\boldsymbol{w}) \leq \epsilon$ for some tolerance $\epsilon > 0$.

c) The corresponding complexity for accelerated gradient is $\mathcal{O}(\epsilon^{-1/2}$. This method is based on combining gradient steps with momentum terms, that correspond to the step taken at the previous iteration.

d) i) A strongly convex function has a unique local minimum.

ii) It suffices to notice that the objective function is nonnegative, and that

$$\frac{1}{2}\sum_{(i,j)\in\mathcal{S}} (\boldsymbol{W}_{ij} - \boldsymbol{X}_{ij})^2 = 0 \quad \Leftrightarrow \quad \boldsymbol{W}_{ij} = \boldsymbol{X}_{ij} \ \forall (i,j).$$

As a result, $\boldsymbol{X}$ is the unique global minimum of the problem. *NB: When we do not observe all entries of $\boldsymbol{X}$, in general there are many global optima!*

iii) The heavy-ball method, like accelerated gradient, has a better complexity than gradient descent on strongly convex quadratic problems.

Solutions for Exercise 4

a) The objective of problem (5) has a finite-sum structure, where each term in the sum depends on a different part of a dataset. This is the setting in which stochastic gradient are interesting algorithms to apply

b) $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \frac{\alpha}{k+1}\nabla f_{i_k}(\boldsymbol{w}_k)$, where $\alpha > 0$ and i_k is an index drawn randomly in $\{1, \dots, n\}$.

c) A stochastic gradient iteration costs one access to a data point. By contrast, a gradient descent computes $\nabla f(\boldsymbol{w}_k) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\boldsymbol{w}_k)$, which costs n accesses to data points.

d) i) Let $\boldsymbol{w}_K^{GD}$ and $\boldsymbol{w}_K^{SG}$ denote the final iterates of gradient descent and stochastic gradient, respectively. To compute $\boldsymbol{w}_K^{GD}$, one needs to run K iterations of gradient descent or, equivalently, $n\,K$ accesses to data points. Meanwhile, computing $\boldsymbol{w}_K^{SG}$ requires only K accesses to data points. As a result, a comparison between $f(\boldsymbol{w}_K^{GD})$ and $f(\boldsymbol{w}_K^{SG})$ is unfair to stochastic gradient, because gradient descent has used a lot more accesses to data points.

ii) Since $K = n\,E$, stochastic gradient has been run for E epochs, where one epoch corresponds to the cost of accessing n data points. It is thus fairer to compare $f(\boldsymbol{w}_K^{SG}) = f(\boldsymbol{w}_{nE}^{SG})$ to $f(\boldsymbol{w}_E^{GD})$, since $\boldsymbol{w}_E^{GD}$ is the iterate output by gradient descent at the cost of E epochs.

e) i) $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \alpha\,\frac{1}{n_b}\sum_{i\in\mathcal{S}_k} \nabla f_i(\boldsymbol{w}_k)$, where $\alpha > 0$, and $\mathcal{S}_k$ is a set of n_b random indices drawn with or without replacement in $\{1, \dots, n\}$.

ii) When $n_b = r$, one can leverage the presence of r processors and evaluate the gradients $\nabla f_i(\boldsymbol{w}_k)$ in the batch in parallel.s

iii) When $r \approx n$, using $n_b = r$ corresponds to a large batch regime, and the performance of the method resembles that of gradient descent. It is thus likely that the method will converge more slowly than vanilla stochastic gradient ($n_b = 1$).

iv) Mini-batching might lead to improvement compared to vanilla stochastic gradient, in that it uses more data points to build a gradient estimator (thus this estimator has less variance). On the other hand, batch methods with $n_b > 1$ can be more sensitive to redundancies in the dataset, and might be outperformed by vanilla stochastic gradient in such a setting.
Other possible elements of justification:

- (positive) In presence of parallelism, the computational cost of mini-batch methods is comparable to that of vanilla stochastic gradient.
- (negative) The per-iteration cost of a batch method is necessarily higher than that of vanilla stochastic gradient, hence the method will perform less updates of the iterate.

f) Adam differs from vanilla stochastic gradient in two aspects. First, a momentum term is incorporated in the iteration, that involves a geometric average of all previous steps. Secondly, a diagonal scaling of the stepsize is performed, using again a geometric average of past stochastic gradients (thus a different stepsize is used for each coordinate of $\boldsymbol{w}$).