Stochastic methods for optimization and sampling

Exam 2025

The precision and conciseness of the justifications are part of the evaluation. All exercises are independent and can be addressed in the order of your choice. Within a specific exercise, some questions may depend on the answers of the previous ones. If you did not solve a given question, you may still use its result in subsequent questions. It is not necessary to have completed all the questions to get an excellent grade.

Exercise 1: MCMC

Let p and q be two probabilities on a finite state space E, with $0 < p(x) \leq cq(x)$ for some constant $c > 0$. Let $(Y_n)_{n\geq 0}$ be independent random variables with the same distribution q. We define a sequence of random variables $(X_n)_{n\geq 0}$ as follows:

- X_0 is a random variable with distribution q, independent of the sequence $(Y_n)_{n\geq 0}$;
- X_{n+1} is defined from X_n by drawing a random variable U_n uniformly on $[0,1]$, independent of the other previously defined variables, and we define

$$X_{n+1} = Y_{n+1} \quad \text{if} \quad U_n \leq \frac{p(Y_{n+1})}{cq(Y_{n+1})}$$

$$X_{n+1} = X_n \quad \text{if} \quad U_n > \frac{p(Y_{n+1})}{cq(Y_{n+1})}.$$

1. Justify that the sequence $(X_n)_{n\geq 0}$ is an irreducible Markov chain (short answer expected).

   The law of X_{n+1} only depends on the value of X_n, so the sequence of random variables $(X_n)_n$ forms a Markov chain.

   Moreover, for any two states $x, y \in E$, assuming $X_n = x$, the probability of transitioning from x to y is at least the probability that $Y_{n+1} = y$ and that the transition is accepted, that is $U_n \leq \frac{p(y)}{cq(y)}$. Both events have a positive probability, since $q(y) > 0$ and $\frac{p(y)}{cq(y)}$ by assumption. Hence, there exists a non-zero probability path of 1 between any two states x, y, which ensures the chain is irreducible.

2. Show that the transition probability $P(x,y) = \mathbb{P}(X_{n+1} = y | X_n = x)$ of $(X_n)_{n\geq 0}$ for $x, y \in E$ satisfies

$$P(x,y) = \begin{cases} \dfrac{p(y)}{c} & \text{if } x \neq y \\ 1 - \dfrac{1}{c} \displaystyle\sum_{x' \neq x} p(x') & \text{otherwise.} \end{cases}$$

   Let $x, y \in E$ be such that $x \neq y$. We have

$$\begin{aligned} P(x,y) &= \mathbb{P}(X_{n+1} = y | X_n = x) \\ &= \mathbb{P}\left(Y_{n+1} = y \text{ and } U_n \leq \frac{p(Y_{n+1})}{cq(Y_{n+1})} \,\middle|\, X_n = x \right) \\ &= \mathbb{P}\left(Y_{n+1} = y \text{ and } U_n \leq \frac{p(y)}{cq(y)} \,\middle|\, X_n = x \right) \end{aligned}$$

$$= \mathbb{P}\left(Y_{n+1} = y \,\middle|\, X_n = x\right) \mathbb{P}\left(U_n \leq \frac{p(y)}{cq(y)} \,\middle|\, X_n = x\right)$$
$$= q(y) \cdot \frac{p(y)}{cq(y)}$$
$$= \frac{q(y)}{c}.$$

Now, if $y = x$, we use the property $\sum_{z \in E} P(x, z) = 1$ for any $x \in E$ to obtain

$$P(x, x) = 1 - \frac{1}{c} \sum_{x \neq x'} p(x').$$

3. What is the difference between this sampling procedure and the rejection method?

   In the rejection method, we define iid random variables $(Y_n)_n$ with probability distribution q satisfying $p \leq cq$ for some $c \geq 1$ and iid uniform random variables $(U_n)_n$ on $[0, 1]$. We keep the Y_n's such that $U_n \leq \frac{p(Y_{n+1})}{cq(Y_{n+1})}$ and discard all the other ones, which yields a sequence of iid random variables with distribution p. In contrast, the algorithm studied here defines a Markov chain, not a sequence of iid random variables. The random variables $(Y_n)_n$ also represent proposed samples that are accepted or rejected based on whether $U_n \leq \frac{p(Y_{n+1})}{cq(Y_{n+1})}$ is true or not, but when a sample is rejected, the Markov chain remains at the current point.

4. Recall the Metropolis-Hastings algorithm and discuss the differences with the algorithm studied here.

   To sample from a target probability density p, the Metropolis-Hastings proceeds by defining an auxiliary Markov transition matrix Q and iterating the steps outlined in the pseudo-code below

   Algorithm 1: Metropolis-Hastings algorithm

   (a) Initialize X_0 according to any initial law;

   (b) For i from 1 to n,

   Draw $Y_i \sim Q(X_{i-1}, \cdot)$

   If $U_i < \frac{p(Y_i)Q(Y_i, X_{i-1})}{p(X_{i-1})Q(X_{i-1}, Y_i)}$ then $X_i = Y_i$

   Otherwise, $X_i = X_{i-1}$

   (c) Output $(X_0, \ldots, X_n)$.

   In the Metropolis-Hastings algorithm, the law of the random variable Y_{n+1} depends on X_{i-1} in general. The special case where it does not, which is equivalent to the sequence $(Y_n)_n$ being iid, is when $Q(x, \cdot)$ does not depend on x, that is, when Q is of the form

$$Q = \begin{pmatrix} q(1) \ldots q(d) \\ \ddots \\ q(1) \ldots q(d) \end{pmatrix}.$$

   But in this case, the acceptance ratio of the Metropolis-Hastings algorithm can be written as

$$\min\left(1, \frac{p(Y_i)Q(Y_i, X_{i-1})}{p(X_{i-1})Q(X_{i-1}, Y_i)}\right) = \frac{p(Y_i)q(X_{i-1})}{p(X_{i-1})q(Y_i)},$$

   which is different from the acceptance ratio of the algorithm studied in this exercise.

5. Show that p is the stationary probability measure of the chain, that is $\sum_{y \in E} P(y, x)p(y) = p(x)$, for any $x \in E$. Is the chain reversible?

   We directly show that p is reversible, which implies that it is stationary. We have for any $x \neq y$:

$$p(x)P(x, y) = p(x)\frac{p(y)}{c} = p(y)\frac{p(x)}{c} = p(y)P(y, x),$$

If $x = y$, then the relation $p(x)P(x,y) = p(y)P(y,x)$ is clear. Therefore, the Markov chain is reversible, hence stationary.

6. Deduce an estimator $\hat{p}(x)$ of $p(x)$ for any $x \in E$ and justify its almost sure convergence toward $p(x)$.
For any $x \in E$, we define

$$\hat{p}(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{X_n = x\}.$$

The Markov chain X_n is defined on a finite space and is irreducible, and p is its stationary measure. Therefore, by the ergodic theorem, we have that

$$\hat{p}(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{X_n = x\} \xrightarrow{a.s.} \sum_{z\in E} \mathbf{1}\{z = x\}p(z) = p(x).$$

Exercise 2: Mini-batch SGD in the interpolation setting

Let $n, d \geq 1$ be integers, and consider the optimization problem

$$\min_{\theta\in\mathbb{R}^d} F(\theta), \quad \text{where} \quad F(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta), \quad \forall\theta \in \mathbb{R}^d \tag{1}$$

for convex and differentiable functions f_i. In modern machine learning, a special case of the optimization problem above has gained considerable importance in the past few years. Specifically, it corresponds to the following restriction for the functions f_i.

Assumption A (Interpolation setting): *there exists a vector θ^* that minimizes all the functions f_i simultaneously:*

$$\exists\theta^* \in \mathbb{R}^d,\ \forall i \in \{1,\dots,n\}: \quad \nabla f_i(\theta^*) = 0_{\mathbb{R}^d}.$$

It is immediate that the vector θ^* from Assumption A is also a solution to the optimization problem (1).
Notation. For any subset $v \subseteq \{1,\dots,n\}$ of cardinality m, we write for ease

$$f_v = \frac{1}{m}\sum_{i\in v} f_i.$$

In particular, if $m = n$ and $v = \{1,\dots,n\}$, then $f_v = F$. We recall the pseudo-code below

Algorithm 2: Mini-batch SGD

Start from $\theta_0 \in \mathbb{R}^d$.

Until termination condition, iterate

Draw a subset $v_{k+1} \subseteq \{1,\dots,n\}$ of cardinality m u.a.r., independent of the past

$$\theta_{k+1} = \theta_k - \gamma\nabla f_{v_{k+1}}(\theta_k)$$
$$\Big(= \theta_k - \frac{\gamma}{m}\sum_{i\in v_{k+1}} \nabla f_i(\theta_k)\Big).$$

The exercise aims to study the performance of the mini-batch SGD algorithm in the *interpolation setting*.

Theorem 1. *Let $\beta, \lambda, \mu > 0$ and assume that*

1. *F is μ-strongly convex*
2. *∇F is λ-Lipschitz*

3. *Each ∇f_i is β-Lipschitz*

4. *Assumption A holds.*

Take any constant step size $\gamma \in \left(0, \frac{m(n-1)}{\lambda n(m-1)+\beta(n-m)}\right]$. Then the iterates of minibatch SGD satisfy

$$\mathbb{E}\|\theta_k - \theta^*\|^2 \leq (1-\gamma\mu)^k \|\theta - \theta_0\|^2.$$

We will also use the following lemma proved in class

Lemma 1. *Let $h : \mathbb{R}^d \to \mathbb{R}$ be such that ∇h is L-Lipschitz. Then it holds that*

$$\|\nabla h(x) - \nabla h(y)\|^2 \leq 2L\big(h(x) - h(y)\big), \quad \forall x, y \in \mathbb{R}^d.$$

Questions

1. For $n = 2$, give an example of an optimization problem (1) where Assumption A holds, and one where it does not (short answer expected).

   Example where Assumption A holds: $f_1 = f_2$.

   Example where Assumption A does not hold: $f_1(x) = \|x\|^2$ and $f_2(x) = \|x - a\|^2$ for some $a \in \mathbb{R}^d$ such that $a \neq 0$.

2. Justify that, for any integer $k \geq 0$,

$$\|\theta_{k+1} - \theta^*\|^2 = \|\theta_k - \theta^*\|^2 - 2\gamma \left\langle \theta_k - \theta^*, \nabla f_{v_{k+1}}(\theta_k) \right\rangle + \gamma^2 \left\|\nabla f_{v_{k+1}}(\theta_k)\right\|^2.$$

   By definition, we have

$$\begin{aligned}\|\theta_{k+1} - \theta^*\|^2 &= \left\|\theta_k - \theta^* - \gamma \nabla f_{v_{k+1}}(\theta_k)\right\|^2 \\ &= \|\theta_k - \theta^*\|^2 - 2\gamma \left\langle \theta_k - \theta^*, \nabla f_{v_{k+1}}(\theta_k) \right\rangle + \gamma^2 \left\|\nabla f_{v_{k+1}}(\theta_k)\right\|^2.\end{aligned}$$

3. Writing $\mathbb{E}_k$ for the expectation conditional on θ_k, prove that:

$$\mathbb{E}_k\left\|\theta_{k+1} - \theta^*\right\|^2 \leq (1-\gamma\mu)\left\|\theta_k - \theta^*\right\|^2 - 2\gamma\left[F(\theta_k) - F(\theta^*)\right] + \gamma^2 \mathbb{E}_k \left\|\nabla f_{v_{k+1}}(\theta_k)\right\|^2.$$

   (One can use the strong convexity of F: $\forall x, y \in \mathbb{R}^d : F(x) - F(y) \geq \langle \nabla F(y), x - y\rangle + \frac{\mu}{2}\|x - y\|^2$).

   We use the strong convexity property of F with $x = \theta^*$ and $y = \theta_k$

$$\begin{aligned}\mathbb{E}_k\left\|\theta_{k+1} - \theta^*\right\|^2 &= \left\|\theta_k - \theta^*\right\|^2 - 2\gamma\left\langle \theta_k - \theta^*, \nabla F(\theta_k)\right\rangle + \gamma^2 \mathbb{E}_k \left\|\nabla f_{v_{k+1}}(\theta_k)\right\|^2 \\ &= \left\|\theta_k - \theta^*\right\|^2 + 2\gamma\left\langle \nabla F(\theta_k), \theta^* - \theta_k\right\rangle + \gamma^2 \mathbb{E}_k \left\|\nabla f_{v_{k+1}}(\theta_k)\right\|^2 \\ &\leq \left\|\theta_k - \theta^*\right\|^2 + 2\gamma\left[F(\theta^*) - F(\theta_k) - \frac{\mu}{2}\|\theta^* - \theta_k\|^2\right] + \gamma^2 \mathbb{E}_k \left\|\nabla f_{v_{k+1}}(\theta_k)\right\|^2 \\ &= (1-\gamma\mu)\left\|\theta_k - \theta^*\right\|^2 - 2\gamma\left[F(\theta_k) - F(\theta^*)\right] + \gamma^2 \mathbb{E}_k \left\|\nabla f_{v_{k+1}}(\theta_k)\right\|^2.\end{aligned}$$

4. For any $m \in \{1, \ldots, n\}$, we denote by $\mathcal{D}_m$ the uniform distribution over subsets of $\{1, \ldots, n\}$ whose cardinality is m, and for any $\theta \in \mathbb{R}^d$, we define $\sigma_m^2(\theta) = \mathbb{E}_{v \sim \mathcal{D}_m}\left\|\nabla f_v(\theta)\right\|^2$. How can one rewrite $\sigma_1^2(\theta)$ and $\sigma_n^2(\theta)$ without expectations and only using sums?

   Remark: Note that here we are considering $\sigma_m^2(\theta)$, not $\sigma_m^2(\theta^*)$ as in the first exercise session.

For $m = 1$, the distribution $\mathcal{D}_m = \mathcal{D}_1$ is the uniform distribution over subsets of $\{1, \dots, n\}$ of size 1, which is exactly the uniform distribution over $\{1, \dots, n\}$. Therefore, we have

$$\begin{aligned}\sigma_1^2(\theta) &= \mathbb{E}_{I \sim \mathrm{Unif}(\{1,\dots,n\})} \left\|\nabla f_I(\theta)\right\|^2 \\ &= \frac{1}{n}\sum_{i=1}^n \left\|\nabla f_i(\theta)\right\|^2\end{aligned}$$

For $m = n$, the distribution $\mathcal{D}_m = \mathcal{D}_n$ is the uniform distribution over subsets of $\{1, \dots, n\}$ of size n. There is only one such subset, namely $\{1, \dots, n\}$ itself, hence a random variable v with distribution $\mathcal{D}_n$ is equal to the set $\{1, \dots, n\}$ almost surely. Therefore,

$$\begin{aligned}\sigma_n^2(\theta) &= \mathbb{E}_{v=\{1,\dots,n\}} \left\|\nabla f_v(\theta)\right\|^2 \\ &= \mathbb{E}_{v=\{1,\dots,n\}} \left\|\frac{1}{n}\sum_{i\in v} \nabla f_i(\theta)\right\|^2 \\ &= \left\|\frac{1}{n}\sum_{i=1}^n \nabla f_i(\theta)\right\|^2.\end{aligned}$$

5. Justify that, for any $\theta \in \mathbb{R}^d$ and any $m \in \{1, \dots, n\}$, we have

$$\sigma_m^2(\theta) = \frac{1}{m^2}\sum_{i,j=1}^n \Big\langle \nabla f_i(\theta), \nabla f_j(\theta)\Big\rangle \, \mathbb{P}_{v\sim\mathcal{D}_m}\big(i \in v \text{ and } j \in v\big).$$

We have

$$\begin{aligned}\sigma_m^2 &= \mathbb{E}_{v\sim\mathcal{D}_m}\left\|\nabla f_v(\theta)\right\|^2 = \mathbb{E}_{v\sim\mathcal{D}_m}\left\|\frac{1}{m}\sum_{i\in v}\nabla f_i(\theta)\right\|^2 \\ &= \mathbb{E}_{v\sim\mathcal{D}_m}\left[\frac{1}{m^2}\sum_{i,j\in v}\big\langle \nabla f_i(\theta),\, \nabla f_j(\theta)\big\rangle\right] \\ &= \mathbb{E}_{v\sim\mathcal{D}_m}\left[\frac{1}{m^2}\sum_{i,j=1}^n \big\langle \nabla f_i(\theta),\, \nabla f_j(\theta)\big\rangle \mathbf{1}\{i,j\in v\}\right] \\ &= \frac{1}{m^2}\sum_{i,j=1}^n \Big\langle \nabla f_i(\theta), \nabla f_j(\theta)\Big\rangle \, \mathbb{P}_{v\sim\mathcal{D}_m}\big(i \in v \text{ and } j \in v\big).\end{aligned}$$

6. Justify that for any integers $i, j \in \{1, \dots, n\}$, we have

$$\mathbb{P}_{v\sim\mathcal{D}_m}\big(i \in v \text{ and } j \in v\big) = \begin{cases} \frac{m}{n} & \text{if } i = j \\ \frac{m(m-1)}{n(n-1)} & \text{if } i \neq j. \end{cases}$$

If $i = j$, we have

$$\mathbb{P}_{v\sim\mathcal{D}_m}\big(i \in v \text{ and } j \in v\big) = \mathbb{P}_{v\sim\mathcal{D}_m}\big(i \in v\big) = \frac{m}{n}.$$

Next, if $i \neq j$, we have

$$\mathbb{P}_{v\sim\mathcal{D}_m}\big(i \in v \text{ and } j \in v\big) = \mathbb{P}_{v\sim\mathcal{D}_m}\big(i \in v \;\big|\; j \in v\big)\, \mathbb{P}_{v\sim\mathcal{D}_m}\big(j \in v\big) = \frac{m-1}{n-1}\frac{m}{n}.$$

7. Deduce that

$$\sigma_m^2(\theta) = \frac{1}{n-1}\Big(\frac{n}{m}-1\Big)\cdot\frac{1}{n}\sum_{i=1}^n \big\|\nabla f_i(\theta)\big\|^2 + \frac{n(m-1)}{m(n-1)}\big\|\nabla F(\theta)\big\|^2.$$

Combining Questions 5 and 6, we obtain

$$\begin{aligned}
\sigma_m^2 &= \frac{1}{m^2}\sum_{i,j=1}^n \Big\langle \nabla f_i(\theta), \nabla f_j(\theta)\Big\rangle \mathbb{P}_{v\sim\mathcal{D}_m}\big(i\in v \text{ and } j\in v\big)\\
&= \frac{1}{m^2}\sum_{1\le i\neq j\le n}\Big\langle \nabla f_i(\theta), \nabla f_j(\theta)\Big\rangle\frac{m(m-1)}{n(n-1)} + \frac{1}{m^2}\sum_{i=1}^n\big\|\nabla f_i(\theta)\big\|^2\frac{m}{n}\\
&= \frac{1}{m^2}\frac{m(m-1)}{n(n-1)}\Bigg[\underbrace{\sum_{1\le i,j\le n}\Big\langle \nabla f_i(\theta), \nabla f_j(\theta)\Big\rangle + \sum_{i=1}^n\big\|\nabla f_i(\theta)\big\|^2}_{=\left\|n\nabla F(\theta)\right\|^2} - \sum_{i=1}^n\big\|\nabla f_i(\theta)\big\|^2\Bigg]\\
&\quad + \frac{1}{m^2}\sum_{i=1}^n\big\|\nabla f_i(\theta)\big\|^2\frac{m}{n}\\
&= \frac{n(m-1)}{m(n-1)}\big\|\nabla F(\theta)\big\|^2 + \frac{n-m}{mn(n-1)}\sum_{i=1}^n\big\|\nabla f_i(\theta)\big\|^2\\
&= \frac{n(m-1)}{m(n-1)}\sigma_n^2(\theta) + \frac{n-m}{m(n-1)}\sigma_1^2.
\end{aligned}$$

8. Using Assumption A and Lemma 1, deduce that

$$\sigma_m^2(\theta) \le 2\big(F(\theta)-F(\theta^*)\big)\,\frac{\beta(n-m)+\lambda n(m-1)}{m(n-1)}.$$

We have

$$\begin{aligned}
\sigma_m^2(\theta) &= \frac{n-m}{m(n-1)}\sigma_1^2 + \frac{n(m-1)}{m(n-1)}\sigma_n^2(\theta)\\
&= \frac{n-m}{m(n-1)}\frac{1}{n}\sum_{i=1}^n\|\nabla f_i(\theta)\|^2 + \frac{n(m-1)}{m(n-1)}\|\nabla F(\theta)\|^2\\
&= \frac{n-m}{m(n-1)}\frac{1}{n}\sum_{i=1}^n\|\nabla f_i(\theta)-\nabla f_i(\theta^*)\|^2 + \frac{n(m-1)}{m(n-1)}\|\nabla F(\theta)-\nabla F(\theta^*)\|^2 \quad \text{by Assumption A}\\
&\le \frac{n-m}{m(n-1)}\frac{1}{n}\sum_{i=1}^n 2\beta\big(f_i(\theta)-f_i(\theta^*)\big) + \frac{n(m-1)}{m(n-1)}\cdot 2\lambda(F(\theta)-F(\theta^*)) \quad \text{by Lemma 1}\\
&= \frac{n-m}{m(n-1)}2\beta\big(F(\theta)-F(\theta^*)\big) + \frac{n(m-1)}{m(n-1)}\cdot 2\lambda(F(\theta)-F(\theta^*))\\
&= 2\big(F(\theta)-F(\theta^*)\big)\left[\frac{\beta}{n-1}\Big(\frac{n}{m}-1\Big) + \frac{n(m-1)}{m(n-1)}\lambda\right].
\end{aligned}$$

9. Using Question 3, conclude that if γ is chosen as specified in the theorem, then

$$\mathbb{E}_k\big\|\theta_{k+1}-\theta^*\big\|^2 \le \big(1-\gamma\mu\big)\big\|\theta_k-\theta^*\big\|^2.$$

We study the sign of $-2\gamma\big[F\left(\theta_k\right)-F\left(\theta^*\right)\big]+\gamma^2\mathbb{E}_k\left\|\nabla f_{v_{k+1}}\left(\theta_k\right)\right\|^2$. We have

$$-2\gamma\big[F\left(\theta_k\right)-F\left(\theta^*\right)\big]+\gamma^2\mathbb{E}_k\left\|\nabla f_{v_{k+1}}\left(\theta_k\right)\right\|^2$$

$$
\begin{aligned}
&= -2\gamma\big[F\left(\theta_k\right) - F\left(\theta^*\right)\big] + \gamma^2\sigma_m^2(\theta_k) \\
&= -2\gamma\big[F\left(\theta_k\right) - F\left(\theta^*\right)\big] + \gamma^2 \cdot 2\left(F(\theta) - F(\theta^*)\right)\left[\frac{\beta}{n-1}\left(\frac{n}{m} - 1\right) + \frac{n(m-1)}{m(n-1)}\lambda\right] \\
&= 2\gamma\underbrace{\left(F(\theta_k) - F(\theta^*)\right)}_{\geq 0}\left[-1 + \gamma\left(\frac{\beta(n-m)}{m(n-1)} + \frac{n(m-1)}{m(n-1)}\lambda\right)\right] \\
&\leq 0
\end{aligned}
$$

provided $\gamma \leq \frac{m(n-1)}{\beta(n-m)+\lambda n(m-1)}$. Hence

$$
\begin{aligned}
\mathbb{E}_k\left\|\theta_{k+1} - \theta^*\right\|^2 &\leq \left(1 - \gamma\mu\right)\left\|\theta_k - \theta^*\right\|^2 + 0 \\
&= \left(1 - \gamma\mu\right)\left\|\theta_k - \theta^*\right\|^2.
\end{aligned}
$$

10. Conclude the proof of the theorem.

    Applying total expectations on both sides, we get

    $$\mathbb{E}\left\|\theta_{k+1} - \theta^*\right\|^2 \leq \left(1 - \gamma\mu\right)\mathbb{E}\left\|\theta_k - \theta^*\right\|^2.$$

    This relation is true for any $k \in N$. Iterating it, we obtain by induction

    $$E\left\|\theta_k - \theta^*\right\|^2 \leq \left(1 - \gamma\mu\right)^k \left\|\theta_0 - \theta^*\right\|^2.$$

11. This theorem shows that minibatch SGD achieves an exponential rate of convergence in the interpolation regime even when $m = 1$, where the algorithm reduces to SGD. In comparison, what is the classical rate of SGD for Problem (1) if we only assume that F is μ-strongly convex and L Lipschitz, but do not assume that Assumption A holds? How should the step size be chosen in this case?

    Under these assumptions, the rate achieved by SGD is $O(\frac{1}{\mu^2 k})$, which is much slower than in the interpolation setting. The step size should be chosen as $\gamma_k = \frac{C}{\mu(k+1)}$, for some constant $C > 0$, whereas we can take a constant step size γ in the interpolation setting.

12. Among all favorable values $\gamma \in \left(0, \frac{m(n-1)}{\lambda n(m-1)+\beta(n-m)}\right]$, which value γ^* yields the fastest convergence?

    The convergence guarantee proved in this exercise is $\mathbb{E}\|\theta_k - \theta^*\|^2 \leq (1 - \gamma\mu)^k\|\theta - \theta_0\|^2$. The rate is therefore all the more rapid as γ is increased, and we should choose

    $$\gamma^* = \frac{m(n-1)}{\lambda n(m-1) + \beta(n-m)}.$$

13. Assume that $\beta = \lambda$. How does γ^* simplify? What optimal batch size m^* should one choose in this case to minimize the runtime of the algorithm? How does γ^* compare with the optimal step size of the gradient descent algorithm?

    If $\beta = \lambda$, then

    $$
    \begin{aligned}
    \gamma^* &= \frac{m(n-1)}{\lambda n(m-1) + \beta(n-m)} \\
    &= \frac{m(n-1)}{\lambda(mn - n + n - m)} \\
    &= \frac{1}{\lambda}.
    \end{aligned}
    $$

    Interestingly, the optimal step size, and therefore the rate, no longer depend on m. To minimize runtime, it is therefore best to choose $m^* = 1$ (SGD). Here, the gradient ∇F is λ-Lipschitz, and the optimal step size is $\gamma^* = 1/\lambda$. It is the same step size we would choose if we were running gradient descent (that is, if we had taken $m = n$).

14. We are interested in computing the runtime of mini-batch SGD with mini-batches of size m and step size γ^* to reach precision ε in the interpolation setting. Assuming that computing one gradient takes 1 second, show that the runtime is proportional to

$$\frac{m \log\left(\frac{\|\theta_0 - \theta^*\|^2}{\varepsilon}\right)}{\log\left(1 - \gamma^* \mu\right)}.$$

Each iteration requires computing m gradients, and we need to run the algorithm for k_m steps, where

$$k_m = \frac{\log\left(\frac{\|\theta_0 - \theta^*\|^2}{\varepsilon}\right)}{\log\left(1 - \gamma^* \mu\right)}.$$

The runtime is proportional to mk_m, that is

$$mk_m = \frac{m \log\left(\frac{\|\theta_0 - \theta^*\|^2}{\varepsilon}\right)}{\log\left(1 - \gamma^* \mu\right)}.$$

15. Assuming that μ is small enough to use the approximation $\log(1 - \gamma^* \mu) \approx -\gamma^* \mu$ for any m, simplify this expression and minimize it as a function of $m \in \{1, \dots, n\}$. Note that the result should be different depending on whether $\lambda n > \beta$ or $\lambda n \leq \beta$. What do you notice? How does the optimal batch size m^* differ from the classical optimal batch size if the interpolation setting is not in force?

Using this approximation, our expression for the runtime can be simplified as

$$\begin{aligned}
mk_m &= \frac{m \log\left(\frac{\|\theta_0 - \theta^*\|^2}{\varepsilon}\right)}{\log\left(1 - \gamma^* \mu\right)} \\
&\approx \frac{m \log\left(\frac{\|\theta_0 - \theta^*\|^2}{\varepsilon}\right)}{-\gamma^* \mu} \\
&= \frac{m \log\left(\frac{\|\theta_0 - \theta^*\|^2}{\varepsilon}\right)}{\mu} \frac{\lambda n(m-1) + \beta(n-m)}{m(n-1)} \\
&= \frac{\log\left(\frac{\|\theta_0 - \theta^*\|^2}{\varepsilon}\right)}{\mu(n-1)} \left(\lambda n(m-1) + \beta(n-m)\right) \\
&= \frac{\log\left(\frac{\|\theta_0 - \theta^*\|^2}{\varepsilon}\right)}{\mu(n-1)} \left(m(\lambda n - \beta) - \lambda n + \beta n\right).
\end{aligned}$$

The only dependence on m is in the term $m(\lambda n - \beta) - \lambda n + \beta n$. To minimize runtime, it suffices to minimize $m(\lambda n - \beta)$ as a function of m. If $\lambda n > \beta$, we should choose $m = 1$. Otherwise, we should choose $m = n$. We therefore highlight an "all-or-nothing" phenomenon, where m should either take the smallest or the largest possible value. This is in contrast with the minibatch SGD algorithm seen in class, where the parameter m controls a tradeoff between fast convergence rate and high per-iteration cost. As a reminder, the optimal batch size when the interpolation regime is not in force is

$$m^* = \frac{2n\sigma_1^2(\theta^*)}{2\sigma_1^2(\theta^*) + \lambda(n-1)\varepsilon\mu}$$

which is generally strictly between 1 and n, as seen in the exercise session.