+0/1/60+

Exam Optimization for Machine Learning – CS-439
Prof. Martin Jaggi

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Fr. 6. July 2018 - 16h15 to 19h15, in CE1515

ID

STUDENT NAME

SCIPER : **SCIPER** Signature :

Wait for the start of the exam before turning to the next page. This document is printed double sided, 18 pages.

- This is a closed book exam. No electronic devices of any kind.
- Place on your desk: your student ID, writing utensils, one double-sided A4 page cheat sheet (handwritten or 11pt min font size) if you have one; place all other personal items below your desk or on the side.
- You each have a different exam.
- For technical reasons, **do use black or blue pens for the MCQ part, no pencils!** Use white corrector if necessary.

Respectez les consignes suivantes | Observe this guidelines | Beachten Sie bitte die unten stehenden Richtlinien

choisir une réponse | select an answer Antwort auswählen	ne PAS choisir une réponse | NOT select an answer NICHT Antwort auswählen	Corriger une réponse | Correct an answer Antwort korrigieren

ce qu'il ne faut **PAS** faire | what should **NOT** be done | was man **NICHT** tun sollte

First part, multiple choice

There is **exactly one** correct answer per question.

Newton-Raphson method

An easy method for computing the square root of a real number $y > 0$ by hand is as follows:

(i) find x_0, such that $x_0^2 \approx y$ (e.g. $y = 17$, $x_0 = 4$).

(ii) Calculate the difference $d = y - x_0^2$ (e.g. $d = 17 - 4^2 = 1$).

(iii) Output $x_1 = x_0 + \frac{d}{2x_0}$ (e.g. $x_1 = 4 + \frac{1}{8} = 4.125$).

(iv) repeat (ii)–(iii) for higher accuracy.

This is an instance of the Newton-Raphson method, which defines the sequence $\{x_t\}_{t\geq 0}$ of real numbers by the following equation:

$$x_{t+1} := x_t - \frac{f(x_t)}{f'(x_t)}\,. \tag{1}$$

Question 1 What is the function $f(z)$ of which we aim to find a zero in the example above?

- ☐ $\sqrt{z - 17}$
- ☐ z^2
- ☐ $z^2 - 17$
- ☐ $\sqrt{z}$

Question 2 Now, suppose we are not happy with the solution $x_1 = 4.125$, because $x_1^2 = 17.015625$ is not accurate enough. What is the next iterate x_2 in the sequence (for $y = 17$, $x_0 = 4$ as above)? Use the following values: $\frac{17.015625}{4.125} = 4.125$, $\frac{0.015625}{4.125} \approx 0.0038$.

- ☐ 4.1288
- ☐ 4.1269
- ☐ 4.1212
- ☐ 4.1231

Question 3 How many iterations do you (roughly) have to perform to compute the correct 16 significant digits in the above example ($y = 17$, $x_0 = 4$).

- ☐ 10^{16}
- ☐ $10^{\sqrt{16}} = 10^4$
- ☐ 16
- ☐ $\frac{16}{2} = 8$
- ☐ $\sqrt{16} = 4$

Question 4 The Newton-Raphson method to find zeros of f can be interpreted as a second-order optimization method. Of course, one could also use the gradient method instead. How would the iterates of this scheme look like? (For carefully chosen stepsize γ).

- ☐ $x_{t+1} := x_t - \gamma f(x_t)$
- ☐ $x_{t+1} := x_t - \gamma f'(x_t)$
- ☐ $x_{t+1} := x_t - \gamma x_t$
- ☐ $x_{t+1} := x_t - \gamma (f'(x_t))^{-1} f(x_t)$
- ☐ $x_{t+1} := x_t - \gamma (f''(x_t))^{-1} f'(x_t)$

Newton's second-order optimization method

Question 5 As studied in the class, the update step for Newton's optimization method for an objective function $g : \mathbb{R}^n \to \mathbb{R}$ is given by

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \nabla^2 g(\mathbf{x}_t)^{-1} \nabla g(\mathbf{x}_t)$$

For $n = 1$, how does this optimization method relate to the Newton-Raphson method from Equation (1) from the previous section?

- ☐ $f = g'$
- ☐ $f'' = g$
- ☐ $f' = g$
- ☐ $f = g''$

Question 6 Given a quadratic function $g : \mathbb{R}^n \to \mathbb{R}$ of the form $g(\mathbf{x}) = -\frac{1}{2}\mathbf{x}^\top A\mathbf{x} + \mathbf{b}^\top \mathbf{x} + c$ where $A \in \mathbb{R}^{n\times n}$ is a symmetric matrix. What are necessary and sufficient conditions for g to be convex?

- ☐ $-A$ positive semidefinite, and $\mathbf{b}$ is non-negative
- ☐ The Hessian of g is negative definite for all $\mathbf{x}$, and $\mathbf{b}$ is non-negative
- ☐ A positive semidefinite
- ☐ The Hessian of g is negative definite for all $\mathbf{x}$
- ☐ The Hessian of g is positive definite for all $\mathbf{x}$
- ☐ A positive semidefinite, and $\mathbf{b}$ is non-negative
- ☐ $-A$ positive semidefinite
- ☐ The Hessian of g is positive definite for all $\mathbf{x}$, and $\mathbf{b}$ is non-negative

Coordinate Descent

Question 7 Consider the least squares objective function

$$f(\mathbf{x}) := \frac{1}{2} \|A\mathbf{x} - \mathbf{b}\|^2 \;, \tag{2}$$

for A a $m \times n$ matrix, $A = [\mathbf{a}_1, \dots \mathbf{a}_n]$ with columns $\mathbf{a}_i$.
What is the gradient $\nabla f(\mathbf{x})$?

- ☐ A
- ☐ $A^\top A\mathbf{x}$
- ☐ $A^\top(A\mathbf{x} - \mathbf{b})$
- ☐ $A(A^\top\mathbf{x} - \mathbf{b})$

Question 8 We are now interested in the complexity of computing the gradient of f as in Equation (2). Each addition or multiplication of two real numbers counts as one operation. How expensive is it to compute the full gradient, given $\mathbf{x}$:
(Note here $\Theta(k)$ refers to a function growing at least and at most as fast as k in the variables of concern)

- ☐ $\Theta(n + m)$
- ☐ $\Theta(n^2m^2)$
- ☐ $\Theta(n)$
- ☐ $\Theta(mn)$
- ☐ $\Theta(m^2n)$
- ☐ $\Theta(mn^2)$
- ☐ $\Theta(m)$

Question 9 How expensive is it to compute just a single coordinate of the gradient of f as in Equation (2), given $\mathbf{x}$:
(Note here $\Theta(k)$ refers to a function growing at least and at most as fast as k in the variables of concern)

- ☐ $\Theta(n)$
- ☐ $\Theta(m^2n)$
- ☐ $\Theta(m)$
- ☐ $\Theta(n^2m^2)$
- ☐ $\Theta(n + m)$
- ☐ $\Theta(mn)$
- ☐ $\Theta(mn^2)$

Question 10 The complexity of Coordinate Descent depends on the coordinate-wise smoothness constants L_i. What is L_i for f as in Equation (2)?

- ☐ $L_i = \lambda_{\max}(A^\top A)$
- ☐ $L_i = \|\mathbf{a}_i\|^2$
- ☐ $L_i = \lambda_{\max}(A^\top A)/n$
- ☐ $L_i = \|A^\top A\|$
- ☐ $L_i = \|\mathbf{a}_i\|$

Frank-Wolfe

Consider the linear minimization oracle (LMO) for matrix completion, that is for

$$\min_{Y \in X \subseteq \mathbb{R}^{n \times m}} \sum_{(i,j) \in \Omega} (Z_{ij} - Y_{ij})^2$$

when $\Omega \subseteq [n] \times [m]$ is the set of observed entries from a given matrix Z. Our optimization domain X is the unit ball of the trace norm (or nuclear norm), which is known to be the convex hull of the rank-1 matrices

$$X := conv(\mathcal{A}) \;\text{ with }\; \mathcal{A} := \left\{ \mathbf{u}\mathbf{v}^\top \;\middle|\; \substack{\mathbf{u} \in \mathbb{R}^n,\, \|\mathbf{u}\|_2 = 1 \\ \mathbf{v} \in \mathbb{R}^m,\, \|\mathbf{v}\|_2 = 1} \right\}.$$

Question 11 Consider the LMO for this set X for a gradient at iterate $Y \in \mathbb{R}^{n \times m}$ (derive it if necessary). Compare the computational operation (or cost) needed to compute the LMO, as opposed to computing the *projection* onto X?
Hint: Assume that the Singular Value Decomposition of a $n \times m$ matrix takes time $\Theta(n^2 m)$, and computing the top singular vector takes time $\Theta(nm)$.

- ☐ LMO and projection both take $\Theta(n^2 m)$
- ☐ LMO takes $\Theta(nm)$, and projection takes $\Theta(n^2 m)$
- ☐ LMO takes $\Theta(n^2 m)$, and projection takes $\Theta(nm)$

Smoothness and Strong Convexity

Consider an iterative optimization procedure.

Question 12 Which one of the following three inequalities is valid for a *smooth* convex function f:

- ☐ $f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) \leq \nabla f(\mathbf{x}_t)^\top (\mathbf{x}_{t+1} - \mathbf{x}_t) + \frac{L}{2} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2$
- ☐ $f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) \leq \nabla f(\mathbf{x}_t)^\top (\mathbf{x}_{t+1} - \mathbf{x}_t) - \frac{L}{2} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2$
- ☐ $f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) \leq \nabla f(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) + \frac{L}{2} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2$

Question 13 Which one of the following three inequalities is valid for a *strongly convex* function f:

- ☐ $f(\mathbf{x}_t) - f(\mathbf{x}^\star) \geq \nabla f(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{x}^\star) + \frac{\mu}{2} \|\mathbf{x}_t - \mathbf{x}^\star\|^2$
- ☐ $f(\mathbf{x}_t) - f(\mathbf{x}^\star) \leq \nabla f(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{x}^\star) - \frac{\mu}{2} \|\mathbf{x}_t - \mathbf{x}^\star\|^2$
- ☐ $f(\mathbf{x}_t) - f(\mathbf{x}^\star) \leq \nabla f(\mathbf{x}_t)^\top (\mathbf{x}_t - \mathbf{x}^\star) + \frac{\mu}{2} \|\mathbf{x}_t - \mathbf{x}^\star\|^2$

Random search

Question 14 Consider derivative-free random search, with line-search, as discussed in the lecture.

- ☐ For strongly convex functions, random search converges as $\mathcal{O}(L \log(1/\varepsilon))$
- ☐ For convex functions, random search converges as $\mathcal{O}(dL/\varepsilon)$
- ☐ For convex functions, random search converges as $\mathcal{O}(dL \log(1/\varepsilon))$

Empirical comparison of different methods

Donald Duck's three nephews Huey, Dewey, and Louie have enrolled in CS 439. For their course project, they analyzed three different algorithms, namely *Gradient Descent*, *Accelerated Gradient Method* and *Newton's second-order optimization method* on a strongly convex optimization problem and plotted the performance of the algorithms on a graph. However, as it turns out they forgot to put a legend in their graph and due to some bug in their code, they plotted a line which corresponds to none of the algorithms. Can you help them in labelling their unlabelled graph?

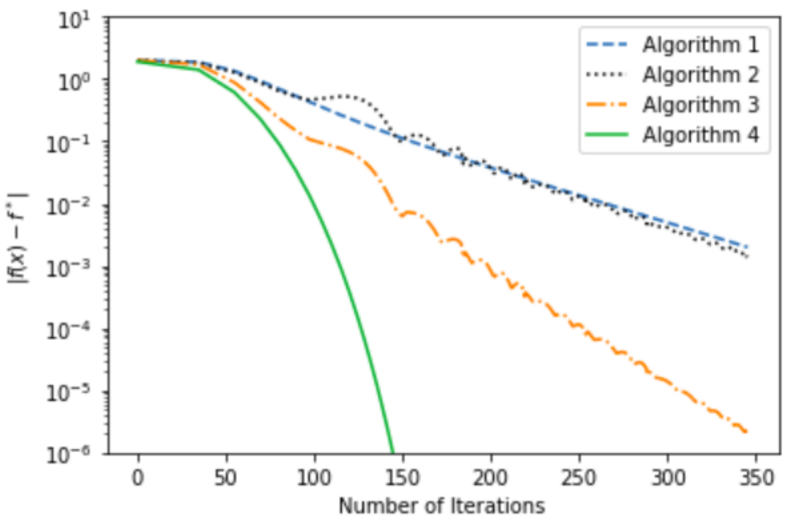

Figure 1: Performance of different optimization algorithms.

Question 15 Which optimization method corresponds to the error-curve for **Algorithm 1**?

- ☐ None
- ☐ Gradient Descent (with correct stepsize)
- ☐ Accelerated Gradient Method (with correct parameters)
- ☐ Newton's optimization method

Question 16 Which optimization method corresponds to the error-curve for **Algorithm 2**?

- ☐ None
- ☐ Newton's optimization method
- ☐ Accelerated Gradient Method (with correct parameters)
- ☐ Gradient Descent (with correct stepsize)

Question 17 Which optimization method corresponds to the error-curve for **Algorithm 3**?

- ☐ Accelerated Gradient Method (with correct parameters)
- ☐ Newton's optimization method
- ☐ None
- ☐ Gradient Descent (with correct stepsize)

Question 18 Which optimization method corresponds to the error-curve for **Algorithm 4**?

- ☐ Newton's optimization method
- ☐ None
- ☐ Accelerated Gradient Method (with correct parameters)
- ☐ Gradient Descent (with correct stepsize)

Second part, true/false questions

Question 19 (Convexity) The *epigraph* of a function $f : \mathbb{R}^d \to \mathbb{R}$ is defined as

$$epi(f) := \{(\mathbf{x}, \alpha) \in \mathbb{R}^{d+1} \,|\, \mathbf{x} \in \mathrm{dom}(f), \alpha \leq f(\mathbf{x})\},$$

☐ TRUE ☐ FALSE

Question 20 (Convexity) The triangle inequality and homogeneity of a norm together imply that any norm is convex.

☐ TRUE ☐ FALSE

Question 21 (Convex Sets) We consider C_1 and C_2 two convex sets in $\mathbb{R}^d$.
We define $C_1 + C_2 := \{x_1 + x_2, x_1 \in C_1, x_2 \in C_2\}$.
Is $C_1 + C_2$ a convex set?

☐ TRUE ☐ FALSE

Question 22 (Differentiability) The function $\big(\max(0, x)\big)^2$ is differentiable over $\mathbb{R}$.

☐ TRUE ☐ FALSE

Question 23 (Coordinate Descent) Consider Coordinate Descent on a strongly convex and smooth objective function. depending on the coordinate-wise smoothness constants L_i (i.e. different Lipschitz constants for each gradient coordinate).
If we sample coordinate i with probability proportional to L_i, and use stepsize L_i, convergence is typically faster than uniform CD

☐ TRUE ☐ FALSE

Question 24 (Coordinate Descent) In the same setting, if we sample coordinate i uniformly, and use stepsize $1/L_i$, convergence is typically faster than CD with fixed stepsize

☐ TRUE ☐ FALSE

Third part, open questions

Answer in the space provided! Your answer must be justified with all steps. Do not cross any checkboxes, they are reserved for correction.

Intersection of Convex Sets

We are given n convex sets $\mathcal{C}_0 = \{C_1, \ldots, C_n\}$ where each set $C_i \subseteq \mathbb{R}^d$. We want to design an algorithm which can check if the intersection of *all* of these sets is null i.e. we want to check if

$$\bigcap_{C_i \in \mathcal{C}_0} C_i = \varnothing \,.$$

However we do not care about solving this problem exactly, but have a small leeway of magnitude $\varepsilon \geq 0$. To make this more mathematically precise, let us define some notation.
The distance between a set $C \subseteq \mathbb{R}^d$ and any point $\mathbf{y} \in \mathbb{R}^d$ is defined as

$$d(C, \mathbf{y}) \overset{\text{def}}{=} \min_{\mathbf{w} \in C} \|\mathbf{w} - \mathbf{y}\|_2 \,.$$

We only want to distinguish between the following two cases for any $\varepsilon > 0$:

(N) The intersection of the sets is non-empty, i.e. $\bigcap_{i=1}^n C_i \neq \varnothing$.

(E) For any point $\mathbf{x} \in \mathbb{R}^d$, $\max_{i \in \{1,\ldots,n\}} d(C_i, \mathbf{x}) \geq \varepsilon$.

We want to solve this problem using calls to an oracle which can compute the projection onto $C_i \in \mathcal{C}$. Let us define the projection oracle $P_i(\mathbf{x})$ for any $i \in \{1, \ldots, n\}$ and $\mathbf{x} \in \mathbb{R}^d$ as

$$P_i(\mathbf{x}) := \operatorname*{argmin}_{\mathbf{y} \in C_i} \|\mathbf{y} - \mathbf{x}\|_2 \,.$$

We want to make as *few* calls to the projection oracle as possible. Our strategy will be to i) define a loss function and ii) run gradient descent. Then using our knowledge of convergence of gradient descent, we can argue about the number of oracle calls required.

First Approach.

Inspired by the condition in case (E), let us define the following loss function:

$$g(\mathbf{x}) := \max_{i \in \{1,\ldots,n\}} d(C_i, \mathbf{x}) \,.$$

Question 24: *2 points.* Is the function $g(\mathbf{x})$ convex? Is it Lipschitz?

Hint: maximum of convex functions is also convex.

☐0 ☐1 ☐2

Question 25: *5 points.* What is the sub-gradient of g? How many calls to the gradient oracle are needed to compute $\partial g(\mathbf{x})$ and $g(\mathbf{x})$?
Hint: Show that for two convex functions $g_1(x)$ and $g_2(x)$, $\partial g_i(x)$ is a subgradient in the set $\partial \max(g_1(x), g_2(x))$ where $g_i(x) := \max(g_1(x), g_2(x))$.

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 26: *5 points.* Assume you are given a starting point $\mathbf{x}_0$ and a constant R such that $\|\mathbf{x}_0 - \mathbf{x}^\star\|_2 \leq R$. Give the update step of gradient descent with an appropriate step-size. Show using the convergence of gradient descent we proved in class that for any optimum $\mathbf{x}^\star$ of g,

$$\min_{t \in \{0,\dots,T\}} g(\mathbf{x}_t) - g(\mathbf{x}^\star) \leq \frac{R}{\sqrt{T}}\,.$$

☐0 ☐1 ☐2 ☐3 ☐4 ☐5

Question 27: *4 points.* Using the result from the previous question, show that $\mathcal{O}(n/\varepsilon^2)$ calls to the projection oracle is sufficient to distinguish between case (N) and case (E) for our problem.

☐0 ☐1 ☐2 ☐3 ☐4

Question 28: *6 points.*

$\square_0$ $\square_1$ $\square_2$ $\square_3$ $\square_4$ $\square_5$ $\square_6$

The convergence of the Frank-Wolfe algorithm was analyzed in class for only *smooth* functions. In this question we will examine if smoothness is necessary. Consider the following non-smooth function $f : \mathbb{R}^2 \to \mathbb{R}$:

$$f(w, v) := \max\{w, v\} ,$$

restricted to a ball of radius 2 around the origin. We are then interested in finding

$$(w^\star, v^\star) := \underset{w^2+v^2 \leq 2}{\operatorname{argmin}} \left(\max\{w, v\}\right) .$$

Suppose we start at the origin $(0, 0)$ and run the Frank-Wolfe algorithm (with any step size rule). Since the function is not smooth, we will call the LMO oracle using an arbitrary subgradient instead of the gradient. Does this algorithm converge to the optimum?

Hint: First show that the iterates of Frank-Wolfe always lie in the convex hull of the starting point and the solutions of the LMO oracle.

Newton's second-order optimization method

As studied in the class, the update step for Newton's optimization method for an objective function $g : \mathbb{R}^n \to \mathbb{R}$ is given by

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \nabla^2 g(\mathbf{x}_t)^{-1} \nabla g(\mathbf{x}_t)$$

Question 29: *2 points.* What happens when Newton's optimization method is run on a convex quadratic function? Explain.

☐ 0 ☐ 1 ☐ 2

Question 30: *2 points.* **Affine Invariance of the Newton's method**

Consider $h(\mathbf{x}) := g(M\mathbf{x})$ where $M \in \mathbb{R}^{n\times n}$ is invertible where g is some convex function. Show that the Newton steps for h and g are also related by the same linear transformation, i.e., $\Delta\mathbf{x}_t = M\Delta\mathbf{y}_t$ where $\Delta\mathbf{x}_t$ and $\Delta\mathbf{y}_t$ are the Newton steps at the t^{th} iteration for h and g respectively. We assume $\mathbf{x}_0 = M\mathbf{y}_0$ are the starting iterates for h and g respectively.

☐ 0 ☐ 1 ☐ 2

Coordinate Descent

Question 31: *2 points.* Given a matrix A, we define $\lambda_{\min}(A^\top A)$ and $\lambda_{\max}(A^\top A)$ to be the smallest and largest eigenvalues of $A^\top A$.

Show that for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$,

$$\lambda_{\min}(A^\top A)\|\mathbf{x}-\mathbf{y}\|^2 \leq \|A(\mathbf{x}-\mathbf{y})\|^2 \leq \lambda_{\max}(A^\top A)\|\mathbf{x}-\mathbf{y}\|^2.$$

☐0 ☐1 ☐2

Question 32: *3 points.* Show that for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, for any $\mathbf{b} \in \mathbb{R}^n$:

$$\|A\mathbf{x}-\mathbf{b}\|^2 \leq \|A\mathbf{y}-\mathbf{b}\|^2 + \left[A^\top(A\mathbf{y}-\mathbf{b})\right]^\top(\mathbf{x}-\mathbf{y}) + \frac{\lambda_{\max}(A^\top A)}{2}\|\mathbf{x}-\mathbf{y}\|^2.$$

What does that imply for $f(\mathbf{x}) := \|A\mathbf{x}-\mathbf{b}\|^2$?

☐0 ☐1 ☐2 ☐3

Question 33: *2 points.* For $f(\mathbf{x}) := \|A\mathbf{x} - \mathbf{b}\|^2$, we now perform one step of coordinate descent. I.e. for a given point $\mathbf{x}_t \in \mathbb{R}^n$ we do a step of the form

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma_t (\nabla f(\mathbf{x}_t))_i \cdot \mathbf{e}_i$$

where $\mathbf{e}_i \in \mathbb{R}^n$ denotes a standard unit vector. For i fixed, compute the best γ_t.

☐0 ☐1 ☐2

Smooth strongly convex SGD

We consider a function $f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$ on $\mathbb{R}^d$, and we assume that the functions f_i are convex, differentiable.
We furthermore assume that f is L-smooth, that is that ∇f is L-Lipschitz.
We consider SGD defined as the following algorithm: Let $\mathbf{x}_0 \in \mathbb{R}^d$, and for any $t \geq 1$, for a sequence of step sizes γ_t, define

$$\mathbf{x}_{t+1} := \mathbf{x}_t - \gamma_t \mathbf{g}_t .$$

We first consider $\mathbf{g}_t := \nabla f_{i_t}(\mathbf{x}_t)$, with i_t uniformly and independently sampled from $\{1, \dots, n\}$.

Question 34: *2 points.* Show that $\mathbf{g}_t$ is an unbiased estimator of the gradient $\nabla f(\mathbf{x}_t)$.

☐0 ☐1 ☐2

Question 35: *6 points.* Combining the two valid equations of smoothness and strong convexity (as also stated in Questions 12 and 13), prove in detailed steps that, if $\gamma_t \leq \frac{1}{L}$, SGD in this setting converges as

$$\mathbb{E}\left[f(\mathbf{x}_{t+1}) - f(\mathbf{x}^\star)\right] \leq \gamma_t \mathbb{E}\left[\|\mathbf{g}_t - \nabla f(\mathbf{x}_t)\|^2\right] + \frac{(1-\gamma_t\mu)\,\mathbb{E}\left[\|\mathbf{x}_t - \mathbf{x}^\star\|^2\right] - \mathbb{E}\left[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2\right]}{2\gamma_t}. \tag{3}$$

For comparison, recall the following result from Lecture 6 (slide 6):

$$\mathbb{E}\left[f(\mathbf{x}_{t+1}) - f(\mathbf{x}^\star)\right] \leq \frac{\gamma_t B^2}{2} + \frac{(1-\gamma_t\mu)\,\mathbb{E}\left[\|\mathbf{x}_t - \mathbf{x}^\star\|^2\right] - \mathbb{E}\left[\|\mathbf{x}_{t+1} - \mathbf{x}^\star\|^2\right]}{2\gamma_t}. \tag{4}$$

under the bounded gradient assumption $\mathbb{E}[\|\mathbf{g}_t\|^2] \leq B^2$.

How do the two results compare?

☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6

Question 36: *4 points.* Recall the possible choices of learning rate (γ_t) in the situation of the previous question. What is the resulting rate of convergence? Which estimator do we eventually consider? Comment on the assumption $\gamma_t \leq \frac{1}{L}$. Is it a restriction? Which choice of step size could be used,

a) for getting $\mathcal{O}(\log(t)/t)$ convergence, and

b) for getting $\mathcal{O}(1/t)$?

☐0 ☐1 ☐2 ☐3 ☐4