Name:

EPFL
EE-556: Mathematics of Data
Mock Exam

Instructions:

- You are not allowed to use calculators, computers, phones, the internet, or any other computing device.
- You are not allowed to use your textbook or course notes.
- You are allowed to use handwritten notes (not typed, printed, or photocopied) on **BOTH** sides of **ONE** sheet of A4 paper.
- Please show and **EXPLAIN** all of your work. For example, it is important to write down any formula you are using so that we can give you partial credit if you make a small mistake along the way.
- Problems marked with $*$ are difficult; attempt them last.
- In some questions, the answer to one part may depend on the answers to previous parts. You can get full credit for the latter part even if the answer to the first part is wrong. In these cases, it is especially important to show your working for all parts.
- If you have any questions about the English vocabulary, please do not hesitate to ask us.
- Once you begin the exam, please write your name on all pages.
- If you run out of space, you may write on the extra blank pages at the end of the exam sheets.

Problem	**Possible Points**	**Score**
1	20	
2	20	
3	20	
4	20	
5	20	
Total:	100	

Name:

PROBLEM 1: DIFFUSION MODELS

Let $(x_1, \ldots, x_n)$ be a dataset of n images. Let X_0 is be a random clean image sampled uniformly from $x_1, \ldots, x_n$. A random noisy image at time t ($t > 0$), denoted by X_t, is defined as

$$X_t = e^{-t} X_0 + \sqrt{1 - e^{-2t}} Z,$$

where Z is a standard Gaussian variable. Clearly, given this setup, the higher the time t, the noisier the image X_t will be.

Diffusion models learn how to obtain X_0 from observing X_t. In other words, a denoising network `net` can be obtained by optimizing the following loss, given X_0 and X_t from some unknown distribution $\mathbb{P}$:

$$\min_{\texttt{net}} \mathcal{L}(\texttt{net}) := \mathbb{E}_{(X_0, X_t) \sim \mathbb{P}}[\|\texttt{net}(X_t) - X_0\|_2^2].$$

As a result, the optimized denoising network learns to approximate the conditional expectation

$$\texttt{net}(X_t) \approx \mathbb{E}[X_0 | X_t].$$

(a) (5 points) Let us assume that $X_0 \sim p(x_0)$ is sampled from the uniform distribution over $(x^1, \ldots, x^n)$. Show that the distribution of X_t is $p(x_t) = \frac{1}{n} \sum_{i=1}^{n} p(x_t | x_i)$.

Name:

(b) (5 points) Compute the gradient of $\log p(x_t)$ and show that

$$\nabla \log p(x_t) = \sum_{i=1}^{n} \left(\frac{e^{-t} x_i - x_t}{1 - e^{-2t}} \right) p(x_i | x_t).$$

Hint: From the definition of $p(x_t|x_i)$, X_t is a Gaussian with mean $e^{-t} x_i$ and variance $1 - e^{-2t}$.

Name:

(c) (5 points) One can show that the Hessian is given by

$$\nabla^2 \log p(x_t) = \frac{e^{-2t}}{(1-e^{-2t})^2}\text{Cov}[X_0|X_t = x_t] - \frac{1}{1-e^{-2t}}I,$$

where $\text{Cov}[X_0|X_t]$ is the covariance matrix of $X_0|X_t = x_t$, and I is the identity matrix. Recalling that a covariance matrix is always positive semidefinite (psd), what can you say about the Hessian of $\log p_t$ as $t \to \infty$? Does it become psd, negative definite, or non-definite?

Name:

(d) (2 points) Deduce from your previous answer if $\log p(x_t)$ becomes more convex, more concave or neither as a function of x as $t \to \infty$? If you could not solve part (c), assume its solution (psd, negative definite, non-definite) and provide an answer here.

(e) (3 points) The conditional expectation $\mathbb{E}[X|X_t]$ can be thought of an average of all the possible candidates X_t that could have generated X. If the noise is large then the conditional expectation is an average of many candidates from $1, \ldots, n$. The next image illustrates this idea. Recalling that given a denoiser, at test time, the basic recipe for generating a new sample is the following:

1. Generate a noisy sample $X_t = e^{-t}X + \sqrt{1 - e^{-2t}}Z$
2. Denoise to obtain X

Figure 1: Denoising MNIST from various noise levels, here $\sigma = \sqrt{1 - e^{-2t}}$

With this in mind, choose the correct sentences and justify your choice:

☐ If the noise $t \to \infty$, the distribution of X_t is very close to being a Gaussian

☐ If the noise $t \to \infty$, step 1 of creating a noisy sample becomes tractable but denoising from this high noise level using a conditional expectation becomes innacurate.

☐ If the noise $t \to \infty$, the distribution of X_t has exactly two modes.

Name:

PROBLEM 2: SHARPNESS-AWARE MINIMIZATION (SAM)

A machine learning team is designing a binary classifier using a linear function $h_{\mathbf{x}}$ with parameter $\mathbf{x} \in \mathbb{R}^d$. They train a classifier using empirical risk minimization on a labeled dataset containing only one sample $\{(\mathbf{a}^{(1)}, b^{(1)})\}$ where $\mathbf{a} \in \mathbb{R}^d$ and $b \in \{\pm 1\}$. Their SAM framework incorporates an ℓ_2-norm constraint on the parameter perturbations. As a result,

- The SAM objective involves finding an adversarial perturbation δ that maximizes the loss, subject to the constraint $\|\delta\|_2 \leq \epsilon$.
- The team's goal is to determine the parameter $\mathbf{x}$ that minimizes the sharpness-aware loss over the dataset.

(a) (2 points) For a given loss function of the form $L(\mathbf{x}, \mathbf{a}, b)$, Fill in the blanks below to write the corresponding SAM optimization problem.

$$\underline{\quad\min\quad} \; \underline{\quad\max\quad} \; \underline{\qquad\qquad\qquad}$$

(b) (3 points) Recall that we assume a linear function for the classification problem above

$$h_{\mathbf{x}}(\mathbf{a}) = \langle \mathbf{x}, \mathbf{a} \rangle,$$

and now we assume a loss function as the **logistic loss**, defined as:

$$L(\mathbf{x}, \mathbf{a}, b) = \log\left(1 + \exp(-b \langle \mathbf{x}, \mathbf{a} \rangle)\right).$$

Using the inner maximization problem from part (a), reformulate the SAM problem considering the logistic loss. Given this loss function, is the minimization problem in the minmax problem above convex?

Name:

(c) (4 points) Suppose that the data given is $\mathbf{a}^{(1)} = (1, -1)$, $b^{(1)} = 1$, the initial iterate is $\mathbf{x}_1 = (1, 0)$, and $\epsilon = 0.5$. Find the set of possible perturbations that can maximize the inner SAM objective, i.e., solve the maximization problem corresponding to the perturbation. Show that the solution is unique.

Name:

Consider the same setting as in part (c), where we have

$$\mathbf{a}^{(1)} = (1, -1), \quad b^{(1)} = 1, \quad \mathbf{x}_1 = (1, 0), \quad \epsilon = 0.5.$$

(d) (2 points) Based on the formulation and results from part (b), does the convex variant of Danskin's Theorem applies to this problem? If yes, check the conditions for the convex variant. If not, check the conditions of the general variant. (See the end of the question for their definitions.)

Does the convex variant of Danskin's Theorem apply? □ Yes □ No

Check the conditions:

(e) (4 points) Using the appropriate Danskin's theorem, is the SAM's objective function differentiable at the point $\mathbf{x} = \mathbf{x}_1$? If the function is differentiable, compute the gradient at $\mathbf{x} = \mathbf{x}_1$. If the function is not differentiable, find a subgradient and provide a justification for your answer.

Is the function differentiable? □ Yes □ No

gradient or subgradient =

Name:

(f) (5 points) Write a single iteration of the stochastic SAM training update in pseudo-code. Then, using the values obtained in part (c) and the computed gradient from part (e), calculate the updated weights $\mathbf{x}_2$ for the adversarial training objective. Assume a learning rate of $\eta = 1$. Show your work.

- Model parameters: $\mathbf{x}$
- Learning rate: η
- Input data: $(\mathbf{a}, b)$
- Loss function: $L(\mathbf{x}, \mathbf{a}, b)$
- Perturbation size: ϵ

```
Input: Initial weights x(1), data point a(1), label b(1),
    perturbation budget epsilon,learning rate eta,
    loss function L, gradient function grad_x.
Output: Updated weights x(2).

1. Initialize x(1) and define loss function L(x, a, b).
2. Compute perturbation delta: Answer 1
3. Compute gradient: grad = grad_x L(Answer 2)
4. Update weights using gradient descent: x(2) = Answer 3
```

Name:

Danskin's Theorem. (Convex variant) Let $\Phi(\mathbf{x}, \mathbf{y}) : \mathbb{R}^p \times \mathcal{Y} \to \mathbb{R}$ be a continuous function, where $\mathcal{Y} \subset \mathbb{R}^m$ is a compact set and define $f(\mathbf{x}) := \max_{\mathbf{y} \in \mathcal{Y}} \Phi(\mathbf{x}, \mathbf{y})$. Suppose that for each $\mathbf{y}$ in the compact set $\mathcal{Y}$, $\Phi(\mathbf{x}, \mathbf{y})$ (as a function of $\mathbf{x}$) is an extended real-valued closed proper convex function.

Define $\mathcal{Y}^\star(\mathbf{x}) := \arg\max_{\mathbf{y} \in \mathcal{Y}} \Phi(\mathbf{x}, \mathbf{y})$ as the set of maximizers (for a fixed value of $\mathbf{x}$ and $\mathbf{y}_\mathbf{x}^\star \in \mathcal{Y}^\star$ as an element of this set. We have

1. $f(\mathbf{x})$ is a convex function.
2. If $\mathbf{y}_\mathbf{x}^\star = \arg\max_{\mathbf{y} \in \mathcal{Y}} \Phi(\mathbf{x}, \mathbf{y})$ is unique, then the function $f(\mathbf{x}) = \max_{\mathbf{y} \in \mathcal{Y}} \Phi(\mathbf{x}, \mathbf{y})$ is differentiable at $\mathbf{x}$:
$$\nabla_\mathbf{x} f(\mathbf{x}) = \nabla_\mathbf{x} \left(\max_{\mathbf{y} \in \mathcal{Y}} \phi(\mathbf{x}, \mathbf{y}) \right) = \nabla_\mathbf{x} \Phi(\mathbf{x}, \mathbf{y}_\mathbf{x}^\star).$$
3. If $\mathbf{y}_\mathbf{x}^\star = \arg\max_{\mathbf{y} \in \mathcal{Y}} \Phi(\mathbf{x}, \mathbf{y})$ is not unique, then the subdifferential $\partial_\mathbf{x} f(\mathbf{x})$ of f is given by
$$\partial_\mathbf{x} f(\mathbf{x}) = \operatorname{conv} \left\{ \partial_\mathbf{x} \Phi(\mathbf{x}, \mathbf{y}_\mathbf{x}^\star) : \mathbf{y}_\mathbf{x}^\star \in \mathcal{Y}^\star(\mathbf{x}) \right\}.$$

Danskin's Theorem. (General) Let $\Phi(\mathbf{x}, \mathbf{y}) : \mathbb{R}^p \times \mathcal{Y} \to \mathbb{R}$ be a continuous function, where $\mathcal{Y} \subset \mathbb{R}^m$ is a compact set and define $f(\mathbf{x}) := \max_{\mathbf{y} \in \mathcal{Y}} \Phi(\mathbf{x}, \mathbf{y})$ and let $\mathcal{Y}^\star(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}} \Phi(\mathbf{x}, \mathbf{y})$.
Suppose that for each $\mathbf{y}$ in the compact set $\mathcal{Y}$, $\Phi(\mathbf{x}, \mathbf{y})$ (as a function of $\mathbf{x}$) is differentiable and that $\nabla_\mathbf{x} \Phi(\mathbf{x}, \mathbf{y})$ exists and is continuous. Then f is continuous, directionally differentiable and its directional derivatives satisy
$$f'(\mathbf{x}, u) = \sup_{\mathbf{y}_\mathbf{x}^\star \in \mathcal{Y}^\star(\mathbf{x})} u^T \nabla_\mathbf{x} \Phi(\mathbf{x}, \mathbf{y}_\mathbf{x}^\star)$$

In particular, if for some $\mathbf{x} \in \mathbb{R}^p$ the set $\mathcal{Y}^\star(\mathbf{x}) = \mathbf{y}_\mathbf{x}^\star$ is a singleton (contains only one element) then f is differentiable and we have
$$\nabla_\mathbf{x} f(\mathbf{x}) = \nabla_\mathbf{x} \Phi(\mathbf{x}, \mathbf{y}_\mathbf{x}^\star)$$

Corollary to Danskin's Theorem Suppose the conditions of Danskin's Theorem (General) hold.
If $\mathbf{y}_\mathbf{x}^\star \in \mathcal{Y}^\star(\mathbf{x})$, then as long as $-\nabla_\mathbf{x} \Phi(\mathbf{x}, \mathbf{y}_\mathbf{x}^\star)$ is nonzero then it is a descent direction for $f(\mathbf{x})$.

Name:

PROBLEM 3: CONVERGENCE RATE AND PER-ITERATION COMPLEXITY TRADEOFFS (20 points)

Fourier and András are trying to solve some optimization problems and are in disagreement on who is correct. In problems (a)–(d) below,

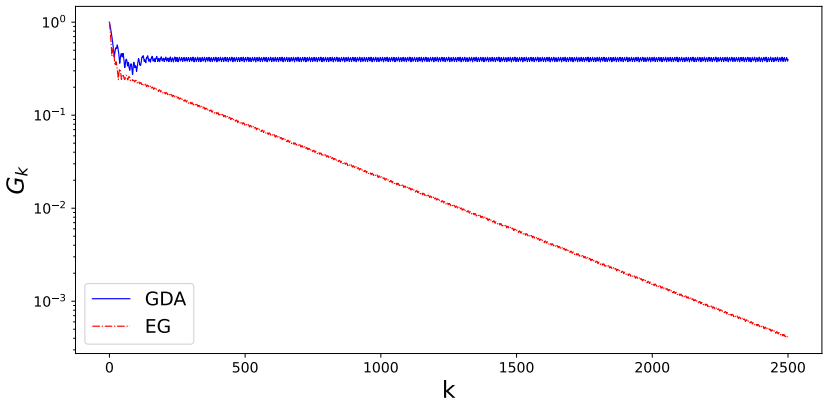
If the function f is smooth, we assume that the Lipschitz constant L of $\nabla f(\mathbf{x})$ is known.

If the function f is strongly convex, we assume that the strong convexity parameter μ is known.

(a) (5 points) András was comparing two optimization algorithms w.r.t. how they solve the following min-max optimization problem:

$$\min_{\mathbf{x}\in\Delta_5} \max_{\mathbf{y}\in\Delta_5} f(\mathbf{x},\mathbf{y}) := \mathbf{x}^\top \mathbf{I}\mathbf{y},$$

where Δ_5 is the simplex in 5 dimensions and $\mathbf{I}$ is the 5×5 identity matrix. He obtained the linear-linear scale convergence plots below after running Gradient Descent Ascent (GDA) and ExtraGradient (EG). On the y-axis in the plots, the duality gap G over time is shown, $G_k := G(\mathbf{x}_k,\mathbf{y}_k) = \max_{\mathbf{y}\in\Delta_5} f(\mathbf{x}_k,\mathbf{y}) - \min_{\mathbf{x}\in\Delta_5} f(\mathbf{x},\mathbf{y}_k)$.



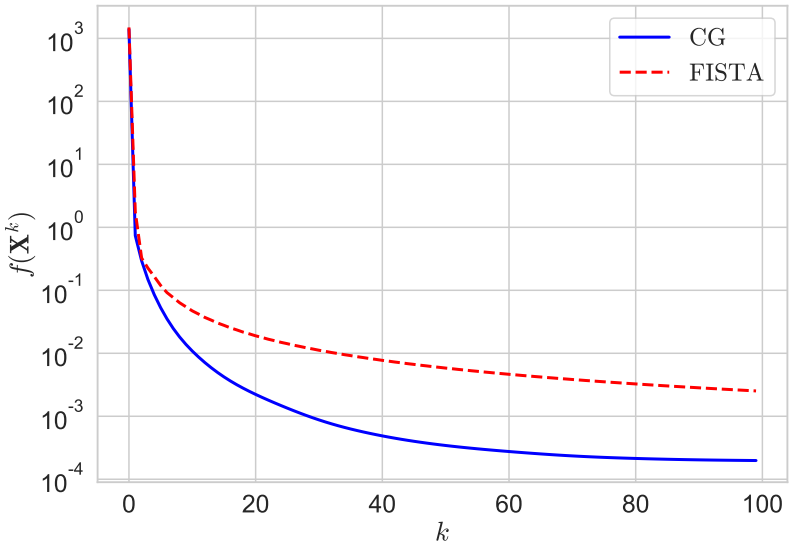
Fourier says that there must be a bug in the code (i.e., the results are not reasonable, so there is something wrong). Do you agree or disagree with Fourier? Explain your answer. (No credits will be given unless you provide a clear explanation, and we will accept any reasonable answers.)

Name:

(b) (5 points) András was comparing two algorithms on how they solve a semidefinite programming (SDP) problem, namely:

$$\min_{\mathbf{X}, \|\mathbf{X}\|_F \leq 1, \mathbf{X} \succcurlyeq 0} f(\mathbf{X}) := \|\mathbf{A}(\mathbf{X}) - \mathbf{b}\|_F^2,$$

where $\mathbf{A}$ is a smooth and convex operator. He obtained the linear-log scale convergence plots of the **objective values** below after running Fast Proximal Gradient Descent (FISTA) and Frank-Wolfe / Conditional Gradient (CG).



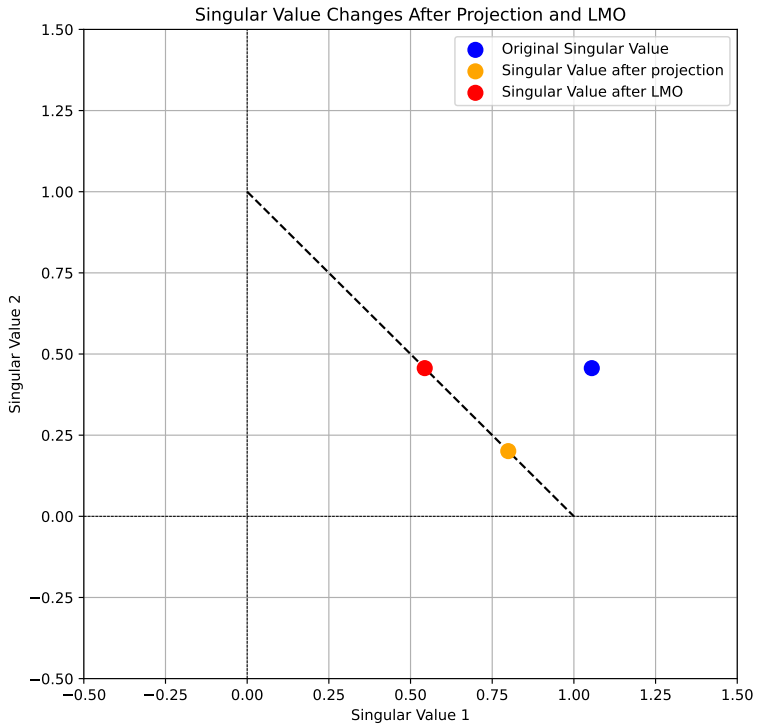
Fourier says that there must be a bug in the code (i.e., the results are not reasonable, so there is something wrong). Do you agree or disagree with Fourier? Explain your answer. (No credits will be given unless you provide a clear explanation, and we will accept any reasonable answers.)

Name:

(c) (5 points) András and Fourier were comparing the output of two oracles for a nuclear-norm-constrained optimization problem, specifically:

$$\min_{\mathbf{X}, \|\mathbf{X}\|_* \leq 1} f(\mathbf{X}) := (\langle \mathbf{A}, \mathbf{X} \rangle - b)^2,$$

where $\mathbf{A}$ is dimensionality-reducing. They implemented the proximal / projection operator and the Linear Minimization Oracle (LMO) for the nuclear norm ball to test their results.
Hint: The definitions of these operators are provided on the next page.
For a simple 2×2 example matrix, they visualized the singular values before and after applying each oracle in the x-axis and y-axis of a scatter plot, shown below:



Fourier says that there must be a bug in the code. Do you agree or disagree with Fourier? Explain your answer. (No credit will be awarded without a clear and reasonable explanation. Both correct and incorrect answers are accepted as long as they are well justified.) You can check the hints in the next page.

Name:

Recall: The definitions of the nuclear norm ball projection and LMO operators are as follows:
1. Projection: The projection of a matrix $\mathbf{X}$ onto the nuclear norm ball $\{\mathbf{X} \mid \|\mathbf{X}\|_* \leq 1\}$ is performed by truncating its singular values to ensure the nuclear norm constraint holds. If the singular value decomposition (SVD) of $\mathbf{X}$ is $\mathbf{X} = \mathbf{U}\,\mathrm{diag}(\sigma)\mathbf{V}^\top$, then the projection is computed as:

$$\mathrm{proj}(\mathbf{X}) = \mathbf{U}\,\mathrm{diag}(\sigma')\mathbf{V}^\top,$$

where σ' is the vector of singular values thresholded such that $\sum \sigma' \leq 1$ and $\sigma' \geq 0$.
2. Linear Minimization Oracle (LMO): The LMO for the nuclear norm ball solves the linear problem:

$$\min_{\|\mathbf{Y}\|_* \leq 1} \langle \mathbf{C}, \mathbf{Y} \rangle,$$

where $\mathbf{C}$ is any matrix. The solution is given by $\mathbf{Y} = -\mathbf{u}\mathbf{v}^\top$, where $\mathbf{u}$ and $\mathbf{v}$ are the left and right singular vectors corresponding to the largest singular value of $\mathbf{C}$.

Name:

(d)* (5 points) Neural Tangent Kernel: András wants to derive the NTK matrix for a neural network with squared loss:

$$L(\mathbf{w}(t)) = \frac{1}{2}\sum_{i=1}^{n}(f(\mathbf{x}_i;\mathbf{w}(t)) - y_i)^2,$$

under gradient flow:

$$\frac{d\mathbf{w}(t)}{dt} = -\nabla_{\mathbf{w}} L(\mathbf{w}(t)).$$

András claims that the NTK matrix $\mathbf{H}(t) \in \mathbb{R}^{n\times n}$ can be derived as follows. For the time evolution of the output $\mathbf{f}(t) = (f(\mathbf{x}_i;\mathbf{w}(t)))_{i=1}^{n}$, he writes:

1. Differentiating $\mathbf{f}(t)$ with respect to time:

$$\frac{d\mathbf{f}(t)}{dt} = \frac{\partial \mathbf{f}(t)}{\partial \mathbf{w}(t)} \cdot \frac{d\mathbf{w}(t)}{dt}.$$

2. The gradient of the loss with respect to $\mathbf{w}(t)$ is:

$$\nabla_{\mathbf{w}} L(\mathbf{w}(t)) = \sum_{i=1}^{n} (f(\mathbf{x}_i;\mathbf{w}(t)) - y_i)\, \frac{\partial f(\mathbf{x}_i;\mathbf{w}(t))}{\partial \mathbf{w}}.$$

3. Substituting the gradient flow equation:

$$\frac{d\mathbf{w}(t)}{dt} = -\sum_{i=1}^{n} (f(\mathbf{x}_i;\mathbf{w}(t)) - y_i)\, \frac{\partial f(\mathbf{x}_i;\mathbf{w}(t))}{\partial \mathbf{w}}.$$

4. Substituting this into $\frac{d\mathbf{f}(t)}{dt}$, András writes:

$$\frac{d\mathbf{f}(t)}{dt} = -\frac{\partial \mathbf{f}(t)}{\partial \mathbf{w}(t)} \sum_{i=1}^{n} (f(\mathbf{x}_i;\mathbf{w}(t)) - y_i)\, \frac{\partial f(\mathbf{x}_i;\mathbf{w}(t))}{\partial \mathbf{w}(t)}.$$

5. Using the definition of the NTK matrix, he concludes:

$$\frac{d\mathbf{f}(t)}{dt} = -\mathbf{H}(t)\left[\mathbf{f}(t) - \mathbf{y}\right], \quad \text{where } H_{ij}(t) = \left\langle \frac{\partial f(\mathbf{x}_j;\mathbf{w}(t))}{\partial \mathbf{w}(t)}, \sum_{i=1}^{n} \frac{\partial f(\mathbf{x}_i;\mathbf{w}(t))}{\partial \mathbf{w}(t)} \right\rangle.$$

Fourier says that there is a mistake in the step 5. Do you agree or disagree with Fourier? Explain your answer. (No credit will be awarded without a clear and reasonable explanation. Both correct and incorrect answers are accepted as long as they are well justified.)

Hint: You can check the dimensions after each step of the derivation.

Name:

PROBLEM 4: LIPSCHITZ CONSTANT AND PROXIMAL OPERATOR

In this question, we consider Reinforcement Learning from Human Feedback (RLHF) and reward modeling. We will see how the the Lipschitz constant can be related to RLHF.

Particularly, we consider the Lipschitz constant with respect to the ℓ_2-norm. Given a prompt a, associated with a preferred answer $b^{(1)}$ and a non-preferred answer $b^{(2)}$, the loss in reward modelling is given by:

$$f(a, b^{(1)}, b^{(2)}, \boldsymbol{x}) = -\log \sigma \Big(r(a, b^{(1)}, \boldsymbol{x}) - r(a, b^{(2)}, \boldsymbol{x}) \Big), \tag{1}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function and r is the reward model with learnable parameters $\boldsymbol{x}$.

(a) (5 points) Assume r is an L-Lipschitz function with respect to $\boldsymbol{x}$, define $h(\boldsymbol{x}) := r(a, b^{(1)}, \boldsymbol{x}) - r(a, b^{(2)}, \boldsymbol{x})$, then show that h is $2L$-Lipschitz continous with respect to $\boldsymbol{x}$.

Hint: Let $f : Q \to \mathbb{R}$ where $Q \subseteq \mathbb{R}^p$. Then, f is L-Lipschitz continuous if there exists a constant value $L \geq 0$ such that:

$$|f(\mathbf{y}) - f(\mathbf{x})| \leq L\|\mathbf{y} - \mathbf{x}\|_2, \quad \forall \mathbf{x},\ \mathbf{y} \in Q.$$

(b) (5 points) Based on the condition of (a), calculate an informative upper bound of the Lipschitz constant of the function f (defined in Eq.(1)) with respect to $\boldsymbol{x}$.

Hint: The Lipschitz constant of the composition of two or more functions can be bounded by the product of the Lipschitz constants of the individual functions

Name:

(c) (5 points) Suppose the answers $b^{(1)}$ and $b^{(2)}$ are real number, we define $\boldsymbol{b} = [b^{(1)}, b^{(2)}]^\top \in \mathbb{R}^2$. Assume that we use a trivial reward model, i.e., $r(a, b^{(i)}, \boldsymbol{x}) = b^{(i)}$, for $i = 1, 2$. Is f an 3/2 Lipschitz-continous function with respect to $\boldsymbol{b}$?

(d) (3 points) Gradient descent is an optimization method used for learning the reward model. Next, we will explore how the proximal operator can be interpreted as a gradient descent step. We have learnt that:

Definition 1. *Let $f \in \mathcal{F}(\mathbb{R}^d)$, $\boldsymbol{x} \in \mathbb{R}^d$ and $\lambda > 0$. The proximal operator of f is defined as:*

$$\text{prox}_{\lambda f}(\mathbf{y}) \equiv \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} \left\{ f(\boldsymbol{x}) + \frac{1}{2\lambda} \|\mathbf{y} - \boldsymbol{x}\|_2^2 \right\}. \tag{2}$$

Next, we introduce Bregman Proximal operator which is based on the Bregman divergence:

$$\text{prox}_{\lambda f}^{D_h}(\mathbf{y}) \equiv \arg\min_{\mathbf{x} \in \mathbb{R}^d} \left\{ f(\mathbf{x}) + \frac{1}{\lambda} D_h(\mathbf{y}, \mathbf{x}) \right\}, \tag{3}$$

where the Bregman divergence is defined as follows: Let h be a continuously-differentiable and strictly convex function. The **Bregman divergence** associated with h for points $\mathbf{x}$ and $\mathbf{y}$ is:

$$D_h(\mathbf{y}, \mathbf{x}) = h(\mathbf{y}) - h(\mathbf{x}) - \langle \nabla h(\mathbf{x}), (\mathbf{y} - \mathbf{x}) \rangle$$

Show that when $h(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$, the Bregman Proximal operator reduces to the proximal operator as seen in the lecture.

$\text{prox}_{\lambda f}(\mathbf{y}) =$

Name:

(e) (2 points) Next, when $f(\mathbf{x})$ is convex and we replace $f(\mathbf{x})$ with its lower bound $f(\mathbf{x}_t) + \langle \nabla f(\mathbf{x}_t), \mathbf{x} - \mathbf{x}_t \rangle$ inside the proximal operator, prove that the new optimization problem results in a gradient descent step with step size λ.

Name:

Problem 5: Constrained Convex Optimization (20 points)
Consider the following convex optimization problem

$$\min_{X \in \mathbb{R}^{p \times n}} f(X) := \frac{1}{2}\|AX - b\|_F^2 \quad \text{subject to} \quad \|X\|_{S_\infty} \leq 1,$$

where $A \in \mathbb{R}^{m \times p}$ and $b \in \mathbb{R}^{m \times n}$ are known and $X \in \mathbb{R}^{p \times n}$ and $\|\cdot\|_{S_\infty}$ is the Schatten infinity norm or equivalently the $\|\cdot\|_{2\to 2}$ operator norm:

$$\|X\|_{2\to 2} = \max_{u \in \mathbb{R}^n : \|u\|_2 \leq 1} \|Xu\|_2. \tag{4}$$

Recall that Frank-Wolfe's method applies to this problem as follows:

Algorithm 1 Frank-Wolfe Method for Convex Optimization

1: **Input:** $x^0 \in \mathbb{R}^p$
2: **for** $k = 1, 2, \ldots$ **do**
3: $\quad \hat{X}^k \in \arg\min_X \left(\nabla f(X^k), X\right) \quad$ subject to $\quad \|X\|_{S_\infty} \leq 1$
4: $\quad X^{k+1} = (1 - \gamma_k)X^k + \gamma_k \hat{X}^k$, where $\gamma_k := \frac{2}{k+1}$
5: **end for**

(a) (5 points) Show that f is a smooth function with respect to the S_∞ norm, in the sense that its gradient is Lipschitz continuous, with respect to this norm, i.e., $|f(X) - f(Y)| \leq L\|X - Y\|_{2\to 2}, \forall X, Y$. Find L in this equation.

(b) (5 points) Let r the rank of $\nabla f(X^k)$ and $\sigma_1, \ldots, \sigma_r$ and $u_1, \ldots, u_r$ and $v_1, \ldots, v_r$ the singular values, the left and right singular vectors corresponding to the r non zero singular values of $\nabla f(X^k)$. Show that we can choose an output of the linear minimization oracle, $\text{lmo}(\nabla f(X^k))$, as the following matrix:

$$\hat{X}^k = -\sum_{i=1}^{r} u_i v_i^\top$$

Hint: The dual of the S_∞ norm is the nuclear norm of the matrix.

Name:

Consider the following convex optimization problem:

$$\min_{x \in \mathbb{R}^p} f(x) \quad \text{subject to} \quad Ax = b, \tag{15}$$

where $A \in \mathbb{R}^{n \times p}$ and $b \in \mathbb{R}^n$ are known.
(c) (3 points) Write down the Lagrangian function corresponding to formulation (15) with the dual variable $\lambda \in \mathbb{R}^n$. Write down an equivalent formulation of (15) using the Lagrangian function.

$L(x, \lambda) =$

(d) (5 points) Write down the augmented Lagrangian (AL) of formulation (15) with quadratic penalty. Use μ as the parameter for the penalty funciton. Given the definition of the Lagrange dual function as:

$L_\mu(x, \lambda) =$

Name:

Consider the following convex optimization problem:

$$\min_{x \in \mathbb{R}^p} c^T x \quad \text{subject to} \quad Ax = b,\ \|x\|_1 \leq 1, \tag{16}$$

where $A \in \mathbb{R}^{n \times p}$, $b \in \mathbb{R}^n$, and $c \in \mathbb{R}^p$ are known.

(e)* (2 points) Derive the dual function of formulation (16). Simplify the expression of $d(\lambda)$ so that it does not involve a min or max. (Hint: You may use Hölder's inequality)

$d(\lambda) =$

Name:

This page is extra space to write. Please make sure to refer to the problem on which you are working.

Name:

This page is extra space to write. Please make sure to refer to the problem on which you are working.

Name:

This page is extra space to write. Please make sure to refer to the problem on which you are working.

Name:

This page is extra space to write. Please make sure to refer to the problem on which you are working.

Name:

This page is extra space to write. Please make sure to refer to the problem on which you are working.