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.

Name:

PROBLEM 1: LOGISTIC REGRESSION WITH BINARY DATA

In medical diagnostics, we often predict binary outcomes from patient features. For example, given genome data, we may want to predict whether a patient has breast cancer ($s_i = 1$) or not ($s_i = 0$).

We observe n independent binary outcomes.

1. **Observation**: At position i, we observe $s_i \in \{0, 1\}$ where:

$$P(s_i = 1 \mid a_i, x) = \pi_i, \qquad P(s_i = 0 \mid a_i, x) = 1 - \pi_i$$

2. **Model**: We model the success probability with:

$$\pi_i = \sigma(a_i^\top x), \qquad \sigma(u) = \frac{1}{1 + e^{-u}}$$

   where $a_i \in \mathbb{R}^p$ are known feature vectors and $x \in \mathbb{R}^p$ is the parameter to estimate.

3. **Data**: We observe pairs $\{(s_i, a_i)\}_{i=1}^n$ where s_i is the binary outcome and a_i are features.

(a) (4 points) **Log-Likelihood Derivation**
Write the negative log-likelihood $L(x)$ as:

$$L(x) = \sum_{i=1}^{n} f_i(x)$$

Find: The explicit form of $f_i(x)$ in terms of s_i, a_i, and $\sigma(\cdot)$.

(b) (4 points) **Gradient Computation**
Derive $\nabla_x L(x) \in \mathbb{R}^p$ in terms of s_i, π_i, and a_i. Simplify the expression as much as possible.

Name:

(c) **Hessian Analysis and Convexity**
(i) (3 points) Compute $\nabla_x^2 L(x)$ and show it has the form $\sum_{i=1}^n \alpha_i(x) \cdot a_i a_i^\top$ for some scalar $\alpha_i(x)$. Give the explicit formula for $\alpha_i(x)$ in terms of π_i, s_i.

(ii) (3 points) Is $L(x)$ convex? Justify.

(d) **Lipschitz Smoothness and Bounds**
Assume $\max_i \|a_i\|_2 \leq R$.
(i) (2 points) Find an upper bound for $\alpha_i(x)$ that holds for all i and all x. **Hint**: The function $p(1-p)$ is maximized at $p = 1/2$.

Name:

(ii) (2 points) Find an upper bound for $\|\nabla^2 L(x)\|_2$. **Hints**: - For any vector a: $\|aa^\top\|_2 = \|a\|_2^2$ - Triangle inequality: $\|\sum_i M_i\|_2 \leq \sum_i \|M_i\|_2$

(iii) (2 points) Conclude the Lipschitz constant L_{Lip} of ∇L. **Hint**: For twice-differentiable functions, $L_{\text{Lip}} = \sup_x \|\nabla^2 L(x)\|_2$.

Name:

PROBLEM 2: IMPLICIT BIAS OF GRADIENT DESCENT

Consider data points $\{(\mathbf{a}_i, b_i)\}_{i=1}^n$, where $\mathbf{a}_i \in \mathbb{R}^p$ and $b_i \in \{\pm 1\}$. Define the log-sum-exp margin loss:

$$f(\mathbf{x}) = \log\left(\sum_{i=1}^{n} \exp\left(-b_i\, \mathbf{a}_i^\top \mathbf{x}\right)\right).$$

Assume the data are linearly well-separable, i.e., there exists vectors $\mathbf{x}$ such that $b_i\, \mathbf{a}_i^\top \mathbf{x} > 0$ for all i.

(a) (2 points) Is the optimization problem convex? Is it strongly convex?

f is ☐ convex.
☐ concave.
☐ strongly convex.

(b) (2 points) Find a constant $L > 0$ such that f is L-smooth with respect to the Euclidean norm, i.e., such that

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq L\|\mathbf{x} - \mathbf{y}\|_2 \quad \text{for all } \mathbf{x}, \mathbf{y} \in \mathbb{R}^p.$$

(c) (4 points) Consider gradient descent on $f(x)$ with constant step-size $\eta \in (0, 2/L)$ and initialization $x_0 = 0$:

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

Assume the iterates diverge in norm but converge in direction:

$$\frac{x_k}{\|x_k\|} \to \hat{x}.$$

Among all separating hyperplanes, $\hat{x}$ corresponds to the one that minimizes the Euclidean norm. Fill in the optimization problem that characterizes $\hat{x}$:

$$\hat{x} = \arg\min_x \_\_\_\_\_ \quad \text{subject to} \quad b_i\, a_i^\top x \geq \_\_\_\_\_ \quad \text{for all } i.$$

Name:

In adaptive gradient methods such as AdaGrad, the update direction is scaled coordinate-wise based on past gradients. AdaGrad maintains a diagonal matrix

$$\mathbf{H}_k = \operatorname{diag}\left(\sum_{i=0}^{k-1} \mathbf{g}_i \odot \mathbf{g}_i\right), \qquad \mathbf{g}_i = \nabla f(\mathbf{x}_i),$$

and performs the update

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta\, \mathbf{H}_k^{-1/2} \mathbf{g}_k.$$

As k grows, $\mathbf{H}_k$ typically converges to a positive diagonal matrix $\mathbf{H}^\star$. This suggests that AdaGrad behaves like gradient descent in a geometry determined by a diagonal matrix $\mathbf{H}$.

To understand this geometry, consider the constrained optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^p} \frac{1}{2}\mathbf{x}^\top \mathbf{H}\mathbf{x} \qquad \text{such that} \qquad b_i\, \mathbf{a}_i^\top \mathbf{x} \geq 1,\ i = 1, \ldots, n, ,$$

where $\mathbf{H}$ is a diagonal matrix with strictly positive entries.

(d) (2 points) Write the Lagrangian $\mathcal{L}(\mathbf{x}, \lambda)$ of this problem using dual variable λ.

$\mathcal{L}(\mathbf{x}, \lambda) =$

(e) (4 points) Write the KKT conditions and solve them to obtain the optimal solution $\mathbf{x}_H^\star$ and $\lambda^\star$. Show your work. (*Hint:* See the next page for a description of the KKT conditions.)

KKT conditions:

$\mathbf{x}_H^\star =$

$\lambda^\star =$

Name:

(f) (6 points) Suppose AdaGrad converges and $\mathbf{H}_k \to \mathbf{H}^\star$, so that its limit is the point $\mathbf{x}^\star_{H^\star}$ from part (e). Briefly explain what you expect the implicit bias of AdaGrad to be (i.e., which norm it prefers among all separating solutions), and state for which diagonal matrices $\mathbf{H}^\star$ AdaGrad and standard GD would be expected to converge to the same separating direction.

Name:

KKT CONDITIONS:
Given a problem

$$\arg\min_{\mathbf{x}\in\mathbb{R}^p} f(\mathbf{x}) \quad \text{subject to } g_i(\mathbf{x}) \leq 0, i = 1, \dots, n \text{ and } h_j(\mathbf{x}) = 0, j = 1, \dots, p.$$

with dual variables $\lambda \in \mathbb{R}^n$ and $\mu \in \mathbb{R}^n$, the KKT conditions state that the triplet of points $(\mathbf{x}^\star, \lambda^\star, \mu^\star)$ is a solution to the problem only if

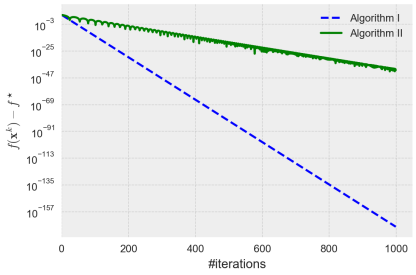
1. (Primal feasibility) We have $\forall i = 1, \dots, n, \quad g_i(\mathbf{x}^\star) \leq 0$ and $\forall j = 1, \dots, p, \quad h_i(\mathbf{x}^\star) = 0$.
2. (Dual feasibility) We have $\lambda_i^\star \geq 0$.
3. (Stationarity) We have
$$\nabla f(\mathbf{x}^\star) + \sum_{i=1}^{n} \lambda_i^\star \nabla g_i(\mathbf{x}^\star) + \sum_{j=1}^{n} \mu_j^\star \nabla h_j(\mathbf{x}^\star) = 0.$$
4. (Complementary slackness) Either $\lambda_i^\star = 0$ or $g_i(\mathbf{x}^\star) = 0$.

Conversely, if Slater's conditions hold, then any point verifying these 4 conditions is necessarily a solution to the problem.

Name:

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

(a) (6 points)



Wanda and Tony are discussing what they learned in Math of Data class for convergence of algorithms. Wanda shows Tony a plot like above, saying that it is a convergence plot she obtained from optimizing one function with two optimization algorithms that she learned from the course.

(1) Wanda asks Tony to guess which type of function she was dealing with and what Algorithm I was. She gave Tony three options:

(A) function: L-smooth, convex; algorithm: accelerated gradient descent

(B) function: L-smooth, μ-strongly convex; algorithm: gradient descent

(C) function: L-smooth, convex; algorithm: AMSGrad

Which one/ones do you think is the correct answer, and why (explain briefly)? If you need a brush-up on accelerated gradient descent and AMSGrad, please refer to the next page.

(2) Wanda reveals that Algorithm II is ADAM. Tony knows that ADAM is the most widely used optimizer these days and is not convinced by Wanda. He suggests that there might be a bug in Wanda's implementation because the curve of Algorithm II is not monotonic and it seems to lag far behind Algorithm I. Do you think Tony's rationale is well-founded? If you need a brush-up on ADAM, please refer to the next page.

Name:

Reminder.

Accelerated Gradient Descent

Input. Step size α and momentum coefficients $\{\gamma_k\}_{k\in\mathbb{N}}$

1. Choose $\mathbf{x}^0 = \mathbf{y}^0 \in \mathrm{dom}(f)$

2. For $k = 0, 1, \ldots,$ iterate

$$\begin{cases} \mathbf{x}^{k+1} &= \mathbf{y}^k - \alpha \nabla f(\mathbf{y}^k) \\ \mathbf{y}^{k+1} &= \mathbf{x}^{k+1} + \gamma_{k+1}(\mathbf{x}^{k+1} - \mathbf{x}^k) \end{cases}$$

Output : $\mathbf{x}^{k+1}$ of the last iteration.

AMSGrad

Input. Step size γ, exponential decay rates $\beta_1, \beta_2 \in [0, 1)$

1. Set $\mathbf{m}_0 = 0, \mathbf{v}_0 = 0$ and $\mathbf{v}_0^{\max} > 0$

2. For $k = 1, 2, \ldots,$ iterate

$$\begin{cases} \mathbf{g}_k &= \nabla f(\mathbf{x}^k) \\ \mathbf{m}_k &= \beta_1 \mathbf{m}_{k-1} + (1-\beta_1)\mathbf{g}_k \leftarrow \text{1st order estimate} \\ \mathbf{v}_k &= \beta_2 \mathbf{v}_{k-1} + (1-\beta_2)\mathbf{g}_k^2 \leftarrow \text{2nd order estimate} \\ \mathbf{v}_k^{\max} &= \max\{\mathbf{v}_{k-1}^{\max}, \mathbf{v}_k\} \\ \hat{\mathbf{m}}_k &= \mathbf{m}_k/(1-\beta_1^k) \leftarrow \text{Bias correction} \\ \hat{\mathbf{v}}_k &= \mathbf{v}_k^{\max}/(1-\beta_2^k) \leftarrow \text{Bias correction} \\ \mathbf{H}_k &= \mathrm{diag}(\sqrt{\hat{\mathbf{v}}_k}) + \epsilon I \\ \mathbf{x}^{k+1} &= \mathbf{x}^k - \gamma \mathbf{H}_k^{-1} \hat{\mathbf{m}}_k \end{cases}$$

(Every vector operation is an element-wise operation)

Output : $\mathbf{x}^{k+1}$ of the last iteration.

ADAM

Input. Step size γ, exponential decay rates $\beta_1, \beta_2 \in [0, 1)$

1. Set $\mathbf{m}_0, \mathbf{v}_0 = 0$

2. For $k = 1, 2, \ldots,$ iterate

$$\begin{cases} \mathbf{g}_k &= \nabla f(\mathbf{x}^k) \\ \mathbf{m}_k &= \beta_1 \mathbf{m}_{k-1} + (1-\beta_1)\mathbf{g}_k \leftarrow \text{1st order estimate} \\ \mathbf{v}_k &= \beta_2 \mathbf{v}_{k-1} + (1-\beta_2)\mathbf{g}_k^2 \leftarrow \text{2nd order estimate} \\ \hat{\mathbf{m}}_k &= \mathbf{m}_k/(1-\beta_1^k) \leftarrow \text{Bias correction} \\ \hat{\mathbf{v}}_k &= \mathbf{v}_k/(1-\beta_2^k) \leftarrow \text{Bias correction} \\ \mathbf{H}_k &= \mathrm{diag}(\sqrt{\hat{\mathbf{v}}_k}) + \epsilon I \\ \mathbf{x}^{k+1} &= \mathbf{x}^k - \gamma \mathbf{H}_k^{-1} \hat{\mathbf{m}}_k \end{cases}$$

(Every vector operation is an element-wise operation)

Output : $\mathbf{x}^{k+1}$ of the last iteration.

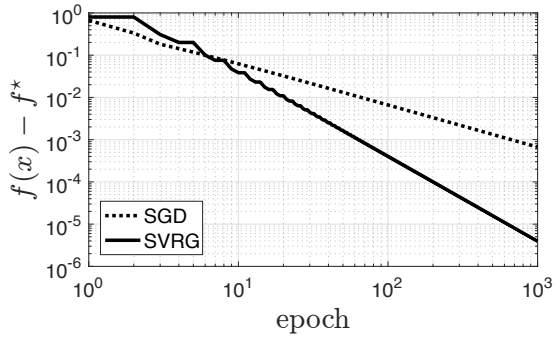
Name:

(b) (5 points) Alban and Volfgang are trying to solve an unconstrained optimization problem:

$$f^\star := \min_{\mathbf{x}\in\mathbb{R}^p} \quad f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}),$$

where each f_i is a smooth and strongly convex function. He implements stochastic gradient descent (SGD) with decreasing step size $\gamma_k = \frac{1}{k}$ and stochastic variance reduced gradient method (SVRG) with fixed step size $\gamma = \frac{1}{5L}$. For the number of iterations in the inner loop of SVRG, Alban uses n. Alban draws convergence plots with respect to epochs, where 1 epoch means that the algorithm either used n stochastic gradients (∇f_i) or 1 full gradient (∇f).

After running his implementations, he obtains the following plots. Volfgang 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 Volfgang? Explain your answer. (No credits will be given unless you provide a clear explanation.)



Name:

(c) (4 points) John and Kim study the constrained optimization problem

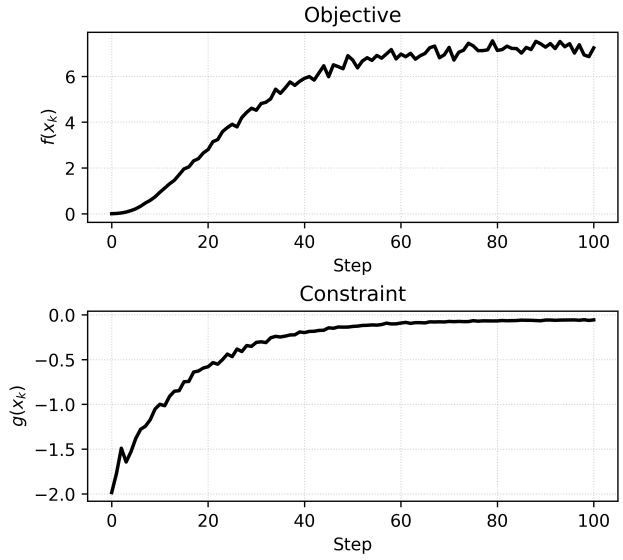
$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{x}) \quad \text{subject to } g(\mathbf{x}) = 0,$$

where $f, g \in C^\infty(\mathbb{R}^d)$. To enforce the constraint, they decide to instead solve the unconstrained problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} F(\mathbf{x}) := f(\mathbf{x}) + 100\, g(\mathbf{x})^2,$$

using gradient descent.

Let $\{\mathbf{x}^k\}_{k\geq 0}$ be the iterates produced by gradient descent on F. During training, John and Kim record the sequences $f(\mathbf{x}^k)$ and $g(\mathbf{x}^k)$ and obtain the following diagnostic plot:



John concludes that there must be a bug in their implementation, since (i) $f(\mathbf{x}^k)$ does not decrease and (ii) $g(\mathbf{x}^k)$ is never exactly zero. Kim is not convinced and argues that the plot alone does not prove there is a mistake.

Which person do you agree with more? Give a clear choice accompanied by a short justification.

Name:

(d) (5 points) Jameson and Drew are training a neural network on an image classification dataset. Drew implemented two adaptive gradient methods: **AdaGrad** and **RMSProp**. Drew followed the simplified flavors of these two algorithms as shown in the boxes below.

AdaGrad (Simplified)

Input. Initial point $\mathbf{x}^0$, step size α

1. Set accumulator $\mathbf{h}_0 = \mathbf{0}$

2. For $k = 1, 2, \ldots,$ iterate

$$\begin{cases} \mathbf{g}_k & = \nabla f(\mathbf{x}^k) \\ \mathbf{h}_k & = \mathbf{h}_{k-1} + \mathbf{g}_k^2 \\ \mathbf{H}_k & = \mathrm{diag}(\mathbf{h}_k) \\ \mathbf{x}^{k+1} & = \mathbf{x}^k - \alpha\, \mathbf{H}_k^{-1/2} \mathbf{g}_k \end{cases}$$

Output : $\mathbf{x}^{k+1}$ of the last iteration.

RMSProp (Simplified)

Input. Initial point $\mathbf{x}^0$, step size α, decay $\beta \in (0, 1)$

1. Set accumulator $\mathbf{h}_0 = \mathbf{0}$

2. For $k = 1, 2, \ldots,$ iterate

$$\begin{cases} \mathbf{g}_k & = \nabla f(\mathbf{x}^k) \\ \mathbf{h}_k & = \beta\, \mathbf{h}_{k-1} + (1-\beta)\, \mathbf{g}_k^2 \\ \mathbf{H}_k & = \mathrm{diag}(\mathbf{h}_k) \\ \mathbf{x}^{k+1} & = \mathbf{x}^k - \alpha\, \mathbf{H}_k^{-1/2} \mathbf{g}_k \end{cases}$$

Output : $\mathbf{x}^{k+1}$ of the last iteration.

Notation. In the above algorithms

(i) For a vector $\mathbf{v} \in \mathbb{R}^d$, $\mathbf{v}^2$ denotes the *componentwise square*, i.e. $(\mathbf{v}^2)_i = v_i^2$.

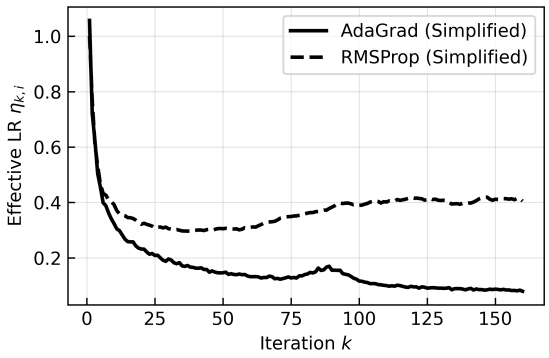
(ii) For a vector $\mathbf{h} \in \mathbb{R}^d$, $\mathrm{diag}(\mathbf{h})$ denotes the diagonal matrix in $\mathbb{R}^{d \times d}$ where the diagonal entries are equal to $\mathbf{h}$, elementwise.

(iii) For a diagonal matrix $\mathbf{H} = \mathrm{diag}(\mathbf{h})$, $\mathbf{H}^{-1/2}$ denotes the diagonal matrix whose ith diagonal entry is $1/\sqrt{h_i}$.

(iv) $\mathbf{x}^k$ represents the parameter vector (e.g. the neural network weights) at iteration k.

(v) The gradient $\nabla f(\mathbf{x}^k)$ is computed over data samples or mini-batches, which are kept implicit in the notation for brevity.

During training, Drew monitored the *effective learning rate* at one coordinate i, computed as $\eta_{k,i} = \alpha \left(\mathbf{H}_k^{-1/2}\right)_{ii}$. The plot below shows $\eta_{k,i}$ as a function of iteration k for both algorithms.



Jameson saw this plot and claimed that there must be a bug in the implementation of the optimizers. Namely, he thought Drew did not strictly follow the two algorithms in the above two boxes. Do you think Jameson is correct? Please give a clear yes-no answer accompanied by a short justification.

Hint. You can check whether the effective learning rates of the two algorithms should be monotonic in theory.

Name:

PROBLEM 4: THE PROXIMAL OPERATOR AND THE PROXIMAL POINT METHOD
Recall the following definitions from the lectures:

Definition 1. *Let $f : \mathbb{R}^p \to \mathbb{R}$ be a continuous and differentiable function. The proximal operator of f is defined as:*

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

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

$$\text{prox}_{\lambda f}^{D_h}(\mathbf{y}) \equiv \arg\min_{\mathbf{z}\in\mathbb{R}^p} \left\{ f(\mathbf{z}) + \frac{1}{\lambda} D_h(\mathbf{y},\mathbf{z}) \right\}, \tag{2}$$

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 $\mathbf{z}$ and $\mathbf{y}$ is:

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

(a) (5 points) Show that when $h(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$, the Bregman proximal operator reduces to the proximal operator.

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

(b) (5 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.

Does the objective value always decrease in such update step? That is, does $f(\mathbf{x}_{k+1}) \le f(\mathbf{x}_k)$ always hold? Provide a proof if yes or a counterexample if not.

(c) (5 points) For minimization, the proximal point method (PPM) update is given by

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}} \left\{ f(\mathbf{x}) + \tfrac{1}{2\lambda}\|\mathbf{x}-\mathbf{x}_k\|_2^2 \right\}.$$

As discussed in the homework, the same idea applies to min–max problems.
Consider $\min_{\mathbf{x}\in\mathbb{R}^p} \max_{\mathbf{y}\in\mathbb{R}^p} \Phi(\mathbf{x},\mathbf{y}) = a\,\mathbf{x}^\top\mathbf{y}, \quad a > 0.$
Define

$$\mathbf{z}_k = \begin{pmatrix} \mathbf{x}_k \\ \mathbf{y}_k \end{pmatrix}, \qquad G(\mathbf{z}) = \begin{pmatrix} \nabla_{\mathbf{x}}\Phi(\mathbf{x},\mathbf{y}) \\ -\nabla_{\mathbf{y}}\Phi(\mathbf{x},\mathbf{y}) \end{pmatrix}.$$

The PPM update is

$$\mathbf{z}_{k+1} = \mathbf{z}_k - \lambda\, G(\mathbf{z}_{k+1})\,.$$

Name:

Starting from the PPM step above, write the updates for $\mathbf{x}_{k+1}$ and $\mathbf{y}_{k+1}$ explicitly in terms of $\mathbf{x}_k$ and $\mathbf{y}_k$.
Hint: The inverse of $\begin{pmatrix} I & \alpha I \\ -\alpha I & I \end{pmatrix}$ is $\frac{1}{1+\alpha^2}\begin{pmatrix} I & -\alpha I \\ \alpha I & I \end{pmatrix}$.

(d) (5 points) Replacing $f(\mathbf{x})$ by its first-order lower bound $f(\mathbf{x}) \approx f(\mathbf{x}_k) + \langle \nabla f(\mathbf{x}_k), \mathbf{x} - \mathbf{x}_k \rangle$ inside the Bregman proximal objective leads to the update

$$x_{k+1} = \arg\min_{\mathbf{x}} \left\{ \langle \nabla f(\mathbf{x}_k), \mathbf{x} \rangle + \frac{1}{\eta} D_h(\mathbf{x}, \mathbf{x}_k) \right\},$$

This update is known as the mirror descent step.
Recall the Fenchel conjugate introduced in class: $h^*(\mathbf{y}) = \max_{\mathbf{x} \in \mathbb{R}^p} \{\langle \mathbf{y}, \mathbf{x} \rangle - h(\mathbf{x})\}$.

Using the first-order optimality condition and the following hint to show that the update is

$$\mathbf{x}_{k+1} = \nabla h^*(\nabla h(\mathbf{x}_k) - \eta \nabla f(\mathbf{x}_k)).$$

[Hint] [not need to prove] ∇h and ∇h^* are inverse maps.

Name:

PROBLEM 5: CONSTRAINED CONVEX OPTIMIZATION (20 POINTS)
Consider the following convex optimization problem:

$$\min_{x \in \mathbb{R}^p} f(x) := \frac{1}{2}\|Ax - b\|_2^2 \quad \text{subject to} \quad \|x\|_1 \leq 1,$$

where $A \in \mathbb{R}^{m \times p}$ and $b \in \mathbb{R}^m$ are given.
Let $A_{1,:}, A_{2,:}, \ldots, A_{m,:} \in \mathbb{R}^p$ denote the rows of A, and define the row norm:

$$\|A\|_{\infty,2} := \max_{1 \leq i \leq m} \|A_{i,:}\|_2.$$

The Frank–Wolfe method for this problem is given by:

$$\hat{x}^k \in \arg\min_{\|x\|_1 \leq 1} \langle \nabla f(x^k), x \rangle,$$

$$x^{k+1} = (1 - \gamma_k)x^k + \gamma_k \hat{x}^k, \quad \text{where } \gamma_k = \frac{2}{k+1}.$$

(a) [10 points] Show that $f(x)$ has a Lipschitz continuous gradient and prove that:

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq m \|A\|_{\infty,2}^2 \|x - y\|_2 \quad \text{for all } x, y \in \mathbb{R}^p.$$

Frank–Wolfe lies as a core of Scion.

(b) [5 points] Show that a solution to the linear minimization problem

$$\hat{x}^k \in \arg\min_{\|x\|_1 \leq 1} \langle \nabla f(x^k), x \rangle$$

is given by the one-sparse vector:

$$(\hat{x}^k)_i = \begin{cases} -\text{sign}((\nabla f(x^k))_i), & i = i_{\max}, \\ 0, & \text{otherwise}, \end{cases}$$

where

$$i_{\max} \in \arg\max_{1 \leq i \leq p} |(\nabla f(x^k))_i|.$$

Name:

(c) [5 points] Suppose one wants to bound the smoothness of $f(x) = \frac{1}{2}\|Ax - b\|_2^2$ in terms of different matrix norms:

- the row norm $\|A\|_{\infty,2} = \max_i \|A_{i,:}\|_2$,
- the Frobenius norm $\|A\|_F$,
- the column norm $\|A\|_{2,\infty} = \max_j \|A_{:,j}\|_2$.

Discuss which of these norms lead to tighter convergence guarantees for the Frank–Wolfe method on the feasible set $\{x : \|x\|_1 \leq 1\}$, and explain which ones are less suitable and why.

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.