ACSYON
Master of Applied Mathematics

ONLINE OPTIMIZATION, LEARNING, AND GAMES
FINAL EXAM (13.01.2022)

INSTRUCTORS: PANAYOTIS MERTIKOPOULOS AND D. QUAN VU

Exam duration: 3 hours

Material allowed: All notes taken in class, as well as all material provided by the instructors.

Grading Scheme: There are 3 problems. Each problem is worth 20 points, for a total of 60 points.

Show your work, pace yourselves – *and good luck!*

Problem 1: Games and their dynamics. This question has two main parts: the first concerns dominated strategies, the second concerns the replicator dynamics.

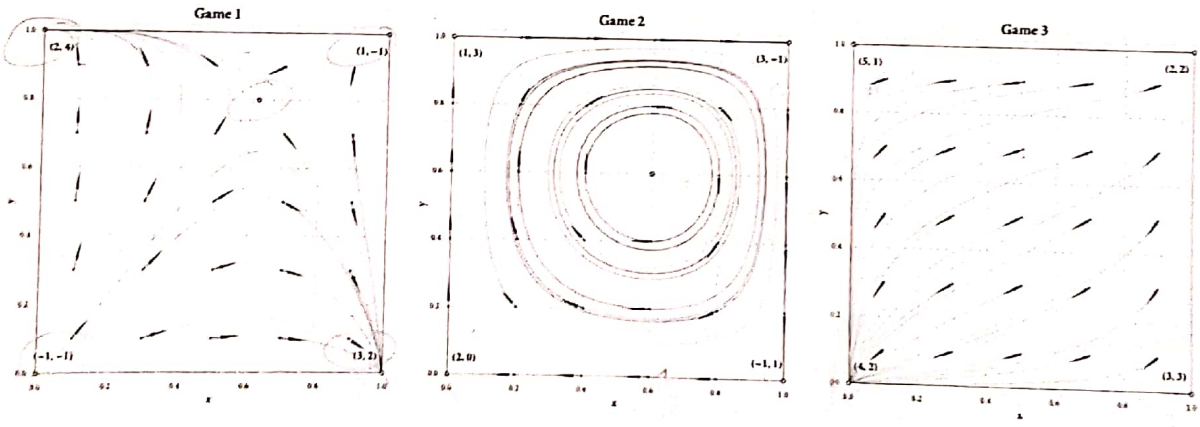
(Q1) Consider the bimatrix game given below:

$(4,7)$	$(6,5)$	$(9,4)$	$(7,4)$
$(6,9)$	$(3,8)$	$(8,7)$	$(5,7)$
$(3,5)$	$(2,6)$	$(1,5)$	$(9,2)$
$(1,12)$	$(2,4)$	$(5,12)$	$(2,3)$

Perform the procedure of iterated elimination of dominated strategies. Is the game dominance-solvable?

[5 points]

(Q2) Each of the diagrams below represents a phase portrait for the replicator dynamics in a 2×2 game.



In each diagram, stationary points have been marked with a blue dot (there are 5 stationary points in the first two games, and 4 in the last one). For each stationary point, indicate:

(a) If it is a Nash equilibrium.
(b) If it is Lyapunov stable.
(c) If it is asymptotically stable.

[15 points total; each diagram is worth 5 points]

Problem 2: Online optimization with strongly convex objectives. Consider the following online optimization problem: at each stage $t = 1, 2, \dots$, the learner selects an action x_t from some compact, convex subset $\mathcal{X}$ of $\mathbb{R}^d$, and incurs a loss $\ell_t(x_t)$. The objective functions $\ell_t : \mathcal{X} \to \mathbb{R}$ are assumed to be continuously differentiable and *strongly convex* on $\mathcal{X}$, i.e., there exists some constant $K > 0$ such that

$$\ell_t(x') - \ell_t(x) \geq \langle \nabla \ell_t(x), x' - x \rangle + \frac{K}{2}\|x' - x\|^2, \tag{SC}$$

for all $x, x' \in \mathcal{X}$.

Suppose now that the learner runs the *online gradient descent* policy

$$x_{t+1} = \Pi_{\mathcal{X}}(x_t - \gamma_t \nabla \ell_t(x_t)), \tag{OGD}$$

where

$$\Pi_{\mathcal{X}}(y) = \arg\min_{x \in \mathcal{X}} \|y - x\|$$

denotes the Euclidean projection of y on $\mathcal{X}$ and $\gamma_t > 0$ is a variable step-size sequence. The objective of this problem is to calculate the worst-case regret incurred by the learner.

(Q1) Fix a benchmark action $p \in \mathcal{X}$ and let $D_t = \|p - x_t\|^2/2$. Show that:

$$D_{t+1} \leq D_t + \gamma_t \langle \nabla \ell_t(x_t), p - x_t \rangle + \frac{1}{2}\gamma_t^2 \|\nabla \ell_t(x_t)\|^2.$$

[4 points]

(Q2) Show that there exist constants $A, B > 0$ such that

$$\sum_{s=1}^{t} \langle \nabla \ell_s(x_s), x_s - p \rangle \leq A + \sum_{s=2}^{t} \left(\frac{1}{\gamma_s} - \frac{1}{\gamma_{s-1}} \right) D_s + \frac{B}{2} \sum_{s=1}^{t} \gamma_s$$

online gradient descent.

[6 points]

(Q3) Use the strong convexity of ℓ_t to conclude that, if (OGD) is run with a step-size sequence of the form $\gamma_t = \gamma/t$ for some $\gamma \geq 1/K$, the learner's regret is bounded as

$$\mathrm{Reg}(T) = \mathcal{O}(\log T).$$

[8 points]

(Q4) Would you recommend this step-size against losses that are *not* strongly convex?

[2 points]

Notation. In the above, $\langle u, v \rangle = \sum_{i=1}^{d} u_i v_i$ denotes the ordinary Euclidean product between $u, v \in \mathbb{R}^d$ and $\|u - v\|^2 = \sum_i (u_i - v_i)^2$ denotes the squared Euclidean distance between u and v.

Useful facts. You may use the following facts for free:

- Projections are nonexpansive:

$$\|\Pi_{\mathcal{X}}(y) - \Pi_{\mathcal{X}}(x)\| \leq \|y - x\| \qquad \text{for all } x, y \in \mathbb{R}^d.$$

- The harmonic series grows logarithmically:

$$\sum_{t=1}^{T} \frac{1}{t} \leq 1 + \log T$$

Problem 3: The doubling trick. You may recall that the regret guarantees of most algorithms studied in class (EXP3, FTRL, ...) depend on properly tuning the learning rate as a function of the learning horizon T (i.e., the total number of rounds the algorithm is run for). The goal of this problem is to introduce the so-called *doubling trick*, a "wrapper" that allows us to run a no-regret algorithm *without* knowing the horizon T beforehand. The procedure is as follows:

Algorithm 1 The doubling trick

Require: Algorithm A whose parameters depend on the time horizon

for all $m = 0, 1, \ldots$ **do**

run A with parameters tuned for $T_m = 2^m$ on the time window $t = 2^m, \ldots, 2^{m+1} - 1$

break when t exceeds the number of rounds to be played

end for

(Q1) Suppose that the doubling trick is applied to an algorithm A guaranteeing at most $\alpha\sqrt{T}$ regret if its parameters are tuned with prior knowledge of T. Show that the total regret of the resulting method against an *unknown* horizon T is at most $C \cdot \alpha\sqrt{T}$, where $C > 0$ is a "universal constant" (like π or $\sqrt{2}$).

[10 points]

(Q2) What would happen above if the windows were taken to be of the form $\{w^m, \ldots, w^{m+1} - 1\}$, $m = 1, 2, \ldots$ for some positive integer $w > 2$? [e.g., "tripling" instead of "doubling", etc.]

[6 points]

(Q3) Which choice of w would minimize asymptotically the resulting anytime bound?

[4 points]

FACULTÉ DES SCIENCES ET TECHNIQUES DE LIMOGES — JANUARY 6, 2022

M2 ACSYON
Deep Learning
Final Exam

Problem 1 [4 points] **Binary classification and stochastic gradient method**
Let us consider a set of points

$$C = \{(x_j, y_j) \in \mathbb{R}^d \times \{-1, 1\} : j = 1, \ldots, n\}.$$

The problem is to find $w \in \mathbb{R}^d$ such that

$$\forall j \in \{1, \ldots, n\} \quad y_j(w^\top x_j) > 0. \tag{1}$$

The data are assumed to be linearly separable, i.e., (1) holds for a separating vector $\bar{w} \in \mathbb{R}^d$.

1. [1 point] Let us consider the function

$$L(w) = \frac{1}{n} \sum_{j=1}^{n} \max\{0, -y_j(w^\top x_j)\}.$$

Show that L is a loss function related to the binary separation problem, i.e., show that L is of the form $L(w) = \frac{1}{n} \sum_{j=1}^{n} \ell(f(x_j, w), y_j)$ and that it is minimized for any separating vector $\bar{w}$.

2. [1 point] Show that the perceptron algorithm can be interpreted as a stochastic gradient algorithm applied to the minimization of L.

3. [2 points] The ReLU function used in the definition of the loss L can be approximated by a function of the form

$$\max\{0, t\} \simeq \frac{1}{a} \log(1 + e^{at}),$$

for some parameter $a > 0$. We then consider now the loss function

$$F(w) = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{a} \log(1 + e^{-a y_j (w^\top x_j)}).$$

Design a binary separation algorithm based on the minimization of F by means of the stochastic gradient method. Describe each step of the algorithm, including the detailed formulas used at each iteration for updating the separation vector w .

Problem 2 [6 points] **Classification of even and odd numbers in the MNIST dataset**

1. [1 point] Download the Jupyter Notebook `NAME_M2EXAM_MNIST.ipynb` from the platform. Import the Pyton stuff that will be useful for the exercise. Import the Keras data set MNIST. Sort the data set into two classes: "class zero" for the even numbers and "class one" for the odd numbers. Reshape the data set in order to apply a convolutional neural network. Each image must be reshaped into a $28 \times 28 \times 1$ tensor. Normalize the data.

2. [2 points] Build a convolutional neural network with the following characteristics:

   - Input layer $28 \times 28 \times 1$.
   - A first convolution layer with 5 kernels of size 3×3 and a ReLU activation function.
   - A 2×2 max-pooling layer.
   - A second convolution layer with 10 kernels of size 3×3 and a ReLU activation function.
   - A second 2×2 max-pooling layer.
   - A fully connected dense layer with 50 neurons and a ReLU activation function.
   - One output layer and a sigmoid activation function.

3. [1 point] Choose the optimizer adam and an appropriate loss function. Train the network with a batch size equals to 512 and 16 epochs. Plot the values of the accuracy and loss function on the training and test sets, along the iterations of the learning process.

4. [1 point] What is the total number of parameters of this network? Detail the number of parameters of each layer (convolution layers, max-pooling layers and dense layer) and show how to determine these values.

5. [1 point] Show the values of the loss and accuracy on the test set. Show all the prediction errors made on the test set of images. How to recover the value of the accuracy?

Problem 3 [10 points] **Text classification and word embeddings interpretation**

1. [0.5 points] Download the Jupyter Notebook `NAME_M2EXAM_20NEWSGROUPS.ipynb` from the platform. Import the Pyton stuff that will be useful for the exercise. Import the Keras data set 20newsgroups with 3 categories:. `'rec.sport.baseball'`, `'sci.electronics'`, ap`'comp.grhics'`. Preprocess the data set in order to remove punctuation, transform all words to a lowercase, replace all digits by `'[DGT]'`, and remove all words which contain just one letter.

2. [0.5 points] Build a vocabulary of size 3000 using `TextVectorization` class from Keras library.

3. [1 point] Build a bidirectional LSTM recurrent neural network for text classification with the following characteristics:

   - `TextVectorization` layer that was prepared in the previous step.
   - Embedding layer of dimension 64.
   - LSTM based Bidirectional layer of dimension 32.
   - Dropout layer with 25% dropping rate.
   - Dense layer with 32 output dimension and ReLU activation function.
   - Dense layer with the output dimension equals to the number of classes and Softmax activation function.

4. [1 point] Choose the optimizer Adam and an appropriate loss function. Train the network with a batch size equals to 128, 10 epochs, and shuffling during training. Save the best model parameters during training.

5. [1 point] Load the best network parameters and obtain the word embeddings matrix. Please, use `model.layers` list to find the Embedding layer and then `embeddings` parameter to retrieve the matrix. Transform the obtained matrix from `tf.Variable` type to `np.array` by calling `.numpy()` method.

6. [2 points] Apply Singular Values Decomposition to find all principal components that spans the word embeddings linear space. Analyze the singular values to define the appropriate dimension of the subspace that contains the biggest part of the information.

7. [1 point] Select the first 2 principal components. Project all word embeddings onto this subspace to obtain the matrix of dimension (3000, 2). Visualize all elements in two-dimensional plot. You can use `plt.scatter` function of the `matplotlib` package for this goal.

8. [1 point] Analyze the obtained plot to understand which 2-dim points represent the most informative words of each category. You can retrain the text classification model with only two categories `'rec.sport.baseball'`, `'sci.electronics'` to see how the plot will change.

9. [2 point] Obtain the most informative words of each category by applying `KMeans` algorithm from the `scikit-learn` library.

Faculté des Sciences et Techniques
Applied Multilinear Algebra

Université de Limoges
M2 – ACSYON – 2021/2022

Final Exam

Exercice 1 Find the SVD of the following matrix $A = \begin{pmatrix} -1 & 0 & -1 \\ 1 & -1 & 0 \end{pmatrix}$. Find the best rank 1 approximation of the matrix A (which we denote by B) and calculate the distance $\|A - B\|_F$ (Frobenius norm).

Exercice 2

Cite one method to find least squares solutions of an overdetermined linear system $Ax = b$.

Apply this method to the system $Ax = b$ where

$$A = \begin{pmatrix} 1 & -2 \\ 0 & -1 \\ -1 & -1 \\ 2 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} -1 \\ 2 \\ -2 \\ 3 \end{pmatrix}.$$

Exercice 3 Let $u, v \in \mathbb{R}^n$ be two nonzero vectors, and let $A = uv^\top$ be the corresponding rank 1 matrix. Prove that the nonzero singular value of A is $\|u\|_2\|v\|_2$.

Exercice 4 For $\alpha = (\alpha_1, \alpha_2) \in \mathbb{N}^2$, let

$$\sigma_\alpha = \begin{cases} 1 & \text{if } \alpha_1 \equiv \alpha_2 \equiv 0 \bmod 2 \\ 0 & \text{otherwise} \end{cases}$$

For $d \in \mathbb{N}$, let

$$T = \sum_{\alpha_1+\alpha_2 \le d} \sigma_\alpha \binom{d}{\alpha} x_0^{d-\alpha_1-\alpha_2} x_1^{\alpha_1} x_2^{\alpha_2}$$

where $\binom{d}{\alpha} = \frac{d!}{(d-\alpha_1-\alpha_2)!\alpha_1!\alpha_2!}$. The objective is to find a Waring decomposition of T of the form

$$T = \sum_{i=1}^{r} \omega_i (x_0 + \xi_{i,1} x_1 + \xi_{i,2} x_2)^d$$

for some minimal r, some weights ω_i and points $\xi_i = (\xi_{i,1}, \xi_{i,2}) \in \mathbb{C}^2$.

We recall that the *apolar product* on symmetric tensors $T = \sum_{|\alpha| \le d} \sigma_\alpha \binom{d}{\alpha} x_0^{d-|\alpha|} x_1^{\alpha_1} x_2^{\alpha_2}$, $T' = \sum_{|\alpha| \le d} \sigma'_\alpha \binom{d}{\alpha} x_0^{d-|\alpha|} x_1^{\alpha_1} x_2^{\alpha_2}$ of order d is defined by

$$\langle T, T' \rangle = \sum_{|\alpha| \le d} \sigma_\alpha \sigma'_\alpha \binom{d}{\alpha}.$$

For $B, B' \subset \mathbb{R}[x_0, x_1, x_2]$, the *catalecticant matrix* of T associated to B, B' is defined by

$$H_T^{B,B'} = (\langle T, bb' \rangle)_{b \in B, b' \in B'}$$

with the convention that if bb' is not homogeneous of degree d then $\langle T, bb' \rangle = 0$. We denote by $H_T^{d',d''}$ the catalecticant matrix of T associated respectively to all the monomials of degree d' and d''.

1. For $\alpha \in \mathbb{N}^2$ with $|\alpha| \leq d$, what is $\langle T, x_0^{d-|\alpha|} x_1^{\alpha_1} x_2^{\alpha_2} \rangle$? $= \sum_{|\alpha| \leq d} \sigma_\alpha \binom{d}{\alpha}$

2. For $d' \geq 2$ and $d - d' \geq 2$, what is the catalecticant matrix $H_T^{B,B'}$ for

   $$B = \{x_0^{d-d'}, x_0^{d-d'-1}x_1, x_0^{d-d'-1}x_2, x_0^{d-d'-2}x_1x_2\}, B' = \{x_0^{d'}, x_0^{d'-1}x_1, x_0^{d'-1}x_2, x_0^{d'-2}x_1x_2\}?$$

   (hint: "set" $x_0 = 1$ to simplify the notation).

3. Show that for $d \geq d' \geq 2$, the elements $(x_0^2 - x_1^2)p$, $(x_0^2 - x_2^2)q$ with $\deg(p) = \deg(q) = d' - 2$ are in $\ker H_T^{d-d',d'}$.

4. Show that $\ker H_T^{d-d',d'}$ is spanned by the elements $(x_0^2 - x_1^2)p$, $(x_0^2 - x_2^2)q$ with $\deg(p) = \deg(q) = d' - 2$.

5. What are the solution points (x_0, x_1, x_2) of the equations $x_0 = 1$, $x_0^2 - x_1^2 = 0$, $x_0^2 - x_2^2 = 0$?

6. Deduce the Waring decomposition of T?

Université de Limoges

Academic Year 2021-2022

Master ACSYON, Final Exam
Fast Algorithmic Methods for Optimization and Learning

January 11, 2022 - 3 hours

Documents are not allowed.

Part 1: Convergence Analysis of the Gradient Descent Method

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a given differentiable function and consider the following unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x). \tag{1}$$

We denote by $x^* \in \operatorname{argmin}(f) \neq \emptyset$ and by $f^* = f(x^*)$ the optimal value of (1).
We say that f is L-smooth if there is exists $L >$ such that

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \ \forall x, \ y \in \mathbb{R}^n. \tag{2}$$

We say that f is strongly convex if there exists $m > 0$ such that

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{m}{2}\|x - y\|^2, \ \forall x, \ y \in \mathbb{R}^n.$$

For a given positive parameter $s > 0$, we define the gradient descent operator $G_{s,f} : \mathbb{R}^n \to \mathbb{R}^n, \ x \mapsto G_{s,f}(x)$ by

$$G_{s,f}(x) = x - s\nabla f(x).$$

The gradient descent method with a constant stepsize $s > 0$ is defined by

$$(\text{GDM}) \begin{cases} x_1 \in \mathbb{R}^n \text{ a given starting point,} \\ x_{k+1} = G_{s,f}(x_k), \ k \geq 1 \end{cases}$$

1. Suppose that f is L-smooth, show that for every $x, \ y \in \mathbb{R}^n$, we have

$$\Big| f(x) - f(y) - \langle \nabla f(y), x - y \rangle \Big| \leq \frac{L}{2}\|x - y\|^2. \tag{3}$$

Hint. Use the following Taylor's formula with integral remainder

$$f(x) - f(y) = \int_0^1 \langle \nabla f(y + \theta(x - y)), x - y \rangle d\theta.$$

2. Show that if f is **convex** and L-smooth, then

$$0 \leq f(x) - f(y) - \langle \nabla f(y), x - y \rangle \leq \frac{L}{2}\|x - y\|^2. \tag{4}$$

3. Show, that if f is convex and L-smooth, then

$$f(x) - f(y) \leq \langle \nabla f(x), x - y \rangle - \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2. \tag{5}$$

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \frac{1}{L}\|\nabla f(x) - \nabla f(y)\|^2. \tag{6}$$

4. Show that if f is convex and L-smooth, then

$$f\Big(G_{s,f}(x)\Big) - f(x) \leq -\frac{s}{2}\|\nabla f(x)\|^2, \ \forall x \in \mathbb{R}^n \ \text{ and } s = \frac{1}{L}. \tag{7}$$

5. In this question, we show a convergence rate for the values of the iterates of (GDM), in the convex case.

   (a) Using (6), show that if f is convex and L-smooth, then

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \frac{1}{L^2}\|\nabla f(x_k)\|^2. \tag{8}$$

   (b) Deduce the the sequence real sequence $\left(\|x_k - x^*\|^2\right)_{k\in\mathbb{N}^*}$ is decreasing.

   (c) Show that if f is convex and L-smooth, then

$$f(x_{k+1}) - f(x_k) \leq -\frac{1}{2L}\|\nabla f(x_k)\|^2. \tag{9}$$

   (d) For each $k \in \mathbb{N}^*$, we set $e_k = f(x_k) - f^*$ the error on the values of f at the iterate x_k. Show that

$$e_{k+1} \leq e_k - \frac{1}{2L\|x_1 - x^*\|^2} e_k^2. \tag{10}$$

   (e) Set $\omega = \dfrac{1}{2L\|x_1 - x^*\|^2}$. Show that $\dfrac{1}{e_{k+1}} - \dfrac{1}{e_k} \geq \omega$ and deduce that $\dfrac{1}{e_k} \geq \omega(k-1)$.

   (f) Deduce that

$$f(x_{k+1}) - f^* \leq 2L\|x_1 - x^*\|^2 \frac{1}{k}. \tag{11}$$

6. Assume that the function f is m-strongly convex and L-smooth, show that for every $x,\ y \in \mathbb{R}^n$, we have

$$\langle \nabla f(x) - \nabla f(y), x - y\rangle \geq \frac{mL}{m+L}\|x - y\|^2 + \frac{1}{m+L}\|\nabla f(x) - \nabla f(y)\|^2. \tag{12}$$

Hint. Apply (6) to the convex function function $x \mapsto g(x) = f(x) - \dfrac{m}{2}\|x\|^2$ (by justifying its smoothness).

7. In this question, we show a convergence rate for the values of the iterates of (GDM), in the strongly convex case. In the following questions we assume that f is m-strongly convex and L-smooth. The condition number related to f is defined by $\kappa = \dfrac{m}{L}$.

   (a) Show that

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - 2s\left[\frac{mL}{m+L}\|x_k - x^*\|^2 + \frac{1}{m+L}\|\nabla f(x_k)\|^2\right] + s^2\|\nabla f(x_k)\|^2, \tag{13}$$

   with $s > 0$ to be chosen later.

   (b) Deduce that

$$\|x_{k+1} - x^*\|^2 \leq \left[1 - \frac{2smL}{m+L}\right]\|x_k - x^*\|^2 + \left[s^2 - \frac{2s}{m+L}\right]\|\nabla f(x_k)\|^2. \tag{14}$$

   (c) Let us set $s = \dfrac{2}{m+L}$ so that the last term in (14) is null. Using the fact that $1 - x \leq e^{-x}$, show that

$$\|x_{k+1} - x^*\|^2 \leq \exp\Big(-\frac{4}{\kappa+1}\Big)\|x_k - x^*\|^2. \tag{15}$$

   (d) Deduce from (15), that

$$\|x_{k+1} - x^*\|^2 \leq \exp\Big(-\frac{4k}{\kappa+1}\Big)\|x_1 - x^*\|^2. \tag{16}$$

   (e) Show that

$$f(x_{k+1}) - f^* \leq \frac{L}{2}\|x_1 - x^*\|^2 \exp\Big(-\frac{4k}{\kappa+1}\Big). \tag{17}$$

   Compare (11) and (17).