ΕΠ08 Αναγνώριση Προτύπων – Μηχανική Μάθηση
1η Εργασία

Τύπος εργασίας: ***Ατομική***

Ημερομηνία παράδοσης: ***Τετάρτη 22/05/2024, 23:55 (Δεν θα δοθεί παράταση)***

Τρόπος παράδοσης: ***Αποκλειστικά μέσω του eclass***

Σύνολο βαθμών: **100** (20% του τελικού βαθμού του μαθήματος)

Η εργασία είναι **ατομική** και αποτελείται από 3 ερωτήματα. Συνιστάται ιδιαίτερα, να αφιερώσετε χρόνο ώστε να κατανοήσετε το θεμελιώδη λογισμό και τη λογική πίσω από τα ερωτήματα της εργασίας και να αποφύγετε την αναζήτηση έτοιμων λύσεων στο διαδίκτυο. Αν ωστόσο συμβουλευτείτε ή/και χρησιμοποιήσετε οποιοδήποτε υλικό ή/και κώδικα που είναι διαθέσιμα στο διαδίκτυο, πρέπει να αναφέρεται σωστά τη πηγή ή/και το σύνδεσμο στην ιστοσελίδα από όπου αντλήσατε πληροφορίες. Σε κάθε περίπτωση, η αντιγραφή τμήματος ή του συνόλου της εργασίας δεν είναι αποδεκτή και στη περίπτωση που διαπιστωθεί αντιγραφή θα μηδενιστούν στο μάθημα όλα τα εμπλεκόμενα μέρη. Θα υπάρξει προφορική εξέταση της εργασίας.

Θα πρέπει να υποβάλετε **ένα μόνο αρχείο Interactive PYthon NoteBook (Jupyter Notebook) μέσω του εργαλείου "Εργασίες" του eclass**, ακολουθώντας την εξής σύβαση ονομασίας για το αρχείο σας: *Επώνυμο_ΑριθμόςΜητρώου.ipynb*

Τόσο ο κώδικας Python όσο και οι απαντήσεις σας στις αναλυτικές/αριθμητικές ερωτήσεις πρέπει να είναι ενσωματωμένα στο ίδιο IPython notebook. Οι μαθηματικές πράξεις μπορούν να ενσωματωθούν στο IPython notebook είτε χρησιμοποιώντας LaTeX σημειογραφία είτε ως εικόνες (π.χ. φωτογραφία χειρόγραφου). Μπορείτε να χρησιμοποιήσετε κελιά επικεφαλίδας για να οργανώσετε περαιτέρω το έγγραφό σας.

Σημαντικό: Το IPython notebook που θα παραδώσετε θα πρέπει βεβαιωθείτε ότι ανοίγει και εκτελείται στο Google Colab.

[Ερώτημα 1: Υπολογισμός Παραγώγων] (10 βαθμοί)

Ένα μεγάλο ποσοστό αλγορίθμων μηχανικής μάθησης απαιτούν την επίλυση κατάλληλων προβλημάτων βελτιστοποίησης στα οποία ο υπολογισμός παραγώγων συναρτήσεων πολλών μεταβλητών είναι κεντρικής σημασίας. Για την εξοικείωσή σας με τον υπολογισμό παραγώγων και την αναλυτική επίλυση προβλημάτων βελτιστοποίησης καλείστε να απαντήσετε στα παρακάτω υποερωτήματα.

Αυτό το ερώτημα αποτελεί καλή ευκαιρία για να ξεκινήσετε να χρησιμοποιείτε το *The Matrix Cookbook* στην πράξη, το οποίο είναι διαθέσιμο στον παρακάτω σύνδεσμο:

https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf

1.1. Έστω μια συνάρτηση $f : \mathbb{R}^d \to \mathbb{R}$ με $f(x) = (x - \mathbf{a})^\mathsf{T}\mathbf{A}(x - \mathbf{a})$, όπου $\mathbf{A} \in \mathbb{R}^{d\times d}$ είναι ένας συμμετρικός πίνακας (δηλ. $\mathbf{A}^\mathsf{T} = \mathbf{A}$) και $\mathbf{a} \in \mathbb{R}^d$. Να υπολογίσετε αναλυτικά, βήμα προς βήμα με το χέρι, την παράγωγο ∇f της f (ως προς το διάνυσμα $x \in \mathbb{R}^d$) παραθέτωντας όλα τα ενδιάμεσα βήματα.

Υπενθύμιση: Η παράγωγος $\nabla f : \mathbb{R}^d \to \mathbb{R}^d$ της f υπολογισμένη στο $\mathbf{x} \in \mathbb{R}^d$ δίνεται ως :

$$\nabla f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_d} \end{bmatrix}$$

1.2. Έστω $\mathbf{A}, \mathbf{B}, \mathbf{X} \in \mathbb{R}^{d\times d}$ πίνακες με τάξη (rank) ίση με d. Να βρείτε, αναλυτικά και σε κλειστή μορφή (closed form solution), το ολικό ελάχιστο του παρακάτω προβλήματος ελαχιστοποίησης :

$$\min_{\mathbf{X}} \|\mathbf{A} - \mathbf{X}\mathbf{B}\|_\mathrm{F}^2$$

Να εξηγήσετε, με λεπτομέρεια, τη διαδικασία για την εύρεση του ελαχίστου την οποία ακολουθήσατε και να παραθέσετε όλα τα ενδιάμεσα βήματα των πράξεων.

[Ερώτημα 2: Gradient Descent] (30 βαθμοί)

Ο αλγόριθμος Gradient Descent αποτελεί την πιο διαδεδομένη μέθοδο βελτιστοποίησης στη μηχανική μάθηση. Σε αυτό το ερώτημα καλείστε να υλοποιήσετε σε Python τον αλγόριθμο αυτό, με σκοπό την ελαχιστοποίηση συναρτήσεων πολλών μεταβλητών.

Συγκεκριμένα για τα δύο σύνολα δεδομένων (σύνολα εκπαίδευσης) που θα βρείτε στο αρχείο *hw1_data.ipynb*, το οποίο είναι διαθέσιμο στο φάκελο "Δεδομένα 1ης εργασίας" του καταλόγου "Έγγραφα" στο eclass, καλείστε να μάθετε (εκτιμήσετε) την πραγματική απεικόνιση μεταξύ δεδομένων εισόδου και στόχου χρησιμοποιώντας, αντίστοιχα για το καθένα από τα σύνολα δεδομένων, τις κλάσεις (χώρους υποθέσεων) :

- $f_1(x;w) = \left(1 + e^{-(w_0 + w_1 x)}\right)^{-1}, \; w \in \mathbb{R}^2$
- $f_2(x;w) = w_0 + w_1 \sigma(w_2 + w_3 x), \; w \in \mathbb{R}^4,$ όπου η $\sigma : \mathbb{R} \to \mathbb{R}$ δίνεται ως

$$\sigma(x) = \begin{cases} x, & x > 0 \\ 0 \end{cases}$$

καθώς και τις ακόλουθες συναρτήσεις κόστους :

- $L_1(w) = -\dfrac{1}{n} \sum_{i=1}^{n} \Big(y_i \log_2 f_1(x_i;w) + (1 - y_i) \log_2 \big(1 - f_1(x_i;w)\big) \Big)$
- $L_2(w) = \dfrac{1}{n} \sum_{i=1}^{n} (y_i - f_2(x_i;w))^2$

όπου με $\{(x_i, y_i)\}_{i=1}^{n}$ συμβολίζονται τα αντίστοιχα δεδομένα εκπαίδευσης.

Σημείωση: Παρατηρήστε ότι η σ δεν είναι παραγωγίσιμη στο $\mathbf{x} = 0$, καθώς τα πλευρικά όρια δε συμπίπτουν. Ωστόσο, θεωρούμε ότι η παράγωγος της δίνεται ως

$$\frac{d\sigma(\mathbf{x})}{dx} = \begin{cases} 1, & \mathbf{x} > 0 \\ 0 \end{cases}$$

2.1. Να υλοποιήσετε σε Python και να εκτελέσετε τον αλγόριθμο Gradient Descent για κάθε μία από τις παραπάνω περιπτώσεις, ως εξής :

1. Αρχικοποιήστε τις τιμές των παραμέτρων ως $[0., 1.]^\intercal$ και $[.5, 1., 2.2, 3.4]^\intercal$ αντίστοιχα.
2. Χρησιμοποιήστε ένα σταθερό ποσοστό εκμάθησης (learning rate) $\eta = 10^{-3}$ για $T = 150$ επαναλήψεις και στις δυο περιπτώσεις.
3. Χρησιμοποιώντας τη βιβλιοθήκη *matplotlib*, για κάθε περίπτωση, να σχεδιάσετε :

α. ένα γράφημα που να απεικονίζει την τιμή της συνάρτησης κόστους (κάθετος άξονας) ως συνάρτηση των επαναλήψεων (οριζόντιος άξονας), και

β. ένα (διδιάστατο) γράφημα στο οποίο απεικονίζονται τα δεδομένα εκπαίδευσης (ως σημεία) και το γράφημα του μοντέλου για την τελική εκτίμηση των τιμών των παραμέτρων (ως συνεχής καμπύλη).

Σχολιάστε τα ευρήματά σας. Παρατηρείτε κάποια διαφορά αν χρησιμοποιήσετε $\eta = 10^{-4}$;

2.2. Να εκτελέσετε εκ νέου τον αλγόριθμο Gradient Descent για κάθε μία από τις παραπάνω περιπτώσεις, χρησιμοποιώντας τις αρχικές τιμές παραμέτρων και τον αριθμό επαναλήψεων που δίνονται στο προηγούμενο υποερώτημα, αλλά με αυτή τη φορά με μεταβλητό ποσοστό εκμάθησης. Συγκεκριμένα :

1. Χρησιμοποιήστε για τις πρώτες 100 επαναλήψεις $\eta = 10^{-3}$, για τις επόμενες 40 επαναλήψεις $\eta = 5 \cdot 10^{-4}$ και τέλος $\eta = 10^{-4}$ για τις τελευταίες 10 επαναλήψεις.

2. Χρησιμοποιήστε αρχικά $\eta = 10^{-3}$ και το οποίο να μειώνετε σε κάθε επόμενη επανάληψη κατά 10^{-6} (από αυτό της προηγούμενης της).

Συγκρίνετε και προσπαθήστε να εξηγήσετε συνοπτικά τις ενδεχόμενες διαφορές στα αποτελέσματα αυτά, τόσο μεταξύ τους όσο και με αυτά του προηγούμενου υποερωτήματος.

[Ερώτημα 3: Αναγνώριση Προσώπων (Face recognition)] *(60 βαθμοί)*

Σε αυτό το ερώτημα θα εφαρμόσετε τη μέθοδο Eigenfaces (δηλαδή συνδυασμό PCA για εξαγωγή χαρακτηριστικών και ταξινομητή πλησιέστερου γείτονα για την αναγνώριση προσώπων). Θα χρησιμοποιήσετε εικόνες προσώπων από τη βάση δεδομένων προσώπων Yale B στην οποία υπάρχουν 10 πρόσωπα που φωτογραφήθηκαν κάτω από 64 διαφορετικές συνθήκες φωτισμού. Χρησιμοποιώντας την υλοποίησή σας, θα αξιολογήσετε την ικανότητα του αλγορίθμου Eigenfaces να χειρίζεται συνθήκες φωτισμού των εικόνων ελέγχου (test set) οι οποίες διαφέρουν από αυτές στις εικόνες εκπαίδευσης (training set).

Τα δεδομένα είναι διαθέσιμα στο αρχείο faces.zip στο κατάλογο Έγγραφα στο eclass. Η μέθοδος Eigenfaces για την αναγνώριση προσώπων περιλαμβάνει 3 βασικά βήματα:

Βήμα 1: Κάθε εικόνα διάστασης 50 x 50 pixels του συνόλου εκπαίδευσης μετατρέπεται σε διάνυσμα διάστασης 2500 στοιχείων και αποθηκεύεται ως στήλη στον πίνακα δεδομένων εκπαίδευσης X. Στη συνέχεια εφαρμόζουμε principal component analysis (PCA) στον πίνακα δεδομένων εκπαίδευσης και εξάγουμε τις d κύριες συνιστώσες (principal components). Τα d ιδιοδιανύσματα (eigenvectors) όταν μετατραπούν και απεικονιστούν ως εικόνες ονομάζονται Eigenfaces.

Βήμα 2: Προβάλουμε τις εικόνες των συνόλων εκπαίδευσης και ελέγχου στο χώρο d διαστάσεων και με αυτόν το τρόπο εξάγουμε χαρακτηριστικά χαμηλής διάστασης (d-dimensional features). Ο χώρος χαμηλής διάστασης d ονομάζεται ιδιοχώρος (eigenspace).

Βήμα 3: Η αναγνώριση των προσώπων γίνεται στον eigenspace χρησιμοποιώντας ταξινομητή (ενός) πλησιέστερου γείτονα με Ευκλείδεια απόσταση ως μετρική.

Από το σύνολο δεδομένων προσώπων Yale B θα χρησιμοποιήσετε τα παρακάτω υποσύνολα:

Set_1: person*_01.png έως person*_07.png (δηλαδή τις 7 πρώτες εικόνες κάθε προσώπου)

Set_2: person*_08.png έως person*_19.png
Set_3: person*_20.png έως person*_31.png
Set_4: person*_32.png εως person*_45.png

Set_5: person*_46.png έως person*_64.png

Ζητούμενα:

1. Να γράψετε μία συνάρτηση loadImages(path, set_number) η οποία παίρνει ως είσοδο το path στο οποίο βρίσκεται ο φάκελος των εικόνων π.χ. loadImages("C:/images", "Set_1"), διαβάζει τις εικόνες και επιστέφει έναν πίνακα δεδομένων ανάλογα με το set_number, όπου κάθε εικόνα αναπαρίσταται ως διάνυσμα στήλη. Η συνάρτηση επιστέφει επίσης τις κατηγορίες (labels) στις οποίες ανήκουν οι διαφορετικές εικόνες κωδικοποιημένες με ακεραίους (π.χ. 0 για φωτογραφίες που ανήκουν στο person_0, 1 για τις φωτογραφίες που ανήκουν στο person_1 κτλ).
2. Να εκπαιδεύσετε την μέθοδο Eigenfaces με d = 30 χρησιμοποιώντας όλες τις εικόνες στο Set_1 (70 εικόνες). Στην συνέχεια χρησιμοποιήστε τα Eigenfaces για να ανακατασκευάσετε μία τυχαία εικόνα του Set_1. Τι παρατηρείτε;
3. Εκπαιδεύστε την μέθοδο Eigenfaces για 10 διαφορετικές τιμές του d στο διάστημα [2, 200]. Στην συνέχεια, για κάθε d, ανακατασκευάστε όλες τις εικόνες του Set_1 και υπολογίστε την μέση τιμή του error_i = $||X_i - \hat{X}_i||_2$ όπου X_i η αρχική εικόνα και $\hat{X}_i$ η ανακατασκευασμένη. Απεικονίστε σε διάγραμμα την τιμή του σφάλματος για διαφορετικά d. Τι παρατηρείτε;
4. Να απεικονίσετε (σε μορφή εικόνας) τα 9 κύρια ιδιοδιανύσματα (9 top eigenvectors) που προέκυψαν αφού εκπαιδεύσατε την μέθοδο Eigenfaces στο Set_1. Τι παρατηρείτε; Τι θα μπορούσαμε να πούμε ότι εκφράζουν τα διαφορετικά ιδιοδιανύσματα;

5. Χρησιμοποιήστε τα 10 διαφορετικά eigenfaces που έχετε εκπαιδεύσει για να γνωρίσετε τα πρόσωπα στα Set_1 έως Set_5. Για κάθε Set δημιουργείστε ένα διάγραμμα της απόδοσης ταξινόμησης ως προς των αριθμό συνιστωσών (d). Απαντήστε και σχολιάστε τα ακόλουθα ερωτήματα:
   a. Για ποιο d έχουμε την καλύτερη απόδοση στο Set_2; Πως η απόδοση αυτή αλλάζει στα υπόλοιπα sets και γιατί;
   b. Ποιο d έχει την σταθερότερη απόδοση στα διαφορετικά sets;
   c. Ποιο d θα επιλέγατε για το μοντέλο σας;
6. Χρησιμοποιείστε τα Eigenfaces για το d που επιλέξατε για να ανακατασκευάσετε μια τυχαία εικόνα από κάθε ένα από τα 5 Sets. Να απεικονίσετε τόσο τις αρχικές εικόνες όσο και τις ανακατασκευασμένες. Να σχολιάσετε την ποιότητα ανακατασκευής κάθε εικόνας.
7. Να απεικονίσετε τα 9 κύρια singular vectors που προκύπτουν αφού εφαρμόσετε SVD στον πίνακα δεδομένων του Set_1. Διαφέρουν τα singular vectors από τα αντίστοιχα ιδιοδιανύσματα; Αν ναι, γιατί;

Σημείωση: Μπορείτε να χρησιμοποιήσετε είτε έτοιμες υλοποιήσεις της PCA είτε να την υλοποιήσετε χρησιμοποιώντας ιδοανάλυση (συναρτήσεις τύπου eig) στον πίνακα συνδιακύμανσης. Προτείνεται να προ-επεξεργαστείτε κάθε εικόνα αφαιρώντας τη μέση τιμή της και διαιρώντας με την τυπική απόκλιση των τιμών της.