

למידה חישובית 1 (096411)

חורף תשפ"ה 2025

תרגיל בית 2

תאריך אחרון להגשה: 18/12/2024 בשעה 23:59

Submission guidelines

- Submission is in pairs, only one team member should submit.
- You are to submit **two files**:
 - **HW2_ID1_ID2_wet.py** – file containing the code for question 2.
 - **HW2_ID1_ID2_dry.pdf** – solution for all the written exercises, including screenshots of the code and plots for the questions in the wet part.
- All code (inside notebooks and .py files) must be clear and concise (documented, using meaningful variable names, etc.).
- Every plot must contain at least the following: a meaningful title, axis labels, ticks, and a legend.
- There are no python/packages version requirements.
- Please use the HW forum for questions since these could be helpful for other classmates.



שאלה 1

בשאלה הזו, נעבוד עם דאטה-סט המגיע מניתוח כימי של יינות שיוצרו באותו המחוז באיטליה ע"י שלושה יקבים שונים. הדאטה-סט מכיל 178 תצפיות של יינות, כל תצפית (x, y) מורכבת מווקטור $x \in \mathbb{R}^{13}$ המתייחס לתכונות של היין ומהלייבל המתאים לו $y \in \{0, 1, 2\}$. המתייחס ליקב בו יוצר היין.

נתחיל בטעינת הדאטה-סט לאובייקט מסוג DataFrame:

```
from sklearn.datasets import load_wine
# Read the wine dataset
dataset = load_wine()
df = pd.DataFrame(data=dataset['data'], columns=dataset['feature_names'])
df = df.assign(target=pd.Series(dataset['target']).values)
```

בסעיפים הבאים נתמקד רק בשתי תכונות של כל אחד מהיינות – אחוז האלכוהול (alcohol) וכמות המגנזיום (magnesium). בנוסף, בשאלה זו נעסוק בבעיית סיווג בינארית ולכן נתבונן רק על יקבים 1 ו-2.

```
# Filter the irrelevant columns
df = df[['alcohol', 'magnesium', 'target']]
# Filter the irrelevant label
df = df[df.target != 0]
```

כעת, נפריד את הדאטה-סט למדגם אימון ומדגם ולידיה באופן הבא:

```
train_df, val_df = train_test_split(df, test_size=30, random_state=3)
```

1. הציגו את שני המדגמים בגרף מסוג *scatter plot* (כל מדגם בגרף נפרד). הקפידו לצבוע את היינות המגיעים מיקב 1 בצבע אחד ואת היינות המגיעים מיקב 2 בצבע אחר. אם נפעיל את אלגוריתם ה-*hard-SVM* על מדגם האימון, מהו הפתרון שיוחזר לנו? הסבירו.

בסעיפים הבאים נשתמש באלגוריתם ה-*soft-SVM* של ספריית *scikit-learn*.

```
from sklearn.svm import SVC
```

לאלגוריתם יש מספר פרמטרים ובשלב זה אנחנו נתייחס לשלושה מהם – *'kernel'* שמגדיר את פונקציית מיפוי הנתונים, *'C'* שהוא פרמטר הרגולריזציה שהוצג בתרגול, ו-*'degree'* שקובע את מעלת הפולינום בקרנל פולינומי (*'kernel='poly'*). לדוגמה, עבור מודל עם *kernel* לינארי ופרמטר רגולריזציה 1 נגדיר:

```
model = SVC(kernel='linear', C=1.0)
```

2. הריצו את אלגוריתם ה-*soft-SVM* עם קרנל לינארי על מדגם האימון עבור $C \in \{0.01, 0.05, 0.1\}$. הציגו שני גרפים לכל אחד משלושת המודלים (סה"כ שישה גרפים): הראשון יכלול את מדגם האימון והשני את מדגם הוולידציה. בכל אחד מן הגרפים הציגו את מישור ההחלטה שהתקבל ואת שוליו. בנוסף, בגרף המציג את מדגם האימון, הדגישו את ה-*support vectors* באופן דומה לדרך שבה הודגשו במחברת הקוד שצורפה לשקפי תרגול 4.

3. בהרצאה ובתרגול ראינו כי ניתן לכתוב את בעיית ה-*hard-SVM* כבעיית אופטימיזציה ריבועית. היעזרו בייצוג זה כדי להראות שה-*margin* של מישור ההחלטה המוגדר ע"י \hat{w} ($\text{margin} = \min_{i \in [m]} |\langle \hat{w}, x_i \rangle|$) שווה ל- $\frac{1}{\|\hat{w}_0\|}$, כאשר w_0 הוא פתרון הבעיה ו- $\hat{w} = \frac{w_0}{\|w_0\|}$.

4. ניתן להראות כי המסקנה מן הסעיף הקודם נכונה גם עבור בעיית ה-*soft-SVM*. הציגו גרף קווי של ה-*margin* כפונקציה של C והסבירו אותו. בהסברכם, התייחסו לתפקידו של כל אחד מן הרכיבים בפונקציית המטרה ולטרייד-אוף ביניהם.

5. הציגו גרף קווי של שגיאת האימון ושגיאת הוולידציה ($\text{Error} = 1.0 - \text{Accuracy}$) כפונקציה של C והסבירו אותו. בהסברכם התייחסו לתוצאות הסעיף הקודם.
6. הריצו את אלגוריתם ה-soft-SVM עם קרנל פולינומי על מדגם האימון עבור $\text{degree} \in \{2, \dots, 8\}$ (ו- $C = 1$). הציגו גרף קווי של שגיאת האימון ושגיאת הוולידציה ($\text{Error} = 1.0 - \text{Accuracy}$) כפונקציה של degree . האם ייצוג נתונים במרחב גבוה יותר בהכרח עוזר לשיפור הדיוק על הוולידציה? למה לדעתכם זה קורה?
7. עבור שני ערכי ה- degree שנותנים את השגיאה הנמוכה ביותר והגבוה ביותר, חזרו על השרטוט מסעיף 2 (סה"כ 4 גרפים).

שאלה 2

בשאלה זו עליכם לממש אלגוריתם Perceptron הפותר בעיית סיווג רב-מחלקתית (multiclass). קיימת הרחבת multiclass טבעית לפרספטרוני הבינארי שראינו בהרצאה וה- Pseudo-code של הרחבה זו מצורף בסוף השאלה.

לתרגיל זה מצורף קובץ בשם **HW2_ID1_ID2.py** בו תמצאו מחלקה בשם **PerceptronClassifier**. מחלקה זו מכילה 2 מתודות שעליכם לממש: **fit** ו-**predict**.

- **fit** מקבלת סט נתונים מתויג $S_{train} = \{(x_i, y_i)\}_{i=1}^{m_{train}}$ ומשתמשת בו בכדי לאמן מסווג Perceptron רב מחלקתי.
 - **predict** מקבלת סט נתונים לא מתויג $S_{test} = \{x_i\}_{i=1}^{m_{test}}$ ועבור כל נקודה x_i בסט זה, המתודה חוזרת את ערך התווית y_i של הנקודה.
- מובטח כי לאחר שאובייקט מסוג **PerceptronClassifier** אותחל, קריאה ל **predict** תתבצע רק לאחר קריאה ל **fit**. יש להיעזר בתיעוד בתוך **HW2_ID1_ID2.py** עבור דרישות קלט/פלט מדויקות.

הערות:

- אפשר להניח כי כל סט נתונים שיוספק ל- **fit** הינו ניתן להפרדה לינארית (linearly separable).
- ייתכן כי $m_{test} \neq m_{train}$.
- מובטח כי כל קובץ csv / מערך נתונים שיוספק לריצת התכנית שלכם לא יכיל שורת כותרת (header row).
- מובטח כי העמודה האחרונה של כל סט נתונים לאימון (כזה שיינתן ל **fit**) יכיל ערכים נומריים $(0, 1, \dots, k)$ של תוויות (לייבלים).
- שימו לב כי סט הנתונים יכול להכיל יותר מ $k = 2$ מחלקות ועל המימוש שלכם לתמוך במספר שרירותי של מחלקות.

הקובץ **HW2_ID1_ID2.py** שלכם אמור לרוץ מה- **command line** באופן הבא:

```
python HW2_ID1_ID2.py <path_to_csv>
```

- **<path_to_csv>** - נתיב לקובץ csv מקומי שמכיל סט נתונים. לדוגמה, הרצת הסקריפט מה- **command line** (כאשר הקובץ **iris_sep.csv** נמצא באותה תיקייה כמו הקובץ **HW2_ID1_ID2.py**) תראה כך:

```
python HW2_ID1_ID2.py iris_sep.csv
```

קריאה זו תריץ (fit and predict) מסווג Perceptron רב מחלקתי על סט הנתונים **iris_sep.csv**.

שאלה 2 – דגשים חשובים:

- עליכם למלא את מס' הסטודנט שלכם במתודה **__init__** בהתאם לתיעוד בקובץ **HW2_ID1_ID2.py**.
- מותר לכם לייבא את המודולים הבאים בלבד:
 - **numpy**
 - **pandas**
 - ספריות פייתון בסיסיות (os, argparse וכו'). ספריות שמגיעות עם התקנה נקיה של סביבת פייתון. אם אינכם בטוחים אם ספרייה היא ספרייה בסיסית או לא, אנא שאלו בפורום תרגילי הבית).
 - בפירוש אסור לכם לייבא ספרייה כדוגמת **sklearn** (או כל ספרייה היורשת ממנה).
- אתם יכולים להוסיף ולערוך את המתודות בתוך המחלקה **PerceptronClassifier**. עם זאת, אסור לכם לשנות את החתימה של המתודות **__init__**, **fit** ו-**predict**.

4. מחלקת ה- **PerceptronClassifier** שתכתבו תיקרא ע"י סקריפט אחר, לכן שינוי של בלוק ה- **__main__** בקובץ **HW2_ID1_ID2.py** לא ישפיע על ביצועי הקוד שלכם. בדיקת הקוד שלכם תיעשה באופן שדומה לזו שמתרחשת בבלוק ה- **__main__** אבל הוא שם רק לנוחותכם.
5. הקוד שלכם צריך לרוץ בזמן ריצה סביר (פחות מדקה להרצה של **fit** ו-**predict** יחדיו עבור סט של 100 דוגמאות).
- אי עמידה בדרישות 1-5 תגרור ציון של 0 על שאלה זו.
 - קובץ **HW2_ID1_ID2.py** שזורק שגיאה מכל סוג או לא מצליח לרוץ מה **command line** יגרור ציון של 0 על שאלה זו.

Algorithm 3 MULTICLASS PERCEPTRON

Require: Number of classes K , number of rounds T .

Require: Inner product space $(V, \langle \cdot, \cdot \rangle)$.

Initialize $w_1^{(1)} = w_2^{(1)} = \dots = w_K^{(1)} = 0$

for $t = 1, 2, \dots, T$ **do**

 Observe feature vector $x_t \in V$

 Predict $\hat{y}_t = \operatorname{argmax}_{i \in \{1, 2, \dots, K\}} \langle w_t^{(i)}, x_t \rangle$

 Observe $y_t \in \{1, 2, \dots, K\}$

if $\hat{y}_t \neq y_t$ **then**

 Set $w_i^{(t+1)} = w_i^{(t)}$

for all $i \in \{1, 2, \dots, K\} \setminus \{y_t, \hat{y}_t\}$

 Update $w_{y_t}^{(t+1)} = w_{y_t}^{(t)} + x_t$

 Update $w_{\hat{y}_t}^{(t+1)} = w_{\hat{y}_t}^{(t)} - x_t$

else

 Set $w_i^{(t+1)} = w_i^{(t)}$ **for all** $i \in \{1, 2, \dots, K\}$

Multiclass perceptron pseudo-code [source]. Please note that your implementation should not use a number of rounds (T) as input, as our data is guaranteed to be separable. That is, your for loop should not be bounded and an output should be returned.

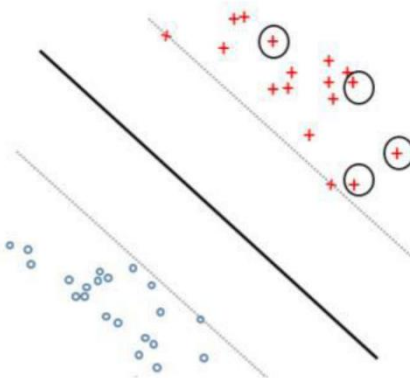
שאלה 3

נרצה לפתור בעיית סיווג בינארי $y \in \{-1, 1\}$ עבור מדגם $S = \{(x_i, y_i)\}_{i=1}^m$, כאשר $x_i \in \mathbb{R}^d$. כלל ההחלטה לסיווג הינו $\text{sign}(\langle w, x \rangle)$ כאשר $w \in \mathbb{R}^d$.

1. מתי המסווג יוכל להשיג שגיאת סיווג אפס, ומתי לא? לכל אחד משני המקרים, ציירו דוגמה של מדגם ומישור מפריד שמשגיג שגיאת סיווג אפס (אם קיים): עבור $d = 1$ ועבור $d = 2$.

2. נניח כי המדגם מכיל שתי נקודות $x_1 = (p, 0)$, $x_2 = (0, q)$ עם תיוגים $y_1 = -1$, $y_2 = 1$, בהתאמה. כתלות בערכי p, q מצאו מתי קיים פתרון ל-hard-SVM ומה הוא הפתרון כאשר קיים? מצאו את w .

3. נחזור למדגם S ונניח שהחלטתם לאמן את המודל שלנו באמצעות hard-SVM, אולם חלק מהנקודות ב- S הלכו לאיבוד ולכן לא התאמנתם עליהן. בציור הבא מופיע המישור המפריד שהתקבל, כאשר הנקודות המוקפות הן הנקודות שהלכו לאיבוד (בהן לא השתמשתם בזמן האימון). אם הייתם מאמנים את המודל שוב, הפעם עם כלל מדגם האימון S , האם הייתם מקבלים מפריד שונה? הסבירו.



4. כזכור, בהינתן פרמטר רגולריזציה $\lambda > 0$, בעיית ה-soft-SVM מוגדרת באופן הבא:

$$w^* = \underset{w}{\operatorname{argmin}} \left\{ \frac{1}{m} \sum_{i=1}^m \max \{0, 1 - y_i \langle w, x_i \rangle\} + \lambda \|w\|^2 \right\}$$

בשרטוטים הבאים מופיעים שני מפרידים לינאריים שהתקבלו ע"י שימוש בערכי λ שונים. ידוע כי שרטוט אחד התקבל ע"י שימוש ב- $\lambda = 2$ ושרטוט אחר התקבל על ידי שימוש ב- $\lambda = 200$. הנקודות המוקפות בכל שרטוט הן ה-support vectors המתאימים של כל מפריד. איזה ערך λ מתאים לכל שרטוט? בתשובתכם התייחסו למשמעות של הפרמטר λ .

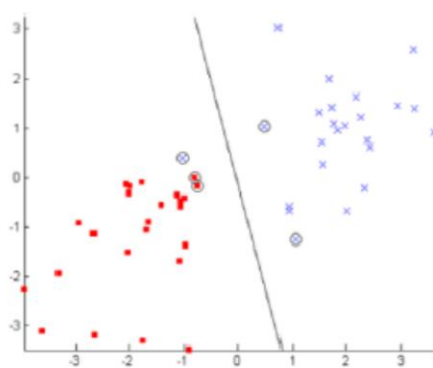


Figure 1

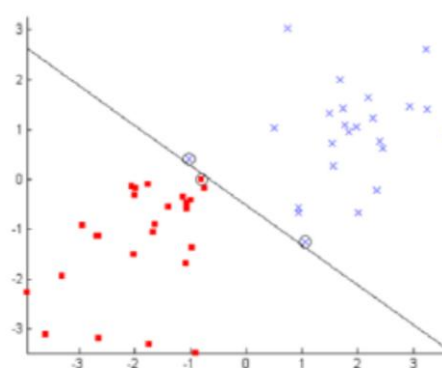


Figure 2

שאלה 4

בשאלה זו עליכם לשנות את אלגוריתם הפרספטון הבינארי כך שיעבוד עם פונקציית kernel.

תהי $X \rightarrow F$ פונקציה הממפה דוגמאות נתונות למרחב פיצ'רים כלשהו F (כאשר נדרוש ש- F הוא תת קבוצה של מרחב הילברט). אזי הפונקציה $K: X \times X \rightarrow R$ המוגדרת על ידי

$$K(x, x') = \langle \psi(x), \psi(x') \rangle$$

היא פונקציית kernel.

בהינתן מדגם אימון $S = \{(x_i, y_i)\}_{i=1}^m$, כאשר $x_i \in R^d =: X$, $y_i \in \{-1, 1\}$, נמפה את הדוגמאות למדגם אימון ב- F , $S^\psi = \{(\psi(x_i), y_i)\}_{i=1}^m$.

1. כתבו פסאודו-קוד של אלגוריתם הפרספטון על מדגם האימון S^ψ . שימו לב ש- ψ לא ידועה ורק K נתון. כהדרכה לסעיף זה, ענו גם על השאלות הבאות:

- מהו הפלט של האלגוריתם? (לא ניתן להחזיר מישור מפריד ב- F מכיוון ש- ψ אינה ידועה).
 - בהינתן הפלט של האלגוריתם, הביעו את כלל ההחלטה, $h: X \rightarrow \{-1, 1\}$, באמצעות K (ללא ψ).
2. הוכיחו שהאלגוריתם שכתבתם מתכנס אם ורק אם S^ψ פריד לינארית ב- F .