עומר גבאי: 318256120
יותם פלצמן: 315717181

Introduction to machine learning
Exercise 3

Winter 2024

Submission guidelines, **read and follow carefully**:

- The exercise **must** be submitted in pairs.
- Submit via Moodle.
- The submission should include two separate files:
  1. A pdf file that includes your answers to all the questions.
  2. The code files for the python question. You must submit a copy of the shell python file provided for this exercise in Moodle, with the required functions implemented by you. **Do not change the name of this file!** In addition, you can also submit other code files that are used by the shell file.
- Your python code should follow the course python guidelines. See the Moodle website for guidelines and python resources.
- Before you submit, **make sure that your code works in the course environment**, as explained in the guidelines. Specifically, **make sure that the test `simple_test` provided in the shell file works**.
- You may only use python modules that are explicitly allowed in the exercise or in the guidelines. If you are wondering whether you can use another module, ask a question in the exercise forum. No module containing machine learning algorithms will be allowed.
- For questions, use the exercise forum, or if they are not of public interest, send them via the course requests system.
- Grading: Q1: 17 points. Q2:18 points. Q3:18 points. Q4:12 points. Q5a:10. Q5b,c:5 points each. Q6:15 points.

Question 1. Implement the soft-margin kernel SVM routine described in class, using `cvxopt` quadratic problem solver you used in Question 1 Assignment 2. The algorithm should use the polynomial kernel. You will implement the function `softsvmpoly` in the shell file "softsvmpoly.py" which is provided for this exercise in Moodle. function details:

```
def softsvmpoly(l, k, trainX, trainY)
```

The input parameters are:

- `l` - the parameter λ of the soft SVM algorithm.
- `k` - the degree of the polynomial kernel.
- `trainX` - a 2-D matrix of size $m \times d$. Row i in this matrix is a vector with d coordinates that describes example x_i from the training sample.
- `trainY` - a column vector of length m. The i's number in this vector is the label y_i from the training sample. You can assume that each label is either -1 or 1.

The function returns the column vector `alpha` $\in \mathbb{R}^m$, which contains the coefficients found by the algorithm.

Question 2. For this question, use the data file `EX3q2_data.npz` provided on the course website, which contains data points in the domain $\mathcal{X} = \mathbb{R}^2$ and labels in $\{-1, 1\}$, split into a training set and test set.

(a) We would like to use soft SVM to learn a predictor for this problem. Plot the points in the training set in $\mathbb{R}^2$, and color them by their label. Explain why it may be a better idea use kernel soft SVM and not the linear (non-kernel) soft SVM. You can use the function `matplotlib.pyplot.scatter`.

(b) Run your Polynomial soft SVM code on the training set. Perform 5-fold cross-validation to tune λ and k. Try the values $\lambda \in \{1, 10, 100\}$ and $k \in \{2, 5, 8\}$ — a total of 9 parameter pairs to try. Report the 9 average validation error values for each of the pairs (λ, k). Report which pair was selected by the cross validation, rerun the training using this pair on the entire training set, and report the test error of the resulting classifier.

(c) For a general classification problem, give one reason why a polynomial SVM might get a better validation error than linear soft SVM, and one reason why it might get a worse validation error.

(d) Set $\lambda = 100$ and consider $k \in \{3, 5, 8\}$. For these values, run the polynomial soft SVM on the training set, and plot the resulting predictor in $\mathbb{R}^2$ as follows: Define a fixed region (roughly the region in which the training data resides), divide it into a fine grid, and color the grid points red or blue, depending on the label predicted by the classifier for each point. You can use the the function `matplotlib.pyplot.imshow` to plot this.

Question 3. **Kernel functions.** Consider a space of examples $\mathcal{X} = \mathbb{R}^d$. Let $x, x' \in \mathcal{X}$.

(a) Prove that the following function *cannot be* a kernel function for any feature mapping:

$$K(x, x') := (2x(7) + x(3)) \cdot x'(2).$$

Hint: what property of inner products does this function violate? How can you prove it?

(b) Prove that the following function *cannot be* a kernel function for any feature mapping:

$$K(x, x') := 5 - (x(1) - x(2))(x'(1) - x'(2)).$$

Hint: consider the case $x = x'$.

(c) Show that the following function f is a Kernel function, by providing a possible feature mapping Ψ such that $f(x, x') = \langle \Psi(x), \Psi(x') \rangle$:

$$f(x, x') = (x(1)x'(1))^6 + e^{x(3)+x(5)+x'(3)+x'(5)} + 1/(x(1)x(1')) + (x(4)+x(6))(x'(4)+x'(6)).$$

Question 4. Consider a linear combination of $k \in \{1, 2, \dots\}$ **convex** functions $f_i : \mathbb{R}^d \to \mathbb{R}$ for $i \in \{1, \dots, k\}$:

$$g(u) = \sum_{i=1}^{k} a_i f_i(u)$$

where $a_1, \dots, a_k \in \mathbb{R}$ and $u \in \mathbb{R}^d$.

(a) Prove that g is not necessarily a convex function for any $a_1, \dots, a_k \in \mathbb{R}$.

(b) Prove that g is a convex function if $a_1, \dots, a_k \geq 0$.

Question 5. Let $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$ where the input domain is $\mathcal{X} = \mathbb{R}^d$ and the labels are from $\mathcal{Y} = \mathbb{R}$. Consider learning of a linear predictor using the following optimization for $\lambda > 0$:

$$\underset{w \in \mathbb{R}^d}{\text{minimize}}\ \lambda \|w - v\|_2^2 + \sum_{i=1}^{m} (\langle w, x_i \rangle - y_i)^2.$$

where $v \in \mathbb{R}^d$ is a given (fixed) vector.
Organize the training examples in S by defining the input data matrix $\mathbf{X} = \left[x_1, x_2, \dots, x_{\cancel{d}\,m}\right]$ and the labels vector $\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{\cancel{d}\,m} \end{bmatrix}$.

(a) Formulate the closed-form expression for the w that solves the optimization in this question. In the closed-form expression you can use only $\mathbf{X}$, $\mathbf{y}$, λ, v that were defined above.

(b) Suppose that Gradient Descent is run on S with a step size η. Calculate the the formula for $w^{(t+1)}$ as a function of $w^{(t)}$ and η (where in addition you can use $\mathbf{X}$, $\mathbf{y}$, λ, v). Explain the steps of your derivation.

(c) What would the update step for $w^{(t+1)}$ be in Stochastic Gradient Descent for the same objective?

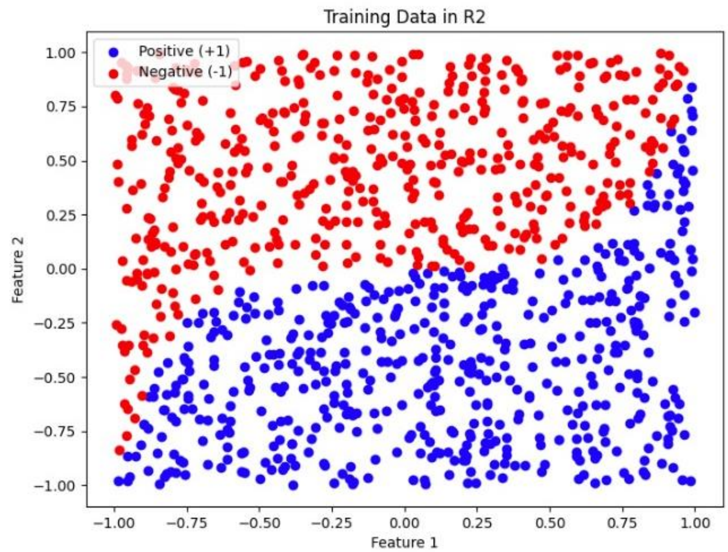
Question 6. (Do this question after we learn about PCA in class)
In an experiment, several measurements were taken at times $t = 1, 2, \dots, m$. At each time t, the measurements taken were $x_t(1), x_t(2), x_t(3), x_t(4)$. This created a data set $S = \{x_1, \dots, x_m\}$, where x_t is a vector in $\mathbb{R}^4$ which includes all the measurements from time t. PCA was performed on the data set S to reduce its dimensionality from 4 to 2.

(a) In one experiment, it turned out that in all times t, $x_t(3) = 3x_t(1) + x_t(2)$, and $x_t(4) = 2x_t(2) - 4x_t(3)$. What will be the error of the PCA in this case? Prove your claim.

(b) In another experiment, it turned out that at all times t, $x_t(3) = (x_t(1))^2 + (x_t(2))^3$, and $x_t(4) = (x_t(3) - x_t(1))^2$. Show an example of experiment results that satisfy these equations such that the error of the PCA is larger than the error you showed for the experiment in (a). You may choose m as you like.

<u>שאלה 2:</u>

(**a**)



באופן כללי, kernel soft SVM יכול למפות נתונים למימד גבוה יותר שבו הם עשויים להיות ניתנים להפרדה באמצעות מפריד לינארי.
לכן ניתן לראות ע"פ הגרף לעיל שהנקודות לא ניתנות להפרדה באמצעות מפריד לינארי ולכן סביר להניח שkernel-based soft SVM יגרום לביצועים טובים יותר.

(**b**) להלן התוצאות לאחר הרצת Polynomial soft SVM על הטריינינג סט:

<u>9 average validation error values for each of the pairs (λ, k):</u>

Validation error: 0.063 for {'lambda': 1, 'k_pol_degree': 2}
Validation error: 0.062 for {'lambda': 1, 'k_pol_degree': 5}
Validation error: 0.0579 for {'lambda': 1, 'k_pol_degree': 8}
Validation error: 0.064 for {'lambda': 10, 'k_pol_degree': 2}
Validation error: 0.064 for {'lambda': 10, 'k_pol_degree': 5}
Validation error: 0.061 for {'lambda': 10, 'k_pol_degree': 8}
Validation error: 0.064 for {'lambda': 100, 'k_pol_degree': 2}
Validation error: 0.063 for {'lambda': 100, 'k_pol_degree': 5}
Validation error: 0.063 for {'lambda': 100, 'k_pol_degree': 8}

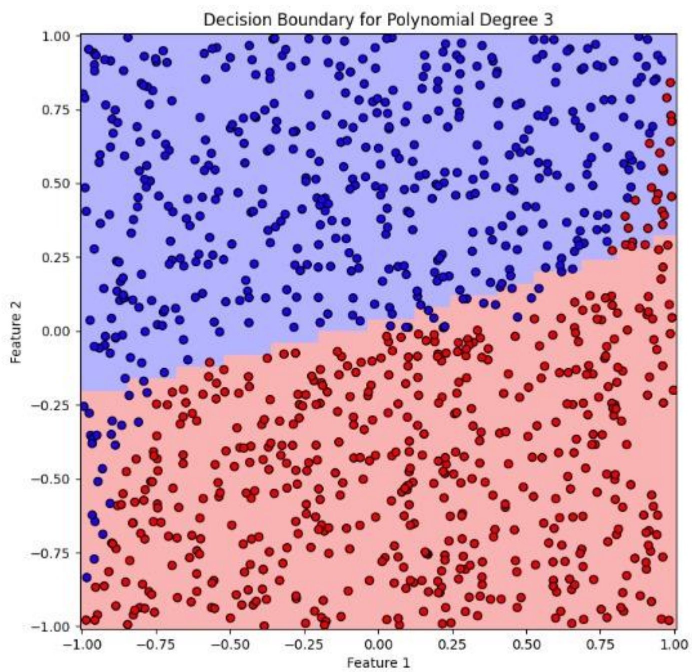
Best alpha: {'lambda': 1, 'k_pol_degree': 8} → Test error of the resulting classifier: 0.01

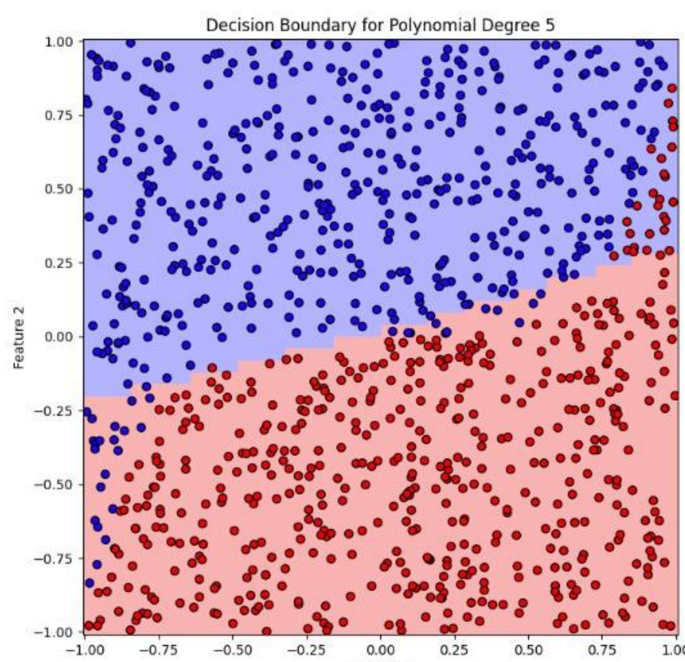
(**c**)

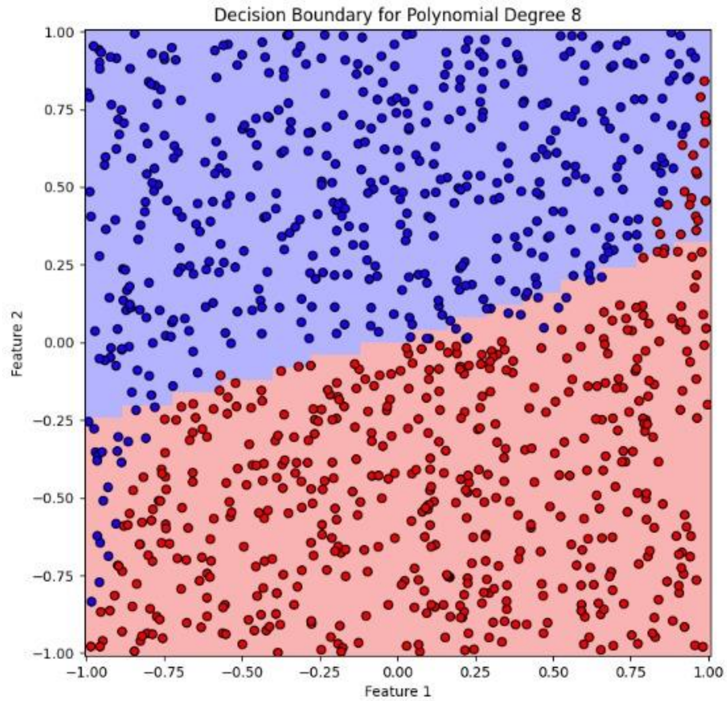
- SVM פולינומי יקבל שגיאת ולידציה נמוכה יותר מ SVM לינארי כאשר הדאטה לא ניתן להפרדה לינארית, כלומר לא קיים קו ישר שיכול להפריד בצורה טובה את הדאטה.
במקרים אלו, SVM פולינומי עשוי להפריד את הדאטה בצורה טובה יותר ע"י שימוש במימד גבוה יותר.

- SVM פולינומי יקבל שגיאת ולידציה גבוהה יותר מSVM לינארי בעקבות over-fitting.
over-fitting נוצר כאשר הSVM הפולינומי לומד לא רק את הדפוס הנדרש אלא גם רעשים הקיימים בדוגמאות האימון, ולכן מודל לינארי עשוי במקרים מסוימים לבצע הכללה טובה יותר עבור דוגמאות כאלו.
כמו כן, מחלקת ההיפותזות של מפרידים פולינומויאליים עשירה הרבה יותר מאשר של מפרידים לינאריים ועל כן ייתכן שהמסווג שיימצא בשלב האימון יהיה תואם למדגם האימון אך יניב שגיאה גבוהה על דוגמאות שלא ראה בעבר.

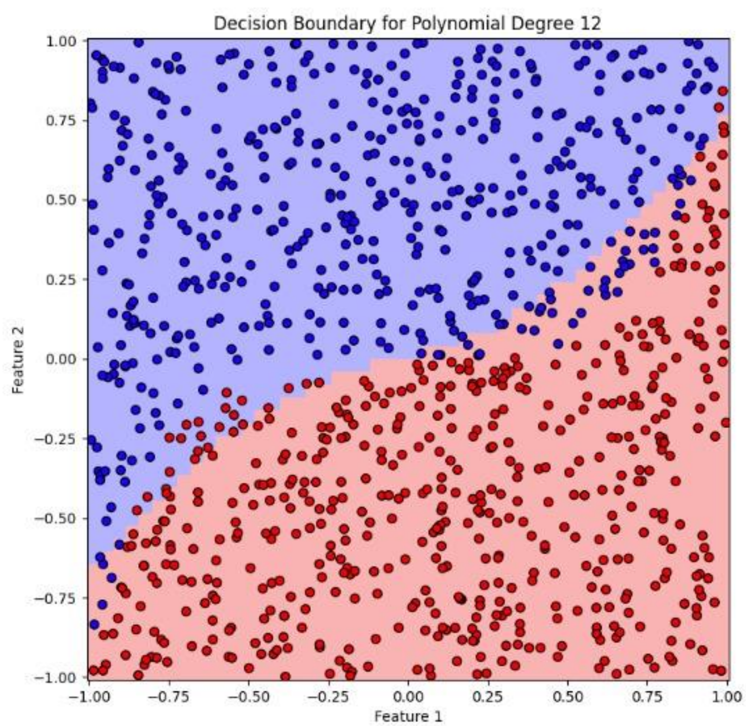
(**d**) בגרפים הבאים ניתן לראות כי שככל שאנחנו מגדילים את דרגת הפולינום של פונקציית הקרנל, כך המודל חוזה בצורה טובה יותר את התוויות עבור הדאטה סט הנל.







כדי להבליט את חשיבות הדרגה של k נשים לב שעבור k=12 קיבלנו התאמה שממש קרובה לתוצאות האמיתיות:



שאלה 3:

(a) נתון כי: $k(x,x'):=(2x(7)+x(3))\cdot x'(2)$, צ"ל: לכל $\psi(x)$ הפונ' $K(x,x')$ לא יכולה להיות פונ' קרנל.

לפי ההגדרה של פונקציית הקרנל $K(x,x'):=\langle\psi(x),\psi(x')\rangle$.

לפי ההגדרה של מכפלה פנימית, תכונת הסימטריות חייבת להתקיים, כלומר:

$$\forall x,x'\in\mathbb{R} \qquad K(x,x'):=\langle\psi(x),\psi(x')\rangle=\langle\psi(x'),\psi(x)\rangle=:K(x',x)$$

נניח בשלילה כי קיימת feature-map $\psi(x)$ עבורה $K(x,x')$ הינה פונ' קרנל חוקית ולכן מקיימת סימטריות.

נבחר $x=(0,1,0,\dots,0)\in\mathbb{R}^d$, $x'=(1,0,1,\dots,1)\in\mathbb{R}^d$

אז: $K(x,x')=(2x(7)+x(3))\cdot x'(2)=(2\cdot0+0)\cdot0=0$

$K(x',x)=(2x'(7)+x'(3))\cdot x(2)=(2\cdot1+1)\cdot1=3$ בסתירה לתכונת הסימטריות של מכפלה פנימית.

(b) נתון כי: $k(x,x'):=5-(x(1)-x(2))(x'(1)-x'(2))$, צ"ל: לכל $\psi(x)$ הפונ' $K(x,x')$ לא יכולה להיות פונ' קרנל.

לפי ההגדרה של מכפלה פנימית, תכונת החיוביות לחלוטין מתקיימת, כלומר:

$$\forall x\in\mathbb{R} \qquad K(x,x):=\langle\psi(x),\psi(x)\rangle\geq0$$ (שוויון קיים אמ"מ $x=0$)

נניח בשלילה כי קיימת feature-map $\psi(x)$ עבורה $K(x,x')$ הינה פונ' קרנל חוקית ולכן מקיימת חיוביות לחלוטין.

נבחר $x=(3,0,\dots,0)\in\mathbb{R}^d$

אז: $K(x,x)=5-(x(1)-x(2))(x(1)-x(2))=5-(3-0)(3-0)=5-9\not\geq0$

בסתירה לתכונת החיובית לחלוטין של מכפלה פנימית.

(c) נתון כי:
$$f(x,x')=(x(1)x'(1))^6+e^{x(3)+x(5)+x'(3)+x'(5)}+\frac{1}{x(1)x'(1)}+(x(4)+x(6))(x'(4)+x'(6))$$

צ"ל: f היא פונ' קרנל ע"י קיום ψ feature-map כך ש- $f(x,x')=\langle\psi(x),\psi(x')\rangle$.

נגדיר את ψ כך ש-
$$\psi(x):=\begin{bmatrix}x(1)^6\\ e^{x(3)}\\ e^{x(5)}\\ e^{x(3)}\\ x(1)^{-1}\\ x(4)+x(6)\end{bmatrix}$$
, לכן המכפלה הפנימית $f(x,x')=\langle\psi(x),\psi(x')\rangle$, כנדרש.

$\Big($ $\psi(x)(1)=x(1)^6$, $\psi(x)(2)=\psi(x)(4)=e^{x(3)}$, $\psi(x)(3)=e^{x(5)}$

$\psi(x)(5)=x(1)^{-1}$ $\quad$ $\psi(x)(6)=x(4)+x(6)$ $\Big)$

שאלה 4: (Convexity)

עבור פונקציות קמורות $f_i:\mathbb{R}^d\to\mathbb{R}$ $i\in\{1,\dots,k\}$ כך ש- $k\in\{1,2,\dots\}$ נגדיר:

$g(u)=\sum_{i=1}^{k}\alpha_i f_i(u)$, $u\in\mathbb{R}^d$, $\alpha_1,\dots,\alpha_k\in\mathbb{R}$.

(a) כדי להוכיח כי g לא בהכרח פונ' קמורה לכל $\alpha_1,\dots,\alpha_k\in\mathbb{R}$. נראה דוגמה נגדית:

עבור $k=2$, $f_1(u)=u^2$, $f_2(u)=(u-1)^2$.

נשים לב כי f_1,f_2 פונ' קמורות מאחר וקל לראות כי הנגזרת השנייה שלהן חיובית לכל $u\in\mathbb{R}$.

נבחר $\alpha_1=1$, $\alpha_2=-2$ אז עפ"י הגדרת $g(u)$:

$$g(u)=\sum_{i=1}^{k}\alpha_i f_i(u)=1\cdot u^2-2\cdot(u-1)^2=u^2-2u^2+4u-2=-u^2+4u-2$$

פונ' קמורה f מקיימת כי לכל $\lambda\in[0,1]$ ולכל $u,v\in\mathbb{R}^d$, $f(\lambda u+(1-\lambda)v)\le\lambda f(u)+(1-\lambda)f(v)$.

אז נראה כי $g(u)$ לא מקיימת א-שיוויון זה עבור $\lambda=0.5$, $u=0$, $v=2$:

$$g(\lambda u+(1-\lambda)v)=g(0.5\cdot0+0.5\cdot(2))=g(1)=-1+4-2=1$$

$$\lambda\cdot g(u)+(1-\lambda)g(v)=0.5\cdot g(0)+0.5\cdot g(2)=0.5(-2+(-4+8-2))=0.5\cdot0=0$$

מאחר וקיבלנו כי $g(\lambda u+(1-\lambda)v)>\lambda\cdot g(u)+(1-\lambda)g(v)$ נסיק כי g אינה פונקציה קמורה.

(b) נוכיח כי g פונ' קמורה עבור $\alpha_1,\dots,\alpha_k\ge0$.

יהיו $u,v\in\mathbb{R}^d$ ותהא $\lambda\in[0,1]$.

מאחר ונתון כי f_i פונ' קמורה, היא מקיימת את א-השיוויון הבא:

$$f_i(\lambda u+(1-\lambda)v)\le\lambda f_i(u)+(1-\lambda)f_i(v)$$

נסכום את כל ה f_i ונכפול את הא-שיוויון ב- $\alpha_i\ge0$ ונקבל:

$$\sum_{i=1}^{k}\alpha_i f_i(\lambda u+(1-\lambda)v)\le\sum_{i=1}^{k}\alpha_i[\lambda f_i(u)+(1-\lambda)f_i(v)]=\sum_{i=1}^{k}\alpha_i\lambda f_i(u)+\sum_{i=1}^{k}\alpha_i(1-\lambda)\cdot f_i(v)$$

עפ"י הגדרה נקבל:

$$g(\lambda u+(1-\lambda)v)\le\lambda\cdot g(u)+(1-\lambda)g(v)$$

לכן הפונ' g הינה פונ' קמורה לכל $\alpha_i\ge0$. ■

שאלה 5: (Gradient Descent)

$$\underset{w\in\mathbb{R}^d}{\text{mimize}} \quad \lambda\|w-v\|_2^2+\sum_{i=1}^{m}(\langle w,x_i\rangle-y_i)^2$$

עבור בעיית האופטימיזציה הנתונה לעיל נגדיר את פונ המטרה שלנו להיות

$$f(w,s)=\lambda\|w-v\|_2^2+\sum_{i=1}^{m}(\langle w,x_i\rangle-y_i)^2$$

(a) ננסח את ביטוי הנוסחה הסגורה עבור w שפותר את בעיית האופטימיזציה.

נשים לב שבעיית האופטימיזציה מורכבת מסכום של פונ קמורות ולכן היא בעצם קמורה אזי כל מינימום לוקאלי שנמצא הינו גם מינימום גלובלי. ולכן כדי למצוא את ביטוי הנוסחה הסגורה עבור w, נחשב את הגרדיאנט ונשווה ל-0:

$$\nabla f(w)=2\lambda(w-v)+\sum_{i=1}^{m}2\cdot(\langle w,x_i\rangle-y_i)\cdot x_i=0$$

$$\lambda w+\sum_{i=1}^{m}\langle w,x_i\rangle x_i-\lambda v-\sum_{i=1}^{m}y_ix_i=0$$

$$\lambda w+X^TXw-\lambda v-X^T\cdot y=0$$

$$(\lambda I+X^TX)w=\lambda v+X^Ty \implies \boxed{w=(\lambda I+X^TX)^{-1}(\lambda v+X^Ty)}$$

(b) נחשב את הנוסחה של $w^{(t+1)}$ עפ"י אלגו' GD: $w^{(t+1)}\leftarrow w^{(t)}-\eta\cdot\nabla f(w^{(t)})$

נגדיר את הפונ הבאות:

$$g(w)=\lambda\|w-v\|_2^2$$

$$h(w)=\sum_{i=1}^{m}(\langle w,x_i\rangle-y_i)^2=\|X^Tw-y\|^2$$

עכשיו נחשב את הגרדיאנט של כל אחת מהפונקציות:

$$g(w)=\lambda\left(\sqrt{(w_1-v_1)^2\cdots+(w_d-v_d)^2}\right)^2=\lambda\left(|(w_1-v_1)^2+\ldots+(w_d-v_d)^2|\right)\Rightarrow\nabla g(w)=2\lambda(w-v)$$

$$h(w)=(X^Tw-y)^T(X^Tw-y)=(w^TX-y^T)(X^Tw-y)=w^TXX^Tw-w^TXy-y^TX^Tw+y^Ty$$

$$\nabla h(w)=XX^Tw+(w^TXX^T)^T-Xy-(y^TX^T)^T=2XX^Tw-2Xy=2X(X^Tw-y)$$

בסה"כ קיבלנו כי

$$\nabla f(w)=\nabla g(w)+\nabla h(w)=2\lambda(w-v)+2X(X^Tw-y)$$

ולכן

$$w^{(t+1)}=w^{(t)}-\eta\cdot\nabla f(w^{(t)})=w^{(t)}-\eta\left(2\lambda(w^{(t)}-v)+2X(X^Tw^{(t)}-y)\right)$$

(c) לפי אלג' SGD: $w^{(t+1)} = w^{(t)} - \eta\left(\nabla R(w^{(t)}) + \nabla \ell(w^{(t)},(x_i,y_i))\right)$

במקרה שלנו $\ell(w,(x_i,y_i)) = (\langle w,x_i\rangle - y_i)^2$, $R(w) = \lambda\|w-v\|_2^2$

כלומר, במקרה זה אנו לא סוכמים את m הדוגמאות אלא בוחרים דוגמא אחת באקראי.

כעת כמו בסעיף הקודם נחשב את הגרדיאנטים של שתי הפונ':

$$\nabla R(w) = 2\lambda(w-v)$$

$$\nabla \ell(w,(x_i,y_i)) = 2(w^T x_i - y_i)\cdot x_i$$

בסה"כ קיבלנו כי: $\nabla f(w) = \nabla R(w) + \nabla \ell(w,(x_i,y_i)) = 2\lambda(w-v) + 2x_i(w^T x_i - y_i)$

ולכן $w^{(t+1)} = w^{(t)} - \eta\cdot\left(\nabla R(w^{(t)}) + \nabla \ell(w^{(t)},(x_i,y_i))\right) = w^{(t)} - \eta\cdot\left(2\lambda(w-v) + 2x_i(w^T x_i - y_i)\right)$

שאלה 6, סעיף ב:

נתון כי
$$X_t(3) = 3X_t(1) + X_t(2)$$
$$X_t(4) = 2X_t(2) - 4X_t(3)$$

לכן תכונות 4 ו-3 הן צירוף לינארי של תכונות 1, 2 ולכן הן אינן מכילות מידע נוסף.

כלומר ניתן לייצג הכל על ידי המידע של תכונות 1,2.

בניית PCA למימד 2 לא יגרור איבוד בגלל האמור לעיל ולכן השגיאה תהיה אפס.

נראה כי כל התכונות הן צירוף לינארי של $X(1), X(2)$:

$\alpha = 1, \beta = 0, \gamma = 0, \delta = 0: \quad X_1 = \alpha X_1 + \beta X_2 + \gamma X_3 + \delta X_4$

$\alpha = 0, \beta = 1, \gamma = 0, \delta = 0: \quad X_2 = \alpha X_1 + \beta X_2 + \gamma X_3 + \delta X_4$

$\alpha = 3, \beta = 1, \gamma = 0, \delta = 0: \quad X_3 = \alpha X_1 + \beta X_2 + \gamma X_3 + \delta X_4$

$\alpha = (-12), \beta = (-4), \gamma = 0, \delta = 0: \quad X_4 = \alpha X_1 + \beta X_2 + \gamma X_3 + \delta X_4$

שאלה 6 סעיף ט:

נראה שבמקרה ש $X(3)$, $X(4)$ הם לא ליניאריות ב- $X_1(t)$ ו-ב $X_2(t)$

השגיאה גדולה מאפס

$t = 1, 2, 3$, $m = 3$

$$X_t(3) = (X_t(1))^2 + (X_t(2))^3$$

$$X_t(4) = (X_t(3) - X_t(1))^2$$

t	$X_t(1)$	$X_t(2)$	$X_t(3)$	$X_t(4)$
1	1	2	9	64
2	2	1	5	9
3	3	3	36	1089

בדוגמה הזו תהליך ה PCA לא יצליח לייצג את השונות באופן מלא ב- $X_t(3)$! $X_t(4)$ ולכן צמצום המימד ל- $\mathbb{R}^2$ יוביל לאיבוד מידע כלומר השגיאה של ה PCA תהיה גדולה מאפס (סעיף א).