1. Go over the code in the SVMtrial function handed to you, debug it using the various input sets and observe the values of the variables calculated inside the function. Answer the following questions :
2. What is the role of the variable Lambda? describe the effect of changing the value of Lambda. Hint – read the documentation of the **quadprog** function:

תפקידו של המשתנה Lambda הוא שווה ערך למשתנה C במודל SVM. כפי שניתן לראות המגבלה בSVM היא ואילו בקוד המגבלה היא . על כן שינוי משתנה ה Lambda הוא כמו עונש במודל SVM. (הסבר נוסף ראה בשאלה 3).

כאשר Lambda גדול ( גדול) לא מאפשר טעויות .

ואילו כאשר Lambda קטן ( קטן) מאפשר טעויות לסיווג טוב יותר.

1. Describe the input and output (in particular, describe the input H) of the quadprog function and explain the function:

*ראה* [*שאלה 3*](#שאלה3) *להסבר מפורט.*

|  |  |  |  |
| --- | --- | --- | --- |
| הערות | SVM | quadprog | Input |
| המטריצה סימטרית | K- קרנל. | מטריצת mxm כך ש : | H |
|  |  | מטריצת mx1 של (-1,…..,-1) | F |
|  | ריק | ריק | A |
|  | ריק | ריק | B |
|  | סיווגים | מטריצת 1xm של yi | Aeq |
|  |  | 0 – לצורך המגבלה Aeq\*x = Beq | Beq |
|  | 0 | וקטורי אפסים לצורך גבול תחתון. | Pb |
|  | C | Lambda- גבול עליון. | Ub |

פונקציית **quadprog** פותרת את המשוואה :

*כאשר זה ה ו H זהו הקרנל : כולל המכפלה בסיווגים: .*

1. How does this function contribute to the SVM algorithm?

*(ראה* [*שאלה 3*](#שאלה3) *וסעיף b).*

*פונקציית* ***quadprog*** *פותרת בעיות אופטימיזציה ריבועיות עם מגבלות .*

*הבעיה הזו זהה לDual SVM עם RBF Kernel . כלומר פונקציה זו פותרת את בעיית* ***dual svm*** *( מציאת ) .*

1. Describe how the bias is estimated:

*על מנת לחשב את הbias צריך למצוא את כל הנקודות שנמצאות ליד הגבולות (near boundary ) אשר בעזרתן מחשבים את ה margin . לאחר חישוב הפונקציה quadprog אנחנו מקבלים a עם ערכי עבור כל דוגמה. כל המסווגת 0 או קרובה ל 0 (לאחר בדיקת tol) לא משפיעות על ה margin. כל המסווגת כ Lambda או קרובה לLambda מסווגת כטעות ועל כן אינה משפיעה .*

*לכן לאחר סינון (nb) מחשבים לפי הנוסחה :*

חישוב זה מגיע מהאילוץ : עבור כל הxi שנמצאים על השוליים.

כלומר :

1. Write pseudocode describing the given SVM algorithm:

SVM\_RBF(X,Y,Kw,Lambda):

FOR each x in X

Normalization x by mean(x) and std(X)

Create variables F,Aeq,Beq,Lb

Ub <- Create Lambda – if scalar -> create vector of Lambda

` Create H -> H(i,j)= y(i)\*y(j)\*exp(-(d\*d')/kw)

H(j,i)=H(i,j)

α <- min (0.5\* sum(sum(αi\* αj\*yi\*yj)\*H))

tolerance <- 1\*exp \*8

sv <- all (xi,yi,αi) s.t αi>= tolerance

nb <- all (xi,yi) s.t ε=<αi<=Γi- ε

bias <- mean(b+= bias theorem in loop)

return sv,b

**Usage of SVM algorithm:**Prediction(X):

Sv,b <- SVM\_RBF(X,Y,Kw,Lambda)

If sum(αi\*yi\* 1/exp(((x-xi)^2)/kw)))+b >= 0

return 1

return -1

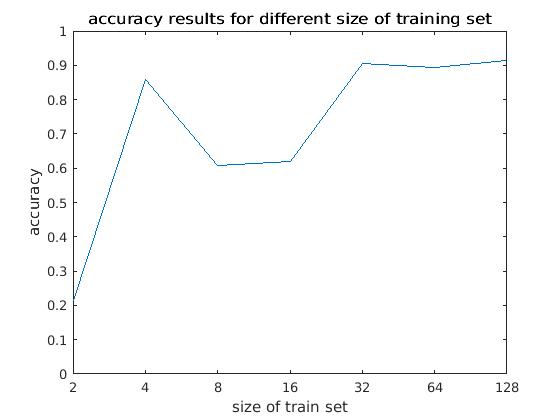
1. Use the fourth option for input data, and follow the instructions in the following questions:
2. Write a function that separates the dataset generated into training and validation sets using the K-fold cross validation method.

*הפונקציות הבאות ממומשות בקוד:*

* **KFold** *- הפונקציה בונה קבוצות אימון וולידציה.*
* **SVMtrail***- הקוד שקיבלנו, יש לשים לב להערה למעלה (שינינו את ה- λ ל λ= λ)*
* **EvaluateModel***- פונקציה שחלקים ממנה נלקחו מ SVMtrail המשתמשת בו וגם קוראת לפונקציה Kfold ובודקת את קבוצת הוולידציה כקבוצת מבחן ומחזירה את הדיוק בהתאם.*
* **Run***- פונקציה שרצה על λ משתנה ועל קבוצות אימון משתנות.*

1. Write a function to evaluate the model created by SVMtrial. Use the model, and the validation sets created by the function from 3a. use K=3, what is the average accuracy of the model?

*לאחר הרצת הפונקציה EvaluateModel קיבלנו :* ***0.9133*** *. (ראה הרצה בקוד).*

1. Test different sizes of training sets (2,4,8,16…) while keeping the size of validation sets constant (50). Plot the average accuracy of the model as a function of the size of the training set. Explain your results:

*מלבד החריגות שמופיעות עבור קבוצת אימון בגודל 4, אפשר להבחין בעלייה מתמדת של אחוזי הדיוק.*

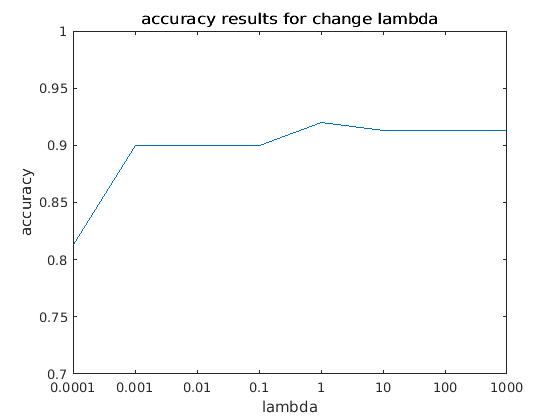
*קבוצת הוולידציה (test) היא בגודל 50, ולכן שימוש בקבוצות אימון קטנות יותר אינו מוביל לדיוק בהפרדה בין הקבוצות.*

*נסביר את התוצאה בקבוצה בגודל 4 כך שלמרות שהשתמשנו בכלים על מנת ליצור רנדומליות, התוצאות בקוד יוצאות זהות עבור כל אימון. ייתכן כי הקבוצה בגודל 4 יצרה הפרדה טובה בין הקבוצות וולידציה. לאור הירידה לאחר מכן אנו מניחים כי מדובר בחריגות וקבוצת אימון בגודל 4 אינה מספיקה במקרה הזה.*

***חשוב לציין שהשתמשנו בקבוצת מבחן של 50 ולכן כאשר רשום 128 מדובר בקבוצה של 100!***

*ככל שגודל קבוצת האימון עולה כך הדיוק על קבוצת המבחן משתפר (ניתן לראות שיפור משמעותי בגודל 32).*

1. Set the training set to 10 samples, change the value of Lambda from very small values to very large values. Plot the average accuracy of the model as a function of Lambda, explain your results.



***על מנת לקבל את הגרף המצורף השתמשנו ב-קבוצות אימון של 100 דגימות ולא 10 כמו שהתבקשנו בשאלה.*** *התחלנו לבדוק מ λ נמוכה עד ל λ גבוהה.*

*אם נשווה את התוצאות בסעיף זה לסעיף הקודם נבחין כי שינוי ה λ משפיע פחות משינוי גודל קבוצות האימון.*

*ננסה להסביר את התוצאות:*

*λ מאוד קטנה כמעט ולא מענישה את הנקודות המסווגות לא נכון בהתאם למפריד (margin) שנוצר ולכן המפריד רחב מאוד. לעומת זאת, λ מאוד גדולה מענישה את הטעיות הרבה יותר ועל כן גורמת למפריד צר. לכן, ניתן לראות אחוזי דיוק נמוכים יותר עבור למדות קטנות. עם זאת, כבר ב-0.001 (למדה קטנה) ניתן להבחין באחוז דיוק גבוה.*

*השערתנו היא ש λ בגדלים 0.1-10 משרתות בצורה הטובה ביותר את מטרתן (הגבלת הטעויות).*

*λ גדולה 🡨 מפריד צר 🡨 מאפשר טעויות רבות יותר בסט המבחן.*

*λ קטנה 🡨 מפריד רחב 🡨 עלול לגרום לטעויות רבות להיות בטווח של המפריד.*

*ההחלטה על ה λ תלויה מאוד בData ובנסיבות.*

1. Write the equation solved by the quadprog function, show that it is equivalent to the SVM problem we showed in class:

*פונקציית quadprog מוגדרת בצורה הבאה :*

*הפונקציה מקבלת את הפרמטרים כפי שהסברנו בשאלה* [***b1***](#b) *.*

*הפונקציה פותרת את בעיית האופטימיזציה הריבועית עם מגבלות ובפרט את המשוואה הבאה :*

*נפרט את המשוואה על מרכיביה :*

*לאחר פישוט המשוואות נקבל את המשוואה הבאה:*

*עם המגבלות :*

*נשים לב כי מקסום של SVM הוא :*

quadprog

*נסביר כיצד המגבלות זהות לSVM:*

*- מגבלה זו מתקבלת עבור . קיימת בSVM וקיימת ב quadprog.*

*– מגבלת כופלי לגרנג'.*

*– בSoft margin כאשר אנחנו רוצים לאפשר טעות שישלמו עליה משתמשים בנוסחה:*

*נכניס לכופלי לגרנג' ונקבל :*

*נגזור לפי w ונקבל* :

*נגזור לפי b ונקבל :*

*ונגזור לפי ונקבל :* =0

*בהצבה בloss נקבל את המשוואה עם אותה מגבלה . לכן ,b ו לא משפיעים ומה שנותר הוא למקסם עבור α – היות ו b חיובי (מגבלת לגרנג') => כמו ב quadprog .*

*לסיכום, הראנו כי בעיית האופטימיזציה שפותרת הפונקציה quadprog שקולה לבעיית dual svm שנלמדה בכיתה.*