

Computational Methods in Neuroscience

Assignment 5

Rotem Alt 305610750 and Omer Havakook 203345087

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 :

a. 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 λ הוא שווה ערך למשתנה C במודל SVM. כפי שניתן לראות המגבלה במודל SVM היא $\alpha \leq C$ ואילו בקוד המגבלה היא $x \leq ub$. על כן שינוי משתנה λ הוא כמו עונש C במודל SVM. (הסבר נוסף ראה בשאלה 3).

כאשר λ גדול (C גדול) לא מאפשר טעויות .
ואילו כאשר λ קטן (C קטן) מאפשר טעויות לסיווג טוב יותר.

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

ראה [שאלה 3](#) להסבר מפורט.

Input	quadprog	SVM	הערות
H	מטריצת $m \times m$ כך ש: $H_{ij} = Y_i * Y_j * \frac{1}{e^{-k \cdot x_i - x_j ^2}}$	$H_{ij} = Y_i * Y_j * K(x_i, x_j)$ K-קרנל.	המטריצה סימטרית
F	מטריצת $m \times 1$ של $(-1, \dots, -1)$	$-\sum \alpha_i$	
A	ריק	ריק	
B	ריק	ריק	
Aeq	מטריצת $1 \times m$ של y_i	סיווגים	
Beq	$Aeq * x = Beq$ – לצורך המגבלה 0	$\sum \alpha_i * y_i = 0$	
Pb	וקטורי אפסים לצורך גבול תחתון.	0	
Ub	λ - גבול עליון.	C	

פונקציית **quadprog** פותרת את המשוואה :

$$\min(x) = 0.5 * x' * H * x + F' * x$$

$$= \frac{1}{2} * \sum \sum \alpha_i * \alpha_j * y_i * y_j * K(x_i * x_j) - \sum \alpha_i$$

Computational Methods in Neuroscience

Assignment 5

Rotem Alt 305610750 and Omer Havakook 203345087

$$\text{s.t. : } \sum \alpha_i * y_i = 0, 0 < \alpha_i < \lambda_i$$

כאשר x_i זה α_i ו H זהו הקרנל: $K(x_i, x_j)$ כולל המכפלה בסיווגים: $Y_i * Y_j$.

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

(ראה [שאלה 3](#) וסעיף b).

פונקציית **quadprog** פותרת בעיות אופטימיזציה ריבועיות עם מגבלות. הבעיה הזו זהה ל**Dual SVM** עם **RBF Kernel**. כלומר פונקציה זו פותרת את בעיית **dual svm** מציאת α .

d. Describe how the bias is estimated:

על מנת לחשב את ה **bias** 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 α עבור כל דוגמה. כל המסווגת 0 או קרובה ל 0 (לאחר בדיקת tol) לא משפיעות על ה **margin**. כל α המסווגת כ **Lambda** או קרובה ל **Lambda** מסווגת בטעות ועל כן אינה משפיעה. לכן לאחר סינון (**nb**) מחשבים לפי הנוסחה:

$$b_i = \frac{1}{y_i} - \sum_{x_j \in sv} \alpha_j * y_j * \frac{1}{e^{\frac{||x_i - x_j||^2}{kw}}}$$

חישוב זה מגיע מהאילוץ: $y_i(w^T x_i + b) - 1 = 0$ עבור כל ה x_i שנמצאים על השוליים. כלומר:

$$b = \frac{1}{y_i} - \sum_{x_j \in sv} \alpha_j y_j K(x_j, x_i)$$

Computational Methods in Neuroscience

Assignment 5

Rotem Alt 305610750 and Omer Havakook 203345087

e. 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 -> $H(i,j) = y(i) * y(j) * \exp(-(d * d') / kw)$

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

$\alpha <- \min(0.5 * \sum(\sum(\alpha_i * \alpha_j * y_i * y_j) * H))$

tolerance <- $1 * \exp * 8$

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

nb <- all (xi,yi) s.t $\epsilon \leq \alpha_i \leq \Gamma_i - \epsilon$

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 $\sum(\alpha_i * y_i * 1 / \exp(((x-x_i)^2) / kw)) + b \geq 0$

return 1

return -1

Computational Methods in Neuroscience

Assignment 5

Rotem Alt 305610750 and Omer Havakook 203345087

2. Use the fourth option for input data, and follow the instructions in the following questions:
 - a. 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.
- **SVMtrial** - הקוד שקיבלנו, יש לשים לב להערה למעלה (שינינו את ה- λ ל $\lambda = \lambda$)
- **EvaluateModel** - פונקציה שחלקים ממנה נלקחו מ *SVMtrial* המשתמשת בו וגם קוראת לפונקציה *Kfold* ובודקת את קבוצת הוולידציה בקבוצת מבחן ומחזירה את הדיוק בהתאם.
- **Run** - פונקציה שרצה על λ משתנה ועל קבוצות אימון משתנות.

- b. 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?

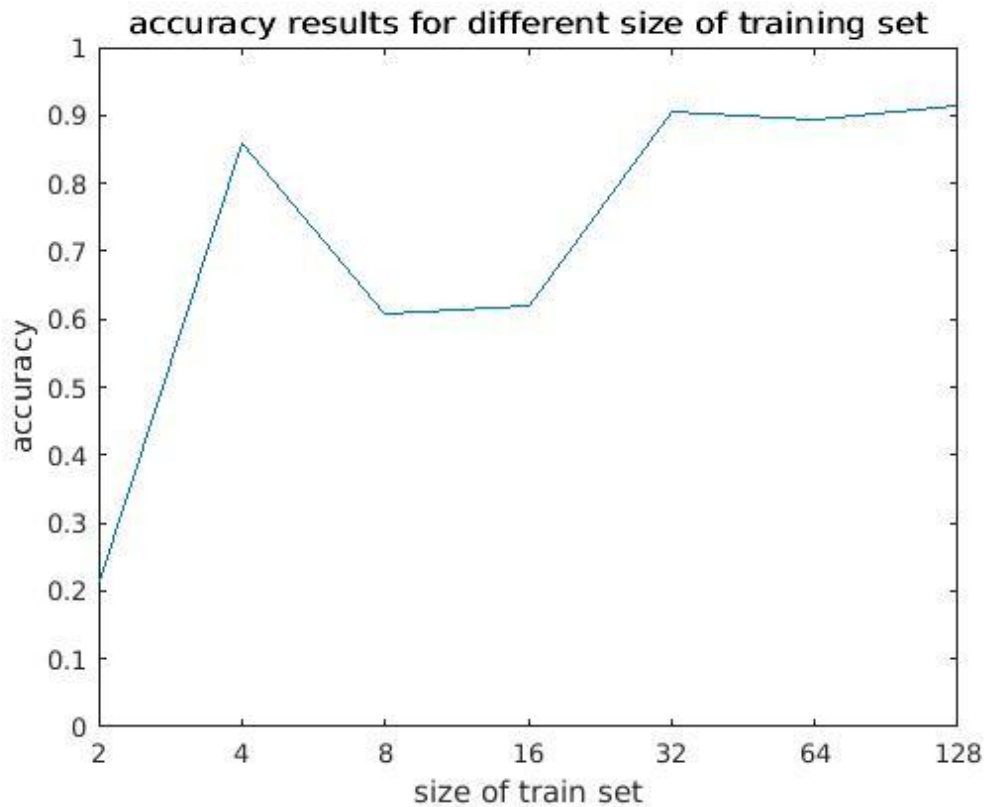
לאחר הרצת הפונקציה *EvaluateModel* קיבלנו : **0.9133** . (ראה הרצה בקוד).

Computational Methods in Neuroscience

Assignment 5

Rotem Alt 305610750 and Omer Havakook 203345087

- c. 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.

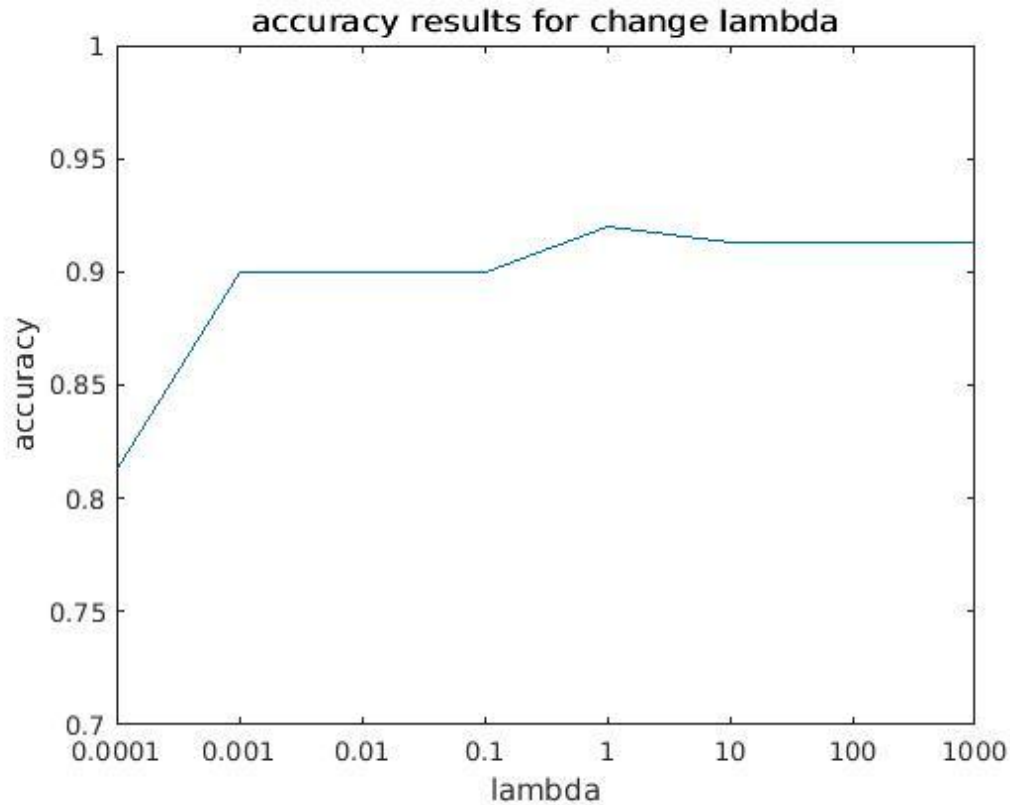
חשוב לציין שהשתמשנו בקבוצת מבחן של 50 ולכן כאשר רשום 128 מדובר בקבוצה של 100!
ככל שגודל קבוצת האימון עולה כך הדיוק על קבוצת המבחן משתפר (ניתן לראות שיפור משמעותי בגודל 32).

Computational Methods in Neuroscience

Assignment 5

Rotem Alt 305610750 and Omer Havakook 203345087

- d. 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 הטעויות 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 \leftarrow מפריד צר \leftarrow מאפשר טעויות רבות יותר בסט המבחן.
 λ קטנה \leftarrow מפריד רחב \leftarrow עלול לגרום לטעויות רבות להיות בטווח של המפריד.

ההחלטה על ה λ תלויה מאוד ב *Data* ובנסיבות.

Computational Methods in Neuroscience

Assignment 5

Rotem Alt 305610750 and Omer Havakook 203345087

- 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 [1b](#).

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

$$\min(X) 0.5 * X' * H * X + F' * X$$

$$\begin{aligned} s. t \\ A * \alpha &\leq B \\ Aeq * \alpha &= Beq \\ Pb &\leq \alpha \leq Ub \end{aligned}$$

נפרט את המשוואה על מרכיביה :

$$\min(X) 0.5 * (\alpha_1, \dots, \alpha_m)$$

$$* \begin{pmatrix} y_1 * y_1 * k(x_1, x_1) & \dots & y_m * y_m * k(x_m, x_m) \\ \vdots & & \vdots \end{pmatrix} * \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{pmatrix}$$

$$+ (-1, \dots, -1) * \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{pmatrix}$$

$$\begin{aligned} s. t \\ (y_1, \dots, y_m) * \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{pmatrix} &= 0 \end{aligned}$$

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \leq \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{pmatrix} \leq \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_m \end{pmatrix}$$

Computational Methods in Neuroscience

Assignment 5

Rotem Alt 305610750 and Omer Havakook 203345087

לאחר פישוט המשוואות נקבל את המשוואה הבאה:

$$\frac{1}{2} * \sum_j \sum_i \alpha_i * \alpha_j * y_i * y_j * K(x_i * x_j) - \sum_i \alpha_i$$

עם המגבלות :

$$\begin{aligned} s. t \\ \sum_i \alpha_i * y_i &= 0 \\ 0 \leq \alpha_i &\leq \lambda_i \end{aligned}$$

נשים לב כי מקסום של SVM הוא :

$$\max \sum_i \alpha_i - \frac{1}{2} * \sum_j \sum_i \alpha_i * \alpha_j * y_i * y_j * K(x_i * x_j) =$$

$$\min - \left(\sum_i \alpha_i - \frac{1}{2} * \sum_j \sum_i \alpha_i * \alpha_j * y_i * y_j * K(x_i * x_j) \right) =$$

$$\text{quadprog} = \min \frac{1}{2} * \sum_j \sum_i \alpha_i * \alpha_j * y_i * y_j * K(x_i * x_j) - \sum_i \alpha_i$$

נסביר כיצד המגבלות זהות לSVM:

$$\sum_i \alpha_i * y_i = 0 \text{ - מגבלה זו מתקבלת עבור } \frac{d\text{loss}}{db} = 0 \text{ . קיימת בSVM וקיימת ב } \text{quadprog}.$$

$$0 \leq \alpha_i \text{ - מגבלת כופלי לגרנג' .}$$

$\alpha_i \leq \lambda_i$ ב *Soft margin* כאשר אנחנו רוצים לאפשר טעות שישלמו עליה משתמשים בנוסחה:

$$\begin{aligned} \frac{1}{2} * |W|^2 + c * \sum_j \varepsilon_i \\ s. t \end{aligned}$$

$$\begin{aligned} y_i * (w^t * x + b) - 1 + \varepsilon_i &\geq 0 \\ 0 \leq \varepsilon_i \end{aligned}$$

Computational Methods in Neuroscience

Assignment 5

Rotem Alt 305610750 and Omer Havakook 203345087

נכניס לכופלי לגרנג' ונקבל :

$$\frac{1}{2} * |W|^2 + * \sum_j \varepsilon_i * c - \sum_i \alpha_i * (y_i * (w^t * x + b) - 1 + \varepsilon_i) - \sum_j b_i \varepsilon_i$$

נגזור לפי w ונקבל: $W = \sum_i \alpha_i * y_i * x_i$

נגזור לפי b ונקבל: $0 = \sum_i \alpha_i * y_i$

ונגזור לפי ε ונקבל: $\bar{\alpha} - \bar{b} - \bar{C} = 0$

בהצבה ב $loss$ נקבל את המשוואה עם אותה מגבלה. לכן, b ו ε לא משפיעים ומה שנותר הוא למקסם עבור α – היות ו b חיובי (מגבלת לגרנג') $\alpha_i \leq C \leq \alpha_i \leq \lambda_i$ כמו ב $quadprog$.

לסיכום, הראנו כי בעיית האופטימיזציה שפותרת הפונקציה $quadprog$ שקולה לבעיית $dual$ svm שנלמדה בביתה.