סיכום

הסתברות:

התפלגות משותפת היא ההתלפגות המשותפת לקבלת כל סיווג של (חיובי, שלילי וכו...) והיא נתונה ע"י: כאשר מסווג או שלילי או חיובי ו- היא ההסתברות לאותו סיווג.

כעת נוכל לסווג תוצאה חיובית בהינתן ע"י חוק בייס: .

**מודל PAC:**

1. הנחה כי ההתפלגות קבועה אך אינה ידועה.
2. הדוגמאות נשלפות i.i.d.
3. בנייה של חוק החלטה ולא הסתברות.

בנייה:

1. ישנה התפלגות משותפת לא ידועה .
2. פונק' המטרה כאשר: .
3. השערות כאשר .
4. השגיאה .
5. ישנו "אורקל" אשר מייצר לנו דוגמאות .

הגדרה:

1. ו- עם מחלקות קונספטים מעל .
2. היא PAC learnable על ע"י אם:
   1. קיים אלגוריתם A כך ש:
      1. עבור התפלגות מעל ו- ועבור כל .
      2. המוצא הוא השערה עם הסתברות ושגיאה .

מחלקת השערות סופית (Realizable case) :

נאמר כי הוא משערך רע אם .

אלגוריתם:

* נדגום מדגם S עם m דוגמאות.
* נמצא כך שהיא קונסיסטנטית – מסווגת את כל הדוגמאות נכון.

האלגוריתם נכשל אם היא .

נתקן השערה שהיא . ההסתברות ש- קונסיסטנטית:  
 .

מחלקת השערות (Non Realizable case):

מה קורה כאשר , יש להגדיר את המטרה מחדש: תהיה בעלת השגיאה הכי קטנה:  
 , המטרה: .

ERM ALGORITHM:

1. עבור כל נסמן את כשגיאה על מדגם S.
2. נחשב את ההסתברות ש: עבור כל הדגימות, סכימה על m דגימות ושימוש בחסם צ'רנוב נקבל כי החסם: עבור כל הדגימות בהשערה אחת
3. כעת עבור כל ההשערות ב- נקבל: וגודל המדגם הדרוש:

מסקנה: .

סיכום PAC:

עבור Realizable case: .

עבור Non Realizable case: .

**VC Dimension:**

הגדרה: C מחלקת קונספטים ו- S הוא המדגם. ההקרנה של C על S: .

ניתוץ: נאמר ש C מנתצת את S אם .

VC-dim מוגדרת כקבוצה הכי גדולה אשרש עבור S מתנתצת: . אם לא קיים מקסימום כזה נאמר שה- VC-dim הוא אינסוף.

**Rademacher Complexity:**

המטרה היא למצוא חסם הדוק יותר ללא תלות בהתפלגות. נסמן: ו:  
.

הגדרה: עבור מגדם S בגודל m: ו- .  
כעת, ה- overfitting הצפוי:   
הוכחה: נוסיף מדגם S’ עבור ובכך הוא יהפוך ל ובכך יש לנו פעמיים שזה קטן מפעמיים .  
כעת, בהסתברות ועבור כל :  
 .

**Perceptron:**

אלגוריתם:

1. נאתחל את המשקולות .
2. בהינתן דגימה נסווג אותה כחיובית אם"ם .
3. כאשר יש שגיאה בזמן t נעדכן את המשקולות: כלומר:
   1. שגיאה לגבי דגימה שלילית (): .
   2. שגיאה לגבי דגימה חיובית (): .

**SVM:**

ה SVM בונה [על מישור](https://he.wikipedia.org/wiki/%D7%A2%D7%9C-%D7%9E%D7%99%D7%A9%D7%95%D7%A8) (Hyperplane) שהוא המפריד הלינארי (מפריד את המרחב לשני [חצאי מרחבים](https://he.wikipedia.org/wiki/%D7%97%D7%A6%D7%99_%D7%9E%D7%A8%D7%97%D7%91) שכל אחד מהם אמור להכיל בעיקר דוגמאות מסוג  אחד), וכן שני על מישורים מקבילים לו, אחד מכל צד, במרחק זהה, אשר מתלכדים עם דוגמת אימון אחת מכל מחלקה. מטרת ה SVM היא להגדיל את המרחק בין המפריד הלינארי ובין כל אחד מהמישורים המקבילים (מרחק המכונה שוליים), ולמעשה למצוא את המפריד בעל השוליים הרחבים ביותר. דוגמאות האימון המתלכדות עם מישורי השוליים נקראות וקטורים תומכים, ומכאן נגזר שם האלגוריתם.

המסווג SVM הוא הפתרון ל: **.**נשתמש בלגרנג'יאן על מנת למצוא את הפתרון:

איך מוצאים את ו- ? פותרים את הבעיה הדואלית: נציב את מה שמצאנו ע"פ הנגזרות החלקיות ונקבל כי:   
ולכן הבעיה הדואלית היא: .  
ע"מ למצוא את b, נוכל לחשב (KKT אז ).

כעת נוסיף אילוץ אשר "משלם" עבור הפרות של המפריד הלינארי והבעיה החדשה:  
כאשר אפשר לכתוב את כ: ולפתור את הבעיה הדואלית בהצבה הזאת. (HINGE LOSS).

Positive Semi-Define Matrices (PSD):

קרנלים קשורים באופן ישיר ל PSD. מטריצה היא PSD אם:

* היא סימטרית.
* .
* כל הערכים העצמיים הם אי-שליליים.
* קיימת מטריצה .

מטריצה סימטרית אם:

* הערכים העצמיים ממשיים.
* הערכים העצמיים הם בסיס אורתוגונאלי: .

עבור מטריצה עם וקטורי עמודה , מטריצה אלכסונית עם ערכים עצמיים באלכסון, נקבל: .

כל וקטור ניתן לרשום כ: : .

מסקנה: המטריצה *היא* PSD.

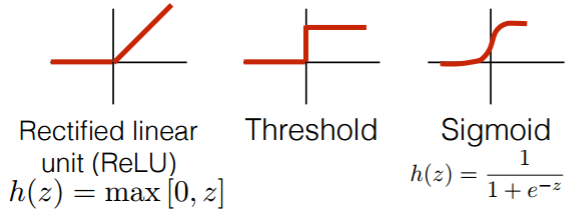
**Gradient Descent:**

*נרצה למצוא מינימום של פונקציה . מתחילים ב- ובכל איטרציה: .*

**SGD for SVM*:***

1. *דוגמים דגימה באופן אחיד מהמדגם.*
2. *אם אז: .*
3. *אחרת: כאשר: .*

**Neural Network:**

*המודל מכיל שכבות ובכל שכבה צמתים: .  
נשתמש ברשתות נויירונים ע"מ לבנות מסווג בינארי: או מסווג רב-מימדי: .*Activation function:   
*כל צומת מקודדת על מישור במרחב כאשר השכבה הראשונה היא על מישור מעל הווקטור קלט .*

VC dimension *עבור רשתות נויירונים: נניח כי ישנם E משקולות שכל אחת מקודדת ע"י D ביטים. החסם על VC הינו: וכאשר המשקולות ממשיות: .*

*Training Neural Nets: נשתמש ב SGD ונרצה להביא למינימום:* ***.***

***Back propagation:***

1. *Forward pass: עבור כל שכבה נחשב את המוצא הלינארי ופונק' ההפעלה הלא לינארית .*
2. *נחשב את ווקטור "השגיאה" עבור כל שכבה מהאחרונה לראשונה:*

***עצי החלטה:***

*בעץ החלטה, כל צומת פנימית היא היפוטזה או והעלים הם הסיווגים השונים.  
המטרה: עץ החלטה קטן המסווג את כמעט כל הדוגמאות נכון.*

*VC dimension לעצי החלטה: כאשר לעץ S עלים ומספר n מאפיינים.  
כאשר המאפיינים בינאריים:   
חסם תחתון: עבור S דוגמאות, נבנה עץ החלטה כאשר כל דוגמא מגיעה לעלה מסוים ולכן .  
חסם עליון: מספר העצים הינו מספר קאטאלן: עבור כל עץ: לכל צומת יש מאפיינים והלייבל עבור עלה הוא . מספר הפונקציות: ולכן נקבל:  
 .*במקרה הכללי כאשר המאפיינים לא-בינארים: .  
עיקרי האלגוריתם:

1. תהליך רקורסיבי:
   1. נבחר מחלק למידע בשורש.
   2. נחלק את המידע באמצעות השורש.
   3. נבנה תת עץ ימני עבור .
   4. נבנה תת עץ שמאלי עבור .

זמן הריצה:   
במקרה הגרוע: במקרה הממוצע: .  
פונק' פוטנציאל:   
אלגוריתם לבניית העץ:

1. אם כל הדוגמאות ב S בעלות אותו סיווג b, ניצור עלה עם הסיווג b ונחזיר את העץ.
2. עבור כל נחשב כאשר: .
3. יהיה .
4. נחלק את S באמצעות ל- .
5. כעת נחזור רקורסיבית ל- 1עם תתי הקבוצות .

קטימה של העץ (Pruning):  
נקבל כקלט עץ T ומדגם S ונוציא עץ T’ כאשר בקטימה הבסיסית נחליף צומת פנימי בעלה ובמקרה המתקדם נחליף אותו באחד מיילדיו.  
Reduced Error Pruning: נחלק את המדגם כאשר עם החלק הראשון נבנה את העץ ובחלק השני נשתמש ע"מ לקטום עת העץ. כעת עבור כל צומת פנימי: נבדוק את השגיאה על אותו צומת ועל מסוים ואם השגיאה של העלה קטנה מזה של הצומת, נחליף את הצומת בעלה.

**Weak Learning**:  
בהינתן אלגוריתם עם ביטחון ושגיאה גרועים: , אפשר לשפר את השגיאה.  
**AdaBoost**: נבנה מסווג לינארי באמצעותת weak learner, בכל פעם נוסיף עוד מסווג ונתאים לדגימה S.  
בכל צעד t, יש היפוטזה , נבחר התפלגות של המדגם. נמצא את המסווג ביחס להתפלגות. נוסיף את המסווג למחלקת ההיפוטזות, נחשב את ו- ונחשב את .

**אלגוריתם:**

1. יש מחלקה של weak learners H: .
2. אתחול: בהינתן מדגם נקבע את ההתפלגות לכל דגימה: .
3. עבור כל צעד :
   1. נקבל בהתאם ל- .
   2. נגדיר: , .
   3. נגדיר: .
4. תחזית: .

אינטואיציה:

1. איך אנחנו משנים את ההתפלגות:
   1. שגיאה המשקולות גדלות.
   2. לא שגיאה המשקולות קטנות.
   3. התמקדות בדוגמאות "קשות".
2. מהם הפרמטרים:
   1. מחלקת ה- weak learners H.
   2. מספר האיטרציות T.

השגיאה של האלגוריתם בהינתן השגיאות של כל איטרציה: (הוכחה ע"י 3 טענות במצגת).

**??Ensemble Methods**

**Regression and PCA:**

המטרה היא למפות קלט X ללייבל Y כאשר Y הוא בינארי או multiclass באמצעות פונק'.  
ההפסד הצפוי: כאשר באה ממחלקת ההשערות F.  
נניח כי הדוגמאות באות מהתפלגות ונרצה להביא למינימום: .

כיוון שאנחנו לא יודעים את הערכים הצפויים, נמצא קירוב: .  
בעיית ה- ERM: **.**

נתחיל בפונק' לינארית: נניח כי , כאשר . תמיד: .  
נוסיף bias: .

**Solving Linear Regression:**  
נפתור: **.**נגדיר: , .  
נרצה להביא למינימום: .  
גרדיאנט: **.**נשתמש ב: .  
נציב שהגרדיאנט שווה אפס ונקבל: . אם הפיכה נקבל: .  
נגדיר ונקבל: המודדת את הקורלציה בין i ל- j. עבור משתנים עם תוחלת אפס, נקבל כי זו מטריצת הקווואריאנס.

מה קורה אם לא הפיכה? יש תלות בין הפיצ'רים ואז נקבל אינסוף פתרונות ל .  
פתרון: Regularized Regression: נוסיף פקטור אשר יהיה נמוך עבור פתרונות שנרצה וגבוה עבור פתרונות שלא נרצה. אפשרויות:  
Ridge: נרצה להביא למינימום: . הנגזרת כמעת זהה למקרה הרגיל:  
(ההפיכה הזו תמיד מתקיימת). כעת נקבל כי:   
אם : אחרת: .

**Linear Subspaces:** תת-מרחב במימד r מוגדר ע"י: . התת-מרחב הוא כל הנקודות כך שקיימים: ו- כך ש הם ייצוג r מימדי של .  
אם: ו- נקבל כי: .  
נניח בלי הגבלת הכלליות שהבסיס אורתוגונלי: (נוכל תמיד לקבל ע"י Gram-Schmidt)  
אז: , .

הקרנה לתת-המרחב: בהינתן נקודה שלא בתת-מרחב נמצא את הנקודה שבתת-המרחב:  
**בעיית PCA:** המטרה היא למצוא תת-מרחב הקרוב ביותר למידע: זוהי בעיית האופטימיזציה ל PCA.  
לפני הרצת PCA יש להפוך את התוחלת של המידע לאפס כיוון שזה האופן האופטימאלי למצוא תת-מרחב "מחובר" וגם כי כך מטריצת הקורלציה הופכת למטריצת הקווואריאנס.  
נקבע: והקלט החדש: .

**פתרון PCA:** מטריצת הקווואריאנס: . הערכים העצמיים והווקטורים העצמיים:   
אז הפתרון ל- PCA הוא: , בעיקרון: ניקח את r הווקטורים העצמיים עם הערכים העצמיים הגדולים ביותר.

**קלסטרים: K-Means:**  
נניח כי מספר הקלסטרים ופרוטוטייפס הוא K והפרוטוטייפס: . הנקודות בקלסטר ה- i צריכות להיות קרובות ככל הניתן ל- .נגדיר את המרחק בין נקודה לפרוטוטייפ: .  
והמטרה היא למצוא המביא למינימום את: . האלגוריתם קטן בכל איטרציה.

אלגוריתם: נעדכן בכל איטרציה את: , נשייך את כל הנקודות לקלסטר הקרוב אליהן ביותר ונעריך  
מחדש את להיות התוחלת של כל הנקודות ששויכו לקלסטר i. נחזור על הפעולה עד להתכנסות.

המטרה: נכתוב את המרחק כ: כאשר הוא המשתנה indicator ה-"בוחר" את הקלסטר הטוב ביותר לנקודה i.  
 בכל איטרציה נעדכן את:  
   
נקבל: בכל איטרציה הפונק' קטנה.