אלגוריתם חיזוי= רגרסיה לינארית:

פונקצית הMSE : מזעור השגיאה

עבור רגרסיה לינארית במשתנה יחיד, MSE היא פרבולה דו מימדית (קערה) :

$$MSE_{(w_0,w_1)} = \frac{1}{n}\sum_{i\in D}(t_i - y_i)^2 = \frac{1}{n}\sum_{i\in D}\left(t_i - (w_1 x_i + w_0)\right)^2$$

מקרה פרטי של הפרבולה, הוא אם ישנה רק משקולת אחת, ואז הפרבולה היא חד מימדית.

$$\Delta w_i = -\frac{\lambda}{m}\sum_{p\in D}(t_p - y_p)\frac{-\partial y_p}{\partial w_i} = \frac{\lambda}{m}\sum_{p\in D}(t_p - y_p)\frac{\partial yp}{\partial w_i} = \frac{\lambda}{m}\sum_{p\in D}(t_p - y_p)\, x_i$$

אלגוריתם זהו off line כיוון שמשתמשים בכל סט האימונים בכל epoch. ורסיה אחרת של האלגוריתם הזו הופכת אותו להיות on line ובו יש עדכון משקולות אחרי כל דוגמא. במקרה זה $\Delta w_i = \lambda(t_p - y_p)x_{pi}$.

בשיטת GD למיזעור $MSE_D(w)$ עבור היפותזה פארמטרית כללית $h_w(x)$ (שאינה בהכרח לינארית), פונקצית loss תיראה בצורה שונה :

$$lossD, h(w) = \frac{1}{2}MSE_D(w) = \frac{1}{2m}\sum_{p=1}^{m}\left(t_p - h_w(x_p)\right)^2$$

נרמול שדות :

דרכי נירמול שונות :

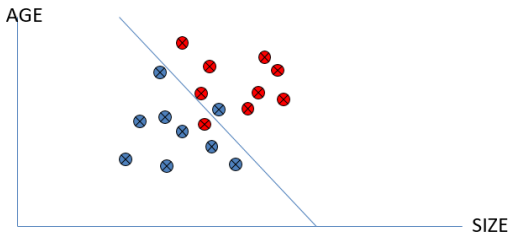
1. Min,max Scaling : יעביר ל [0,1] .
   נמיר את ערכי שדות לערכים מאותו סדר גודל : (x-Min)/range ה Range הוא טווח הערכים (במידה וכולם חיוביים) , למשל (8-2)/(2-#room)
2. Mean-Normalization : נמיר את ערכי השדות כך שיתנו ממוצע 0, (X-mean(X))
3. SD-Scaling : נקבל סטיית תקן 1. נחלק בסטיית התקן : (X/sd(X))
   הערך החדש של המאפיין Feature משקף את המרחק שלו מהממוצע בסטיות תקן

מרצה : ד״ר גדי פנקס
נכתב על ידי : יעלי בר גימלשטיין

אלגוריתם קלסיפיקציה = רגרסיה לוגיסטית:

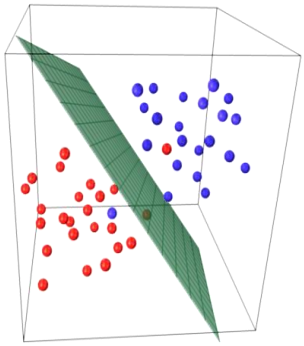
דוגמא – קלסיפיקציה בינארית בשני מימדים:

גבול ההחלטה בדו מימד הוא קו שכל נקודותיו על הישר $w_0 + w_1x_1 + w_2x_2 = 0$.



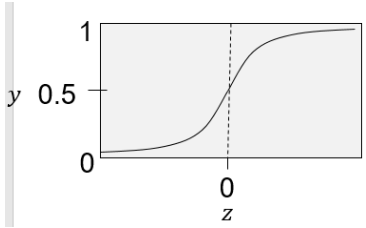
$$p(t = 1|x) \approx$$
$$y = g(z) = \frac{1}{1 + e^{-z}}$$
$$z = w_0 + w_1x_1 + w_2\,x_2$$

בשלושה מימדים, גבול ההחלטה הינו מישור.
בפראקטיקה קיימים הרבה מאפיינים (features) שהם מימדים שיכולים לעזור בסיווג.



$$p(t = 1|x) \approx$$
$$y = g(z) = \frac{1}{1 + e^{-z}}$$
$$z = w_0 + w_1x_1 + w_2\,x_2 + w_3x_3$$

היפותזה עבור רגרסיה לוגיסטית היא מהצורה $y = g(z) = \frac{1}{1+e^{-z}}$, $z = w_0 + \sum_i w_i x_i$.
היפותזה זו משמשת לקלסיפיקציה בינארית ומחזירה הסתברויות. נשים לב שהפונקציה $g(z)$ היא סיגמואיד.



המודל פועל באופן הבא:

- ✓ אימון: בהמלך אימון המודל מחפשים משקולות של מישור מפריד. ככל שהאימון יהיה טוב יותר החיזוי יהיה מדויק יותר.
- ✓ חיזוי: חישוב ההסתברות של נקודה להיות בקטגוריה כלשהיא על פי מרחקה מהמישור המפריד. ההסתברות על הקו היא חצי, ומתחיו ומעליו פחות ויותר בהתאמה.

המרחק בין נקודה x' למישור המפריד $wx = 0$ הוא:

$$\frac{wx'}{||w||} = \frac{z}{||w||} \sim z$$

כשהנורמה שואפת ל-1 המרחק שואף ל-z .

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

פונקציית השגיאה ברגרסיה לוגיסטית:
הפונקציה שבה משתמשים היא Cross Entropy . זוהי פונקציה קמורה, ללא נקודות מינימום מקומיות.
עבור תבנית בודדת $\{\langle x,t\rangle\}$: כאשר המצב החזוי הוא $y=g(z)$:

$$C(y,t)=-t\cdot\log(y)-(1-t)\log(1-y)$$

שגיאת Cross Entropy עבור קבוצת אימון הכוללת m דוגמאות:

$$lossD(w)=\frac{1}{m}\sum_{p=1\ldots m}C(y_p,t_p)$$

כדי למקסם את יעילותו של האלגוריתם נרצה למזער את השגיאה ולקבל את המשקולות בהתאם. לשם כך נרצה למצוא את המשקולות עבורן המישור (או הישר) יהיה מדויק כמה שיותר.
לשם כך, נבצע נגזרות:

$$z=wx,y=h(x)=g(z)=\frac{1}{1+e^{-wx}},\frac{dy}{dz}=y(1-y),\frac{\partial y}{\partial w_i}=y(1-y)x_i$$

נרצה למעזר את פונקציה זו כדי לקבל את המישור הכי מתאים לנתונים שאפשר. זוהי פונקציה קמורה ללא נקודות מינימום מקומיות:

$$\frac{\partial CE_D}{\partial wi}=1/m\sum_{i\in D}(y_p-t_p)x_i$$

כלל עדכון המשקולות = מעזור השגיאה:
ניתן להשתמש בשיטת GD על מנת למצוא את המשקולות, ועידכון המשקולות יהיה באופן הבא:

$$\Delta w_i=\lambda/m\sum_{i\in D}(t_p-y_p)xi$$
$$\Delta w_0=\lambda/m\sum_{i\in D}(t_p-y_p)1$$

***נרצה למקסם את ה*Likelihood *שההיפותזה מסבירה את*D**

$$lD(w)=\prod_{p:t_p=1}y_p\prod_{p:t_p=0}(1-y_p)$$

במקום למקסם מכפלה, ממקסמים את הלוג של המכפלה שהוא סכום של לוגים
- פעולה הסכום מהירה יותר ממכפלה
- בלוג יש פחות אובדן דיוק כאשר ההסתברויות קטנות מאוד.
לעומת ההסתברויות, הלוגים הם מספרים גדולים בערכם המוחלט
- ההסתברות הממוצעת היא ממוצע גיאומטרי (שורש m-י) אבל הופכת לממוצע רגיל של הלוגים

קלסיפיקציה של יותר משתי קטגוריות:
ניתן להעזר בקלסיפיקציה בינארית עבור יותר משתי קטגוריות באופן הבא:
1. גישת one vs rest = נבנה מסווג לכל קטגוריה, כלומר נעביר קו חוצה בינה לבין הקטגוריות האחרות סה"כ k מסווגים. אימון: נלמד להפריד קטגוריה אחת מכל השאר. חיזוי: בהינתן דוגמה לסיווג נבדוק תחזית של כל מסווג ונחזיר את הקטגוריה שלה חוזה הסתברות הגבוהה ביותר.
2. גישת one vs one = נבנה מסווג לכל זוג קטגוריות, כלומר נעביר קו חוצה בין כל זוג קטגוריות. סה"כ $C(k,2)$ מסווגים. אימון: נלמד להפריד קטגוריה אחת מכל השאר. חיזוי: מספר אפשרויות. אחת תהיה הצבעת הרוב, אחרת תהיה חישוב הסתברות ממוצעת לכל קטגוריה ובחירת קטגוריה שמקבלת את ההסתברות המקסימלית.

פונקצית השגיאה:

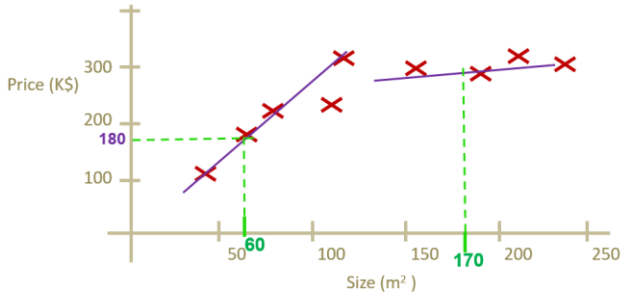
$$MCCE_D(y,t)=-\frac{1}{m}\sum_p\sum_i^k t_{pi}\log(y_{pi})+(1-t_{pi})\log(1-y_{pi})$$

ניתן לראות שכאשר יש הרבה קטגוריות, חישוב הפונקציה לא יעיל במיוחד.

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

שליטה על גמישות רגרסיה לינארית:

על מנת לשלוט בגמישו המודל הלינארי, ניתן להשתמש בשיטת Locally Weighted Linear Regression .
לפי שיטה זו, לא נבנה מודל בזמן האימון, אלא רק בעת השאילתה (query). זוהי שיטה שהיא שילוב של שיטה לא פארמטרית ביחד עם רגרסיה לינארית.



בהינתן שאילתה (וקטור מאפיינים x), נבנה מודל רגרסיה, אך ההשפעה של הנתונים השונים תהיה שונה: הנקודות בסט הנתונים D שקרובות יותר לוקטור x ישפיעו יותר על מודל הרגרסיה מאשר נקודות רחוקות יותר.

אימון:

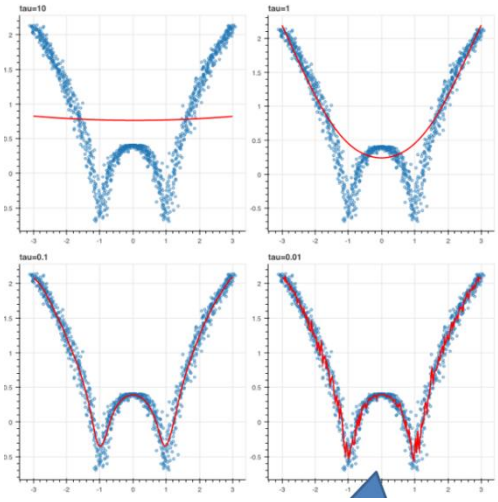
- ✓ שומרים בזיכרון דוגמאות ומשתמשים בהם בכל פעם שזקוקים לחיזוי.

חיזוי:

- ✓ בהינתן וקטור x מחשבים משקולות [0-1] לדוגמאות האימון על פי מרחקיהם מהוקטור x כאשר דוגמה רחוקה- משקלה יתקרב לאפס, ואילו דוגמה קרובה- משקלה יתקרב לאחד.

חישוב המרחק לפי פונקציה גאוסיאנית מסביב לנקודה $\beta_i = e^{\frac{-\|x_i - x\|^2}{2\tau^2}}$. נשים לב כי ככל שהקבוע τ קטן יותר והמרחקים הם גדולים, אז המשקלים המתקבלים הם משקלי אפס. לעומת זאת, ככל שהקבוע τ גדול יותר נלקחות בחשבון גם נקודות רחוקות יותר. פונקציית המרחק מודדת גם אי דמיון.

- ✓ מבצעים רגרסיה עם פונקציית MSE ממושקלת: $WMSE_w = \frac{1}{2m}\sum_i \beta_i (wx_i - t_i)^2$



בעזרת פונקצית המרחק ניתן לשלוט על הגמישות של המודל. ככל שהקבוע τ קטן יותר המשקל של נקודות שהן מרוחקות שואף לאפס ואילו נקודה קרובה משפיעה יותר. ככל שהקבוע τ גדול יותר, כל נקודות האימון משפיעות באופן דומה ומקבלים רגרסיה לינארית שהיא רגילה.

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

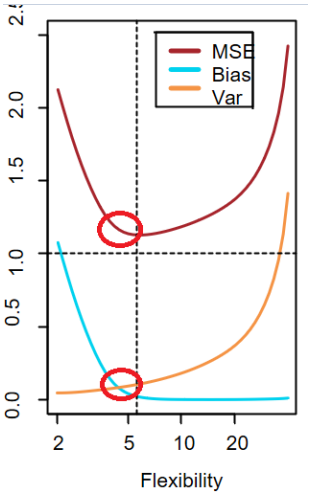
הקשר של פונקצית השגיאה MSE לבין bias , variance :

פירוק מתמטי של פונקצית השגיאה לגורמים bias , variance:
השגיאה שיש למודל על נתוני המבחן מורכבת משלושה סוגי שגיאה :

$$loss = (Bias)^2 + Variance + irreducible$$

ניסוח מתמטי של הגורמים bias , variance:
כפי שנראה בהוכחה, נגדיר באופן מילולי ומתמטי את שני המושגים :

1. הביאס= תוחלת ההפרשים של ההיפותזה והפונקציה $Bias[h,f] = E\left[\left(\overline{h_D(x)} - f(x)\right)^2\right]$, כלומר, מדד הביאס מציין עד כמה ההיפותזה שונה מהפונקציה אותה אנחנו מחפשים. אם ההיפותזה לא דומה כלל לפונקציה אותה מחפשים נקבל under fitting.
2. השונות = תוחלת ההפרשים של ההיפותזה לבין התוחלת שלה $variance[h_D(x)] = E\left[\left(h_D(x) - \overline{h_D(x)}\right)^2\right]$ כלומר, מדד השונות מציין עד כמה רחבה ומגוונת ההיפותזה יכולה להיות. כמובן שהשונות תלויה במרחב המדגם, היות ויכולים להיות מדגמים שונים. אם ההיפותזה מותאמת יותר מידי למדגם מסוים נקבל over fitting.



פשרת The bias – variance tradeoff:
ככל שמעלים גמישות של מודל הביאס יורד והשונות עולה.בפרט, פונקצית MSE עולה או יורדת בתלות לקצב הירידה בביאס וקצב העליה בשונות.

בד"כ בהתחלה, קצב הירידה בביאס מהיר מקצב העליה בשונות ולכן יש ירידה התחלתית בשגיאה ואחכ עליה, ונרצה לבחור את האיזור בו השגיאה היא מינימלית.

כאשר המודל אינו מספיק גמיש ושגיאת Bias גבוהה : משלמים למהנדסים "יצירתיים" כדי שיפיקו טרנספרמציות למאפיינים "טובים" , או שמשתמשים במודלים גמישים ובטכניקות לביצוע טרנספורמציות אוטומטיות (LWLR, עצים, רשתות נוירונים, Kernels).

כאשר המודל גמיש מידי ושגיאת השונות גבוהה : קורה כאשר מימד הקלט גבוה או שהוספנו יותר מידי מאפיינים. ניתן לתיקון על ידי השקעה כספית : קונים חומרה חזקה , משלמים כדי לקבל יותר נתונים מסווגים או משתמשים בטכניקות לצמצום מספר המימדים (למשל feature selection, PCA) או משתמשים בטכניקות לצמצום שגיאת ה variance : מורידים גמישות המודל, משתמשים בטכניקות לרגולריזציה, או מוסיפים (או מחוללים) נתונים.

מרצה : ד"ר גדי פנקס
נכתב על ידי : יעלי בר גימלשטיין

הצלחת המודלים:

נרצה להעריך את מידת ההצלחה של המודל הלומד.

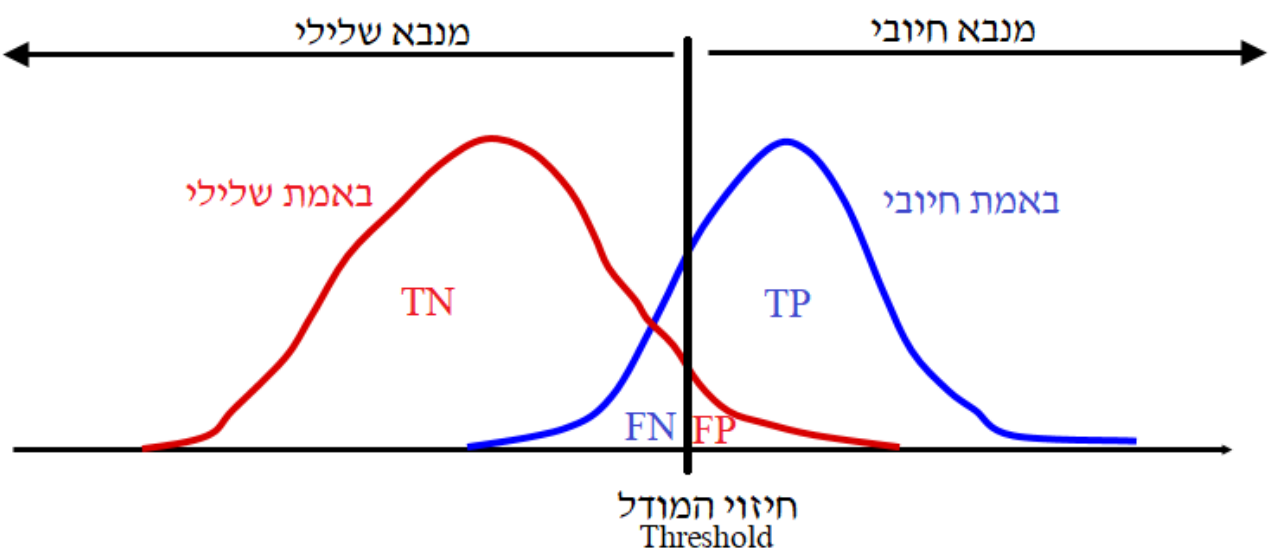
KPI's = Key Performance

עבור משתמשים אלו (עולם האפליקציה) עדיף למדוד ביצועים בדרכים אחרות, KPI's:

sensitivity, recall, hit rate, or true positive rate (TPR)	$\frac{TP}{TP+FN} = 1 - FNR$
specificity, selectivity or true negative rate (TNR)	$\frac{TN}{TN+FP} = 1 - FPR$
precision or positive predictive value (PPV)	$\frac{TP}{TP+FP} = 1 - FDR$
negative predictive value (NPV)	$\frac{TN}{TN+FN} = 1 - FOR$
miss rate or false negative rate (FNR)	$\frac{FN}{FN+TP} = 1 - TPR$
fall-out or false positive rate (FPR)	$\frac{FP}{FP+TN} = 1 - TNR$
false discovery rate (FDR)	$\frac{FP}{FP+TP} = 1 - PPV$
false omission rate (FOR)	$\frac{FN}{FN+TN} = 1 - NPV$
Positive likelihood ratio (LR+)	$\frac{TPR}{FPR}$
Negative likelihood ratio (LR-)	$\frac{FNR}{TNR}$

הדיוק, Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$.

ניתן להציג גם בעזרת גרף:



$$F1\ score = 2 \cdot \frac{P \cdot R}{P+R}$$

הכללה- F measure:

$$F - Measure = \frac{1}{\alpha \cdot \frac{1}{Precision} + (1-\alpha) \cdot \frac{1}{Recall}}$$

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

מדד balanced accuracy למודל שאינו מאוזן:
Balanced accuracy הוא בעצם ממוצע (רגיל) של המדד האיחזור recall של כל קטגוריה.
בקלסיפיקציה בינארית, זהו הממוצע בין האיחזור (recall) לסגוליות (specificity).

Cross Validation:

K-Fold Cross Validation =

תהליך האימון:

1. חלק את הנתונים המתויגים ל K קבוצות
2. שמור 1/K מהנתונים כקבוצת ולידציה: אמן על (K-1)/K מהנתונים ובדוק על הוולידציה
3. בצע K אימונים שונים בכל פעם על קבוצת וולידציה אחרת וקבל K שיערוכים שונים לשגיאה (validation loss, F1....)

בחירת המודל עבור המבחן:

4. מצע את k השיערוכים כדי לקבל הערכה לשגיאת הטסט

למשל, תהליך האימון עבור 10-fold CV

- ✓ נחלק את ה training ל 10 קבוצות: נאמן על 9 ונשתמש בעשירית כ validation.
- ✓ נמשיך כך 10 פעמים, כשבכל פעם בוחרים קבוצת validation אחרת
- ✓ כמובן, כתוצר לוואי ניתן לבנות קלסיפייר שהוא אנסמבל של k קלסיפיירים שונים

סה"כ ישנם k מודלים שיחושבו.

Leave K Out Validation =

תהליך האימון:

1. בהינתן קבוצת דוגמאות מתויגות D בחר קבוצת וולידציה של k דוגמאות כאשר יש v כאלה, אמן על D-v הדוגמאות שנותרו. מספר האימונים יהיה C(m, k) כאשר m=D

בחירת המודל עבור המבחן:

2. שערך את השגיאה ע"י מיצוע של השגיאות על כל הדוגמאות ב D.

סה"כ ישנם C(m, k) מודלים שיחושבו.

Leave One Out Cross validation =

תהליך האימון:

1. בחר דוגמא אחת בלבד כוולידציה, אמן על m-1 הדוגמאות שנותרו
2. עשה זאת עבור כל אחת מדוגמאות האימון ובדוק את המודל על דוגמת הוולידציה שנבחרה.

בחירת המודל עבור המבחן:

3. שערך את השגיאה ע"י מיצוע של השגיאות על כל הדוגמאות ב D.

סה"כ ישנם m מודלים שיחושבו (כמספר הדוגמאות).

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

הקטנת שגיאת השונות:

בחירת פרמטרים feature selection:

דרך מקובלת לפשט את ההיפותזות היא להוריד מימדים באמצעות בחירת מאפיינים. יתרונותיה:

1. הפחתה של שגיאת השונות (מטרה)
2. יותר קל להסביר מודל בעל פחות מימדים
3. מודל בעל פחות מימדים מהיר יותר לחישוב

=Forward feature selection

שיטה חמדנית מבוססת על היוריסטיקה הכללית.

ניצור n קבוצות של מאפיינים כאשר גודל של כל קבוצה הוא $k = 1, 2, \dots, n$. בכל פעם נחפש את המאפיין בעל השגיאה הטובה (הקטנה) ביותר ונוסיף אותו לקבוצות המאפיינים הרצויים.
בסוף נבחר בין כל הקבוצות את זו שיוצרת שגיאת שיערוך מינימלית.

1. Start with empty feature set
2. For $k = 1, \dots, n$:
   a. For all features f_k not used by model:
      i. Try adding each of the missing features, train and save the loss
      ii. Create model M_k by adding the feature that provides the best loss
3. Choose the best $M_k$using cross validation

יווצרו n^2 מודלים (סדרה חשבונית), n שיערוכי שגיאה cross validation.

=Backwards feature selection

שיטה חמדנית מבוססת על היוריסטיקה הכללית.

ניצור n קבוצות של מאפיינים כאשר גודל של כל קבוצה הוא $k = 1, 2, \dots, n$. בכל פעם נחפש את המאפיין בעל השגיאה הרעה (הגדולה) ביותר ונוריד אותו מקבוצות המאפיינים הרצויים.
בסוף נבחר בין כל הקבוצות את זו שיוצרת שגיאת שיערוך מינימלית.

1. Start with full feature set
2. For $k = 1, \dots, n$:
   a. For all features f_k not used by model:
      i. Try removing each of the missing features, train and save the loss
      ii. Create model M_k by removing the feature that provides the best loss
3. Choose the best $M_k$using cross validation

יווצרו n^2 מודלים (סדרה חשבונית), n שיערוכי שגיאה cross validation.

=Hybrid feature selection

שיטה חמדנית מבוססת על היוריסטיקה הכללית.

ניצור n קבוצות של מאפיינים כאשר גודל של כל קבוצה הוא $k = 1, 2, \dots, n$. בכל פעם נחפש את המאפיין בעל השגיאה הטובה (הקטנה) ביותר ונוסיף אותו לקבוצות המאפיינים הרצויים. לאחר מכן, נחפש את המאפיין בעל השגיאה הרעה (הגדולה) ביותר ונוריד אותו מקבוצות המאפיינים הרצויים.
בסוף נבחר בין כל הקבוצות את זו שיוצרת שגיאת שיערוך מינימלית.
נועד לפצות על החמדנות של השיטות הקודמות.

1. Start with full feature set
2. For $k = 1, \dots, n$:
   a. add a feature in Forward selection
   b. remove a feature in Backward selection
   c. save M_k
3. Choose the best $M_k$using cross validation

יווצרו n^2 מודלים (סדרה חשבונית), n שיערוכי שגיאה cross validation.

מרצה: ד״ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

רגולריזציה:

= Ridge regularization (L2)

נעניש את הפונקציית השגיאה על ידי הוספת הריבוע של הנורמה מסדר שני (שהם ריבועים):

$$\|w\|_2 = \sqrt{\sum_{i=1}^{n} w_i^2}$$

ואז, נוסחת השגיאה המעונשת תהיה $regMSE(w) = MSE(w) + \frac{\gamma}{2}\sum_{j=1}^{n} w_j^2$.

אם כך, כלל עדכון המשקולות משתנה בהתאם (גזירה):

$$\Delta w = -\lambda \frac{\partial loss}{\partial w_i} - \gamma w_i$$

חשוב לציין כי שיטה זו היא רגישה למדד של המאפיינים, ולכן עדיף לבצע נורמליזציה לנתונים.

חשוב לעשות למאפיינים Scale normalization ע"י חלוקה בסטית תקן. סטיית התקן לאחר הנירמול היא 1 לכל המאפינים שנורמלו.

יתרונות שיטה זו על פני בחירת מאפיינים:

1. יעילות חישובית
2. מאפיינים בעלי תרומה קטנה, לא יסולקו בהכרח אלא ישתתפו בחיזוי (משקולות קטנים)
3. עמידות טובה יותר לרעשים: רעש בהרבה מאפיינים מתמצע וכן מופחת
4. ניתן להרחבה לכל אלגוריתם למידה פארמטרי (למשל רשתות נוירונים)

החסרון בשיטה זו הוא שנשארים הרבה מאפיינים וקשה יותר להסביר מודל כזה.

= Lasso regularization (L1)

לפעמים יש יתרון ברגורליזציה המתבססת על ערכים מוחלטים (במקום על ריבועי המשקולות).

$$\|w\|_1 = \sqrt{\sum_{i=1}^{n} |w_i|}$$

ואז, נוסחת השגיאה המעונשת תהיה $regMSE(w) = MSE(w) + \gamma\sum_{j=1}^{n}|w_j|$.

אם כך, כלל עדכון המשקולות משתנה בהתאם (גזירה):

$$\Delta w = -\lambda \frac{\partial loss}{\partial w_i} - \gamma$$

שיטה זו מסוגלת לבצע בחירת מאפיינים. ככל שנגדיל את ה"עונש" משתנים יעלמו (יקבלו משקל אפס).

בשיטת Ridge המשתנים יקבלו משקולות נמוכים אבל לא יעלמו לגמרי.

= Elastic net regularization

הנוסחה $\frac{lasso+ridge}{2}$.

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

אנסמבלים:

שיטות ליצירת אנסמבלים:

1. שיטת Bagging = יצירת אנסמבלים על ידי דגימות שונות מתוך סט האימונים. כלומר, מאמנים מודלים שונים על תת קבוצות שונות של הנתונים, ואת הפלט ממצעים או החלטת הרוב. את המודלים השונים ניתן ליצור במספר דרכים:
   a. שיטת boot strap = שיטת דגימה בה מייצרים הרבה קבוצות אימון שונות ע"י דגימה אקראית עם חזרות מתוך קבוצת אימון יחידה. לכל קבוצה מותאמת קבוצת ווליידציה הכוללת דוגמאות שלא נלקחו לדגימה.
   b. שיטת k fold = מיצרים k קבוצות אימון (בדומה ל k-fold cross validation) ע"י שמוציאים בכל פעם 1/k מהנתונים כוולידציה, ומשאירים בקבוצת האימון את היתר k-1/K. קבוצות הוולידציה הן זרות זו לזו. ניתן כמובן "לנצל" k-fold cross validation (שמטרתו לשערך שגיאה ולכוונן היפר-פארמטרים) על מנת לבנות bagging Ensemble בעזרת K המודלים שנוצרו.
2. שיטת boosting = אנסמבלים של מודלים מאומנים על קבוצות אימון שונות כך שכל מודל מתמקד בדוגמאות שהקודמים לו שגו בהן. כלומר, משתמשים בסדרה של מודלים "חלשים" (עם שגיאת ביאס גבוהה) וכל מודל מתרכז בלמידה של המקרים שהמודלים האחרים לא הצליחו לסווג נכון:
   a. משתמשים באותם נתונים, אך מגדילים את משקל הדוגמה (תדירות הופעתה) במידה והמודלים הקודמים לא סיווגו אותה נכון.
   b. מקבלים סידרת מומחים (שכל אחד "חלש" מידי – ולא יכול לעשות Overfitting) שנותנים תחזיות שונות למיקרים קשים.
   c. את המודלים השונים שמתקבלים ניתן לקבל ע"י למשל הצבעת הרוב או מיצוע (גיאומטרי) של הסתברויות– (אפשר לשקלל מודל על פי חשיבות השגיאות שמטופלות על ידו) או על פי מידת הצלחתו על ווליידציה
   d. חיסרון: רגיש ל outliers ועלול לגרום ל high-variance אם המומחים אינם כל כך חלשים.

אלגוריתמים שונים:

אלגוריתם K – nearest neighbors:

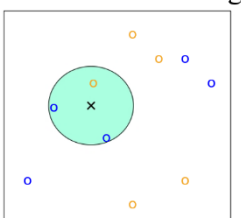
עבור קלסיפיקציה:

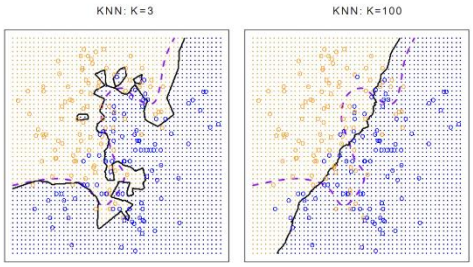
בהינתן אוסף דוגמאות (אימון) D, ודוגמת מבחן חדשה x, נשערך את ההסתברות האפוסטריורית (=המותנית) של כל הקטגוריות $p(y=j|x,D)$ ונבחר בקטגוריה שההסתברות שלה היא המקסימלית.

השיערוך באלגוריתם KNN הוא על פי הצבעת השכנים:

- ✓ נמצא את הקבוצה N_x המכילה את K הדוגמאות מתוך D ה"קרובות" ביותר לדוגמא x
- ✓ ההסתברות לקטגוריה j מחושבת על פי מספר הדוגמאות מקבוצה N_x שסווגו לקטגוריה j באופן הבא: $p(y=j|x,D)=\frac{1}{k}\sum_{i\in N_x} I(y_i=j)$
- ✓ התחזית עבור הדוגמה x היא לכן על פי "הצבעת" הרוב מתוך קבוצת N_x השכנים באופן הבא: $I(y_i=j)=\begin{cases}1 \; if \; y_i=j \\ 0 \; else\end{cases}$

3-Nearest Neighbors (KNN)





K=3 is too flexible, k=100 is almost linear
K=N is the null hypothesis: y= est_prior(+)

גבול ההחלטה (המפריד בין הקטגוריות השונות) אינו חייב להיות לינארי.
הגמישות יורדת ככל שמשתתפים יותר שכנים בהחלטה.

עבור רגרסיה:

בהינתן אוסף דוגמאות (אימון) D, ודוגמת מבחן חדשה x, עבור על k הדוגמאות הכי קרובות והחזר את ממוצע הערכים על פני k הדוגמאות השכנות.

ההחלטה מי קרוב:

עלינו להחליט על סמך איזו שיטה נחליט מי השכנים הקרובים לדוגמא.

1. כאשר המאפיינים נומרים בלבד:
   - ✓ מרחק אוקלידי, $L=2$, $d(q,p)=\sqrt{\sum_{i=1}^{n}(q_i-p_i)^2}$
   - ✓ מרחק מנהטן, $L=1$, $d(q,p)=|q|+|p|$
   - ✓ מרחק נורמת האינסוף $L=\infty$, $d(q,p)=\max(|q_i-p_i|)$

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

2. מרחק קוסינוס: מרחק זה מודד קרבה לפי דמיון, אורנטציה ולא לפי גודל.

$$cosineSim(A,B) = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

כזכור, הקוסינוס של הזווית אפס הוא 1, והוא פחות מאחד עבור כל זווית באינטרוול $[0,\pi]$.
לכן, שני וקטורים מאונכים אחד לשני הדמיון שלהם יהיה אפס, ואילו שני וקטורים עם אותה אורינטציה (זווית אפס), הדמיון שלהם יהיה אחד.

3. מרחק קורלצית פירסון: מרחק זה הוא בעצם מרחק קוסינוס -1 כאשר הוקטורים מנורמלים מבסיס לממוצע שלהם.
$d(A,B) = \frac{\sum_{i=1}^{n}(A_i-\overline{A})(B_i-\overline{B})}{\sqrt{\sum_{i=1}^{n}(A-\overline{A})_i^2}\sqrt{\sum_{i=1}^{n}(B-\overline{B})_i^2}}$. במקרה זה, המרחק נקבע על ידי הדמיון בצורה של פרופילים (למשל פרופיל לקוח).

4. עבור וקטור בעל מאפיינים קטגורים: נעזר במרחק hamming = ספירת מספר הביטים השונים של 2 הוקטורים.

כאמור, דמיון יכול להיות קשור למרחק. ניתן לבצע גם את התהליך ההפוך- הפיכת מדידת מרחק לדמיון בין שתי נקודות:

1. ההופכי למרחק בריבוע: $GravityLaw(x_i,x) = \frac{1}{d(x_i,\ x)^2}$.
2. דמיון גאוסיאני: $GausSimilarity(x_i,x) = e^{\frac{-d(x_i,\ x)^2}{2\tau^2}}$.

טאו שולט על רוחב הגאוסיאן (בדומה לסטית התקן בהתפלגות נורמלית), טאו קטן: רק נקודות מאוד קרובות יקבלו ערך דימיון גדול, נקודות שאינן קרובות יונחתו לאפס.

<u>למידה בייסיאנית Bayesian learning:</u>

חוק בייס לאירועים בדידים בלתי תלויים:
חוק בייס הוא תוצאה בתורת ההסתברות המאפשרת לחשב הסתברות מותנית של מאורע כאשר יודעים את ההסתברויות המותנות ההפוכות.

באופן פורמלי, ההסתברות המותנית של מאורע A בהינתן מאורע B היא הסיכוי להתרחשותו של A בהנחה ש- B אכן התרחש.
קיום ההנחה מצמצם את מרחב המדגם. מרעיון זה נובעת הנוסחה להסתברות המותנית הבאה: $P(B|A) = \frac{P(A\cap B)}{P(A)}$.
הסתברות זו נקראת "אחורית" (posterior). לעומת זאת, ההסתברות למאורע כללי $P(B)$ נקראת "קודמת" (prior).

ואילו, חוק בייס מאפשר לחשב את ההסתברות המותנית ההפוכה- ההסתברות המותנית של B בהינתן A: $P(A|B) = \frac{P(B|A)\cdot P(A)}{P(B)}$.

מתוך נוסחת ההסתברות השלמה, במידה והמאורעות A_i זרים אז $P(B) = \sum_i^k P(B|A_i)P(A_i)$ ואז: $P(A_i|B) = \frac{P(B|A_i)\cdot P(A_i)}{\sum_i^k P(B|A_i)P(A_i)}$.
כלומר, ניתן לחשב הסתברות posterior של A_i בהינתן B בעזרת priors וכן ההסתברות המותנית ההפוכה לה.

חוק בייס עבור קלסיפיקציה:
נגדיר מאורעות זרים שהם הקטגוריות השונות בקלסיפיקציה $1,\dots,k$. ואז, למשל, בהינתן דוגמא x, ההסתברות שלה להיות מסווגת בקטגוריה k היא

$$P(k|x) = \frac{P(x|k)\cdot P(k)}{P(x)} = \frac{P(x|k)\cdot P(k)}{\sum_i^k P(x|i)P(i)} = \frac{P(x\cap k)}{P(k)}\cdot\frac{P(k)}{\sum_i^k P(x|i)P(i)} = \frac{P(x\cap k)}{\sum_i^k P(x|i)P(i)}$$

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

את ההתסברות שהיא priors של הקטגוריות בד"כ קל לשערך. אם היינו יודעים לשערך גם הסתברויות הקלט בהינתן קטגוריה $p(x|k)$, או לחילופין את $p(x \cap k)$ אז היינו יכולים לחשב את ההסתברות הפוסטריורית $p(k|x)$ לכל לקטגוריה k.

לדוגמא $x = fever$ הינו סימפטום (מאפיין), ורוצים לחשב את ההסתברות של להיות חולה בקורונה או בריא כתלות בסימפטום שהוא חום $p(k = corona|fever);\ p(k = healthy|fever)$.

הערכה מקסימלית להסתברות אחורית MAP = Maximal A-Posteriori Estimation:
נרצה לחזות את הסיווג על פי הקטגוריה בעלת ההסתברות המקסימלית מבין כל הקטגוריות האפשריות.

$$y_{MAP} = argmax_k \{p(k\,|x)\}$$

כאשר:

- ✓ k היא קטגורית סיווג (מתוך קבוצה K של קטגוריות)
- ✓ x הוא וקטור של ערכי קלט (מאפיינים) אותו רוצים לסווג

כלומר, אם נשתמש במדד map נצטרך לחשב:

$$argmax_k \{p(k\,|x)\} = argmax_k \left\{\frac{P(x|k) \cdot P(k)}{\sum_i^k P(x|i)P(i)}\right\} = argmax_k\{P(x|k) \cdot P(k)\}$$

לשמחתנו כדי לסווג מהי הקטגוריה בעלת ההסתברות המקסימלית לא צריך לחשב את המכנה. בד"כ המכנה קשה לשיערוך היות וצריך את ההסתברות של כל צירוף מאפיינים. אבל אם נרצה לחשב את ההסתברות הפוסטריורית ממש, נצטרך לחשב את המכנה.

הערכה מקסימלית להסתברות סבירות ML= Maximal Likelihood Estimation :
נרצה לחזות את הסיווג על פי הקטגוריה בעלת ההסתברות המקסימלית מבין כל הקטגוריות האפשריות. אבל, כאשר לא יודעים לשערך את ה Priors של הקטגוריות, בהרבה מיקרים מניחים שהסתברות prior של הקטגוריות שווה זו לזו ולכן ניתן למקסם רק את $P(x|k)$, כלומר, בשונה מהערכת map :

$$y_{ML} = argmax_k \{P(x|k)\}$$

ניתן לחזות את הקטגוריה בקלסיפיקציה ביזיאנית בעזרת ML אך אם הנחת השיוויון ב Priors שגויה, נקבל הבדלים מול חיזוי על פי MAP.

נשים לב כי נתוני recall, precision מספקים מידע הסתברותי חשוב עבור החישובים שמעלה.
דוגמא:
מטופל נבדק במעבדה והבדיקה חוזרת חיובית למחלת הסרטן. האם תבחר לעשות לו ניתוח מורכב כאשר יודעים את הרגישות והספציפיות של הבדיקה?

- ✓ recall: ידוע כי בדיקת מעבדה חיובית תתקבל ב 98% מהמיקרים בהם ישנה המחלה
- ✓ true negative rate: תוצאה שלילית נכונה תתקבל ב 97% מהמיקרים בהם המחלה איננה
- ✓ ידע מוקדם: 0.8% מהאוכלוסיה חולים במחלה

מנתונים אלה אנו מבינים:

- ✓ $P(-|cancer) \approx 0.02, P(+|cancer) \approx 0.98$
- ✓ $P(-|\neg cancer) \approx 0.97\ , P(+|\neg cancer) \approx 0.03$
- ✓ $P(cancer) \approx 0.008, P(\neg cancer) \approx 0.992$

ברור כי לסיווג ישנן שתי קטגוריות $cancer$, $\neg cancer$ וכן ישנו מאפיין אחד בוליאני שהוא תוצאת הבדיקה $+$, $-$.
כעת, נוכל לחשב את הסיווג על פי הקטגוריה בעלת ההסתברות המקסימלית מבין כל הקטגוריות האפשריות, כאמור, בשני אופנים:

1. לפי MAP :
   I. $y_{cancer} = P(cancer|\,+) = P(+|cancer) \cdot P(cancer) = 0.98 \cdot 0.008 = 0.00784$
   II. $y_{\neg cancer} = P(\neg cancer|\,+) = P(+|\neg cancer) \cdot P(\neg cancer) = 0.03 \cdot 0.992 = 0.0298$
2. לפי ML :
   I. $y_{cancer} = P(cancer|\,+) = P(+|cancer) = 0.98$
   II. $y_{\neg cancer} = P(\neg cancer|\,+) = P(+|\neg cancer) = 0.03$

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

ואז, נוכל לבחור היפותזה בהתאם לכל אחת מהגישות:

1. לפי MAP : $0.0298 > 0.00784$ כלומר, הסיווג שיבחר הוא $y_{\neg cancer}$, לא חולה
2. לפי ML : $0.98 > 0.03$ כלומר, הסיווג שיבחר הוא y_{cancer} , חולה

נשים לב כי דוגמה זו לא שיערה את ההסתברות עצמה, אלא רק נתנה סיווג שיבחר.
במידה והיינו רוצים לחשב את השערוך להסתברויות, היינו מחשבים:

1. עלינו לחשב את ההסתברות האחורית של לקבל בדיקה חיובית, ולכן:
   I. $P(+ \cap cancer) = p(+|cancer)p(cancer) \approx 0.98 \cdot 0.008 = 0.00784$
   II. $P(+ \cap \neg cancer) = p(+|\neg cancer)p(\neg cancer) \approx 0.03 \cdot 0.992 = 0.0298$
   III. $P(+) = P(cancer \cap +) + P(\neg cancer \cap +) \approx 0.00784 + 0.0298 = 0.03764$
2. כעת, על פי הנוסחה להתברות מותנית ניתן לחשב את השערוך:
   I. $P(cancer|+) \approx \frac{P(+\cap cancer)}{P(+)} = \frac{0.00784}{0.03764} \approx 0.21$
   II. $P(\neg cancer|+) \approx \frac{P(+\cap \neg cancer)}{P(+)} = \frac{0.0298}{0.03764} \approx 0.79$

נשים לב כי החישוב הנ"ל מאמת את גישת MAP .

וחזרה לשאלה בתחילת הדוגמה- האם תבחרו לעשות ניתוח להוצאת הגידול? התשובה היא שזו החלטה סובייקטיבית, כלכלית, כמה אתם מעריכים את חייכם? וכמה את הסבל הסיכון והעלות הכרוכים בניתוח?

שערוך הסתברויות מתוך מדגם הנתונים:
אם יש מדגם גדול, נוכל לספור ולשערך הסתברות.
למשל, בהינתן מדגם גדול המשקף את האוכלוסיה של חולים ובריאים עם תוצאות הבדיקה, נוכל לספור:

סהכ נבדקים n	כמה חולים n_c	כמה בריאים n_h	כמה חיוביים n_+	כמה שליליים $n - n_+$	לכמה חולים יש בדיקה חיובית $n_{c\cap+}$	לכמה בריאים יש בדיקה חיובית $n_{h\cap+}$

ואז, נוכל לשערך הסתברויות: $P(+|c) = \frac{n_{c\cap+}}{n_c}$, $P(+) = \frac{n_+}{n}$.

הנחות בייס:
שיטות אלו מצויינות כאשר וקטור המאפיינים הוא חד מימדי כפי שראינו בדוגמה, אבל איך נשערך את ההסתברות לקטגוריה בעבור וקטור רב מימדים?

אם x הוא ממימד קטן והמאפינים הם בעלי מספר קטן של ערכים אפשריים, לעיתים יהיו מספיק נתונים בסט הנתונים כדי לשערך ישירות ע"י ספירה $P(x\ |k) = \frac{n_{x\cap k}}{n_k}$. אך כאשר למאפינים יש הרבה ערכים, או כשהמימד גבוה, לא יהיו לנו הרבה דוגמאות בסט הנתונים שזהות לוקטור הקלט x (אם בכלל), השערוך הישיר להסתברות לא יהיה מדויק.

נעזר בהנחות שונות.

הנחת בייס נאיבית:

1. הנחה נאיבית חזקה (אגרסיבית): המאפיינים אינם תלויים זה בזה בהינתן קטגוריה k , כלומר ההסתברות לכל וקטור הקלט הוא פשוט כפל ההסתברויות $p(v_1, v_2, \dots v_n\ |k) \approx \prod_{j=1}^{n} p(x_j = v_j|k)$. בנוסף, ידוע כי $p(k) \approx \frac{n_k}{n}, p(x_j = v_j|k) \approx \frac{n_{v_j,k}}{n_k}$. ואז, הסיווג הינו
$$y_{MAP} \approx argmax_k \{p(k) \cdot \prod_{j=1}^{n} p(x_j = v_j|k)\}.$$
2. סדר המילים אינו חשוב, ההסתברות של מילה מסוימת להופיע בכל פוזיציה היא שווה (איננה תלויה בפוזיציה במסמך) $p(a_i = w_j|k) = p(a_m = w_j|k)$.

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

תחת הנחה זו, נלמד את אלגוריתם הקלסיפיקציה הנאיבי של בייס:

1. הערכת כל ההסתברות הנחוצות – התסברויות אחוריות, הסתברויות של מאפיינים בהתאם לקטגוריה
2. קבלת קלט וחישוב הסיווג שמוחזר.

באופן פורמלי, האלגוריתם נראה כך:

Naïve Bayes Classifier (D, K, x):
For each category k=1…K:

- $P(k) \approx \frac{n_k}{n}$
- For each value v_j of each attribute x_j : $P(v_j|k) = p(x_j = v_j|k) \approx \frac{n_{v_j \cap k}}{n_k}$

Return $predict = argmax_k \{P(k) \cdot \prod_{j=1}^{n} P(v_j|k)\}$

אלגוריתם זה פשוט מאוד, קל לחישוב והבנה, בעל הנחות מפשטות שגויות (בד"כ), ומצד שני, באופן מפתיע, נותן לפעמים תוצאות טובות למדי, גם כאשר הנחות האי-תלות לא מתקיימות.

אלגוריתם זה שימושי בקלסיפיקציה רפואית ועיבוד שפה טבעית. למשל, גוגל השתמשה בהנחות ביזיאניות נאיביות בתחילת דרכה (ואולי עדיין) עבור קלסיפיקציה של מסמכים.

ישנה בעיתיות בהנחת הנאיביות. במקרים רבים מקבלים הערכות להסתברויות אפוסטריוריות קרובות לאחד או לאפס באופן לא ריאלי (בגלל תלות סטטיסטית בין המשתנים או בגלל מידגם קטן מה שמוביל לשיערוך לא מדויק). מכיוון שהסתברות לא יכולה להיות אפס, נהוג לתקן שיערוכים של ערכי הקטגוריות הנדירים.

תיקון לשיערוכי הסתברות עבור ערכים נדירים $m - estimate$:

נהוג לתקן שיערוכים של ערכי הקטגוריות הנדירים בהסתמך על הסתברות הקודמת של ערכים אלו- "כאילו" שקימות עוד מספר קטן m של דוגמאות בקטגוריה שבהן מופיע הערך v בהסתברות של $prior(v)$.

כאשר אין מספיק דוגמאות מקטגוריה $y = k$ ושבהן גם $x_i = v_i$, מוסיפים m דוגמאות וירטואליות מקטגורה k שמתוכם $P(x_i = v_i)$ הם בעלי ערך $x_i = v_i$. $P(x_i = v_i)$ הינו השערוך הקודם של v_i .

m הינו היפר פאראמטר שמצפה על גודל המדגם או על התלויות.

$$\hat{P}(x_i = v_i|k) \approx \frac{n_{i \cap k} + m \cdot P(x_i = v_i)}{n_k + m}$$

אם לא ניתן לשערך את ההסתברות הקודמת $P(x_i = v_i)$ כי למשל סט הנתונים אינו מייצג אוכלוסיה כללית או שהערכים מאוד נדירים, נבחר שערוך אחר.

תיקון לשיערוכי הסתברות עבור ערכים נדירים $Laplace\ Smoothening$:

אם לא ניתן לשערך את ההסתברות הקודמת $P(x_i = v_i)$, אפשר להניח התפלגות אחידה בין הערכים $v_1 \dots v_n$ ולשערך $P(x_i = v_i) \approx \frac{1}{n}$. ואז ההסתברות תהיה

$$\hat{P}(x_i = v_i|k) \approx \frac{n_{i \cap k} + m \cdot \frac{1}{|v|}}{n_k + m}$$

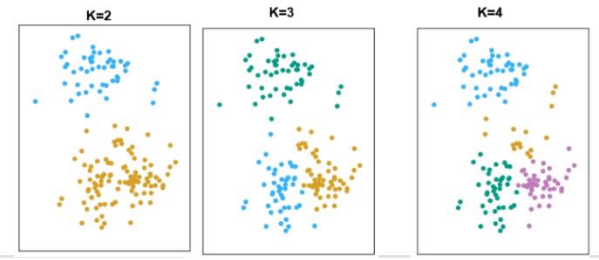
בפרט, אם נוסיף $m = |v|$ דוגמאות פיקטיביות נקבל: $\hat{P}(x_i = v_i|k) \approx \frac{n_{i \cap k} + |v| \cdot \frac{1}{|v|}}{n_k + |v|} = \frac{n_{i \cap k} + 1}{n_k + |v|}$

למשל, מילים נדירות שקיימות במילון אך לא נמצאות כלל בקטגוריה k בסט הנתונים ישוערכו $\frac{1}{n_k + |v|}$.

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

<u>למידה בלתי מונחת:</u>
נלמד שתי שיטות של למידה שאינה מונחת עבור למידת מכונה.
<u>שיטת אשכולות Clustering</u>:

K-Means Clustering:



נתונה קבוצת דוגמאות D ממימד n. נתון מספר ה clusters (אשכולות) K.

- ✓ נחלק את סט הנתונים ל K קבוצות זרות המכסות את כל הסט
- ✓ נרצה כי הדימיון בין הנקודות בתוך כל cluster יהיה הגדול ביותר, או במילים אחרות, השוני (אי-הדימיון) בין הנקודות שבכל אשכול יהיה מינימלי

לשם כך, נזדקק:

- ✓ לפונקציה "אי דמיון" שמהווה מרחק בין וקטורים, למשל מרחק אוקלידי $d(x_{i,}, x_{i'})^2 = ||x_i - x_{i'}||^2 = \sum_{j=1}^{m}(x_{i,j} - x_{i',j})^2$
- ✓ למדד לשוני בין נקודות בתוך האשכול, למשל Within-Cluster-Variation, סכום הממוצעים של מרחקי נקודה משאר הנקודות בקלסטר $WCV(C_k) = \frac{1}{|C_k|}\sum_{i,i' \in C_k} d(x_{i,}, x_{i'})^2$
- ✓ לפונקציה שמוצאת אשכולות שהסכום הכולל של השוני WCV של כל אחד מהם הוא מינימלי, למשל $TWCV = \sum_{k=1}^{K} WCV(C_k)$

כמובן שנרצה להעזר באלגוריתם הקליסטור מבוסס על אופטימיזציה $minimize_{c_1 \dots c_k}\{TWCV\}$.

האלגוריתם הינו:

1. מצרפים רנדומית כל דוגמא לאחד מתוך K הקלסטרים
2. חוזרים שוב ושוב עד שאין שינוי בקלסטרים:
   א. מחשבים את המרכז לכל קלסטר
   ב. מצרפים מחדש את הנקודות על פי קירבתם למרכזי הקלסטרים.

מרכז הקלסטר הוא הווקטור של ממוצעי המאפיינים של הדוגמאות שבקלסטר.
אם נשתמש בנורמה d לחישוב המרחק בין 2 וקטורים, ניתן להוכיח כי בכל איטרציה, סכום המרחקים $TWCV$ אינו גדל ולכן בכל צעד שבו מעבירים דוגמא מקלאסטר לקלאסטר (מבצעים שינוי), משפרים.
אלגוריתם K-means מבצע ירידה בפונקצית המטרה WCV עד לקבלת מינימום מקומי. ניתן לראות שקיים קשר בין ה WCV לבין סכום המרחקים ממרכז הקלסטר:

המרחק של נקודה מהסנטרואיד

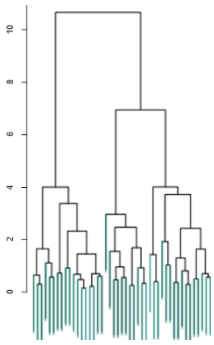
$$\frac{1}{|C_k|}\sum_{i,i' \in Ck}\sum_{j=1}^{m}(x_{ij} - x_{i'j})^2 = 2\sum_{i \in C_k}\sum_{j=1}^{m}(x_{ij} - \bar{x}_{kj})^2$$

בכל איטרציה המרחק בין הנקודות יורד מונוטונית. מחשבים סנטרואידים חדשים ואז מכניסים לקלסטר את הנקודות הקרובות לכל כל סנטרואיד. המרחק של הנקודות מהסנטרואיד יכול רק להשתפר- אם התווספה נקודה לקלסטר חדש הרי שהנקודה יצאה מקלסטר ישן ואז מרחקה ממרכז הקלסטר החדש קטן ממרחקה ממרכז הקלסטר הישן, ולכן ה WCV של הקלסטר החדש גדל פחות מההקטנה ב WCV שקרתה בקלסטר הישן. מתבצעת לכן ירידה בפונקצית המטרה TWCV עד למציאת מינימום מקומי.

פונקצית המטרה אינה קמורה בהכרח ולכן נקודת המינימום שנמצא אינה בהכרח מינימום גלובאלי, לכן רצוי לנסות את האלגוריתם כמה פעמים (התחלה רנדומית שונה) ולבחור WCV מינימלי.

חיסרון גדול של K-means: יש לקבוע מראש את מספר הקלסטרים. קשה מאוד לדעת כמה קלסטרים מראש נצטרך. הפתרון לבעיה זו הוא שימוש בהיררכיה של קלסטרים.

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין



קלאסטרים היררכים:

השיטה המקובלת היא היררכיהמלמטה למעלה: מתחילים כשכל דוגמא בסט הנתונים היא קלסטר (עלה בעץ)

בכל איטרציה מאחדים 2 קלסטרים שהם הכי קרובים עד שמקבלים בקלסטר אחד את כל הסט שוב.

ברגע שינו גרף Den do Grams נוכל ע"י חיתוך מלמעלה בגובה המתאים, לקבל כל מספר של קלסטרים שנרצה.

אלגוריתם Bottom-up לקליסטור היררכי:

נתונה פונקצית מרחק $d(x,y)$ בין 2 ווקטורים ובעזרתה נגדיר "אי דימיון" בין קלסטרים וכן נתונה $Inter-Cluster-DisSimilarity\ (C1,C2)$ המודד כמה 2 קלסטרים שונים זה מזה.

1. הכנס כל דוגמא $x \in D$ לקלסטר נפרד. חשב את מדד אי-הדימיון לכל $\binom{n}{2}$ זוגות הדוגמאות.
2. עבור $i = n, n-1, \dots 2$: (עד שכל הדוגמאות מאוחדות בקבוצה אחת)
   א. בחר זוג קלסטרים בעלי אי-דימיון (ICD) מינימלי בין הזוג ומזג אותם לקלסטר אחד
   ב. חשב את אי הדימיון (ICD) של הקלסטר הממוזג מול כל אחד מהקלסטרים שנותרו
   ג. גובה הקלסטר הממוזג בדנדוגרם הוא ה WCV(C) בין הדוגמאות ששויכו לקלסטר

$$WCV(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} ||x_{i,} - x_{i'}||^2$$

בכל איטרציה מספר הקלסטרים יורד באחד ואילו אי הדימיון בקלסטר האחרון שמוזג גדל.

ישנן כמה וריאציות כיצד נמדד אי דימיון בין קבוצות (כדי למזג):

1. מקסימום אי דימיון:

חשב את כל אי הדמיון עבור כל זוג d(x,y) בין x∈C1 ל-y∈C2 ותן כפלט את השונות המקסימלית.

2. ממוצע:

חשב את כל אי הדמיון עבור כל זוג d(x,y) בין x∈C1 ל-y∈C2 ותן כפלט את ממוצע השונות.

3. מינימום אי דמיון:

חשב את כל אי הדמיון עבור כל זוג d(x,y) בין x∈C1 ל-y∈C2 ותן כפלט את השונות המינימלית. נשים לב כי שיטה זו בעלת נטייה ליצור עץ לא מאוזן.

4. מרכוז:

תן כפלט את השונות (d(center(c1),center(c2) בין המרכזים של שני האשכולות. בדרך כלל מרכז אשכול הוא הנקודה הממוצעת. נשים לב כי שיטה זו בעלת נטייה לייצר בעיות בויזואליזציה.

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

1st Principle Component :
מיקסום השונות בו זמנית עם מזעור המרחק, כלומר, המרחקים האוקלידים של הנקודות מהציר מתמזערים באותו רגע בו מתמקסמת השונות על הציר.

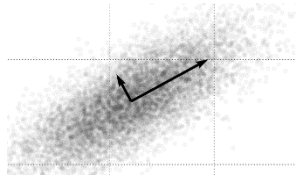
Principle Component Analysis באופן פורמלי :
נתון מדגם D של נתוני קלט n מימדי : $x = (x_1, x_2, \dots, x_D)$. מחפשים טרנספורמציות לינאריות
ששיעבירו את הקלט המקורי x ממימד n ל z בממימד m (קטן מ n) תוך שימור של המידע "החשוב".
הטרנספורמציות שמחפשים הם $\emptyset_1, \emptyset_2, \dots \emptyset_m$. וקטורים ממימד n ,
$z = (z_1, z_2, \dots, z_m)$ הם m קומבינציות לינאריות של n המאפיינים המקוריים של דוגמא x בעזרת הטרנספורמציות. z_i
הוא ההטלה של דוגמא x על הציר ה- i .

The score of the ith PC

$$z_i = \sum_{j=1}^{n} \emptyset_{j,i} x_j$$

$z = (z_1, z_2, \dots, z_m)$ הוא וקטור ההטלות של x על מערכת צירים חדשה בה יש m צירים אורתוגונלים.

נרצה למצוא m<<n טרנספורמציות $\emptyset_i$ שתשמרנה אינפורמציה "חשובה". כל אחת מ m הטרנספורמציות נקראת Principle Component. $\emptyset_i$ הוא הקומפוננט (הציר) ה i. Principle component הראשון הוא הכי משמעותי, ה PC השני הוא שני בחשיבותו וכן הלאה.



הצירים החדשים הם הוקטורים העצמיים של מטריצת ה Covar של D. הציר שנותן את השונות הכי גדולה בין ההטלות של הנקודות על הציר הוא ה Principle Component , הוקטור העצמי עם הערך העצמי הדול ביותר של מטריצת ה covar של הנקודות ב D. הווקטור עם הערך העצמי הבא הכי גדול, הוא הציר השני. מאונך לווקטור העצמי העיקרי.

כלומר, מחפשים טרנספורמציה לינארית למערכת צירים שממקסמת שונות.הכיוון העיקרי שממקסם את האינפרמציה הטמונה בנתונים- ממקסם את השונות בין נקודות וכן ההיפר מישור הכי קרוב לנקודות, ההתאמה הלינארית הכי טובה לקשר בין המאפיינים.

הטרנספורמציה z_1 של 1st PC תחשב את ההטלה של כל נקודה על הציר החדש.

אם נרצה לדחוס את 2 הקלטים למספר אחד, אוסף ההטלות על הציר של הרכיב הראשון, יתן שונות מקסימלית וגם יבטא קורלציה בין 2 המאפינים אם קימת (בדוגמא יש קשר לינארי רועש בין 2 המאפינים).

שינוי מערכת צירים :
1. הווקטורים מנורמלים מסביב לממוצע 0 (ע"י הפחתת הממוצע)
2. הקומפוננט הראשי הראשון הוא טרנספורמציה לינארית המתקבלת ע"י אלגוריתם הממקסם את השונות של ההטלות
3. שונות ההטלות על כל ציר אחר תהיה קטנה יותר
4. ברגע שנוסיף את הציר ההשני (והאחרון), לא נאבד שום אינפורמציה

ייצוג הרכיבים :
פירוש נוסף לרכיב הראשון הוא שדוחסים 2 מספרים למספר אחד שמיצג הכי טוב את 2 המספרים המקוריים.
למשל אם יש קשר לינארי בין 2 המאפינים, הרי המספר הדחוס לא יאבד שום אינפורמציה.

הרכיב השני z2 הוא קומבינציה לינארית של המאפינים המקוריים שאיננה קורלטיבית (אורתוגנלית) עם הרכיב הראשון ויש לה שונות מקסימלית (בכפוף לאילוץ האורתוגנליות).
ב 2 מימדים, הרכיב השני אינו תורם להפחתת המימד, אך במימדים גבוהים, יש לבחור ציר מאונך לציר הראשון כך שהשונות של ההטלות עליו ממוקסמת.

על פי הבניה, הרכיב הראשון מכיל את מקסימום האינפורמציה, הרכיב השני מכיל פחות אינפורמציה, הרכיב השלישי עוד פחות....וכך הלאה.

מרצה : ד"ר גדי פנקס
נכתב על ידי : יעלי בר גימלשטיין

PCA כבעיית אופטימיזציה:
ממקסמים variance (שקול למיזעור מרחק L2). ללא הגבלת כלליות נניח שהתוחלת של כל מאפיין היא 0. נוכל להסב כל משתנה מקורי כך שיהיה מסביב ל 0. מכיוון שישנן אין סוף צירים שקולים (למשל מכפלה בקבוע) נבקש שלצירים תהיה נורמה 1:

$$\sum_{j=1}^{n} \emptyset_{j,p}{}^{2} = 1$$

זו גם בעיית מיקסום, מציאת ה PC הראשון.
ב D יש m דוגמאות שנורמלו מסביב ל 0. כיצד נמצא את ה- PC הראשון: $\emptyset_1 = (\emptyset_{1,1} \ \dots \emptyset_{n,1})$
שיבצע טרנספורמציה על ווקטורים $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ בסט הנתונים.

- $z_{i,1} = \sum_{j=1}^{n} \emptyset_{j,1}\, x_{i,j}$ the scores of the 1st PC
- $Maximize_{\ \emptyset_1} \left\{ \frac{1}{m} \sum_{i=1}^{m} (z_{i,1})^2 \right\}$ מיקסום הוואריאנס
- $subject\ to\ \sum_{j=1}^{n} \emptyset_{j,1}{}^{2} = 1$

הבעיה ניתנת לפיתרון ע"י פירוק מטריצת הקו-וריאנס של D לווקטורים עצמיים . ה PC הראשון הוא הוקטור העצמי עם הערך העצמי הכי גבוה.

אם ניקח את מטריצת ה co-variance בין כל הנקודות ונחשב eigenvectors,נמצא מערכת צירים אורתוגונית שממקסמת את ה variance . הווקטור עם השונות הכי גדולה, הוא הווקטור העצמי עם ערך העצמי הגדול ביותר, הווקטור עם הערך העצמי הבא הכי גדול, הוא הציר השני.

מבט נוסף:
אלגוריתם PCA מבצע דחיסה באמצעות קומבינציות לינאריות שממקסמות את יכולת שיחזור הנקודות ע"י מזעור ריבוע ההפרשים (MSE) בין נקודה לשיחזור שלה. השיחזור הוא שוב ע"י מכפלה במטריצה . הPC הראשון הוא ישר שממזער את ריבועי המרחקים של הנקודות לקו. ה PC הראשון והשני פורשים מישור שממזער את ריבוע מרחקי הנקודות מהמישור.
המרחב הנפרש ע"י m הPC הראשונים ממזער את ריבוע מרחקי הנקודות מהמרחב שנפרש. כלומר, המישור הנפרש ע"י 2 ה PC הראשונים ממזער את סכום ריבועי המרחקים של הנקודות מהמישור.

נרמול הנתונים:
בד"כ עבור PCA ממרכזים את המאפינים מסביב ל 0. נירמול כזה ניתן לעשות ע"י הפחתת הממוצע של כל מאפיין. אין חובה לנרמל גם את סטיית התקן. אבל התוצאות של PCA יהיו מאוד תלויות בScale של המאפינים: אם השונות של אחד המאפינים גבוהה במיוחד, PCA ישקלל מאפיין זו יותר מאחרים ויתן תוצאות שונות לפני/אחרי scaling (הכפלה במספר) להבדיל מרגרסיה לינארית שאיננה רגישה להכפלות.

כדי ש PCA לא יהיה תלוי שרירותית בסקאלת המדידה, נהוג ללבצע Scaling כדי ליצור סטית תקן 1.
אבל במידה ו 2 מאפיינים נמדדים באותה סקאלה, יתכן מאוד ולא נרצה לבצע scaling שונה לכל אחד וזאת כדי שההבדלים ב scale יבואו לידי ביטוי.

אין תשובה נכונה לכמה מימדים נרצה להפחית את המידע, אלא מספיק כדי לשמר אינפורמציה חשובה. עבור ויזואליזציה נרצה 2-3 מימדים.

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

עצי החלטה ויער רנדומי:

האלגוריתם ניתן להכללה גם עבור קלסיפיקציה שאיננה בינארית וכן עבור רגרסיה.

קריטריוני עצירה אפשריים:

1. כאשר ההפרדה מושלמת (תוויות הומוגניות – לא ניתן לפצל יותר)
2. לעיתים לא נוכל להגיע להפרדה מושלמת ואז נעצור אם לא ניתן לשפר את ההומוגניות
3. קריטריונים אחרים – (כמו גודל העץ) - על מנת למנוע התאמת יתר- נעצור כאשר העץ גדול/עמוק מידי

באופו פורמלי:

נתונה קבוצה S של דוגמאות עם מאפיינים בינארים ותווית מטרה בינארית $y = 0/1$.

Grow Tree(S):

all $< x, t > \epsilon S$, return (new leaf(0))

Else if (t=1) for all $< x, t > \epsilon S$, return (new leaf(1))

Else

choose "best" feature x_j

$S_0 = all < x, t > \epsilon S \; with \; x_j = 0$

$S_1 = all < x, t > \epsilon S \; with \; x_j = 1$

return (new node ($x_j < 0.5$,Grow Tree(S_0), Grow Tree(S_1))

if (t=0) for

אם הקבוצה היא הומגנית מבחינת y, ניצור ממנה עלה, ונסמנו על פי התווית. סיימנו

אחרת (S איננה הומגנית), נבחר מאפיין שיפצל "הכי טוב" לשתי קבוצות נפרדות. כמה שיותר הומוגניות ב y

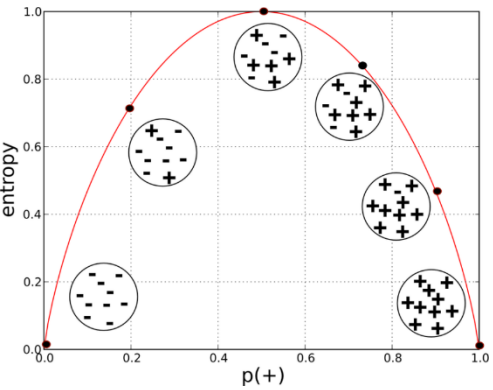
נחזיר תת עץ שהשורש שלו הוא השאלה שנבחרה כדי לפצל, ויש לו 2 ילדים שבצורה רקורסיבית ממשיכים להתפצל לעוד קבוצות יותר ויותר הומוגניות עד שהקבוצות "מספיק" הומוגניות או שלא ניתן יותר לפצל

בחירת היוריסטיקה - אנטרופיה:

בהינתן משתנה אקראי V (בינארי או לא), האנטרופיה שלו היא:

$$H(V) = \sum_{v=0}^{1} -P(V = v) \log P(V = v)$$

זה הוא ההפתעה הממוצעת בתיאור המשתנה. במילים אחרות, מדד לאי ודאות.



כאשר המשתנה המקרי הוא בוליאני האנטרופיה מקסימלית כאשר הסיכוי $V = 1$ הוא 0.5. האנטרופיה מינימלית כאשר הסיכוי הוא 1 או 0 (קבוצה הומגנית). כאשר יש K ערכים, האנטרפיה המקסימלית היא $log(\frac{1}{k})$.

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

דוגמה לחישוב אנטרופיה ובחירת מאפיינים על סמך אנטרופיה:
נבנה עץ החלטה עבור ההיפותזה האם לקוח יקבל הלוואה מהבנק. בהינתן המידע הבא:

Name	age	gender	Balance ($)	Employed	Default
Mike	42	M	200,000	Yes	No
John	37	M	35,000	No	Yes
Mary	40	F	115,000	No	No
Robert	23	M	72,000	Yes	No
Dora	31	F	29,000	No	Yes

האנטרופיה של חיזוי ברירת המחדל הוא:

$$H(default) = -P_{yes}\log P_{yes} - P_{no}\log P_{no} = -\frac{2}{5}\log\frac{2}{5} - \frac{3}{5}\log\frac{3}{5} \approx 0.97$$

Employed	Default
Yes	No
No	Yes
No	No
Yes	No
No	Yes

כעת, הוחלט כי שורש עץ ההחלטה יהיה מאפיין $employed$. נחשב מחדש את האנטרופיה.
מתוך כל מי שהינו מועסק, אף אחד לא יקבל הלוואה, לכן:

$$H(D, E = yes) = -\frac{0}{2}\log\frac{0}{2} - \frac{2}{2}\log\frac{2}{2} = 0$$

מתוך כל מי שאינו מועסק, שניים יקבלו הלוואה ואחד לא יקבל, לכן:

$$H(D, E = no) = -\frac{1}{3}\log\frac{1}{3} - \frac{2}{3}\log\frac{2}{3} = 0.92$$

לפני הפיצול האנטרופיה הייתה 0.97 ואחרי, קיבלנו שתי אנטרופיות 0.92, 0 . נשאלת השאלה האם המאפיין הנבחר הוא מאפיין טוב לפיצול. לא ברור. לשם כך נעזרת באנטרופיה משוכללת.

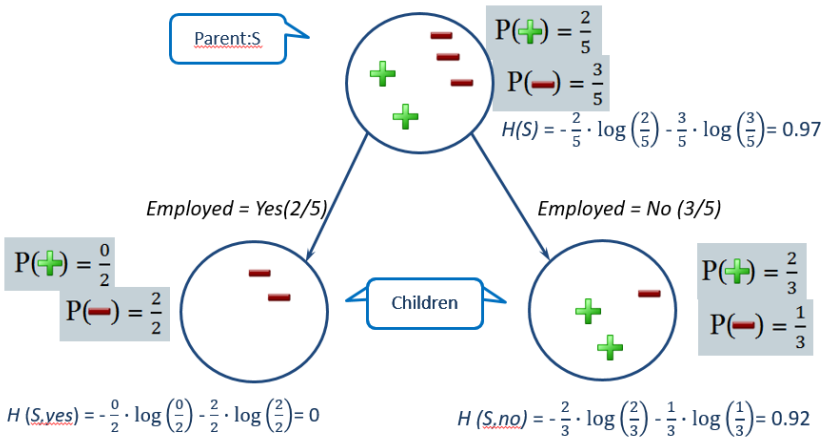
אנטרופיה משוכללת:

המאפיין שנבחר הוא זה שגורם לירידה הגדולה ביותר באנטרופיה המשוכללת.
האנטרופיה המשקללת של מספר קבוצות היא סכום האנטרופיות המשוקללות עי ההסתברות של דוגמא להגיע לכל אחת מקבוצות

$$\sum_{i=1}^{\#children} P(child_i)\cdot H(child_i)$$

את ההסתברות של $P(child_i)$ משערכים באופן הבא:

$$P(child_i) = \frac{|child|}{|parent|}$$



אם נחזור לדוגמה, כעת, נוכל לחשב את האנטרופיה המשוכללת:

$$P(D, E = yes)\cdot H(D, E = yes) + P(D, E = no)\cdot H(D, E = no) = \frac{2}{5}\cdot 0 + \frac{3}{5}\cdot 0.92 = 0.552$$

נשים לב כי חישוב זה הוא בעצם לכפול את האנטרופיה בהסתברות של ההסתעפות בעץ.

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

Information gain = רווח המידע:

$$IG(Parent, children) = H(parent) - \sum_{i=1}^{\#children} P(child_i) \cdot H(child_i)$$

רווח המידע תלוי בשני גורמים- באנטרופיה של הבנים $H(child_i)$, וכן כמה הם דומיננטים $P(child_i)$.

דוגמה עבור אנטרופיה משוכללת ועץ החלטה עבור משתנים רב קטגוריים- האם נשחק טניס:

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

TABLE 3.2
Training examples for the target concept *PlayTennis*.

נבדוק מה יהיה רווח המידע אם נחליט לפצל לפי $wind = \{strong, weak\}$.

נספור ונראה כי:

$$S = \{9\ yes, 5\ no\}$$
$$S_{weak} = \{6\ yes, 2\ no\}$$
$$S_{strong} = \{3\ yes, 3\ no\}$$

ואז נוכל לחשב את רווח המידע:

$$H(S) = -\frac{9}{14}\log\frac{9}{14} - \frac{5}{14}\log\frac{5}{14} = 0.94$$
$$H(S_{weak}) = -\frac{6}{8}\log\frac{6}{8} - \frac{2}{8}\log\frac{2}{8} = 0.811$$
$$H(S_{strong}) = -\frac{3}{6}\log\frac{3}{6} - \frac{3}{6}\log\frac{3}{6} = 1$$
$$IG = 0.94 - \frac{8}{14}H(S_{weak}) - \frac{6}{14}H(S_{strong}) = 0.048$$

נוכל באופן זה להחליט מי המפצל הטוב ביותר של S. נבדוק את רווח המידע עבור כל אחד מהמאפיינים ונקבל:

$IG_{temp} = 0.029$ | $IG_{humidity} = 0.151$ | $IG_{outlook} = 0.246$ | $IG_{wind} = 0.048$

ההסתברות (וגם score) לעלה בעץ החלטה:

עבור עץ החלטה לקלסיפיקציה, לכל עלה נקבעת:

1. קטגורית העלה (ע"פ רוב הדוגמאות שבעלה)
2. ההסתברות לגילוי נכון Accuracy של העלה (ע"פ ספירת מספר הדוגמאות החיוביות מתוך סה"כ הדוגמאות שבעלה)

רצוי להחליק את ההסתברות המחושבת כדי להימנע מהסתברויות של 0 או 1. אם אין מספיק דוגמאות וחוששים מהסתברויות 0 או 1, מבצעים תיקונים של ההסתברות, למשל, בעזרת Laplace Smoothening

ואפשר כמובן לקבל את h או G על העלה ולהשתמש במדד לציון או להסתברות.

עצי רגרסיה:

עד כה התעסקנו עם קלסיפיקציה בעצים. ניתן לאפשר רגרסיה בעץ באופן הבא:

- ✓ ממוצע התוויות של הדוגמאות הנופלות בקבוצה (צומת) נלקח כחיזוי y של הקבוצה. כל עלה, לכן, חוזה את הערך הנומרי הממוצע של הדוגמאות שבעלה.
- ✓ $sum\ squared\ error$ של הדוגמאות בקבוצה נלקח כמדד לשיפור (במקום אנטרופיה). כלומר, סכום ריבועי ההפרשים בין התווית לממוצע של הדוגמאות בצומת יהיה מדד השיפור.

הרווח של כל פיצול מחושב ע"פ השיפור במדד SSE בעקבות הפיצול.

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

היוריסטיקות אלטרנטיביות לבחירת מאפיין לפיצול- מדד $Genie$:
ממד זה דומה למדד האנטרופיה וגם הוא מודד אי סדר.
עבור משתנה אקראי בינארי, המדד יחושב כך $G = \sum_{i=1}^{N} p_i \cdot (1 - p_i)$.

כאשר הקבוצות הומוגיות, $G = H = 0$.
בשונה ממד האנטרופיה, כאשר קבוצה היא הטרוגנית, $\max G = \frac{1}{2}$.

שיטות להפחת שגיאת השונות:
1. עצירה מוקדמת - מפסיקים להצמיח עלים לפני שהגענו לעלים הומוגניים
   - כאשר השיפור באנטרופיה אינו משמעותי
   - כאשר השיפור בשגיאת הווליידציה איננו משמעותי
   - רגולריזציה Complexity: הענשת השיפור באנטרופיה לעצים גדולים או צמתים עמוקים למשל ע"י חלוקת ה IG בגודל העץ: InfoGain/TreeSize.
2. Ensemble: Bagging, Boosting, Random Forest

Bagging using Bootstrap Aggregation=
נבצע דגימה עם חזרות (=bootstrapping) כדי לגדל B עצים שונים המורכבים B מדגמים שונים קצת זה מזה. נמצע את תוצאות החיזוי של העצים השונים:
1. ברגרסיה ממצעים את החיזוי
2. בקלסיפיקציה משתמשים בהחלטת הרוב או מיצוע ההסתברות של העלה אליו הגענו בכל אחד מהעצים

חסרון בשיטה זו הוא שהעצים דומים מידי אחד לשני ולכן שגיאת השונות אינה יורדת משמעותית
מכיוון שדגימות ה bootstrapping הינן קורלטיביות והצמתים (המשמעותיים) בסמוך לשורש, נשארים לרוב קבועים במקומם בעץ.

יערות רנדומיים random forests=
1. נבנה עץ לכל מדגם בשיטת bootstrap
2. בכל פעם שבוחרים צומת לפיצול, נגביל את המאפיינים לבדיקה ל $m' < m$ מאפיינים הנבחרים רנדומית מתוך m המאפינים שלרשותנו. כלומר, אם $m' \ll m$, בכל פעם שבוחרים צומת לפיצול, האלגוריתם לא מורשה להסתכל על רוב המאפינים

העצים שנגדל יהיו שונים, גם בגלל שנוצרו מקבוצות אימון שונות במקצת ובעיקר בגלל שאולצו להשתמש במאפינים שונים.

ללא מגבלה על בחירת המאפיינים, ובהנחה שקבוצות האימון דומות, סביר שברוב העצים היה נבחר אותו מאפיין (דומיננטי ברווח המידע) עבור השורש. לעומת זאת, ביער רנדומי, המאפיין ה"חזק" ביותר לא ילקח בחשבון בסיכוי של $\frac{m-m'}{m}$.
באלגוריתם Random Forest העצים פחות קורלטיביים ככל $m' \ll m$.

<u>Choose Best Feature (S, Feature set, m'):</u>
```
If Feature set is empty return (Null,Null)
Randomly select m' features from Feature set (if |Feature set| < m', select all features in Feature set.
Find the feature Best Feature with the best IG
        remove from Feature set,
        return(Best Feature, Feature set)
```

להלן האלגוריתם המלא לבניית עץ בינארי אקראי:

<u>Grow Random Best Tree (S, Feature set, m'):</u>
```
If t= 0 for all <x,t> in S, new leaf(0)
Else if t=1 for all <x,t> in S , new leaf(1)
Else
  Best Feature, Feature set =Choose Best Feature(S, Feature set, m')
  if Best Feature ==Null, return
  S0=all <x,t> in S, where XBest=0
  S1=all <x,t> in S, where XBest=1
Return New node(XBest,Grow Random Bset Tree(S0, Fset, m'), Grow Random Best Tree(S1, Fset, m'))
```
קל להכליל עבור מאפיינים קטגוריים ונומרים.

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

SVM = Support Vector Machines:

אלגוריתם SVM הוא בעצם קלסיפייר בינארי. לשם הבנת האלגוריתם לעומק נבדיל בין מספר מושגים:

1. Maximum Margin Classifier =
קלסיפייר בינארי לינארי על בסיס מיקסום המרווח בין קטגוריות. דורש כי הנתונים יהיו ניתנים להפרדה לינארית (בדומה לפספטרון).
2. Support Vector Classifier =
הרחבה לקלסיפייר בינארי לינארי שאינו דורש הפרדה לינארית מוחלטת ומאפשר גם חריגים (outliers). בנוסף, ישנו פרמטר $capacity$ המאפשר שליטה על Bias – Variance tradeoff.
3. Support Vector Machines =
הרחבה לקלסיפייר בינארי **לא** לינארי על ידי טריק הגרעין (=kernel).

Maximum Margin Classifier:

בהינתן קבוצת אימון D של דוגמאות מסווגות בינארית $\{+1,-1\}$ ממימד n, הניתנת להפרדה לינארית, רוצים למצוא היפר-מישור ממימד $n-1$ המפריד בין הדוגמאות.

במימד n, ההיפר מישור המפריד הוא אילוץ מהצורה:

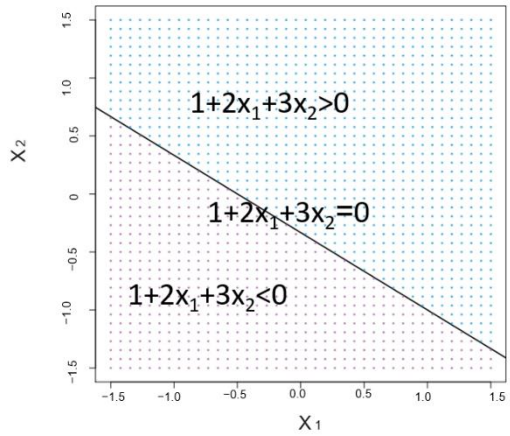
$$h(x) = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n = 0$$

כל נקודה שמקיימת את האילוץ, היא נקודה על המישור המפריד:

- ✓ נקודות מעל למישור יסווגו 1: $w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n > 0$
- ✓ נקודות מתחת למישור יסווגו 0: $w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n < 0$

הסיווג של X הוא לכן: $sign(w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n)$.

דוגמא= בהינתן היפר מישור $1 + 2x_1 + 3x_2 = 0$ הסיווג של כלל הנקודות יראה:



היפר מישור מפריד לינארי:

בהינתן קבוצת אימון D של דוגמאות $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$ מסווגות בינארית $y_i \in \{-1,+1\}$ הניתנת להפרדה לינארית, למישור המפריד יש את התכונה:

$$y_ih(x_i) = y_i(w_0 + w_1x_{i1} + w_2x_{i2} + \cdots + w_nx_{in}) > 0$$

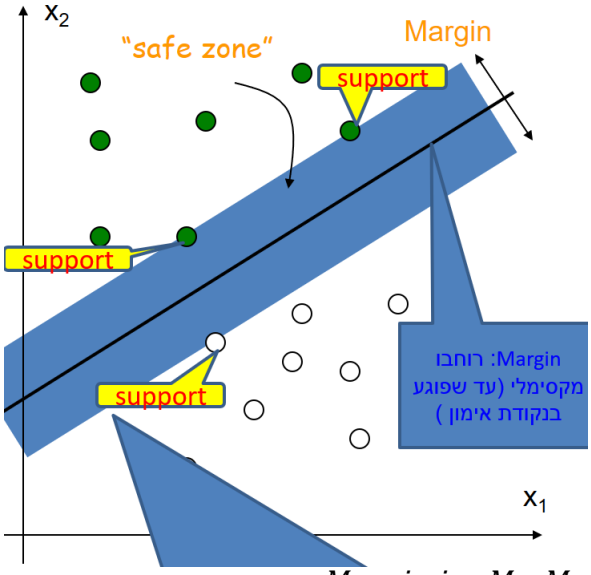
המרחק של נקודה x מהמישור, נותן אינדיקציה לרמת הבטחון בסיווג:

$$\frac{y_i(w_0 + w_1x_{i1} + w_2x_{i2} + \cdots + w_nx_{in})}{\sqrt{\sum_{j=1}^{n} w_j^2}}$$

ישנם אינסוף מישורים המקיימים את האילוצים (לכל נקודה). נשאלת השאלה באיזה מישור נבחר.

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

למשל, בקלסיפיקציה, בחרנו את מישור מפריד שממזער את CE .



עבור SVM נפעיל לוגיקה אחרת:
המישור המפריד בעל השוליים (= Margin) המקסימליים, הוא הטוב ביותר, היות ויש לו יכולת הכללה טובה יותר עבור נקודות מבחן המתקרבות למישור המפריד. בנוסף הוא אינו רגיש למיקום רוב הנקודות שמחוץ לשוליים ולכן פחות התאמת יתר.השוליים מוגדרים כרוחב שבו ניתן להרחיב את ההיפר-מישור המפריד לפני שנתקל בנקודת אימון.

כאשר ממקסמים את השוליים, רק היפר- מישור אחד יכול להיות עם שוליים מקסימליים.

- ✓ אם נקודות האימון מחוץ לשוליים יזוזו, המישור המפריד לא יושפע
- ✓ אם נקודות האימון התומכות יזוזו, המישור המפריד ישתנה

נדמה את מישור ההפרדה (כולל השוליים) כשרוול / צינור ממימד n .

בניית Max Margin Classifier כבעית אופטימיזציה קוודרטית:

$$Maxsimize\ M \quad M,\ w_0, w_1, w_2, \dots, w_n$$
$$subject\ to$$
$$1)\ y_i\ (w_0 + w_1 x_{i1} + w_2 x_{i2} + \cdots + w_n x_{in}) \geq M \quad for\ every \quad (x_i, y_i) \in D$$
$$2)\ \sum_{j=1}^{n} {w_j}^2 = 1$$

משמעות התנאי השני הוא לבחור מבין כל המישורים, מישור כזה עם נורמל היחידה, כלומר המישור מקבל משמעות של מרחק הנקודה מהמישור. הבעיה מנוסחת כך שהפתרון הוא גלובלי ויחיד. כשנקודות אינן ניתנות להפרדה לינארית, אין פיתרון לבעיית האופטימיזציה, הקלסיפייר הזה פשוט לא קיים.

תוספת של נקודה אחת עלולה לגרום לתזוזה דרסטית של המישור המפריד ולשוליים צרים מאוד. זוהי אינדיקציה להתאמת יתר. שוליים צרים מורידים את הבטחון (=המרחק מהמישור) בסיווג של נקודות התמך.

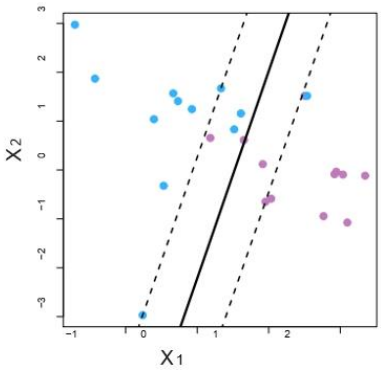
חסרונות בשיטה זו:

1. אם הנתונים לא ניתנים להפרדה לינארית, אין פתרון לבעית האופטימיזציה.
2. הגדרת הבעיה אינה מאפשרת חריגים outliers בכלל.
3. גם אם ניתן להפריד, ישנה רגישות גבוה לנקודות שבחזית (בסמוך למישור המפריד). שינוי במיקום וקטורי התמיכה, כמו תוספת נקודות בגבולות השוליים, יכול לגרום לשינוי דרסטי במישור המפריד ולשוליים צרים מאוד.

Support Vector Classifier:
קלסיפייר שכזה אינו דורש הפרדה מלאה של דוגמאות האימון. כלומר, נחפש soft margin classifier שאינו מכריח כל דוגמא להיות מעבר לשוליים, וזאת על ידי כך שנאפשר שמיעוט דוגמאות תוכלנה להופיע בצד הלא נכון של השוליים מבלי להצר אותו וכן, מיעוט דוגמאות תוכלנה אפילו להופיע בצד הלא נכון של מישור ההפרדה מבלי שישפיעו על מישור ההפרדה ואפילו על השוליים (כלומר, לא נדרוש הפרדה לינארית מושלמת).

על מנת לבצע קלסיפיקציה טובה בנוכחות חריגם ורעש, רצוי לאפשר פרדיקציה שגויה לפעמים. כך נרוויח:
שוליים רחבים יחד עם קלסיפיקציה טובה של רוב דוגמאות האימון, סבילות איתנה לחריגים בודדים ונוכל לאפשר מספר רב יותר של נקודות תמיכה. התוצאה: פחות רגישות לתזוזות ונקודות חדשות.

מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין



הקלסיפייר ה"רך" יסווג את רוב הדוגמאות ברמת ביטחון גבוה (מחוץ לשוליים הרחבים),חלק מן הדוגמאות יסווגו נכון אבל בתוך השוליים ואילו חלק מהדוגמאות יסווגו באופן שגוי.

על כן, יש לנסח מחדש את בעית האופטימיזציה – רוצים למקסם את השוליים אבל עבור שוליים רחבים מוכנים מוכנים "לשלם" שמיעוט נקודות יכנסו לתוך הצינור ואפילו יסווגו כשגויים. הפתרון הוא להוסיף משתנים לבעית האופטימיזציה.

נוסיף משתנה רפיון $Slack\ variable = \varepsilon_i$ לכל דוגמה, אשר מאפשר לה להופיע בצד הלא נכון, כאשר ε_i הוא התשלום שמשלים עבור חריגה מהשוליים. כעת, נרצה למעזר את סכום התשלומים.

- ✓ כאשר $\varepsilon_i = 0$ הדוגמא , תהיה מחוץ לשוליים ותסווג נכון
- ✓ כאשר $0 < \varepsilon_i \leq 1$ הדוגמא תהיה בצד הלא נכון של השוליים אבל בצד הנכון של המישור ותסווג נכון
- ✓ כאשר $\varepsilon_i > 1$ הדוגמא תהיה בצד הלא נכון של המישור המפריד ותסווג שגוי

יהיה לנו תקציב C שאנחנו מוכנים לשלם עבור הנקודות החורגות הללו, ככל שהחריגה גדולה יותר, התשלום ε_i גדול יותר. נרצה להרחיב את השוליים כמה שיותר, אך ובשום פנים לא לחרוג מהתקציב.

הבעיה תנוסח, אם כך, באופן הבא:

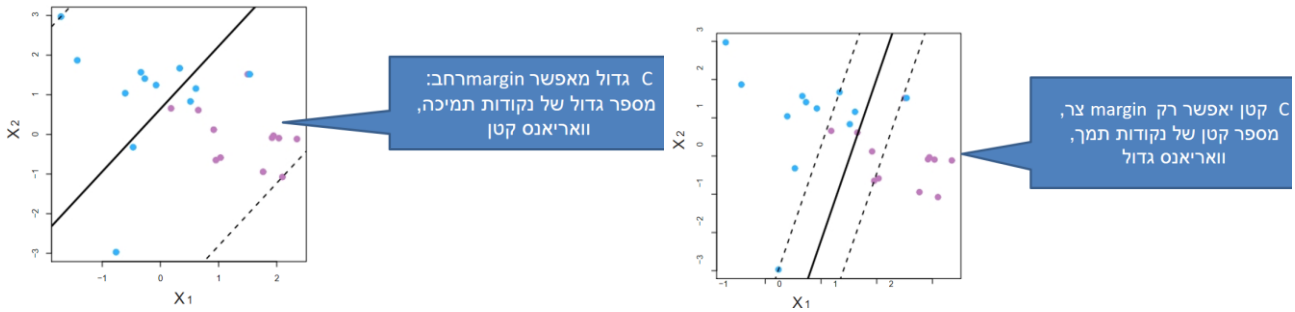
$$Maxsimize\ M \quad M,\ w_0, w_1, w_2, \dots, w_n, \varepsilon_i, \dots, \varepsilon_m$$
$$subject\ to:\ for\ every\ \ (x_i, y_i) \in D$$
$$1)\ y_i\ (w_0 + w_1 x_{i1} + w_2 x_{i2} + \cdots + w_n x_{in}) \geq M(1 - \varepsilon_i)$$
$$2)\ \varepsilon_i \geq 0$$
$$3) \sum_{i=1}^{m} \varepsilon_i < C$$
$$4) \sum_{j=1}^{n} {w_j}^2 = 1$$

בתנאי הראשון, המכפלה $M(1 - \varepsilon_i)$ מקטינה בפועל את השוליים ואפילו הופכת אותם לשליליים אם הדוגמה בצד הלא נכון של המישור.

C הינו היפר פרמטר התקציב המגביל את סכום החריגות מהשוליים M . הוא גם שולט Bias – Variance tradeoff . לא יתאפשרו יותר מאשר C חריגות מהמישור המפריד $\varepsilon_i > 1$:

- ✓ $C = 0$ משמע אין כלל תקציב, כל הדוגמאות צריכות להימצא מהצד הנכון של השוליים. ועל כן חייב להיות $\varepsilon_i = 0$, אם D אינה ניתנת להפרדה לינארית לא יהיה פיתרון
- ✓ נקודה בצד הנכון של המישור אבל בתוך השוליים תיקח מהתקציב $0 < \varepsilon_i < 1$ בהתאם למרחקה מהמישור
- ✓ נקודה בצד הלא נכון של המישור תיקח מהתקציב נתח גדול יותר $\varepsilon_i > 1$ ותסווג לא נכון

תקציב גדול, מאפשר להרבה נקודות לחרוג מהשוליים וכל נקודה שחורגת היא וקטור תמך. כשיש הרבה וקטורי תמך יש פחות רגישות לתזוזות בנקודות אלו, או להוספת נקודות תמך חדשות (שגיאת שונות קטנה יותר). מצד שני, תקציב גדול מאפשר סיווג לא נכון של דוגמאות אימון ולכן עלול להעלות את שגיאת ביאס.



מרצה: ד"ר גדי פנקס
נכתב על ידי: יעלי בר גימלשטיין

במילים אחרות, התקציב $capacity = c$ הינו היפר פארמטר הקובע כמה נתפשר בדוגמאות המפרות את השוליים על מנת לאפשר שוליים רחבים

- ✓ תקציב מאפשר שוליים רחבים הבנויים על הרבה וקטורי תמך למרות שיהיו הרבה דוגמאות שיפרו את השוליים. מספר וקטורי התמך גדול ולפיכך פחות רגישות לשינויים בוקטורי התמך (פחות שגיאת שונות).
- ✓ תקציב קטן, יצור שוליים צרים, הבנויים על מעט וקטורי תמך. גבול ההפרדה יהיה רגיש יותר לחריגים.כל שינוי בוקטורי התמך ישפיע על המישור המפריד ויוביל להתאמת יתר.

העובדה שרוב הנקודות נמצאות מחוץ לשוליין ולכן אינן משפיעות על קביעת המישור נותנת יתרון בהורדת שונות מול שיטות הרגישות לכל הנקודות כמו עצים. ברגרסיה לוגיסטית ניתן להראות שמצב דומה מתקיים וגם שם נקודות רחוקות כמעט ולא משפיעות על גבול ההחלטה וכנ"ל לגבי KNN : נקודות רחוקות אינן משפיעות. יש דימיון רב גם בהתנהגות האמפירית והתיאורטית של 3 השיטות : KNN, SVM ורגרסיה לוגיסטית.

טריק הגרעין The Kernel Trick :

ההיפותזה :

$$h(x) = \alpha_0 + \sum_{i=1}^{m} \alpha_i \langle x, x_i \rangle$$

מכפלה פנימית היא סוג של פונקצית דמיון בין שתי נקודות, לכן נוכל להחליף אותה בהכללה שלה :

$$h(x) = \alpha_0 + \sum_{i=1}^{m} \alpha_i K(x, x_i)$$

גרעין kernel = פונקציה המכמתת דמיון בין שני וקטורים. הפונקציה צריכה לקיים תכונות מתימטיות של גרעין ואולם לצרכים פרקטיים כמעט כל פונקצית דימיון תהיה בעלת תכונות אלו. קיימים המון סוגי גרעינים, אלו הנפוצים ביותר :

1. גרעין ליניארי, דמיון קורלטיבי

$$Linear: K(x, x_i) = x \cdot x_i$$

2. גרעין פולינומיאלי, גם בצורתו הפשוטה ביותר $d = 2$, נותן הרבה כוח

$$Polynomial: K(x, x_i) = (x \cdot x_i)^d$$

3. גרעין גאוסיאני, נותן דמיון בעל צורת פעמון. הווקטורים דומים אם מרחקם האוקלידי קטן. ככל שהסיגמה קטנה, הגאוסיאן צר, ואילו ככל שהסיגמה גדולה, הגאוסיאן רחב

$$Gaussian: K(x, x_i) = e^{\left(-\frac{\frac{1}{2}\|x - x_i\|}{\sigma}\right)}$$

דמיון בין אלגוריתם SVM לאלגוריתמים אחרים:

1. דומה לרגרסיה לוגיסטית עם רגולריזצית $ridge$
2. כשמשתמשים בגרעין גאוסי מקבלים קירוב לאלגוריתם KNN

תנאים לשימוש באלגוריתם זה:

אלגוריתם SVM פופולרי במיוחד כאשר :

1. הבעיה ממימד גבוה
2. יש מעט נתונים וחוששים משונות גבוהה
3. רוצים לבדוק מהר גבולות החלטה לא לינארים, זמינות גבוהה של Kernel Trick

אבל קיימות בעיות ביצועים כאשר D גדול מאוד. מטריצת ערכי K (הגרעינים) היא ריבועית במספר הדוגמאות m. אם כי, ישנן ווריאציות עבור קרנלים מסוימים שהמטריצה K לא תהיה ענקית.

אלגוריתם SVM ליותר משתי קטגוריות:

האלגוריתם נבנה עבור קלסיפיקציה בינארית ולא ניתן להרחבה בקלות ליותר מאשר 2 קטגוריות. הדרכים בו ניתן להרחיב הן :

1. One VS One= בנה $\frac{k(k-1)}{2}$ קלסיפיירים, אחד לכל זוג קטגוריות, בהינתן דוגמא x הפעל את כל הקלסיפייר. פלט החיזוי : הקטגוריה בעלת רוב החיזויים
2. One VS. Rest= בנה k קלסיפיירים שמבדילים אם קלט x שייך לקטגוריה i או לכל יתר הקטגוריות. פלט החיזוי : הקלסיפייר שנותן את הביטחון המקסימלי לקטגוריה שלו : $h_i(x)$ המקסימלי.

מרצה : ד"ר גדי פנקס
נכתב על ידי : יעלי בר גימלשטיין