אלגוריתם Gradient Decent וחוק עדכון המשקולות:

השגיאה היא פונקציה תלוית משקולות. על מנת למזער אותה עלינו להגיע לנקודת המינימום שלה וכך נקבל את המשקולות שמביאים את המינימום.

1. התחל ממשקולות אקראיים קטנים ובצע epochs שוב ושוב עד לקיום תנאי עצירה (השגיאה לא מפסיקה לרדת או מס' איטרציות):
   1.1 בכל epoch עדכן את המשקולות באופן הבא:

$$\Delta w_i = -\lambda \frac{\partial loss(w)}{\partial w_i}$$
$$w_i = w_i + \Delta w_i$$

היפר פרמטר, קבוע הלמידה λ. קיים גם בגרסת online שבו עדכון המשקולות אחרי כל דוגמא.

רגרסיה לינארית:

מודל פרמטרי לשיערוך היפותזה בעזרת משערך לינארי (קו או מישור ישר).

עבור n מאפיינים, ההיפותזה נראת כך:

$$y = wx = w_0 \cdot 1 + w_1 \cdot x_1 + \cdots + w_n x_n$$

השגיאה נראת כך:

$$loss = MSE = \frac{1}{n}\sum_{i \in D}(t_i - y_i)^2$$

מציאת המשקולות:

ישנן 2 דרכים למצוא את המשקולות כך שימזערו את השגיאה:

1. המשוואה הנורמלית $w = (x^T x)^{-1} x^T t$ שיטה מדויקת אך מתאימה למימדים קטנים בלבד
2. אלגוריתם GD כפי שתואר מעלה. לשם כך, הגזירה היא

$$\frac{\partial MSE}{\partial w} = \frac{\partial MSE}{\partial y}\frac{\partial y}{\partial w} = \frac{1}{n}\sum_{i \in D}(t_i - y_i) \cdot x_i$$

אם סט האימונים קטן נעדיף את המשוואה הנורמלית, אחרת את אלגוריתם GD.

נרמול שדות:

פונקצית השגיאה היא קערה, על מנת לוודא שהיא סימטרית יש לנמרל את ערכי המאפיינים:

1. Min max scaling = $\frac{x-min}{range}$ טווח ערכים
2. Mean normalization = $x - mean(x)$
3. SD scaling = $\frac{x}{sd(x)}$ סטיית תקן

רגרסיה פולינומיאלית:

הוספת מאפיינים שהם טרנספורמציות של מאפיינים קיימים (בודד או מורכב). התאמת ההיפותזה = נוסיף מאפיינים לפי דרגת הפולינום שבו מתעניינים להיפותזה (ניתן להתאים פולינומים מכל דרגה) על ידי הוספת כלל האפשרויות המתאימות. לכל מאפיין חדש משקל חדש.

רגרסיה לוגיסטית: (קלסיפיקציה)

מודל פרמטרי לסיווג קטגוריות (קלסיפיקציה) בעזרת קלסיפייר (יוצר גבול הפרדה בין קטגוריות).

עבור n מאפיינים, ההיפותזה נראת כך:

$$z = wx = w_0 \cdot 1 + w_1 \cdot x_1 + \cdots + w_n x_n$$
$$y = g(z) = \frac{1}{1 + e^{-z}}$$

השגיאה נראת כך:

$$loss = CE = -t \cdot \log(y) - (1 - t)\log(1 - y)$$

$$C(y,t) = \begin{cases} -\log(y) & if\ t = 1 \\ -\log(1-y) & if\ t = 0 \end{cases}$$

מציאת המשקולות:

1. אלגוריתם GD כפי שתואר מעלה. לשם כך, הגזירה היא

$$\frac{\partial CE}{\partial w} = \frac{\partial C}{\partial y}\frac{\partial y}{\partial z}\frac{\partial z}{\partial w} = \frac{1}{n}\sum_{i \in D}(t_i - y_i) \cdot x_i$$

סיווג שאינו בינארי בעזרת קלסיפייר בינארי:

גישת one vs rest נבנה קלסיפייר 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.

גישת one vs one נבנה קלסיפייר 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.

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

גמישות מודלים:

גמישות של מודל מתבטאת בדרגת הפולינום שלו.

Under fitting = אין התאמה מספיק טובה בין המודל הנוצר לנתונים.

Over fitting = התאמה מידי טובה בין המודל הנוצר לנתונים. על מנת להגמיש את המודל ניתן: להוסיף מאפיינים, להשתמש בהיפותזות מורכבות וגמישות יותר (רשתות). **הגמשה עבור רגרסיה:** שיטת LWLR. נקודות יותר קרובות ישפיעו יותר מנקודות רחוקות.קרבה מחושבת בעזרת $\beta_i = e^{\frac{-\|x_i - x\|^2}{2\tau^2}}$ ואז פונקצית השגיאה היא $WMSE_w = \frac{1}{2m}\sum_i \beta_i (wx_i - t_i)^2$. ככל שהקבוע τ קטן יותר המשקל של נקודות שהן מרוחקות קטן ואילו של נקודה קרובה גדל, ככל שיותר קטן יותר over fitting. τ גדול משמע רגרסיה לינארית רגילה

הקשחת המודלים:

1. Feature selection

לפי עקרון התער על אוקאם, " אין להעמיד ריבוי ללא צורך" , כלומר העדפת ההסבר הפשוט יותר על פני המורכב. במקרה הזה, העדפה למודלים בעלי פחות פארמטרים. אלגוריתם בחירת פרמטרים יחזיר רק את המאפיינים הטובים ביותר.

=Forward feature selection

1) Start with empty feature set
2) For $k = 1, \dots, n$:
   2.1) For all features f_k not used by model:
      2.1.1) Try adding each of the missing features, train and save the loss
      2.2.2) 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

1) Start with full feature set
2) For $k = 1, \dots, n$:
   2.1) For all features f_k not used by model:
      2.1.1) Try removing each of the missing features, train and save the loss
      2.2.2) 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

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

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

2. רגולריזציה:

הענשת משקולות. נגדיר γ קבוע רגולריזציה, ונוסחת שגיאה חדשה, מעונשת.

= 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:

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

= Lasso regularization (L1)

נעניש את הפונקציית השגיאה על ידי הוספת ריבוע הנורמה מסדר ראשון $\|w\|_1 = \sqrt{\sum_{i=1}^{n} |w_i|}$ ואז, נוסחת השגיאה המעונשת תהיה $regMSE(w) = MSE(w) + \gamma \sum_{j=1}^{n} |w_j|$.

אם כך, כלל עדכון המשקולות משתנה בהתאם:

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

שיטה זו מסוגלת לבצע בחירת מאפיינים (לאפס).

= Elastic net regularization

שילוב בין שתי הנורמות. במשקולות הקטנים לאסו משפיע ואילו במשקולות הגבוהים, הרידג' (אבל הלאסו ממתן קצת). הנוסחה $\frac{lasso+ridge}{2}$.

3. אנסמבלים:

אנסמבלים הם צירופים של הרבה מודלים שונים, כאשר מתוכם יבחר המודל הטוב ביותר. יצירה:

= Bagging יצירת אנסמבלים על ידי דגימות שונות מתוך סט האימונים. מאמנים מודלים שונים על תת קבוצות שונות של הנתונים, ואת הפלט ממצעים או החלטת הרוב.

1) 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 ולידציה הכוללת דוגמאות שלא נלקחו לדגימה.

2) שיטת k fold = מיצרים k קבוצות אימון ע"י שמוציאים בכל פעם 1/k מהנתונים כוולידציה, ומשאירים בקבוצת האימון את היתר. קבוצות הוולידציה הן זרות זו לזו.

שיטת boosting = אנסמבלים של מודלים מאומנים על קבוצות אימון שונות כך שכל מודל מתמקד בדוגמאות שהקודמים לו שגו בהן. משתמשים בסדרה של מודלים "חלשים" (עם שגיאת ביאס גבוהה) וכל מודל מתרכז בלמידה של המקרים שהמודלים האחרים לא הצליחו לסווג נכון. משתמשים באותם נתונים, אך מגדילים את משקל הדוגמה (תדירות הופעתה) במידה והמודלים הקודמים לא סיווגו אותה נכון. מקבלים סידרת מומחים (שכל אחד "חלש" מידי – ולא יכול לעשות Overfitting) שנותנים תחזיות שונות למיקרים קשים.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 ולידציה.

פירוק שגיאת MSE:

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

בד"כ שגיאת ביאס משמע under fitting ואילו שגיאת שונות משמע over fitting . ככל שמעלים גמישות של מודל הביאס יורד והשונות עולה.

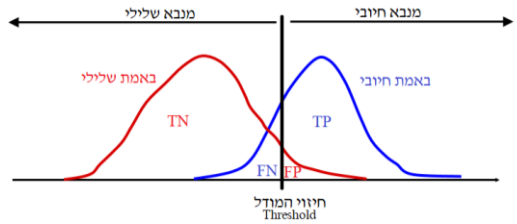
מדידת הצלחת המודלים:

נשתמש KPI's = Key Performance Indicators :

Recall, TPR	$\frac{TP}{TP+FN} = 1 - FNR$
Specificity, TNR	$\frac{TN}{TN+FP} = 1 - FPR$
Precision / PPV	$\frac{TP}{TP+FP} = 1 - FDR$
false negative rate	$\frac{FN}{FN+TP} = 1 - TPR$
false positive rate	$\frac{FP}{FP+TN} = 1 - TNR$
false discovery rate	$\frac{FP}{FP+TP} = 1 - PPV$
Accuracy מדד כללי טוב להצלחת מודל	$\frac{TP+TN}{TP+TN+FP+FN}$
F1 score עוזר כשחלק מהקטגוריות נדירות	$2 \cdot \frac{P \cdot R}{P+R}$
Balanced accuracy	עבור מודל שאינו בינארי הוא הממוצע של recall של כל קטגוריה. אחרת, ממוצע בין recall לבין הספציפיות

ניתן לעזר במטריצת עמימות

Confusion matrix		החיזויים (y) חיזוי חיובי	חיזוי שלילי	
הסיווג האמיתי (t)	באמת חיובי	*True Positive*	*False Negative* שגיאה מסוג 2	$TPR\ or\ recall = \frac{TP}{P}$
	באמת שלילי	*False Positive* שגיאה מסוג 1	*True Negative*	$TNR = \frac{TN}{N}$
		$PPV\ or\ Precision = \frac{TP}{Predicted\ P}$	$NPV = \frac{TN}{Predicted\ N}$	$Accuracy = \frac{TP+TN}{ALL\ Examples}$



נוכל ליצור מטריצת עמימות גם עבור קלסיפיקציה לא בינארית.גודל המטריצה יהיה כמספר הקטגוריות. אם ישנן n קטגוריות אז גודל המטריצה יהיה $n \times n$. ואז יהיה ניתן לחשב את כל המדדים שנלמדו על כל קטגוריה בנפרד.

שיערוך הצלחה בעזרת ואלידציה 33CV 33:

שגיאת הולידציה עבור k דוגמאות היא

$$CVloss = \frac{1}{k}\sum_{v} loss_v$$

1. K-fold CV:

חלק את הנתונים המתויגים ל K קבוצות. שמור $\frac{1}{k}$ מהנתונים כקבוצת ולידציה כל פעם. אמן על $\frac{K-1}{K}$ מהנתונים ובדוק על הוולידציה.

בצע K אימונים שונים בכל פעם על קבוצת וולידציה אחרת. סה"כ K מודלים יחושבו.

2. Leave K out CV:

חלק את הנתונים המתויגים ל K קבוצות. קבוצה אחת כקבוצת ולידציה כל פעם. אמן על $D-K$ מהנתונים ובדוק על הוולידציה. בצע $C\binom{D}{K}$ אימונים שונים בכל פעם על קבוצת ולידציה אחרת. סה"כ $C\binom{D}{K}$ מודלים יחושבו.

3. Bootstrap Method:

יכולות החיזוי והערכת השגיאה היו משתפרות אם היינו יכולים לייצר כמויות גדולות של נתונים מתויגים שאיתם היינו מייצרים הרבה קבוצות אימון. אבל, ברוב המקרים אין נתונים רבים. בשיטה הזו, מיצרים הרבה קבוצות אמון מקבוצת אימון אחת שאותה דוגמים רנדומית עם חזרות.קיים סיכוי $\frac{1}{3}$ שחלק מהדוגמאות בסט לא יופיעו בקבוצת האימון, אותן ניקח כולידציה

אלגוריתמים שונים:

1. **K nearest neighbors =**

עבור קלסיפיקציה: בהינתן אוסף דוגמאות ודוגמת מבחן חדשה, נשערך את הסתברות השייכות של הדוגמא לכל אחת מהקטגוריות על פי החלטת הרוב של k השכנים הקרובים ביותר.

עבור רגרסיה: בהינתן אוסף דוגמאות ודוגמת מבחן חדשה, נשערך את החיזוי של הדוגמא לפי ממוצע הערכים של k השכנים הקרובים ביותר. הגמישות יורדת ככל שיותר שכנים משתתפים בהחלטה. יש להחליט מי קרוב:

עבור מאפיינים נומרים:

- ✓ מרחק מנהטן l1 norm

$$d(A,B) = |x_A - x_B| + |y_A - y_b|$$

- ✓ מרחק אוקלידי l2 norm

$$d(A,B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

- ✓ נורמת האינסוף

$$d(A,B) = \max|A_i| - \max|B_i|$$

עבור מאפיינים שאינם נומרים:

- ✓ מרחק קוסינוס שמודד קרבה לפי דמיון

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

שני וקטורים מאונכים אחד לשני הדמיון שלהם יהיה אפס, ואילו וקטורים עם אותה זוית, דמיון אחד.

✓ מרחק קורלצית פירסון שמודד קרבה לפי קורלציה

$$d(A,B)=\frac{\sum_{i=1}^{n}(A_i-\overline{A})(B_i-\overline{B})}{\sqrt{\sum_{i=1}^{n}(A-\overline{A})_i^2}\sqrt{\sum_{i=1}^{n}(B-\overline{B})_i^2}}$$

k הינו היפר פארמטר עבור האלגוריתם.

שכלול האלגוריתם = weighted k-NN:

הבאת משקל רב יותר לשכנים הקרובים ביותר.

בעזרת דימיון גאוסי, מאפשרת שליטה בגמישות:

$$w_i=e^{\frac{-\|x_i-x\|^2}{2\tau^2}}$$

חסרון האלגוריתם:

רגיש מאוד למספר המימדים. הדמיון יכול להטעות. ניתן לפתור על ידי הורדת מימדים.

2. למידה בייזיאנית=

התבססות על תורת ההסתברות, בפרט על משפט בייס וכן על 2 הנחות נאיביות.

נוסחאת ההסתברות המותנית: $P(B|A)=\frac{P(A\cap B)}{P(A)}$

חוק בייס: $P(A|B)=\frac{P(B|A)\cdot P(A)}{P(B)}$

נוסחאת ההסתברות השלמה:

$P(B)=\sum_i^k P(B|A_i)P(A_i)$ ומתוכה ניתן לשנות את נוסחת בייס: $P(A_i|B)=\frac{P(B|A_i)\cdot P(A_i)}{\sum_i^k P(B|A_i)P(A_i)}$

הסתברות מותנית נקראת posterior ואילו הסתברות של מאורע כלשהו נקראת prior.

עבור המאורע $k|x$ נרצה לשערך את הייתכנות שלו. ישנן 2 דרכים.

הערכה מקסימלית להסתברות posterior= MAP:

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

כלומר, נחשב עבור כל ערכי x את ההערכה ונבחר את המקסימלית. היא השערוך להסתברות הרצויה.

הערכה מקסימלית לסבירות likelihood= ML:

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

כלומר, נחשב עבור כל ערכי x את ההערכה ונבחר את המקסימלית. היא השערוך להסתברות הרצויה.

נשים לב כי אילו הערכות בלבד שאינן מחשבות את ההסתברות עצמה. על מנת לחשב את ההסתברות עצמה יש לחשב את כל הנדרש כדי לקבל:

$$p(k|x)=\frac{k\cap x}{p(x)}$$

הנחות בייס:

✓ המאפיינים אינם תלויים זה בזה בהינתן קטגוריה k ולכן ההסתברות לוקטור קלט כלשהו הוא כפל ההסתברויות של כל המאפיינים בו

$$p(v_1,v_2,\dots v_n\,|k)\approx\prod_{j=1}^{n}p(x_j=v_j|k)$$

✓ סדר ההופעה אינו חשוב

$$p(a_i=w_j|k)=p(a_m=w_j|k)$$

אלגוריתם בייס הנאיבי:

בהתאם להנחות הנאיביות, נבנה האלגוריתם

Naïve Bayes Classifier (D, K, x):
For each prediction 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
$predict=argmax_k\{P(k)\cdot\prod_{j=1}^{n}P(v_j|k)\}$

ישנו צורך לבצע תיקונים לאלגוריתם עבור קטגוריות נדירות m-estimate:

כאשר אין מספיק דוגמאות מקטגוריה k ושבהן גם $x_i=v_i$, מוסיפים m דוגמאות וירטואליות מקטגוריה k שמתוכם $P(x_i=v_i)$ הם בעלי ערך $x_i=v_i$. $P(x_i=v_i)$ הינו השערוך prior של v_i. m הינו היפר פאראמטר שמצפה על גודל המדגם או על התלויות.

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

אם לא ניתן לשערך את ההסתברות הקודמת $P(x_i=v_i)$ נשתמש בתיקון Laplace smoothing:

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

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

בפרט, אם נוסיף מספר דוגמאות כמספר הערכים של הקטגוריה החסרה $m=|v|$ נקבל $\frac{n_{v_i\cap k}+1}{n_k+|v|}$.

3. עצי החלטה ויער רנדומי=

גישה שבה ההיפותזה היא עץ החלטה. נתונה קבוצת אימון עם דוגמאות ממספר קטגוריות. נחפש שאלה על אחד המאפיינים שתפצל את קבוצת האימון לשתיים (או יותר) קבוצות הומוגניות ככל האפשר. ניצור צומת הכולל את השאלה שנבחרה ונפצל את קבוצת הדוגמאות לקבוצות (ילדים בעץ) בהתאם לשאלה. בצורה רקורסיבית, נמשיך לפצל את כל קבוצות הצאצאים עד שנצליח ליצור עלים הומוגניים (הפרדה מלאה) או שנעצור קודם על פי קריטריון אחר.

מאפיין הינו קטגורי, ניתן לפצל בשתי דרכים:

- ✓ פיצול שהוא יהיה כל הקטגוריות האפשריות
- ✓ פיצול שהוא בינארי על ידי בחירת תת קבוצה של ערכים, למשל, קטגוריה אחת מול כל השאר או לקבוצת ערכים כנגד הקבוצה המשלימה

מאפיין נומרי, ניתן לפצל, גם כן, בשתי דרכים:

- ✓ ניתן לפצל אותו באופן בינארי על ידי בחירת סף $x_i < threshold$
- ✓ ניתן להגדיר טווחי ערכים (Bins) ולפצל את העץ כאילו היו קטגוריות בדידות

מרחב ההיפותזות של כל העצים האפשריים הוא עצום.

בהנחה שישנם n משתנים בוליאנים ושהעצים הם מלאים, מספר העצים הוא 2^{2^n}. על כן, נחפש באופן חמדני את העץ הטוב ביותר.

אם הקבוצה הומוגנית, ניצור עלה וסיימנו. אחרת, נבחר מאפיין שיפצל הכי טוב טוב לשתי קבוצות נפרדות. באופן רקורסיבי נמשיך לפצל עוד ועוד בנים עד שהקבוצות יהיו הומוגניות או שלא ניתן לפצל יותר. כדי לחשב מה הוא מאפיין הפיצול הטוב ביותר, נעזר במדד האנטרופיה = מדד להומוגניות, לסדר של קבוצה ביחס לקטגוריות המופיעות בה. ככל שיש יותר ערכים בקבוצה, אי הסדר גדל,

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

כדי להמשיך ולאפיין, נשכלל את האנטרופיה =

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

כלומר, האנטרופיה המשוכללת היא סכום האנטרופיות של הבנים במכפלת הסיכוי לפיצול הזה על העץ. ועכשיו, נוכל לדעת כמה מידע הרווחנו בעזרת פיצול זה על ידי information gain $IG = H(parent) - H_{tot}$

להלן האלגוריתם החמדני:

```
Grow Tree(S):
if (t=0) for all examples, return (new leaf(0))
if (t=1) for all examples, return (new leaf(1))
Else
  choose "best" feature x_j not used yet by
checking entropy and IG
S_0 = all <x,t> ϵ S with x_j = 0
S_1 = all <x,t> ϵ S with x_j = 1
  return (new node (x_j < 0.5,Grow Tree(S0),
Grow Tree(S1))
```

נשים לב כי קיימת בעיתיות עבור מאפיינים שהם נומרים רציפים, למשל טמפרטורה. במקרה כזה, נחפש סף שלפיו ניתן לפצל. נבדוק את רווח המידע עבור מספר מקומות סף שבהם ניתן לפצל פיצול בינארי.

נסדר את ערכי הערכי המאפיין בסדר עולה (וברשימה נפרדת את ערכי התוויות המתאימים). נבדוק את רווח המידע info gain (=ממוצע הערכים במעבר) רק בנקודות שבהם מתחלפת התווית. מובטח לרווח המידע המקסימלי להיות באחד המעברים

Temp:	40	48	60	72	80	90
PlayTennis:	No	No	Ye	Ye	Ye	No

ניתן גם לאפשר רגרסיה בעץ: 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.

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, $\max G = \frac{1}{2}$. כאשר הסתברויות לקטגוריות קרובות לאחד או לאפס, שני המדדים יתנו מספר נמוך (קרוב ל 0 מלמעלה). שני המדדים מקבלים ערך גדול מאפס כאשר יש הרבה קטגוריות בהסתברות זהה, אולם G חסום ב 1 ואילו H יכול להגיע למספרים גבוהים.

עצים יכולים לבצע overfitting ולהתייחס למאפיינים לא חשובים ולכן עדיף להשתמש באנסמבל של עצים.

Bagging using bootstrap:

נבצע דגימה עם חזרות כדי לגדל B עצים שונים המורכבים B מדגמים שונים קצת זה מזה. נעזר בתוצאות החיזוי של העצים השונים: ברגרסיה ממצעים את החיזוי, 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.

Random forest:

נרצה לגדל עצים הנבנים מקבוצות שונות של מאפיינים. נבנה עץ לכל מדגם בשיטת bootstrap. בכל פעם שבוחרים צומת לפיצול, נגביל את המאפיינים לבדיקה ל $m' < m$ מאפיינים הנבחרים רנדומית מתוך m המאפיינים שלרשותנו. כלומר, אם $m' \ll m$,

בכל פעם שבוחרים צומת לפיצול, האלגוריתם לא מורשה להסתכל על רוב המאפיינים. העצים שנגדל יהיו שונים, גם בגלל שנוצרו מקבוצות אימון שונות במקצת ובעיקר בגלל שאולצו להשתמש במאפיינים שונים.

בהינתן קבוצת דוגמאות S, רשימת מאפיינים מותרים והיפר פרמטר m' , להלן האלגוריתם מגבלה אקראית על מאפיינים הניתנים לבחירה:

```
Choose Best Feature (S, Feature set, m'):
1)If Feature set is empty return (Null,Null)
2) Randomly select m' features from
   Feature set (if |Feature set| < m',
   select all)
3) Find the Best Feature with the best IG
    3.1) remove from Feature set,
return(Best Feature, Feature set)
```

להלן האלגוריתם המלא לבניית עץ בינארי אקראי:

```
Grow Rand Best Tree (S, Feature set, m'):
If t= 0 for all <x,t> in S, new leaf(0)
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(S, Grow Rand Best
Tree(S0, Feature set, m'), Grow Rand Best
Tree(S1, Feature set, m'))
```

קל להכליל עבור מאפיינים קטגוריים ונומרים.

גבול ההחלטה של עצים: עץ מחלק את מרחב הקלט למלבנים (היפר קופסאות) לא חופפים. אם המאפיינים נומרים, ניתן להגיע להפרדה מושלמת.

4. Support Vector Machines =

קלסיפייר בינארי שאינו חייב להיות לינארי המבוסס על טריק kernel.

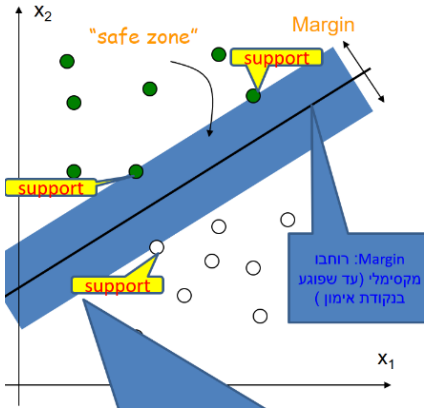
בהינתן קבוצת אימון D של דוגמאות נרצה ליצור מישור מפריד בעל השוליים המקסימליים. כאשר ממקסמים את השוליים, רק היפר- מישור אחד יכול להיות עם שוליים מקסימליים. אם נקודות האימון מחוץ לשוליים יזוזו, המישור המפריד לא יושפע אם נקודות האימון התומכות יזוזו, המישור המפריד ישתנה.נדמה את מישור ההפרדה (כולל השוליים) כשרוול / צינור ממימד n . 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, משתמע מכך שההפרדה אינה לינארית מושלמת.

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.

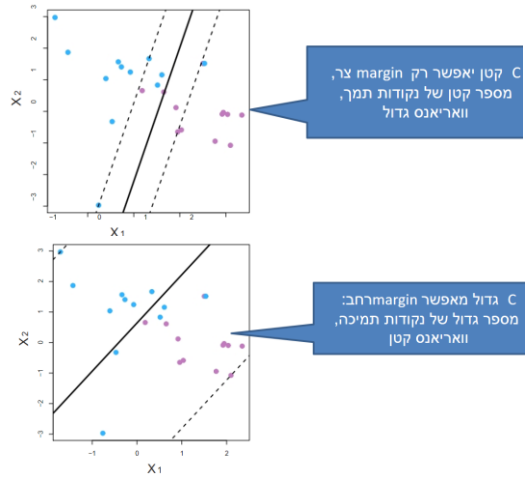
כאשר $\varepsilon_i = 0$ הדוגמא , תהיה מחוץ לשוליים ותסווג נכון . כאשר $0 < \varepsilon_i \leq 1$ הדוגמא תהיה בצד הלא נכון של השוליים אבל בצד הנכון של המישור ותסווג נכון. כאשר $\varepsilon_i > 1$ הדוגמא תהיה בצד הלא נכון של המישור המפריד ותסווג שגוי.

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. C הינו היפר פרמטר שולט Bias – Variance tradeoff

לא יתאפשרו יותר מאשר C חריגות מהמישור המפריד. 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.



להלן ההשפעה של התקציב:



<u>גבולות החלטה לא לינארים, טריק kernel:</u>

שיטת SVM משתמשת בטריק kernel שמאפשר הרחבת המרחב בו משתמש הקלסיפייר באופן שהוא בהרבה מיקרים יעיל יותר מבחינה חישובית וללא המרה בפועל של הווקטורים.

ההיפותזה: $h(x) = \alpha_0 + \sum_{i=1}^{m} \alpha_i \langle x, x_i \rangle$ כלומר, מכפלה פנימית שהיא 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. קיימים המון סוגי גרעינים, הנפוצים:

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

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

<u>דמיון בין אלגוריתם SVM לאלגוריתמים אחרים:</u>

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

<u>אלגוריתם SVM ליותר משתי קטגוריות:</u>

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.

למידה בלתי מונחת:

נציג שני אלגוריתמים של למידה שאיננה מונחת.

✓ **שיטת אשכולות Clustering =**

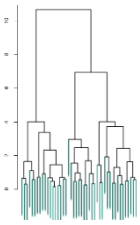
זוהי שיטה בלתי פרמטרית מיועדת עבור קלסיפיקציה. בהינתן קבוצה D של דוגמאות אימון (ללא תיוגים) רוצים לחלק לקבוצות זרות שמכסות את D ומכילות דוגמאות "דומות" כך שהנקודות שבתוך קלסטר תהינה דומות זו לזו ואילו נקודות בקלסטרים שונות תהינה שונות זו מזו.

האלגוריתם k-means clustering:

K- means Classifier (K, D, $distance$):
1) add randomly each example in D to one of the K clusters
2) while there are change in clusters:
   2.1) calculate center c_i of each cluster
   2.2) for each example x, calculate the $distance(x, c_i)$ and add x to the cluster that is closest

מרכז הקלסטר הוא ווקטור ממוצעי המאפיינים של הדוגמאות שבקלסטר. לשם ביצוע האלגוריתם:
1) פונקצית מרחק לחישוב קרבה למרכז קלסטר, למשל מרחק אוקלידי.
2) מדד לשוני בתוך קלסטר Within-Cluster-Variation, למשל סכום הממוצעים של מרחקי נקודה משאר הנקודות בקלסטר.

חיסרון גדול של אלגוריתם זה הוא שיש לקבוע מראש את מספר הקלסטרים. הפתרון לבעיה זו הוא שימוש בהיררכיה של קלסטרים.



קלאסטרים היררכים:

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 Dendograms נוכל ע"י חיתוך מלמעלה בגובה המתאים, לקבל כל מספר של קלסטרים שנרצה.

Hierarchal Classifier (K, D, ICD, WCV):
1) add each example in D to a separate cluster
2) while more than one cluster left:
   2.1) calculate Inter-Cluster-Dissimilarity for each couple of clusters using ICD
   2.2) choose the two clusters with the minimal ICD and merge them into one cluster
   2.3) the height of the new united cluster is the result of the Within-Cluster-Variation using WCV

ישנן מספר דרכים לחשב אי דמיון בין קבוצות, linkage=inter- cluster- dissimilarity:

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.

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

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.

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.

בשימוש באלגוריתמים אלו, יש צורך לנרמל את הנתונים, בשיטת sd. שיטת האשכולות לא תמיד עובדת- כאשר אין הפרדה טובה בין תת הקבוצות או כאשר נפח תת הקבוצות שונה משמעותית.

✓ **הפחתת מימדים PCA=**

בהינתן מדגם D של נתוני קלט n מימדי $x = (x_1, x_2, \dots, x_n)$ מחפשים טרנספורמציות לינאריות ששיעבירו את הקלט המקורי x ממימד n ל z ממימד m (קטן מ n) תוך שימור של המידע "החשוב". האלגוריתם מתבסס על הטלות ועל עקרון מיקסום השונות בו זמנית עם מזעור המרחק, כלומר, המרחקים האוקלידים של הנקודות מהציר מתמזערים באותו רגע בו מתמקסמת השונות על הציר.

הטרנספורמציות שמחפשים הם $\emptyset_1, \emptyset_2, \dots \emptyset_m$ כאשר 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 טרנספורמציות שישמרו מידע חשוב. כל אחת מ m ה$\emptyset_i$ הוא Principle Component. הצירים החדשים הם הוקטורים העצמיים של מטריצת ה Co-variance של D. הציר שנותן את השונות הכי גדולה בין ההטלות של הנקודות על הציר הוא ה Principle Component הראשון שהוא הוקטור העצמי עם הערך העצמי הדול ביותר של מטריצה. הווקטור עם הערך העצמי הבא הכי גדול, הוא הציר השני, כל הצירים מאונכים אילו לאילו. יש לנרמל את הנתונים. התוצאות של האלגוריתם יהיה תלויות בנרמול המאפיינים.

דוגמאות:

Naive Base:

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. $y_{\neg cancer} = P(\neg cancer|+) = P(+|\neg cancer) = 0.03$

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

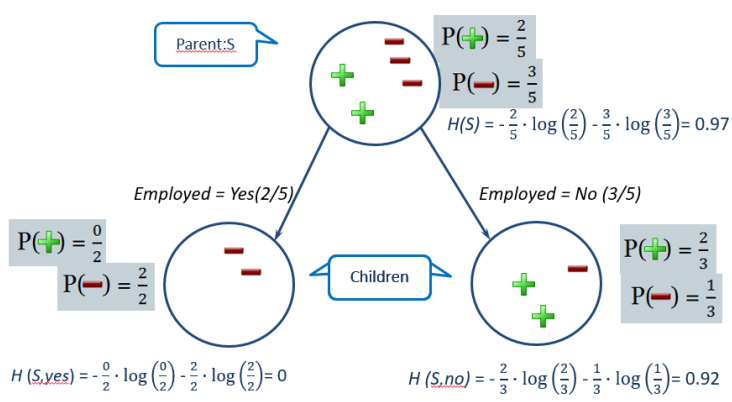
נשים לב כי דוגמה זו לא שיערה את ההסתברות עצמה, אלא רק נתנה סיווג שיבחר.

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 .

עצים:



נוכל לחשב את האנטרופיה המשוכללת:

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

רווח המידע שיחושב הוא $0.97 - 0.552 \approx 0.42$ כלומר, מאפיין "מועסק" הוסיף 0.42 כמות של מידע.

רגרסיה פולינומיאלית:

בהינתן n מאפיינים, להלן הנוסחה למספר המאפיינים שיש להוסיף על מנת להעלות את דרגת הפולינום להיות p בעזרת C:

$$\sum_{p=2}^{P}\binom{p+n-1}{p}$$

שערוך של ההסתברות הממוצעת לחיזוי נכון של קלסיפייר:

בהנחה שהקלסיפייר משתמש בפונקציית CE אז, שיערוך ההסתברות הוא:

1. חשב את השגיאה הממוצעת (מצע את כלל השגיאות על כל הדוגמאות), נסמן ce
2. חשב e^{-ce}

הוכחה הסתברותית לשימוש בנוסחת CE:

הצדקה הסתברותית לשימוש ב CE:

נרצה למצא היפותזה h(w) בעלת ההסתברות המקסימלית בהינתן D.

$$\text{argmax}_w\{P(h_w \mid D)\}$$

בהינתן D, מחפשים היפותזה h_w שמסבירה הכי טוב את D
ז.א. שההסתברות $h_w(x_p)=t_p$ לכל דוגמא p ב D, היא מקסימלית מבין כל ה hים האפשריים

נראה כי מיזעור ה CE (תחת הנחות מסוימות) נותן את ההיפותזה הכי מסתברת

נרצה למקסם את ה Likelihood שההיפותזה מסבירה את D
(ז.א. חוזה נכון את ה t של כל דוגמא ב D)

ה- likelihood של h_w ביחס ל D,
אידיאלית h_w צריך לתת הסתברות 1 לדוגמאות חיוביות ו 0 לדוגמאות שליליות
ואז שיערוך ההסתברות הוא $l_D(w)=1$
אם h_w אינם מדייק בחיזוי, $l_D(w) < 1$
מיקסום ה likelihood מוצא את ה W שבו hw נותנת הסתברויות גבוהות לlabels הנכונים.

$$l_D(w) = \prod_{p:t_p=1} y_p \prod_{p:t_p=0}(1-y_p) \qquad y_p = h_w(x_p)$$

ממוצע גיאומטרי לפי m דוגמאות יתן את ההסתברות הממוצעת לחזות נכון בדוגמא x בעזרת w

אימון: מחפשים w שימקסם את $l_D(w)$,

ניתן למקסם את המכפלה באמצעות מיקסום ה log שלה.

$$w^* = argmax_w\{\log(l_D(w))\}$$

לוג מכפלת ההסתברויות הוא סכום הלוגים

$$loglikelihood_D(w) = \sum_p t_p \log(y) + \sum_p (1-t_p)\log(1-y)$$

24

נרצה למצא היפותזה h(w) בעלת ההסתברות המקסימלית בהינתן D: $\text{argmax}_w\{P(h_w \mid D)\}$

likelihood prior posterior evidence

חוק: bayes

$$p(h_w|D) = \frac{p(D|h_w)P(h_w)}{P(D)}$$

נחשב את ה Likelihood: בהנחת אי תלות בין הדוגמאות:
ההסתברות ש מודל w חוזה נכון כל הדוגמאות ב D שווה למכפלת ההסתברות שכל דוגמא נחזית נכון

$$l_D(h_w) = P(D|h_w) = \prod_{p:D} p(t_p \mid x, h_w) = \prod_{p:t_p=1} p(t=1 \mid x_p, h_w) \prod_{p:t_p=0} p(t=0 \mid x_p, h_w)$$

ההיפותזה מחשבת הסתברות ל t=1

$$= \prod_{p:t_p=1} h_w(x_p) \prod_{p:t_p=0}(1-h_w(x_p))$$

$$p(h_w|D) = \frac{P(h_w)}{P(D)} l_D(h_w)$$

מדוע מיקסום loglikelihood עדיף?

לוג מכפלת ההסתברויות הוא סכום הלוגים

$$loglikelihood_D(w) = \sum_p t_p \log(y) + \sum_p (1-t_p)\log(1-y)$$

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

נרצה למצא היפותזה h(w) בעלת ההסתברות המקסימלית בהינתן D: $\text{argmax}_w\{P(h_w \mid D)\}$

$$p(h_w|D) = \frac{P(h_w)}{P(D)} l_D(h_w)$$

$$l_D(h_w) = \prod_{p:t_p=1} h_w(x_p) \prod_{p:t_p=0}(1-h_w(x_p))$$

מביוון שרוצים למקסם את ההסתברות p(h|D), ניתן להתעלם מה evidence - p(D) הקבוע שאינו תלוי ב w
נתעלם גם מ $P(h_w)$ (הפריור של ההיפותזה): מחוסר ידע, מניחים שלכל ההיפותזות יש הסתברות שווה

$$w^* = argmax_w\{\prod_{p:t_p=1} h_w(x_p) \prod_{p:t_p=1}(1-h_w(x_p))\}$$

Maximizing the LogLikelihood= Minimizing the CE

$$loglikelihood_D(w) = \sum_p t_p \log(y_p) + \sum_p (1-t_p)\log(1-y_p)$$

אם ניקח ממוצע (של הדוגמאות ב D) נקבל את ה Log של ההסתברות הממוצעת לחיזוי נכון של הדוגמאות ב D
נהפוך את הסימן כדי לקבל מספרים חיוביים - שאותם נרצה למזער

פונקציה זו נקראת Cross Entropy

$$CrossEntropy(y,t) = -\frac{1}{m}\left(\sum_p t_p \log(y_p) + \sum_p (1-t_p)\log(1-y_p)\right)$$

מיזעור ה CrossEntropy היא דרך בללית מאוד להתאים היפותזות $y=h_w(x)$ למטרת קלסיפיקציה
ממצעים את ה CE כדי לקבל log– ההסתברות הממוצעת
מיזעור ה CE נקרא גם מיקסום של ה log-likelihood

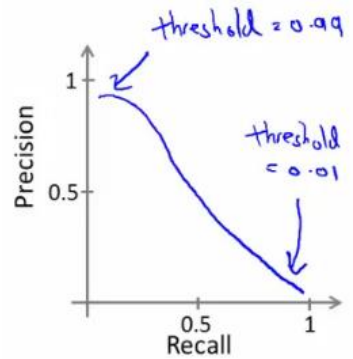
26

הנחות בהוכחה: הקלטים לא תלויים סטטיסטית אחד בשני, מחוסר ידע מוקדם על היפותזות אפשריות הזנחנו את הפריור כלומר לכל ההיפותזות אותה הסתברות. אם ההנחות נכונות, ההיפותזה המתקבלת ממיזעור CE היא ההיפותזה המסתברת ביותר בהינתן D.

Precision – Recall Tradeoff:

ככל שנקטין את הסף (נזיז אותו שמאלה) :

1. Recall יעלה
2. Precision ירד



שיטות רגישות לאי נרמול:

עבור שיטות אלו יש לנרמל תחילה את הנתונים :

1. PCA
2. KNN
3. Clustering
4. GD וכל מי שמשתמש בו (רגרסיה לוגיסטית ולינארית)
5. SVM

במקרה של חריגים, עדיף לנרמל בעזרת std היות ואז הסטיית תקן תהיה 1.

סיבוכיות של המשוואה הנורמלית: הכפלה של X^T ב- X: $O(d^2\|D\|)$ הכפלה של X^T ב- t: $O(d\|D\|)$ מציאת מטריצה הופכית ל- X^TX: $O(d^3)$ הכפלה של $(X^TX)^{-1}$ ב- X^Tt: $O(d^2)$ סה"כ: $O(d^2\|D\|+d\|D\|+d^3+d^2)$ לאחר צמצום: $O(d^2\|D\|+d^3)$ התשובה הנכונה: $O(d^2\|D\|+d^3)$	**סיבוכיות של GD:** ריצה על כל ה- epocs: $O(e)$ ריצה על אוסף הדוגמאות: $O(\|D\|)$ ריצה על הפיצ'רים של דוגמה אחת: $O(d)$ ריצה על כל הפיצ'רים של כל דוגמאות האימון בכל ה- epochs: $O(\|D\|*d*e)$ התשובה הנכונה: $O(\|D\|*d*e)$	**פירוק שגיאת MSE לניתנת להפחתה ולא:** MSE = $E_{x,D}\left[(t_x-h_D(x))^2\right]=$ $E_{x,D}\left[(f(x)+\varepsilon-h_D(x))^2\right]=$ $E_{x,D}\left[(f(x)-h_D(x))^2+2(f(x)-h_D(x))\varepsilon+(\varepsilon-0)^2\right]=$ $E_{x,D}\left[(f(x)-h_D(x))^2\right]+2E_{x,D}[f(x)-h_D(x)]E_{x,D}[\varepsilon]+E_{x,D}\left[(\varepsilon-0)^2\right]=$ $E_{x,D}\left[(h_D(x)-f(x))^2\right]+Var(\varepsilon)=$ ReducableError + IrreducableError
Feature Selection: במידה ולא משתמשים באלגוריתם חמדני, הסיבוכיות תהיה $O(2^n)$ כי נעבור על כל תתי הקבוצות האפשריות	**נוסחת F Score שאיננה מאוזנת:** $F-Measure=\frac{1}{\alpha\cdot\frac{1}{Precision}+(1-\alpha)\cdot\frac{1}{Recall}}$ עבור $\alpha=0.5$ נקבל את $F1\ score$	**פירוק השגיאה הניתנת להפחתה לביאס ושונות:** ReducableError $E_{x,D}\left[(h_D(x)-f(x))^2\right]$ $E_{x,D}\left[(h_D(x)-\overline{h_D(x)}+\overline{h_D(x)}-f(x))^2\right]$ $E_{x,D}\left[(h_D(x)-\overline{h_D(x)})^2+2(h_D(x)-\overline{h_D(x)})(\overline{h_D(x)}-f(x))+(\overline{h_D(x)}-f(x))^2\right]$ $E_{x,D}\left[(h_D(x)-\overline{h_D(x)})^2\right]+2E_{x,D}[(h_D(x)-\overline{h_D(x)})]E_{x,D}[(\overline{h_D(x)}-f(x))]+E_{x,D}\left[(\overline{h_D(x)}-f(x))^2\right]$ $E_{x,D}\left[(h_D(x)-\overline{h_D(x)})^2\right]+(E_{x,D}[h_D(x)-f(x)])^2$ $Variance[h_D(x)]+Bias[h_D(x),f(x)]^2$

רגולריזציה:

ככל שמעלים את קבוע ההענשה, השונות קטנה אילו הביאס גדל - ריבוע הביאס עולה בהתחלה לאט ואח"כ באופן מהיר יותר עד להתיצבותו. השונות יורדת בתמדה. ההכלה יורדת, מגיעה למינימום ועולה כשהביאס הופך לדומיננטי.

אלגוריתם K-means:

אלגוריתם קליסטור של K-means מובטח שיתכנס כיוון שמספר הנקודות בדאטה סופי.