שיעור 2

25 במרץ 2019

1 רקע הסתברותי

נגדיר כמה מושגים בהם נשתמש בהמשך השיעור. נעסוק רק בהסתברות בהקשר דיסקרטי, דבר שיקל עלינו כמה הגדרות.

מרחב המדגם - זהו אוסף האירועים שאנו עשויים לחזות בהם בניסוי. למשל, בהטלת קובייה, האירועים הם "קיבלנו 1", "קיבלנו 2" וכן הלאה. נסמנו ב־Ω.

התפלגות - פונקציה המקצה הסתברות לכל אירוע במרחב המדגם. סכום ההסתברויות הוא בהכרח 1. נסמן $Pr : \Omega \to [0,1]$ כאשר בקוביה הוגנת, $Pr(got\,1) = \frac{1}{6}$.

משתנה מקרי - פונקציה שמחזירה לכל אירוע במרחב המדגם מספר ממשי, $X : \Omega \to \mathbb{R}$. נסמן $Pr(X = n) := Pr(X^{-1}(n))$.

תוחלת של משתנה מקרי - מעין ממוצע משוקלל של הערכים שנמדוד ב־X אם נבצע ניסוי פעמים רבות. מחושב כך:

$$\mathbb{E}(X) = \sum_{n\in\Omega} Pr(n)\cdot X(n) = \sum_{y\in Im(X)} Pr(X^{-1}(y))\cdot y$$

טענה: התוחלת היא פונקציה ליניארית, כלומר $\mathbb{E}(a\cdot X) = a\cdot\mathbb{E}(X)$ וכן $\mathbb{E}(X+Y) = \mathbb{E}(X)+\mathbb{E}(Y)$. נוכיח זאת:

$$\begin{aligned}
\mathbb{E}(X+Y) &= \textstyle\sum_{n\in Im(X+Y)} Pr(X+Y=n)\cdot n = \\
&= \textstyle\sum_{x\in Im(X)}\sum_{y\in Im(Y)} Pr(X=x,Y=y)\cdot(x+y) = \\
&= \textstyle\sum_{x\in Im(X)}\sum_{y\in Im(Y)} Pr(X=x,Y=y)\cdot x + ... = \\
&= \textstyle\sum_{x\in Im(X)} x\cdot\sum_{y\in Im(Y)} Pr(X=x,Y=y) + ... = \\
&= \textstyle\sum_{x\in Im(X)} x\cdot Pr(X=x) + ... = \mathbb{E}(X)+\mathbb{E}(Y)
\end{aligned}$$

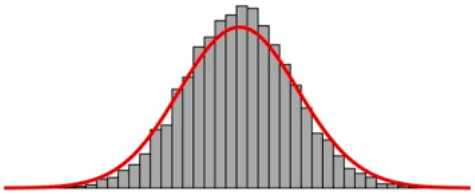
וכן

$$\mathbb{E}(a\cdot X) = \sum_{b\in Im(a\cdot X)} Pr(a\cdot X = b)\cdot b = \sum_{\frac{b}{a}\in Im(X)} Pr(X=\frac{b}{a})\cdot b = \sum_{x\in Im(X)} Pr(X=x)\cdot a\cdot x = a\cdot\mathbb{E}(X)$$

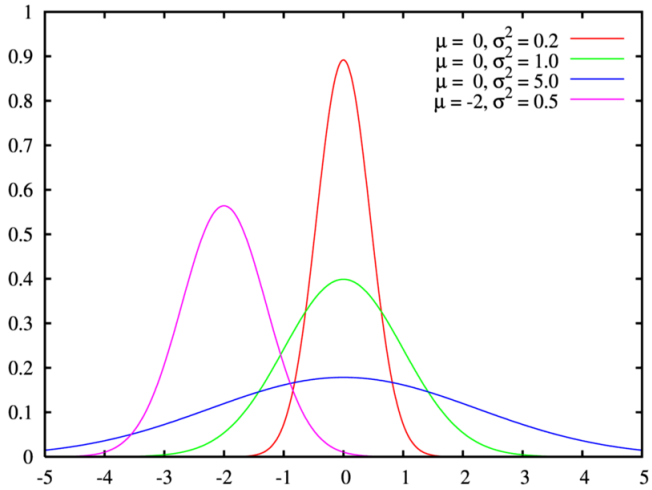
כלומר, התוחלת היא פונקציה ליניארית.

נוסיף הגדרה נוספת:

<u>שונות</u> - השונות של משתנה מקרי היא מדד ל"כמה הוא מפוזר", כלומר, כמה הוא רחוק מהתוחלת שלו. נרצה לדעת, בממוצע, מהו המרחק של המשתנה מהממוצע שלו. ההגדרה נתונה ע"י $Var(X) = \mathbb{E}((X - \mathbb{E}(X))^2)$. היא מתארת את "הרוחב של ההתפלגות".



איור 1:



איור 2:

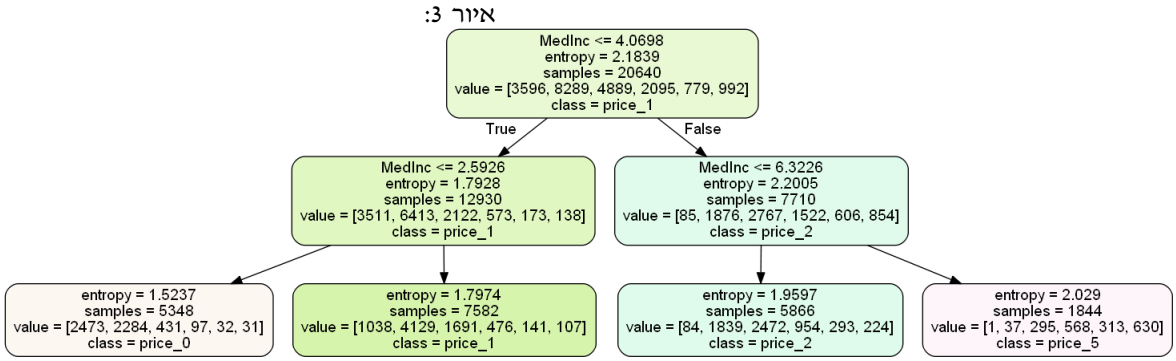
2 עצי החלטה

2.1 עצי החלטה

כעת נראה מודל מורכב יותר, ושימושי מאוד: עץ ההחלטה. ראשית נבין כיצד נראה מודל עץ החלטה מאומן. זהו עץ בינארי, המקבל נקודה במדגם וממיין אותה לפי התכונות שלה: כל צומת בעץ היא התייחסות לאחת התכונות של הדוגמא, ונראית מהצורה, `האם התכונה $x$ גדולה יותר מ־$y$, או קטנה יותר?` בהתאם לשתובה לשאלה הדוגמא ממשיכה לאחד הילדים של הצומת. לכל עלה בעץ מוקצה ערך, בו נשתמש לפרדיקציה עבור הדוגמאות שיגיעו אליו.

כך נראה עץ מאומן:

איור 3:



נעסוק בחלק זה בעץ החלטה לחיזוי בבעיות קטגוריאליות. בדוגמא לעיל, כל דוגמא בסט האימון נשלחת לאחת מ־6 קטגוריות.

כעת נבין כיצד מאמנים עץ החלטה. האימון נעשה באופן חמדני, כאשר המשימה שלנו היא למעשה לבחור, עבור צומת נתונה בעץ, לפי איזו תכונה של ה־data לפצל אותו, ואיפה לקבוע את הסף. לרוב הפיצול מתבצע לפי מדד gini. הוא מוגדר על סט מידע T, והוא מדד של ההתסברות לקבל תיוג לא נכון, אם התיוג של דוגמא יקבע באקראי לפי ההסתברות לכל תיוג אפשרי. כלומר, אם ישנם k תיוגים, עם הסתברות p_i לקבלת התיוג ה־i ב־T, נקבל כי מדד ג'יני הוא

$$gini(T) = p_1 \cdot (1-p_1) + ... + p_k \cdot (1-p_k) = \sum_i p_i(1-p_i) = \sum_i p_i - p_i^2 = 1 - \sum_i p_i^2$$

הערך המקסימלי של מדד ג'יני, בהינתן k תיוגים שונים, הוא

$$1 - \sum_i (\frac{1}{k})^2 = 1 - k \cdot \frac{1}{k^2} = 1 - \frac{1}{k}$$

כאשר הערך המינימלי הוא 0. באימון של עץ החלטה אנו רוצים לפצל את T ל־T_1, T_2, כך ש־ $\frac{|T_1|}{|T|} \cdot gini(T_1) + \frac{|T_2|}{|T|} \cdot gini(T_2)$ יהיה מינימלי. כדי לבנות פיצולים בעץ עלינו לבדוק את כל ה־features האפשריים ואת כל הספים האפשריים, ולבחור את החלוקה האופטימלית. נשים לב כי בבדיקה של כל הספים אנו יכולים לנצל את היות המידע שלנו סופי ודיסקרטי, כך שאין טעם לבדוק את כל ציר הממשיים בחיפוש אחר סף ־ מספיק לבדוק את הערכים השונים שמתקבלים ב־feature אותו אנו בודקים.

כעת אנו יודעים כיצד לבנות פיצולים בעץ. נוכל לפתח את העץ עד לקבלת עלים שהם אחידים בתיוג שמתקבל בהם. התיוג שינתן בעץ בעלים אלו יהיה התיוג שהתקבל בהם על סט האימון. נוכל גם לעצור קודם לכן, ולתת כתיוג את הצבעת הרוב. התיוג שיתקבל ב־test יקבע לכן לפי התיוג שהתקבל בעלה אליו הגענו על ה־train .

נכיר כעת מדד נוסף לפיו ניתן לקבוע כיצד לפצל את המידע בעץ: מדד information gain.
נסמן פיצול ב־feature כלשהו ב־a, ואז

$$IG(T,a) = \underbrace{H(T)}_{entropy\,of\,T} - \underbrace{H(T|a)}_{weighted\,sum\,of\,entropy\,(children)}$$

אנטרופיה היא מדד לאי־סדר. ככל שהאנטרופיה גדולה יותר, כך התיוגים `מעורבבים` יותר, וההתפלגות שלהם מגוונת יותר. האנטרופיה נתונה ע`י

$$H(T) = -\sum_i p_i \cdot log(p_i)$$

ניתן לחשוב על האנטרופיה כך: נרצה לתאר את מרחב ההסתברות שלנו בביטים. בקידוד אידיאלי, כל אירוע יקודד בהתאם להסתברות שלו להתקבל: ככל שאירוע הוא סביר יותר, נרצה לקודד אותו בפחות ביטים. ניתן לראות כי בקידוד אידיאלי, כמות הביטים הדרושה לקידוד אירוע שמתקבל בהסתברות p_i היא $-log(p_i)$. לכן $H(T)$ היא התוחלת של כמות הביטים לקידוד של אירוע ממרחב ההסתברות. ככל ש־$H(T)$ קטנה יותר, כך מרחב ההסתברות `מסודר` יותר, כלומר יש אירועים שהם סבירים בהרבה מאחרים.
הביטוי ל־information gain הוא אם־כן

$$IG(T,a) = -\sum_i p_i log(p_i) + \sum_a p(a) \cdot \sum_i Pr(i|a) \cdot log(Pr(i|a))$$

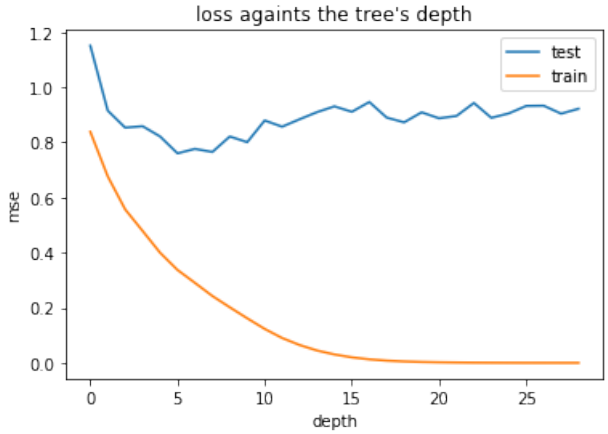
נרצה לבחור את הפיצול עם ה־information gain המקסימלי. כלומר, את הפיצול `שמסדר` את המידע הכי טוב.
נדגים: המידע נתון ע`י

x	y
1	0
2	1
2.5	0
3	0
3.2	1
4	1

נבדוק את הפיצול ב־$x \leq 2.5$:

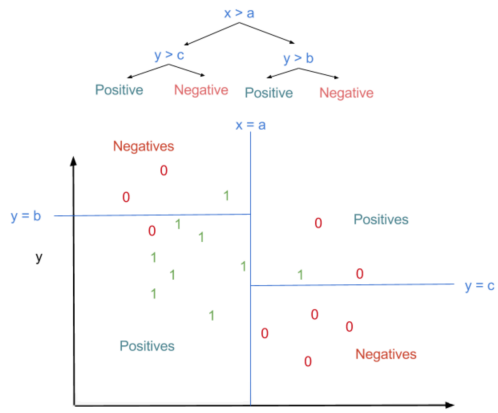
$$IG(T,a) = \underbrace{-(0.5 \cdot log(0.5) + 0.5 \cdot log(0.5))}_{H(T)} +$$
$$\underbrace{+(0.5 \cdot (\frac{2}{3} \cdot log(\frac{2}{3}) + \frac{1}{3} \cdot log(\frac{1}{3})) + 0.5 \cdot (\frac{1}{3} \cdot log(\frac{1}{3}) + \frac{2}{3} \cdot log(\frac{2}{3}))}_{H(T|a)} \approx 0.056$$

ראינו אם כן דרכים שונות לפיצול של העץ. נשים לב לשתי נקודות: האחת, שתי הבניות הן בניות חמדניות: כלומר, כל פיצול נבנה מתוך מטרה להיות הפיצול הטוב ביותר בפני עצמו, אך אין בכך כדי להבטיח כי העץ כולו יהיה העץ האידיאלי.
השנייה היא סכנת ה־overfit במודל המתואר: הרי ברור כי אם אנו מפתחים את עץ ההחלטה עד למצב בו כל עלה מכיל תיוג אחד (במצב קיצוני, כל עלה יכיל גם דוגמא אחת בלבד), אנו עשויים בהחלט להימצא במצב של overfit. נראה זאת בדוגמה הבאה:



איור 4:

בדוגמא ניתן לראות כיצד, על אף שה־Loss על סט האימון הולך וקטן, ה־Loss על ה־test דווקא עולה כתלות בעומק. אנו יודעים כי בעץ בינארי מלא בעומק n יש 2^n עלים. לכן נוכל לקבל מכך חסם נאיבי משוער לעומק העץ: לא נרצה לקבל עצים בהם כל עלה מכיל דוגמא יחידה, כך שלא נרצה עצים בעומק $log_2|dataset|$.
כיצד באופן כללי ניתן להימנע מהתאמת יתר בעצים?
דרך פשוטה במיוחד היא לבחור מראש עומק מקסימלי לעץ. ניתן גם לבחור כמות מינימלית של דוגמאות בכל עלה. הרעיון הכללי של צמצום העץ נקרא pruning, כאשר שיטות מסוימות אלו הן pruning top down. ניתן לבצע גם pruning bottom up, ע`י בחירה של תת־עץ מתוך העץ שמתקבל.
נוסיף הערה לגבי אופן הויזואליזציה של עצים: אחד מהיתרונות המשמעותיים של עצי החלטה הוא הקלות היחסית בה ניתן לחקור אותם, בין השאר ע"י הצגה שלהם בגרף. ראינו קודם לכן הצגה של עץ החלטה כתהליך לוגי, כעץ בינארי של תנאים. ניתן להציג עץ החלטה גם באופן הבא:



איור 5:

כאשר הקווים באיור הם הערכים לפיהם מתבצע פיצול בעץ, והתחומים שהם מגדירים הם העלים שיתקבלו בעץ.

2.1.1 חשיבות הפיצ'רים

עץ החלטה יכול לתת לנו באופן פשוט למדי מדד לחשיבות של הפיצ'רים שלנו למודל: לכל פיצ'ר נוכל להתבונן בכל הפיצולים בהם השתמשנו בו. בכל פיצול חישבנו במהלך הבנייה של העץ את העלייה במדד gini (או את ה־information gain). נוכל לסכום את הגדלים הללו לכל פיצ'ר בנפרד ולקבל עבורו מספר: ככל שהמספר גבוה יותר, פירוש הדבר שהפיצ'ר היה משמעותי יותר בבניית העץ. נוכל לנרמל את הגדלים שקיבלנו כך שסכומם יהיה 1. לגדלים המנורמלים הללו נתייחס כאלל החשיבות של כל פיצ'ר במודל.

מדד זה הוא שימושי ביותר, במיוחד כאשר ברצוננו לבצע feature selection: כלומר, כאשר יש לנו יותר פיצ'רים משהיינו רוצים, ואנו רוצים להישאר רק עם הפיצ'רים הטובים ביותר. פתרון נאיבי (גם אם לא ממש מומלץ) יהיה לדרג פיצ'רים לפי החשיבות שלהם למודל כלשהו, בטרם נאמן על הפיצ'רים הנבחרים מודל אחר.

שימוש נוסף במדד ה־feature importance הוא בניסיון שלנו להסביר לאחרים או לעצמנו את התחזיות של המודל. נוכל לקבל כך תובנות בסיסיות אך חשובות על הבעיה שלנו.

הערה אחרונה על עץ החלטה: עץ החלטה הוא אלגוריתם מאוד גמיש, ולמעשה ניתן לאמן עץ החלטה גם עבור משימות רגרסיה לחיזוי ערך ממשי, ולא רק לחיזוי קטגוריאלי. הרעיון דומה מאוד, כאשר את הפיצול עושים על בסיס מדדים אחרים: לרוב לפי מזעור של mse. את הערך בעלים קובעים לפי הממוצע בהם.

עץ החלטה הוא אלגוריתם מאוד שימושי. מעבר להיותו בעל יכולת ביטוי גבוהה, הוא מאפשר דירוג של חשיבות ה־features והוא מודל פשוט להבנה ולהסבר - יש לתכונות אלה חשיבות רבה בבנייה של מודלים לצרכים מעשיים, כאשר יש לנו הרצון להיות מסוגלים להסביר איך

בפועל המודל מתנהג. עץ החלטה הוא גם אבן היסוד של אלגוריתם נוסף ונפוץ מאוד שנכיר מיד: random forest.

2.2 שיטות ensemble ו־random forest

שיטות ensemble מבוססות על הקיום של `חוכמת המונים`, ובמהותן הן משלבות מודלים פשוטים למתן חיזוי המתבסס על החיזויים של המודלים הפשוטים. נראה כעת את טענת הבסיס להצלחה האפשרית של חוכמת ההמונים, טענה הידועה בשם Condorcet's jury theorem. נשתמש בכמה מושגים מתורת ההסתברות בלי להיכנס להסבר מעמיק שלהם, מתוך הנחה שאם טרם למדתם אותם, עוד תפגשו אותם בעתיד הלא רחוק:
יהיו $c_1, ..., c_k$ מסווגים בלתי־תלויים, כאשר ההסתברות של כל אחד מהם לקבל תשובה נכונה היא $p = \frac{1}{2} + \varepsilon$. נניח כי בתשובה נכונה הם מחזירים את הערך 1 ובתשובה שגויה את הערך 0. נגדיר מסווג חדש, $\tilde{c} = Round(\frac{c_1+...+c_k}{k})$ עם $c = Round(\frac{c_1+...+c_k}{k})$. מהו הסיכוי ש־$\tilde{c} = 1$? זהו בדיוק הסיכוי שנקבל כי

$$Pr(c > \frac{1}{2})$$

נוכיח את אי־ששויון צ'בישב, לפיו

$$Pr(|X - \mathbb{E}(X)| \geq C) \leq \frac{Var(X)}{C^2}$$

ראשית נראה כי

$$\mathbb{E}(X^2) \geq C^2 \cdot Pr(|X| \geq C)$$

אכן,

$$\mathbb{E}(X^2) = \sum_{x \in Im(X)} Pr(X = x) \cdot x^2 \geq \sum_{x \in Im(X), |x| \geq C} Pr(X = x) \cdot x^2 \geq$$

$$\geq \sum_{x \in Im(X), |x| \geq C} Pr(X = x) \cdot C^2 = C^2 \cdot Pr(|X| \geq C)$$

כעת נחליף את המשתנה המקרי X במשתנה המקרי $X - \mathbb{E}(X)$ ונקבל כי

$$\mathbb{E}((X - \mathbb{E}(X))^2) \geq Pr(|X - \mathbb{E}(X)| \geq C) \cdot C^2 \rightarrow Pr(|X - \mathbb{E}(X)| \geq C) \leq \frac{Var(X)}{C^2}$$

כעת נחשב את התוחלת והשונות של c. נשים לב כי

$$Pr(c = \frac{k}{n}) = \binom{n}{k} p^k \cdot (1-p)^{n-k}$$

לכן

$$\mathbb{E}(c) = \frac{1}{n} \cdot \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} \cdot k = \frac{1}{n} \sum_{k=1}^{n} \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k} \cdot k =$$

$$= \frac{np}{n} \sum_{k=1}^{n} \frac{(n-1)!}{(k-1)!(n-k)!} p^{k-1} (1-p)^{n-k} = p \cdot \sum_{k=0}^{n-1} \binom{n-1}{k} p^k (1-p)^{n-1-k} \cdot k =$$

$$= p \cdot (p + (1-p))^{n-1} = p$$

החישוב של התוחלת של c מעט מסובך יותר (אך לא בהרבה), ונקבל כי $Var(c) = \frac{p(1-p)}{n}$. נשתמש באי־שוויון צ'בישב עבור c:

$$Pr(|c-p| \geq \delta) \leq \frac{p(1-p)}{n\delta^2} \underset{n\to\infty}{\to} 0$$

כלומר, ככל ש־n גדול יותר, גדלים הסיכויים ש־c יתקרב ל־p. כיוון ש־$p > \frac{1}{2}$, הסיכוי ש־$c > \frac{1}{2}$ שואף ל־1 כש־n שואף ל־∞. לכן, עבור ensemble שפועל בשיטת הצבעת הרוב, ככל שהוא גדול יותר, כך הוא מדויק יותר. נראה דוגמא: נניח שלושה מסווגים בלתי־תלויים הצודקים בהסתברות $p = 0.7$. אזי ההסתברות לתשובה נכונה היא

$$Pr(correct) = 0.3 \cdot 0.7 \cdot 0.7 + 0.7 \cdot 0.3 \cdot 0.7 + 0.7 \cdot 0.7 \cdot 0.3 + 0.7 \cdot 0.7 \cdot 0.7 \approx 0.78$$

אם נשתמש בחמישה מסווגים נקבל כבר כי

$$Pr(correct) \approx 0.83$$

כלומר, כבר עבור מספר קטן של מסווגים אנו מקבלים שיפור משמעותי. איפה הטענה הזו נופלת במבחן המציאות? בהנחת האי־תלות. אכן, במציאות המסווגים שלנו לעתים רחוקות מאוד יהיו בלתי־תלויים לחלוטין. אם זאת, עקרון מנחה בבנייה של ensemble מוצלח הוא יצירת גיוון גדול ככל האפשר בין המסווגים. יש שיטות רבות לעשות זאת. נדון בהן במסגרת הדיון על random forest. אחר כך נציג שיטה נוספת ללמידת ensemble.

2.2.1 random forest

אלגוריתם זה הוא מהגרסאות הנפוצות ביותר של אלגוריתמים קלאסיים בלמידת מכונה באופן כללי, ובפרט בתחום של ensemble learning. הבנייה של random forest היא פשוטה: בונים אוסף של עצי החלטה, ומחילים חוק של הצבעת הרוב על התחזיות שלהם. האתגר הוא, כפי שטענו קודם לכן, לבנות אוסף עצי החלטה שיהיו מספיק שונים זה מזה, כדי להקטין את התלות ביניהם. ישנן כמה שיטות נפוצות לעשות זאת:

1. אימון כל עץ החלטה על תת־סט אחר של סט האימון.
2. אימון כל עץ על תת־סט שנדגם מתוך סט האימון.

3. אימון כל עץ החלטה עם תת־סט של הפיצ'רים של המידע.

4. בכל פיצול, במקום לבדוק את כל הפיצולים האפשריים, לבדוק פיצולים בחלק אקראי של אוסף הפיצ'רים.

ניתן לחשוב על שיטות רבות נוספות. למשל, ניתן לאמן עצים שונים ביער עם קריטריוני פיצול שונים.

2.2.2 שימושים ב־random forest לבניית מטריקה

מטריקה היא פונקציית מרחק, שעבור שתי נקודות, מחזירה את המרחק ביניהן. על מרחב המידע שלנו מוגדרת באופן טבעי, מטריקה - זו המטריקה האוקלידית. כלומר, מרחקים מוגדרים ע"י

$$d(p,q) = |p-q|$$

פעמים רבות נרצה לעשות שימוש במטריקות נוספות, או בפונקציות "דמויות מטריקה". נעסוק בכך עוד בהמשך. בינתיים נשים לב שבהינתן random forest מאומן, עם עצים $c_1, ..., c_k$, נוכל להגדיר מרחק בין שתי נקודות מידע ע"י

$$d(p,q) = 1 - \frac{\sum_{i=1}^{k} \delta(c_i(p), c_i(q))}{k}$$

כלומר, המרחק בין נקודות יהיה גודל החלק של העצים שלא מסכימים בסיווג שתי הנקודות. לכן, ככל שהדוגמאות "דומות" יותר, נקבל כי המרחק ביניהן אכן קטן יותר, כאשר

$$d(p,p) = 0$$

נקודה נוספת למחשבה: היינו רוצים לחשוב שככל שהאלגוריתם שלנו "יעבוד יותר", כך הוא יקבל תוצאות יותר טובות. ראינו כבר במספר דוגמאות כי תפיסה זו שגויה: קל לנו להגיע למצב בו נותנים לאלגוריתם "לעבוד יותר", למשל, לבנות עץ עמוק ככל האפשר, אך התוצאה המתקבלת היא התאמת יתר על סט האימון, במקום מודל טוב. ה־random forest מאפשר לנו לקחת את האלגוריתם הפשוט של עץ החלטה ולגרום לו "לעבוד יותר" בתמורה למודל טוב יותר.

2.2.3 stacking

דרך נוספת לשלב תוצאות של כמה מודלים היא ע`י אימון מודל חדש על תוצאות המודלים. כלומר, אחרי אימון של המודלים $c_1, ..., c_k$ על סט האימון, נוכל לאמן מודל c, כאשר הפיצ'רים שיתארו את המידע הפעם הם התחזיות של המודלים $c_1, ..., c_k$ עליו. כאשר משתמשים בשיטה זו יש לנקוט משנה זהירות כנגד overfit. מוטב, למשל, שלא לאמן את

המודל c על סט האימון בו השתמשנו לאימון המודלים $c_1, ..., c_k$, אלא להשתמש בסט אימון חדש, שהמודלים שלנו לא התייחסו אליו עד כה. באמצעות שימוש ב־stacking ניתן ליצור מודל סופי מורכב יותר, דבר שלאו דווקא ניתן לעשות עם מודלים פשוטים. למשל, ניסיון ליצור מודל מורכב יותר תוך שימוש בעץ החלטה יחיד יוביל בהכרח להעמקת העץ, דבר שקרוב לוודאי שיביא אותנו להתאמת יתר.

2.3 הרחבת הדיון על overfit

כאמור, תופעה של overfit היא תופעה בה המודל שאימנו מצליח על סט האימון טוב יותר בצורה משמעותית מאשר על סט המבחן. למעשה, פעמים רבות המודל אכן יצליח טוב יותר על סט האימון: זאת כיוון שכמעט תמיד ההתפלגות של סט המבחן שונה מעט מזו של סט האימון, בין אם מתופעות הקשורות בהיות המדגם שלנו סופי, ובין אם ישנו שוני ממשי בין שני הסטים.

נוכל להבחין בתופעה הזו כאשר, כתלות בפרמטר מסוים, השגיאה על סט האימון הולכת וקטנה, בעוד היא אינה משתנה, או הולכת וגדלה, על סט המבחן. ראינו דוגמא לכך בגרף למעלה, בו הצגנו את השגיאות השונות כתלות בעומק העץ. למבחן מסוג זה יש יותר משמעות בבחינה של אלגוריתם שיש בו שלב איטרטיבי כלשהו של אימון, כאשר הדוגמא הבולטת ביותר לכך היא רשתות נוירונים, בהן ניתן לבחון את השגיאה כתלות בשלב בו אנו נמצאים באימון. נרחיב על כך עוד בהמשך.

כיצד ניתן להתמודד עם התאמת יתר? ברמה הבסיסית, באמצעות רגולריזציה על גוניה השונים. ראינו במודל מבוסס gradient descent רגולריזציה על המשקולות, בה דרשנו מהמשקולות להיות קטנות, כתנאי שאמור היה לסייע לנו להגביל את המודל. בעצי החלטה ראינו הגבלות מסוג אחר - על עומק העץ, או על גודל העלים שלו. הרעיונות הללו דומים, ומקורם בניסיון להגביל את המודל שנבנה.

ברוב המודלים שנתעניין בהם התאמת היתר היא סכנה אמיתית. בתהליך שגרתי של התאמת מודל, פעמים רבות נרצה ראשית לראות כי אנו מסוגלים להגיע ל־overfit, כלומר מסוגלים ללמוד את הסט האימון, לפני שנתחיל להגביל את המודל כדי להוריד את התאמת היתר.