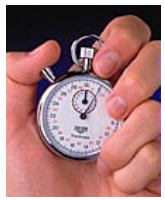
# למידה

# מהי למידה?

- למידה הינו תהליך
- בדרך כלל הקלט של התהליך הינן התנסויות (דוגמאות)
- בעקבות התהליך חל שינוי בלומד
- יתכן שינוי לטובה או לרעה. אם השינוי הוא לטובה, הלמידה נחשבת מוצלחת.
- השינוי נמדד בדרך כלל ביכולת לבצע קבוצת משימות



# מדדים



- יתכנו מדדים רבים לביצוע קבוצת משימות.
- יתכן שהלומד משתפר לפי מדד אחד (למשל אורך הפתרונות הממוצע), אך נעשה גרוע יותר לפי מדד אחר (למשל מהירות ממוצעת של מציאת הפתרון).
- על כן, מדד ההערכה הינו חלק בלתי נפרד מתהליך הלמידה.
- יתכן שנרצה לשפר את יכולתו הפוטנציאלית של הלומד לפתרון כל בעיה אפשרית אך בד"כ יקשה הדבר על תהליך הלימוד.
- לכן, שיפור היכולת נמדד בד"כ לגבי קבוצה (או התפלגות) של משימות.



# הגדרה

למידה הינה תהליך המקבל התנסויות כקלט, ומבצע שינויים בבסיס ידע במטרה לשפר, על פי מדד נתון, את היכולת הפוטנציאלית של פותר הבעיות המשתמש בבסיס הידע, לפתור קבוצת בעיות.

# למידת אינדוקטיבית של מסווגים מדוגמאות

- תהי $X$ קבוצת אוביקטים. תהי $C \subseteq X$ תת קבוצה שלה הנקראת מושג המטרה .
- מסווג הינו פונקציה בוליאנית $f : X \to \{0,1\}$.
- נגדיר את מסווג המטרה כמסווג הבא: $f_c(x) = \begin{cases} 1 & x \in C \\ 0 & x \notin C \end{cases}$
- בהנתן אוביקט $x \in X$, נאמר שהמסווג שוגה כאשר $f(x) \neq f_c(x)$
- בהנתן קבוצת אוביקטים $X_T \subseteq X$ נגדיר את דיוק המסווג כמספר: $\left|\{x \in X_T \mid f(x) = f_c(x)\}\right| / \left|X_T\right|$
- אנו מעונינים במסווג בעל דיוק גבוה על קבוצת הבעיות העתידיות.



# "סיווג" במונחי ההגדרה של מערכות לומדות

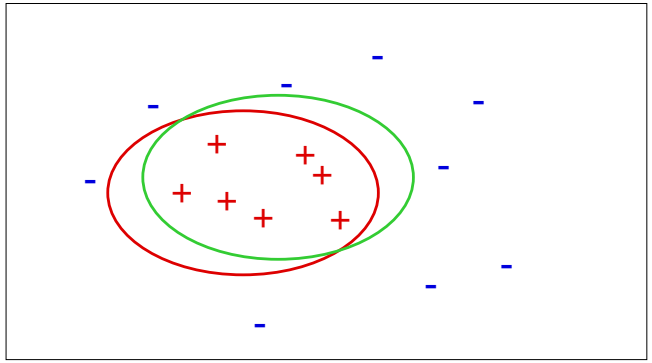
- פותר הבעיות: מסווג.
- קריטריון הערכה: דיוק על קבוצת אוביקטים (נניח כרגע שה"לקוח" מחזיק קבוצת אוביקטים לגביהם הוא יודע את $f_c(x)$ אבל אינו חושף אותה בפנינו. הוא ישתמש בה להערכת המסווג.)



# בעית הלמידה

- נתונה קבוצת דוגמאות מסומנות מתוך קבוצת המטרה $E = \{\langle x_1, f_c(x_1)\rangle, \cdots, \langle x_m, f_c(x_m)\rangle\}$
- אלגוריתם ללמידת מסווגים מקבל כקלט קבוצת דוגמאות מסומנות E ומוציא כפלט מסווג.
- אם E מכיל את כל האוביקטים ב-X אזי ניתן פשוט לשמור אותם בטבלה.
- מכיוון שבד"כ נתונה תת קבוצה של X צריך אלגוריתם הלמידה להכליל : להסיק מתוך דוגמאות שראה לדוגמאות שלא ראה.
- כדי שניתן יהיה להכליל מגדירים בד"כ קבוצת תכונות : אוסף של פונקציות הממפות איברים ב-X לתחום סופי (בינתיים)
- דוגמאות הלמידה הינם זוגות $\langle x_i, f_c(x_i)\rangle$ כאשר $x_i$ מיוצג ע"י ווקטור של ערכי התכונות.





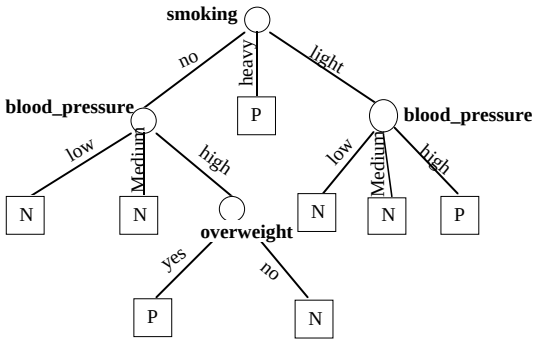
# דוגמא

המושג: "בעל סיכון גבוה להתקף לב"

```
(setf *attributes*
    '(smoking overweight blood_pressure coffee_drinker))
(setf *domains* '((smoking (no light heavy))
                  (overweight (yes no))
                  (blood_pressure (low medium high))
                  (coffee_drinker (yes no))))
(setf *examples* '(((heavy no low yes) p)
((no yes high yes) p) ((light yes low no) n)
((heavy yes low yes) p) ((heavy no low no) p)((light no yes no) n) ((light yes high yes) p)
((heavy yes medium yes) p) ((no no high yes) n)((heavy yes medium no) p) ((no yes medium yes) n)
((light yes medium yes) n) ((no yes medium no) n)((light yes medium no) n)((heavy yes high no) p)
((light yes high no) p)((no no low yes) n)((light no medium no) n)((no no low no) n)
((light no yes yes) n)((no yes high no) p)
((light yes low yes) n) ((heavy yes low no) p)
((no no high no) n) ((heavy yes high yes) p)
((light no medium yes) n)((no yes low yes) n)
((no yes low no) n)((no yes medium yes) n)
((no yes medium no) n)((light medium yes yes) n)
((light medium yes no) n)))
```



# עץ החלטה






# מסווגים

- מסווג מייצג הכללה מעל קבוצת הווקטורים.
- קיימות דרכים רבות ליצוג מסווגים:
  - עצי סיווג
  - קבוצת חוקים
  - תכנית פרולוג (אוסף פסוקיות הורן)
  - רשתות עצביות
  - קבוצת דוגמות מסווגות
- לכל אחד מהיצוגים קיימים אלגוריתמי למידה רבים.

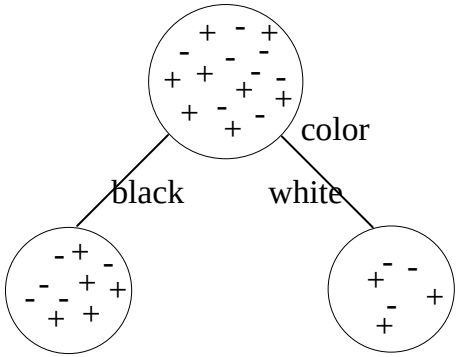


# ID3 - תכנית ללמידת עצי החלטה מדוגמאות

- ID3 הינה התוכנית הידועה ביותר ללמידת מסווגים
- התכנית מקבלת כקלט קבוצת דוגמאות
- התכנית מוציאה כפלט עץ החלטה המסווג נכון את כל הדוגמאות שנצפו
- התוכנית מתחילה משורש העץ ומפתחת תתי עצים.
- ID3 תפצל צומת אם הוא מכיל דוגמאות חיוביות ושליליות ותעצור אם כל הדוגמאות הן מאותו סוג
- כל צומת מתפצל על תכונה מסוימת. התכנית יוצרת תת עץ לכל ערך אפשרי של התכונה. קבוצת הדוגמאות מתחלקת לפי ערכי התכונה ותת-הקבוצות מועברות לתת העצים.
- ההחלטה על איזו תכונה לפצל מתבססת על יוריסטיקה המשתמשת בתורת האינפורמציה: נבחרת התכונה המביאה לתוספת מקסימלית של אינפורמציה.

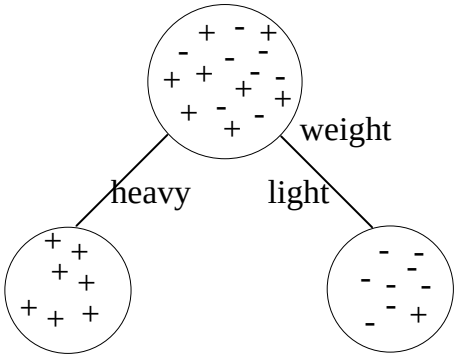
# תכונות אינפורמטיביות

- תכונה שאינה משפיעה על החלוקה של קבוצת הדוגמאות אינה אינפורמטיבית:






- תכונה שמחלקת את קבוצת הדוגמאות בצורה טובה הינה תכונה אינפורמטיבית






# תוספת האינפורמציה

- בכל צומת ID3 בודקת על איזו תכונה כדאי לפצל.
- לכל תכונה נמדדת תוספת האינפורמציה. התכונה המביאה לתוספת האינפורמציה הגדולה ביותר נבחרת לפיצול.
- אם p הינו מספר הדוגמאות החיוביות בצומת ו- n הניו מספר הדוגמאות השליליות אזי ההסתברות ליפול בקבוצת בחיוביים מוגדרת לכל צומת ע"י:

$$\frac{p}{p+n}$$

ובאופן דומה לגבי קבוצת השליליים:

$$\frac{n}{p+n}$$



- אי הוודאות בצומת נמדדת ע"י נוסחאת שנון:

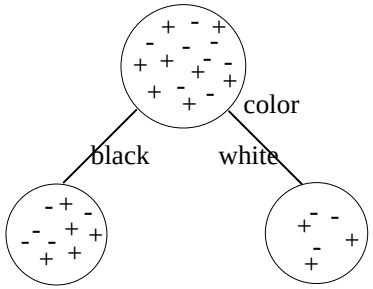
$$I(p,n):=-\frac{p}{p+n}\cdot\log_2\frac{p}{p+n}-\frac{n}{p+n}\cdot\log_2\frac{n}{p+n}$$

- אי הוודאות בבנים נמדדת ע"י הממוצע המשוקלל של אי הוודאות בכל אחד מהם.
- תוספת האינפורמציה מוגדרת ע"י אי הוודאות צומת פחות אי הוודאות בבנים:

$$GAIN=I(p,n)-\sum_{i=1}^{n}\frac{E_i}{|examples|}\cdot I(p_i,n_i)$$

כאשר Ei הוא מספר הדוגמאות שעברו לבן ה-i, pi הינו מספר הדוגמאות החיוביות בבן ה- i ו-ni הינו מספר הדוגמאות השליליות בבן ה-i.

# דוגמא 1




$$I(p,n):=-\frac{8}{8+8}\cdot\log_2\frac{8}{8+8}-\frac{8}{8+8}\cdot\log_2\frac{8}{8+8}=1$$

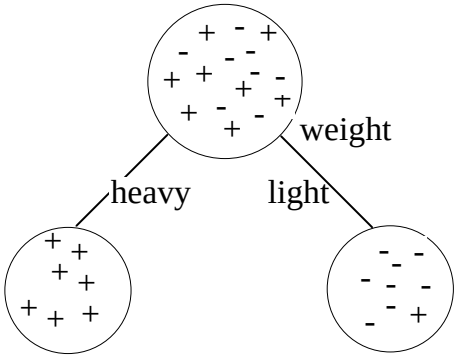
$$I(p_1,n_1):=-\frac{5}{5+5}\cdot\log_2\frac{5}{5+5}-\frac{5}{5+5}\cdot\log_2\frac{5}{5+5}=1$$

$$I(p_2,n_2):=-\frac{3}{3+3}\cdot\log_2\frac{3}{3+3}-\frac{3}{3+3}\cdot\log_2\frac{3}{3+3}=1$$

GAIN=1-(1*6/16+1*10/16)=1-1=0



# דוגמא 2






$$I(p,n):=-\frac{8}{8+8}\cdot\log_2\frac{8}{8+8}-\frac{8}{8+8}\cdot\log_2\frac{8}{8+8}=1$$

$$I(p_1,n_1):=-\frac{1}{8+1}\cdot\log_2\frac{1}{8+1}-\frac{8}{8+1}\cdot\log_2\frac{8}{8+1}=0.50$$

$$I(p_2,n_2):=-\frac{7}{7+0}\cdot\log_2\frac{7}{7+0}-\frac{0}{7+0}\cdot\log_2\frac{0}{7+0}=0$$

GAIN=1-(0.5*9/16+0*7/16)=0.72



# ID3

Let I be the set of training examples
Let $A=A_1,\ldots,A_n$ be all possible attributes
let $V_1,\ldots,V_n$ be domains for attributes
(where $V_i=\{V_{i1},\ldots,V_{iki}\}$)

**ID3 (examples, Att)**

```
if examples = {} then return leaf(null)
P← positive examples N← negative examples
if N = { } then return leaf(P)
if P = { } then return leaf(N)
For each Ai in Att compute gain(Ai)
Select A' such that gain(A',examples) is maximal
for each vi in V' compute
    Ei = {e in examples | A'(e) = vi}
Si = ID3(Ei, Att-{A'})
N ← new-node()
test(N) ← A'
children(N) ← {<Vi,Si> | i=1...n}
Return N
```



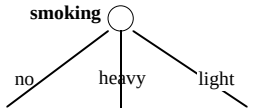
**gain(A, examples)**

```
p← |positive examples|
n← |negative examples|

for each vi in V compute
    Ei = {e ∈ examples | A(e) = vi}
    pi = |positive examples in Ei|
    ni = |negative examples in Ei|

gain ← I(p,n) - E(A)
```



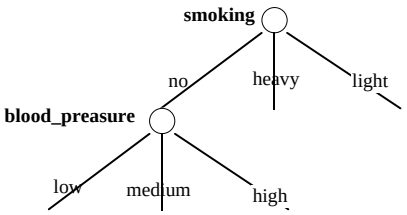
# דוגמא: מעקב אחרי ID3

```
-------------------------------
Choosing from : (SMOKING OVERWEIGHT BLOOD_PRESSURE COFFEE_DRINKER)

SMOKING              p:  12  n:  20  IPN:  0.954
  NO                 p:   2  n:  10  IPN:  0.650
  LIGHT              p:   2  n:  10  IPN:  0.650
  HEAVY              p:   8  n:   0  IPN:  0.000

GAIN:   0.467
-----------------------------

OVERWEIGHT           p:  12  n:  20  IPN:  0.954
  YES                p:  10  n:  12  IPN:  0.994
  NO                 p:   2  n:   8  IPN:  0.722

GAIN:   0.045
-----------------------------

BLOOD_PRESSURE       p:  12  n:  20  IPN:  0.954
  LOW                p:   4  n:   6  IPN:  0.971
  MEDIUM             p:   2  n:  10  IPN:  0.650
  HIGH               p:   6  n:   2  IPN:  0.811

GAIN:   0.204
-----------------------------

COFFEE_DRINKER       p:  12  n:  20  IPN:  0.954
  YES                p:   6  n:  10  IPN:  0.954
  NO                 p:   6  n:  10  IPN:  0.954

GAIN:   0.000
-----------------------------
Selected attribute:  SMOKING
```





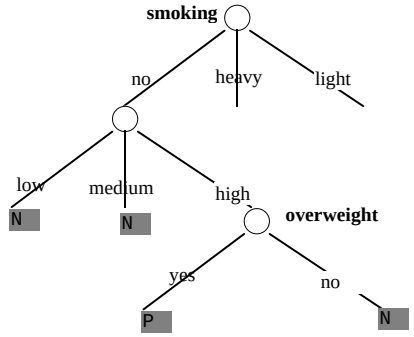

```
-------------------------------
Choosing from : (OVERWEIGHT BLOOD_PRESSURE COFFEE_DRINKER)

OVERWEIGHT           p:   2  n:  10  IPN:  0.650
  YES                p:   2  n:   6  IPN:  0.811
  NO                 p:   0  n:   4  IPN:  0.000

GAIN:   0.109
-----------------------------

BLOOD_PRESSURE       p:   2  n:  10  IPN:  0.650
  LOW                p:   0  n:   4  IPN:  0.000
  MEDIUM             p:   0  n:   4  IPN:  0.000
  HIGH               p:   2  n:   2  IPN:  1.000

GAIN:   0.317
-----------------------------

COFFEE_DRINKER       p:   2  n:  10  IPN:  0.650
  YES                p:   1  n:   5  IPN:  0.650
  NO                 p:   1  n:   5  IPN:  0.650

GAIN:   0.000
-----------------------------
Selected attribute:  BLOOD_PRESSURE
```



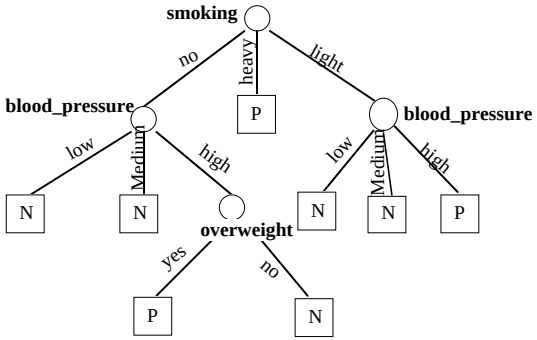
```
Choosing from : (OVERWEIGHT BLOOD_PRESSURE COFFEE_DRINKER)

OVERWEIGHT            p:   2  n:  10  IPN:  0.650
  YES                 p:   2  n:   6  IPN:  0.811
  NO                  p:   0  n:   4  IPN:  0.000

GAIN:   0.109
-----------------------------

BLOOD_PRESSURE        p:   2  n:  10  IPN:  0.650
  LOW                 p:   0  n:   2  IPN:  0.000
  MEDIUM              p:   0  n:   6  IPN:  0.000
  HIGH                p:   2  n:   0  IPN:  0.000

GAIN:   0.650
-----------------------------

COFFEE_DRINKER        p:   2  n:  10  IPN:  0.650
  YES                 p:   1  n:   5  IPN:  0.650
  NO                  p:   1  n:   5  IPN:  0.650

GAIN:   0.000
-----------------------------
Selected attribute:  BLOOD_PRESSURE
```



```
-------------------------------
Choosing from : (OVERWEIGHT COFFEE_DRINKER)

OVERWEIGHT            p:   2  n:   2  IPN:  1.000
YES                   p:   2  n:   0  IPN:  0.000
NO                    p:   0  n:   2  IPN:  0.000

GAIN:   1.000
-----------------------------

COFFEE_DRINKER        p:   2  n:   2  IPN:  1.000
YES                   p:   1  n:   1  IPN:  1.000
NO                    p:   1  n:   1  IPN:  1.000

GAIN:   0.000
-----------------------------
Selected attribute:  OVERWEIGHT
```








# בעיות לתכניות ללימוד מסווגים

- הדוגמאות אינן מייצגות (מוטות)
- קיים רעש בדוגמאות (דוגמאות לא נכונות)
- אין מספיק דוגמאות
- יותר משני סיווגים
- תחומים רציפים של תכונות
- מושגים קשים ללימוד (xor)
- מחירים שונים למבחנים שונים
- ערכים חסרים

# עלים ריקים

- בעיה: יתכן וחלק מהבנים שנוצרים ריקים מדוגמאות (אין דוגמאות המתאימות לחלק מהערכים של התמונה).
- פתרון: כאשר אנו קוראים רקורסיבית לאלגוריתם אנו מעבירים לבן סיווג ברירת מחדל. בדרך כלל זהו הסיווג הדומיננטי בצומת האב.



# רעש בנתונים

- בעיה: יתכן וקים "רעש" בנתונים. קיים עלה בעץ שלא ניתן לפצלו, אך הדוגמאות בו אינן בעלות סיווג אחיד.
- פתרון: בעלה כזה הסיווג יקָבֵע לפי הרוב.



# Gain Ratio

- בעיה: תכונות בעלות קבוצה גדולה של ערכים אפשריים מקבלות "ציון" גבוה ללא הצדקה. דוגמא קיצונית: מס תעודת זהות.
- פתרון: שימוש במדד ש"מעניש" על פיצול למספר רב של בנים. לדוגמא, מדד ה-gain ratio . מדד זה מנרמל את מדד ה-gain ברווח האינפורמציה שאנו מקבלים מעצם הפיצול.

$$\text{split info}(A)=-\sum_{i=1}^{n}\frac{|T_i|}{|T|}\times\log_2\left(\frac{|T_i|}{|T|}\right)$$

$$\text{gain ratio}(A) = \text{gain}(A)/\text{split info}(A)$$

כאשר Ti הינן הקבוצות המתקבלות מפיצול T על ערכי תכונה A.



# טיפול בערכים רציפים (או סדורים)

- ID3 כפי שהוצגה קודם אינה מסוגלת לטפל בערכים רציפים (כגון גיל, משקל, לחץ דם וכו).
- דרך אחת להתגבר על מכשול זה היא דיסקרטיזציה של התחום. לדוגמא, נוכל לחלק את תחום הגיל לתחומים (0-3,3-6,6-12,12-18,18-40,40-120).
- דרך אחרת היא לבצע את הדיסקטיזציה באופן דינמי. כדי להחליט מה ה-gain של תכונה רציפה A נבצע את הפעולות הבאות:

1. תהי X קבוצת הדוגמאות. תהי $\{A(X_{i_1}),...,A(X_{i_n})\}$ קבוצה ממוינת של ערכי הדוגמאות עבור תכונה A.
2. לכל ij בדוק את המבחן : A(x)>A(Xij). מבחן זה מפצל את קבוצת הדוגמאות לשתי קבוצות.

# ערכים חסרים

- בעיה: לחלק מהדוגמאות חסרים ערכים
- פתרונות:
1. ננחש את הערכים החסרים לפי הרוב
2. ננחש לפי הרוב בצומת הנוכחי.
3. ננחש לפי הרוב בצומת הנוכחי בעלי אותה קלסיפיקציה
4. שימוש בהתפלגות - חישוב ההתפלגות יעשה לפי הדוגמאות עבורן ידוע הערך. לדוגמאות עבורן הערך אינו ידוע נותנים ערך אקראי לפי ההתפלגות הנצפית.



# שימוש בתכונות בעלות מחירים שונים

- בעיה: קימים שימושים בהם מחיר חישוב תכונות אינו אחיד. לדוגמא, בדיקת חום ובדיקת צנתור הינן בעלות מחיר שונה (מבחינת סיכון)
- פתרון: בהנתן הגדרת פונקצית מחיר על התכונות, נשקלל את ה-gain במחיר.
- שתי דוגמאות לשקלול כזה במערכות קיימות:
  - $\frac{[Gain(Attribute, Exaples)]^2}{Cost(A)}$
  - $\frac{2^{Gain(A,E)}-1}{[Cost(A)+1]^w}$ כאשר $0 \le w \le 1$ מסמל את החשיבות היחסית של המחיר.



# Overfitting

- פעמים רבות יוצר האלגוריתם עץ גדול מידי בעל כושר פרדיקציה נמוך.
- כדי להתגבר על בעיה זו גוזמים את העץ לאחר יצורו.
- שיטה אחת מניחה בצד חלק מהדוגמאות (לפני הלמידה)
- לאחר הלמידה מנסים לבטל צמתים בעץ. צומת שהסרתו משפרת את הביצועים על קבוצת המבחן נגזם (עם כל תת העץ שתחתיו)
- שיטה דומה אך טובה יותר: הופכים את העץ למערכת חוקים, כאשר כל מסלול מהשורש לעלה מהווה חוק.
- בכל חוק מנסים להוריד את כל אחד מהתנאים ומודדים את הדיוק המוערך לפני ואחרי הגיזום.
- מערכת החוקים לאחר הגיזום יכולה להכיל חוקים סותרים. אחת הדכים להתגבר על כך היא באמצעות מיון של החוקים לפי הדיוק המוערך.



# חלונות

- פעמים רבות מספיק חלק קטן יחסית של קבוצת הדוגמאות כדי ללמוד מסווג טוב
- ID3 משתמשת ב *חלונות* כדי לחסוך במשאבים במקרים כאלו.
- הפרוצדורה שבונה את העץ נקראת עם חלק קטן, אקראי, של קבוצת הדוגמאות (חלק זה נקרא *חלון*)
- העץ שנבנה נבחן על שאר הדוגמאות. אם היתה הצלחה מלאה בסיווג שאר הדוגמאות ID3 עוצרת
- אחרת ID3 מצרפת חלק מהדוגמאות שעץ טעה עליהם לחלון הקודם ויוצרת חלון חדש
- העץ הקודם נזרק ונבנה עץ חדש על פי החלון המורחב.

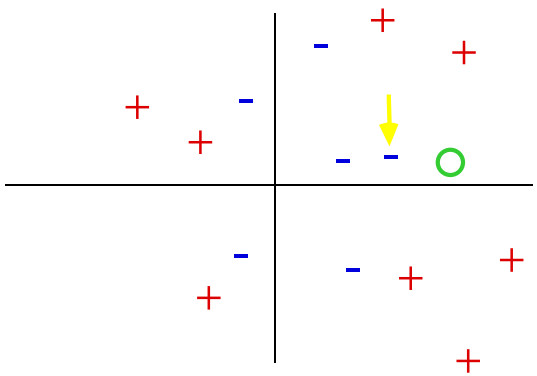
# סיווג מבוסס דוגמאות
# (Instance Based Learning)

- למידה "עצלה": ההכללה מתבצעת בזמן הסיווג.
- ההיפותזה (ההכללה) נעשית לוקלית בסביבת הנקודה (הדוגמא) אותה נרצה לסווג.



# סיווג השכן הקרוב
# (nearest neighbor classification)

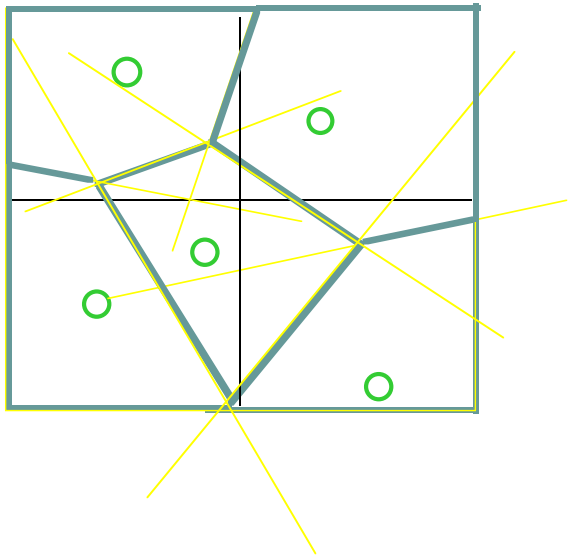
- למידה: שמירת דוגמאות האימון.
- סיווג: בהנתן דוגמא לא מסווגת, סווג אותה על פי השכן המסווג הקרוב ביותר.
- מדד קרבה: מרחק במרחב התכונות.





# דיאגרמת ורונוי
# (Varonoi Diagram)

- דיאגרמה המחלקת את המרחב לתאים
- לכל דוגמא מסווגת מסומן תא המהווה את אוסף הנקודות עבורן הדוגמא הינה השכן הקרוב ביותר.
- הדיאגרמה מבטאת למעשה את ההיפותזה אותה מתאר כלל מסווג השכן הקרוב.





# מדד הדמיון - פונקצית המרחק

- בד"כ משתמשים בפונקצית המרחק האוקלידי:

$$d(x,y)=\sqrt{\sum_{i=1}^{n} h\left(a_i(x),a_i(y)\right)^2}$$

כאשר $a_1(x),\cdots,a_n(x)$ הינן ערכי התכונות של הדוגמא.

- כאשר התכונות מספריות, $h(a_i(x),a_i(y))=a_i(x)-a_i(y)$

כלומר: $d(x,y)=\sqrt{\sum_{i=1}^{n}\left(a_i(x)-a_i(y)\right)^2}$

- כאשר התכונות נומינליות (לא מספריות) נהוג להשתמש ב:

$$h(a_i(x),a_i(y))=\begin{cases}0 & a_i(x)=a_i(y)\\ 1 & a_i(x)\neq a_i(y)\end{cases}$$

# כלל הסיווג ב NN

- נסמן ב-$f(x)$ את הסיווג של דוגמא $x$.
- נסמן ב- $H$ את אוסף הדוגמאות המסומנות.
- $f(y) = f(\arg\min_{x \in H}(d(x,y)))$
- ניתן להשתמש בכלל לפונקצית סיווג בינארית, דיסקרטית עם קבוצת סיווגים **V**, ורציפה.



# נירמול התחומים

- כאשר התכונות הינן ביחידות שונות מקבלות תכונות עם תחום מספרי נרחב משקל גדול ללא סיבה.
- פתרון אפשרי לבעיה הינה נרמול הערכים:
  - נניח שקבוצת הדוגמאות המסווגות הינן $E = \{e_1, \cdots, e_n\}$ כאשר $e_i = \langle \langle v_{1_i}, \cdots, v_{k_i} \rangle, c_i \rangle$ והדוגמא אותה צריך לסווג הינה $\langle v_{1_{n+1}}, \cdots, v_{k_{n+1}} \rangle$
  - נגדיר $\min_j = \min_{i=1,\cdots,n+1} v_{j_i}$ לכל $j = 1, \cdots, k$
  - נגדיר $\max_j = \max_{i=1,\cdots,n+1} v_{j_i}$ לכל $j = 1, \cdots, k$
  - נגדיר $\hat{v}_{j_i} = \dfrac{v_{j_i} - \min_j}{\max_j - \min_j}$
  - נמדוד מרחק על ווקטורי $\hat{v}$

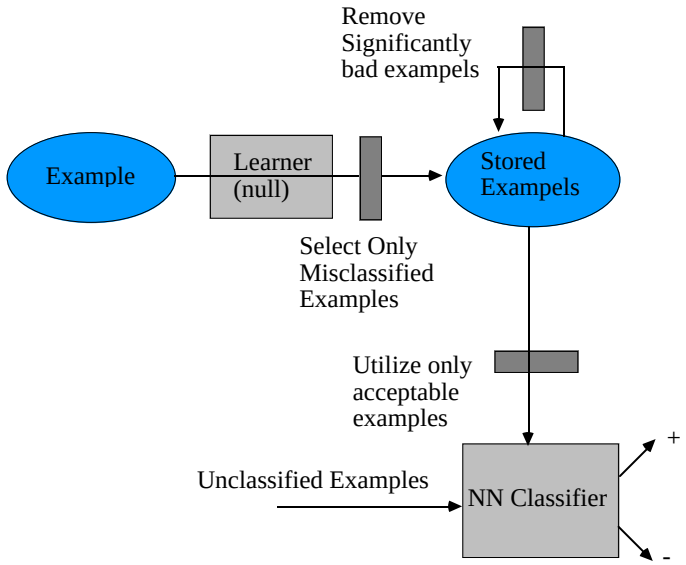


# אלגוריתם IB3

- אלגוריתם הדרגתי (incremental) ללמידת NN.
- האלגוריתם מפעיל מסנן למידה (acquisition filter): רק דוגמאות עליהן טועה המסווג הנוכחי מאוחסנות.
- לכל דוגמא בקבוצת האימון נשמרת הסטורית ההצלחה שלה (כמה דוגמאות סיווגה נכון).
- לכל דוגמא בודקים אם רמת הדיוק שלה גבוהה באופן מובהק מתדירות הסיווג שלה בכלל הדוגמאות.
- אם הדיוק גבוה יותר באופן מובהק הדוגמא מסומנת כקבילה ומשתתפת במסווג.
- אם הדיוק נמוך יותר באופן מובהק הדוגמא מסומנת כבלתי קבילה ונמחקת (זהו סינון שכחה - retention filter).



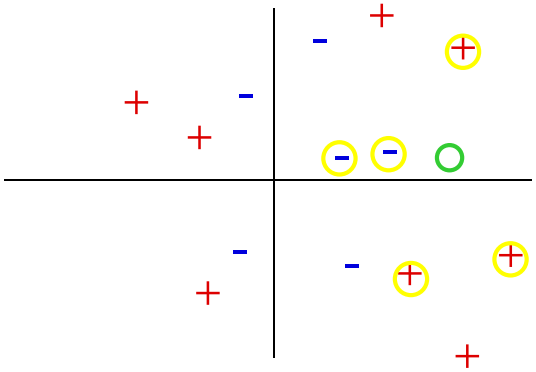
- אם שני המקרים אינם חלים אזי הדוגמא נשמרת "על תנאי".
- היא אינה משתתפת בסיווג, אולם, אם היא קרובה יותר לדוגמא חדשה מאשר השכן הקרוב ביותר מבין הקבילים, בוחנים אותה על הדוגמא החדשה ומעדכנים את ההסטוריה שלה.






# K-nearest Neighbors

- הכללה של אלגוריתם ה-NN.
- הסיווג מוכרע ע"י הצבעה בין k השכנים הקרובים ביותר.



- בדוגמא שלמעלה מסווג השכן הקרוב יחזיר - אולם מסווג 5-NN יחזיר +.



# ביצועי IB3

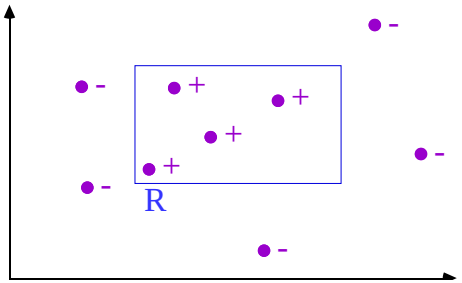
- בקבצי הדוגמאות עליהן נוסה הראה האלגוריתם ביצועים טובים יותר מ-NN עם דרישות זכרון צנועות יותר באופן ניכר.

| | NN | IB3 | IB3 Storage | C4.5 |
|---|---|---|---|---|
| Voting | 91.8 | 91.6 | 0.07 | 95.5 |
| Tumor | 34.7 | 38.6 | 0.16 | 37.8 |
| LED | 70.5 | 71.7 | 0.29 | 68.3 |
| Waveform | 75.2 | 73.8 | 0.15 | 70.7 |
| Cleveland | 75.7 | 78.0 | 0.08 | 75.5 |
| Hungarian | 58.7 | 80.5 | 0.08 | 78.2 |



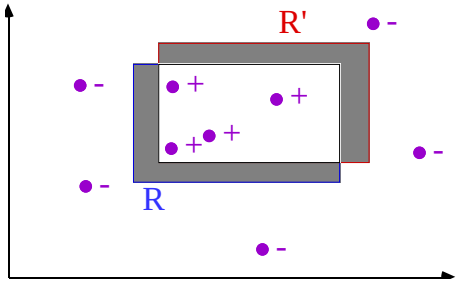
# למידה חישובית

- נניח שברצוננו ללמוד מלבן R במישור (נקרא מלבן המטרה)
- נניח שאנו מקבלים דוגמאות הנדגמות אקראית מהתפלגות D.
- כל דוגמא הינה נקודה בתוספת + או - כדי לציין האם היא בתוך מלבן המטרה.



# שגיאה של השערה

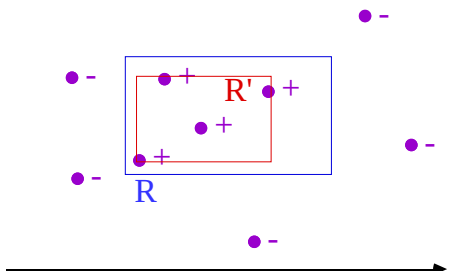
- הקרוב ככל האפשר ל- R' מטרת הלומד לייצר מלבן השערה R.
- נגדיר את השגיאה של R' כהסתברות שנקודה הנדגמת אקראית לפי D תיפול בהפרש הסימטרי של מלבן המטרה ומלבן ההשערה. [ההפרש הסימטרי: $R\Delta R' = (R - R') \cup (R' - R)$]






# אלגוריתם למידה A לחישוב R'

- בהנתן קבוצת דוגמאות, החזר את המלבן הקטן ביותר המכיל את כל הדוגמאות החיוביות.
- כלומר: תהי $S = \{\langle x_1, y_1\rangle, \ldots, \langle x_k, y_k\rangle\}$ קבוצת הדוגמאות החיוביות.
- הפינה השמאלית תחתונה של מלבן ההשערה תהיה $\left\langle \min_{\langle x,y\rangle \in S}(x), \min_{\langle x,y\rangle \in S}(y) \right\rangle$
- הפינה הימנית עליונה של מלבן ההשערה תהיה $\left\langle \max_{\langle x,y\rangle \in S}(x), \max_{\langle x,y\rangle \in S}(y) \right\rangle$
- (אם אין דוגמאות חיוביות החזר את המלבן הריק)






# טענה

- לכל מלבן מטרה R, לכל התפלגות D, לכל $0 < \varepsilon, \delta \le \frac{1}{2}$, קיים מספר m, כך שבהנתן m דוגמאות, נוכל לטעון בהסתברות של לפחות $1-\delta$, שהטעות של R' הנוצר ע"י האלגוריתם A אינה עולה על ε.



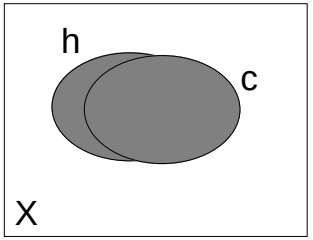
# מודל ה-PAC

הגדרות:

- מרחב הדוגמאות (instance space) הינה קבוצה X של אוביקטים. [בדוגמת המלבן - כל הנקודות במישור]
- מושג (concept) הינו תת קבוצה $c \subseteq X$ של מרחב הדוגמאות.
- ניתן להסתכל על מושג גם כפונקציה בוליאנית $c: X \to \{0,1\}$
- קבוצת מושגים (concept class) הינה קבוצה של מושגים מעל X. [למשל - קבוצת כל המלבנים בעלי צלעות מקבילות לצירים]
- מושג המטרה (target concept) הינו מושג c בקבוצת המושגים C. אנו מניחים שלמתכנן אלגוריתם הלמידה ידועה C.

- התפלגות המטרה D הינה התפלגות על מרחב הדוגמאות X.
- בהנתן מושג מטרה c ומושג h נגדיר את השגיאה של h עבור D:

$$error(h) = \Pr_{x \in D}[c(x) \neq h(x)]$$




- אורקל EX(c,D) הינה פרוצדורה המחזירה דוגמא מסומנת $\langle x, c(x) \rangle$ כאשר x נדגמת אקראית ובאופן בלתי תלוי לפי התפלגות D.



# תכונות רצויות עבור אלגוריתם למידה

- אלגוריתם למידה רשאי לקרוא ל-EX(c,D). נרצה שמספר הקריאות יהיה קטן (פולינומיאלי בפרמטרים שנציין בהמשך).
- זמן החישוב יהיה קטן.
- האלגוריתם יוציא מושג היפותזה h כך ש- error(h) יהיה קטן.



# למידת PAC

תהי C קבוצת מושגים מעל X. נאמר ש-C ניתנת ללמידת PAC אם קיים אלגוריתם L המקיים את התכונה הבאה: לכל מושג $c \in C$, לכל התפלגות D על X, לכל $0 \le \varepsilon \le 1/2$ ולכל $0 \le \delta \le 1/2$, אזי בהסתברות $1-\delta$ , L יחזיר היפותזה $h \in C$ המקיימת $error(h) \le \varepsilon$.

PAC=Probably Approximately Correct

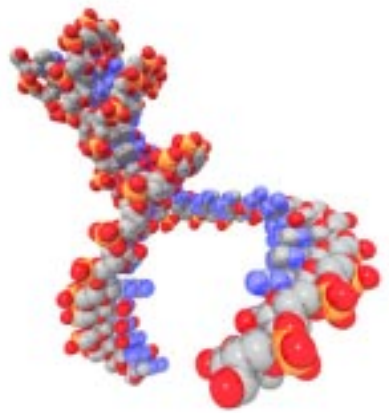
אם L רצה בזמן פולינומיאלי ב-$1/\varepsilon$ ב-$1/\delta$ ב-size(c) וב-n נגיד ש C ניתנת ללמידת PAC יעילה. (size(c) הוא גודל המושג ביצוג הנבחר, n הוא גודל הדוגמאות בקידוד)

- בדוגמת המלבן הוכחנו למעשה שקבוצת המלבנים שצלעותיהם מקבילות לצירים ניתנת ללמידת PAC יעילה.



# אלגוריתמים גנטיים

- אלגוריתמים המחקים תהליכים אבולוציונים על אוכלוסיות של היפותזות.
- הפותזות/פתרונות מיוצגים ע"י מחרוזות ביטים.
- האלגוריתמים הגנטיים מבצעים חיפוש אלומה (beam search) במרחב ההיפותזות.
- האלגוריתמים מניחים קיום פונקצית fitness הנותנת חיזוקים חיוביים להיפותזות טובות.
- היפותזות גרועות נזרקות. היפותזות חדשות נוצרות ע"י הפעלת פעולות גנטיות על היפותזות קיימות.



# פעולות גנטיות

- קיימות פעולות גנטיות רבות. שלוש הנפוצות ביותר הינן:

בחירה (selection): האופרטור בוחר חברים באוכלוסיה (כרומוזומים) לשם התרבות. ההסתברות לבחירה תלויה בחיזוק fitness שקיבל הכרומוזום.

הצלבה (crossover): האופרטור בוחר מקום אקראי, קוטע את שני הכרומוזומים במקום זה ומחליף ביניהם את התחיליות:

00101101 → 00110110

10010110 → 10001101

מוטציה (mutation): האופרטור משנה באופן אקראי ביטים בכרומוזום.



# אלגוריתם גנטי בסיסי

1. התחל עם אוכלוסיה של N כרומוזומים אקאיים באורך L.
2. חשב את פונקצית ה-fitness, f(x) , לכל כרומוזום x באוכלוסיה.
3. חזור על הפעולות הבאות עד שיווצרו N ילדים:
   - א. בחר זוג הורים מהאוכלוסיה הנוכחית בהסתברות יחסית ל-fitness שלהם (הבחירה עם חזרות).
   - ב. בהסתברות Pc בצע הצלבה בין ההורים (אחרת קח את ההורים כמות שהם).
   - ג. בצע מוטציה בכל אחד מהביטים בכל אחד מהילדים בהסתברות Pm.
   - ד. הוסף את שני הילדים החדשים לאוכלוסיה החדשה.
4. חזור לצעד 2.



# סכמות

- הרעיון המרכזי שעמד מאחורי פיתוח האלגוריתמים הגנטיים היה ליצור ולמזג "אבני בנין" של "היפותזות" (או פתרונות).
- סכמה היא מחרוזת מעל האלף-בית {0,1,*}, כאשר * מסמנת don't care. לכן סכמה מייצגת למעשה קבוצת מחרוזות מעל {0,1}.
- דוגמא לסכמה: H=1****1. 100101 היא מופע (instance) של H.
- לסכמה H יש 2 ביטים מוגדרים (ביטים שאינם *). נאמר שהסכמה היא בעלת סדר 2.
- לסכמה יש אורך מגדיר 5 (מרחק בין הביטים המגדירים הקיצוניים).



# עיבוד סכמות

- הטענה הבסיסית של מפתחי האלגוריתמים הגנטיים: הסכמות הן אבני הבנין אותן מעבדים האלגוריתמים באופן אפקטיבי ע" האלגוריתמים הגנטיים.
- כל מחרוזת הינה מופע של $2^L$ סכמות. באוכלוסיה של N כרומוזומים ישנם נציגים של בין $2^L$ ו-$N \cdot 2^L$ סכמות.
- על כן, בעוד שהאלגוריתם הגנטי מחשב באופן ישיר את הכשירות (fitness) של N כרומוזומים, הוא מעריך באופן עקיף את הכשירות הממוצעת של לפחות $2^L$ סכמות.

# משפט הסכמה

- אם היינו פועלים ישירות על סכמות, היינו שואפים שמספר המופעים של סכמה טובה יגדל בדור הבא באופן יחסי לכשירות הממוצעת של הסכמה
- משפט הסכמה טוען שאסטרטגיה זו אכן ממומשת באופן עקיף.
- תהי H סכמה בעלת מופע אחד לפחות בדור t.
- נסמן ב $m(H,t)$ את מספר המופעים של סכמה H בדור t.
- הכשירות הממוצעת של H בדור t הינה: $\hat{u}(H,t) = \sum_{x \in H_t} f(x) \Big/ m(H,t)$



- ההסתברות לכרומוזום x להבחר לדור t+1 הינה $f(x) \Big/ \sum_y f(y)$
- לכן המספר הצפוי של מופעים של כרומוזום x בדור t+1 הינו:

$$\left[ f(x) \Big/ \sum_y f(y) \right] N = f(x) \Big/ \left[ \left( \sum_y f(y) \right) \Big/ N \right] = f(x) \Big/ \hat{f}(t)$$

כאשר $\hat{f}(t)$ הינה הכשירות הממוצעת בדור t.

- בהתעלם מהצלבות ומוטציות, מספר המופעים הצפוי של סכמה H בדור t+1 הינו: $E(m(H,t+1)) = \sum_{x \in H} \frac{f(x)}{\hat{f}(t)} = \frac{1}{\hat{f}(t)} \sum_{x \in H} f(x) = \frac{\hat{u}(H,t)}{\hat{f}(t)} m(H,t)$



- אנו רואים לכן שמספר המופעים של הסכמה גדל או קטן ביחס לכשירות הממוצעת שלה. מספר המופעים של סכמה בעלת כשירות ממוצעת גבוהה מהממוצע יגדל באופן אקפוננציאלי ($\frac{\hat{u}(H,t)}{\hat{f}(t)}$ יהיה גדול מ-1).
- נותר לנו לבדוק את ההשפעות של ההצלבה והמוטציה:
- תהי $P_c$ ההסתברות שהצלבה תופעל על כרומוזום.
- ההסתברות שסכמה H "תשרוד" הצלבה חסומה ע"י: $S_c(H) \geq 1 - P_c \left( \frac{d(H)}{L-1} \right)$

כאשר $d(H)$ הינו האורך המגדיר של H ו- L הינו אורך הכרומוזום (הסיבה שזהו חסם מלמעלה: יתכן שכרומוזום יחתך בתוך הסכמה אבל היא לא תהרס)



- תהי $P_m$ ההסתברות ש-ביט ישונה ע"י מוטציה. ההסתברות שסכמה H תשרוד מוטציה הינה $(1-P_m)^{o(H)}$ כאשר $o(H)$ הינו ה"סדר" של הסכמה - מספר הביטים המגדירים.
- נשלב את השפעות המוטציה וההצלבה ונקבל חסם על המספר הממוצע של מופעי סכמה H בדור t+1:

$$E(m,(H,t+1)) \geq \frac{\hat{u}(H,t)}{\hat{f}(t)} m(H,t) \left( 1 - P_c \frac{d(H)}{L-1} \right) \left[ (1-P_m)^{o(H)} \right]$$

- כלומר: סכמות קצרות בעלות סדר נמוך בעלות כשירות גדולה מהממוצע ישתכפלו באופן אקפוננציאלי.

# מימוש אלגוריתמים גנטיים

**קידוד**

- הקידוד אינו חייב להיות בינארי. קידודים אפשריים אחרים הינם קידודים באלף-בית שאינו בינארי.
- לעתים מקדדים באמצעות ערכים מספריים.
- בתכנות גנטי מקדדים באמצעות עצים.



# שיטות בחירה (selection)

- השיטה הבסיסית ממומשת בד"כ באמצעות "רולטה מוטה": כל כרומוזום מקבל גזרה יחסית לכשירות שלו.
- קיימת בעיה של התכנסות מוקדמת לכרומוזומים שהפגינו, אולי במקרה, כשירות גבוהה יחסית בשלבים מוקדמים של הלמידה.
- שיטת "sigma scaling" מתחשבת גם בסטיית התקן (שהיא גבוהה בהתחלה):

$$ExpVal(x,t)=\begin{cases}1+\dfrac{f(x)-\hat{f}(t)}{2\sigma(t)} & \sigma(t)\neq 0\\ 1.0 & \sigma(t)=0\end{cases}$$

- כך, כרומוזום בעל כשירות הגבוהה בסטיית תקן אחת מעל הממוצע יקבל בממוצע 1.5 ילדים.
- אם הערך יורד מתחת ל-0 משנים אותו שרירותית לערך חיובי קטן.



# שיטות בחירה נוספות

- בחירת בולצמן משנה את הטיית הבחירה עם זמן הלימוד.
- פרמטר "טמפרטורה" יורד עם זמן הלימוד.
- בתחילה הסיכוי של כרומוזומים חלשים להבחר גדולה יותר. עם התקדמות הלמידה גוברת הנטייה לבחור חברים בעלי כשירות גבוהה.

$ExpVal(x,t)=\dfrac{e^{f(x)/T}}{\left\langle e^{f(i)/T}\right\rangle_t}$ כאשר $\left\langle e^{f(i)/T}\right\rangle_t$ הינו הממוצע מעל האוכלוסיה בזמן t, ו-T הינה הטמפרטורה.

- rank selection בוחר איברים לרבייה לפי הסדר שלהם במיון יורד על פי הכשירות.
- לפעמים נוהגים לבחור רק חלק קטן מהאוכלוסיה הטובה להתרבות, ומכניסים את הצאצאים במקום הגרועים.



# אופרטורים גנטיים - הצלבה

- ההצלבה החד-נקודתית הינה הנפוצה ביותר, אולם היא סובלת מכמה בעיות:
- קיימים שילובים של סכמות שלא ניתן להשיגם. למשל לא ניתן לשלב בפעולה אחת את 1***0*11 ו- *1*1**** לסכמה 11*01*11.
- באופן כללי סכמות בעלות מרחק הגדרה גדול לא שורדות.
- בהצלבה חד נקודתית ההורה מעביר בהכרח את אחת מנקודות הקצה - כלומר יש הטיה הקשורה במיקום הגן.

- חלק מהבעיות הנ"ל נפתרות ע"י שימוש בהצלבה דו-נקודתית: שתי נקודות נבחרות אקראית והקטע ביניהן מוחלף.
- קיימות הכללות למספרים שונים של נקודות חציה.
- גישה קיצונית מתירה החלפה של כל ביט בהסתברות מסוימת (בד"כ 0.7-0.8).



# פרמטרים באלגוריתמים גנטיים

- בעיה מרכזית באלגוריתמים גנטיים הינה קביעת הערכים של הפרמטרים השונים.
- נערכו ניסויים רבים מאוד. כולל הפעלה של האלגוריתמים הגנטיים כתכנית המנסה למצוא ערכים אופטימליים אלה (meta-learning).
- הערכים המקובלים כיום:
  - גודל אוכלוסיה: 20-30.
  - הסתברות להצלבה של זוג: 0.75-0.95
  - הסתברות למוטציה: 0.005-0.01
- הערכים ה"נכונים" תלויים בישום הספציפי. קיימות שיטות אדפטיביות המשנות את ההסתברויות עם זמן הלמידה.