Machine Learning from Data – IDC

HW6 – Theory

**Nitai Aharoni - 203626742**

# VC-Dimension

Compute the VC-dimension of the following hypothesis classes:

* 1. (10 pt.) Assume the instance space X satisfies |X | = ∞. The space of binary hypotheses, which given a training set, returns the target *y* of **x** if the pair (**x***, y*) was observed in the training set, and +1 otherwise. Formally, compute *VC*(H) of

H = {*h* : X → {−1*,* +1} : *h* equals − 1 on a finite subset of X and + 1 elsewhere} *.*

* נתון מרחב X כך ש-
* נגדיר את d להיות ה-training set כך ש-(תת קבוצה סופית)
* מהגדרת H, h תנתץ באופן מושלם את קבוצת האימון d
* היות ו- אז ולכן עבור כל תת קבוצה d בכל גודל קיימת h שתנתץ אותה באופן מושלם
* לכן
  1. (15 pt.) *n*-Interval classifiers of length ≥ 2. Let X = R,

H = {*x* → +1 ⇐⇒ *x* ∈ [*a*1*, b*1] ∪ [*a*2*, b*2] ∪ · · · ∪ [*an, bn*] : *a*1 + 2 ≤ *b*1*, . . . , an* + 2 ≤ *bn*} *.*

* *VC*(H)=2n נסביר:
  + נראה
    - נתון מרחב X כך ש-
    - נגדיר את d להיות ה-training set כך ש-(קבוצה סופית)
    - מוגדר לנו classifier המורכב מ-n אינטרוולים על הישר באורך גדול שווה ל-2
    - נחלק את הנקודות ב-d לזוגות - כך שעבור כל הזוגות ממוקמים על הישר במרחק 1 ביניהם ובמרחק 10 מהזוג הבא.
    - *עבור כל זוג ניתן לקבוע אינטרוול [ai, bi] - כך ש- ai + 2 ≤ bi, האינטרוול מנפץ את הזוג, וגם אינטרוול זה לא משפיע על ניפוץ הזוגות השכנים.*
    - לכן בעזרת n אינטרוולים כאלה ניתן לנפץ 2n נקודות ולכן
  + נראה
    - נוסיף נקודה אחת וננסה לנפץ
    - כעת, היות ויש לנו נקודות (כלומר מספר אי זוגי של נקודות) ננסה לנפץ שלישיית נקודות באמצעות אינטרוול אחד
    - נשים לב שאם נרצה לסווג את הנקודות באופן הבא: +,-,+ - לא ניתן לעשות זאת באמצעות אינטרוול אחד
    - כמובן שבאותו אופן גם רביעיית נקודות וכך הלאה לא ניתן לנפץ באמצעות אינטרוול אחד.
  + לכן *VC*(H)=2n
  1. (20 pt.) Linear classifiers in the plain. Let X = R2,

H ={(*x*1*, x*2)->

+1 *w*1*x*1 + *w*2*x*2 + *b >0*

}

−1 *w*1*x*1 + *w*2*x*2 + *b* ≤ 0

: *w*1*, w*2*, b* ∈ R}

Show that VCdim(H) = 3:

* + 1. Find a set of size 3 that H shatters.
* נראה פיזור של 3 נקודות כאלה כך ש-H מנפץ אותן
  + 1. Show that no set of size 4, *A* = (**z**1*,* **z**2*,* **z**3*,* **z**4), **z***i* ∈ R2 can be shattered by H.

**Guidance:** First prove the following lemma:

**Lemma 1.** *Suppose a linear classifier h obtains prediction y* ∈ {−1*,* +1} *on a set of points* **z***,* **z’**∈R2 *(h*(**z**) = *h*(**z’**) = *y). Then it also obtains the same prediction on any intermediate point. Namely,*   
∀*α* ∈ [0*,* 1] *h*((1 − *α*)**z** + *α***z’**) = *y.*

And use it in each of the following 3 possible cases:

* + - * The convex hull of *A* forms a line.
      * The convex hull of *A* forms a triangle.
      * The convex hull of *A* forms a quadrilateral.
* יהי z=(x,y), z’=(x’,y’)
* נתון h(z)=h(z’)=y
* נראה כי לכל *α* ∈ [0*,* 1]  *מתקיים h*((1 − *α*)**z** + *α***z’**) = *y*
* יהיו 4 נקודות:
* נשתמש בכך עבור שלושת המקרים:
  + קו ישר:
    - מהיות קו ישר אזי לכל i,
    - ונניח
    - ו-
    - לפי הלמה שהוכחנו אם וגם אזי בהכרח בסתירה
    - לכן הקבוצה A לא יכולה להיות מנופצת ע"י H
  + משולש:
    - כאשר z1,z2,z3 יוצרות משולש ו-z4 מוכלת במשולש
    - ו-
    - אם z4 נמצא על אחת מצלעות המשולש - בה"כ על הצלע שבין z1 ל-z3:
      * לפי הלמה שהוכחנו אם וגם אזי בהכרח בסתירה
      * לכן הקבוצה A לא יכולה להיות מנופצת ע"י H
    - אם z4 נמצא בתוך המשולש:
      * אזי קיימת נקודה z’ כך ש-z’ שנמצאת על אחת מצלעות המשולש, בה"כ בין z1 ל-z3, כך ש z4 נמצאת על הישר שבין z’ לבין z2.
      * לפי הלמה וגם אזי בהכרח
      * ולכן לפי הלמה שהוכחנו אם וגם אזי בהכרח בסתירה
      * לכן הקבוצה A לא יכולה להיות מנופצת ע"י H
  + מרובע:
    - כאשר z1,z2,z3,z4 יוצרות מרובע
    - מדובר בבעיית XOR אותה לא ניתן לנפץ בעזרת מפריד לינארי כפי שהראינו בכיתה

# Learning Conjunctions of Literals

(30 pt.) Let X = {0*,* 1}*n* (all Boolean strings of length *n*), let *C* = H = the set of all conjunctions on X (e.g. *x*1 ∧ ¬*x*3 ∧ *xn* is in *C* and H). Define an algorithm *L* so that *C* is PAC-learnable by *L* using H. Prove all your steps.

* אלגוריתם:
  + נגדיר
  + נעבור על כל ה-instances:
  + נתבונן ב-instance ה-i.
    - אם C()=1, וכל האלמנטים ב-h הנוכחי קיימים ב-attributes של ה-instance   
      🡨 אז h הוא קונסיסטנטי עם ה-instance הנוכחי.
    - אם C()=1, ואחד או יותר מהאלמנטים ב-h הנוכחי לא מופיעים בו 🡨 אז נסיר   
      מ-h את כל ה-attributes שקיימים ב-h ולא ב-instance.
  + אחרי מעבר על כל ה-instances נקבל h שהוא קונסיסטנטי עם C
  + ואם קיבלנו h כזה אז C הוא PAC-learnable באמצעות L.
* נכונות:
* לומד קונסיסטנטי:
  + עבור C()=1 היות ומדובר בביטוי AND, כל אחד מה-attributes של אותו instance אמורים לחזות את ערכו האמיתי – ולכן אם קיימים ביטויים ב-h שלא מסכימים עם ה-attributes של ה-instance, הם יוסרו מ-h
  + עבור C()=0 היות ומדובר בביטוי AND, אם כל הביטויים ב-h יופיע ב-attributes של ה-instance אז h יהיה קונסיסטנטי עם אותו instance.
    - נניח בשלילה שעבור instance מסוים כל הביטויים ב-h קיימים ב-attributes של ה-instance אזי h לא קונסיסטנטי עם C. אבל נתון ש-H=C ולכן קיים כך ש-h קונסיסטנטי עם C בסתירה.
    - אם עברנו על כל ה-instances ו-h קונסיסטנטי עם כולם אז h קונסיסטנטי עם C.
* sample complexity:
  + - ה-attribute קיים בביטוי כ-true, קיים כ-false או לא קיים. n=attribute num
  + ולכן כלומר sample complexity פולינומיאלי
* בדיקת זמן ריצה:
  + עבור כל instance נשווה את ה-attributes שלו אל מול האלמנטים ב-h – ניתן לבדוק ב-O(n)
  + נעבור על כל ה-instances כלומר m
  + סה"כ זמן ריצה O(n\*m) – פולינומיאלי
* לכן C הוא PAC-learnable באמצעות L

# (Almost) PAC-learnability

(25 pt.) Let *C* denote the class of all possible target concepts defined over a set of instances X. Suppose that H is a space of binary hypotheses containing the constant concept *c*1 defined by *c*1(*x*) = +1 for all *x* ∈ X, and having the property that  
C \ {c1} is PAC-learnable by an algorithm L using H with sample complexity m(δ, ).

Provide a learning algorithm L’ that uses L, so that C (including c1) is PAC-learnable

by L’ using H with sample complexity max{m(δ, ), }. Prove all your steps.

* אלגוריתם L’:
  + נעבור על כל ה-instances ב-D ונבדוק:
  + אם קיים כך ש- 🡨 אז נריץ את L הידוע בתור PAC-learnable ונחזיר את h
  + אם לכל מתקיים 🡨 אז נחזיר h=1
* נראה L’ הינו PAC-learnable:
* ידוע ש-L הינו PAC-learnable עם sample complexity m(δ, ) ולכן אין צורך להראות במקרה בו קיים כך ש-.
* נראה עבור המקרה שלכל מתקיים
  + לומד קונסיסטנטי:
    - ידוע שלכל מתקיים ולכן h=1 קונסיסטנטי עם c
  + sample complexity:

    - ולכן כלומר sample complexity פולינומיאלי
  + בדיקת זמן ריצה:
    - עבור כל נבדוק את לכן מדובר ב-m בדיקות O(m)
* לכן C הוא PAC-learnable באמצעות L’
* ידוע שבמקרה של מתקיים
* לכן ה- sample complexity הוא