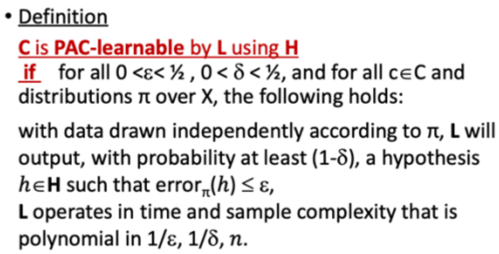
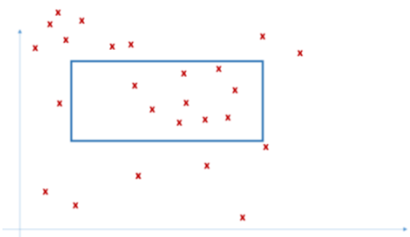
UnSupervised Learning and kMeans – 10 הרצאה :למידה חישובית ממידע

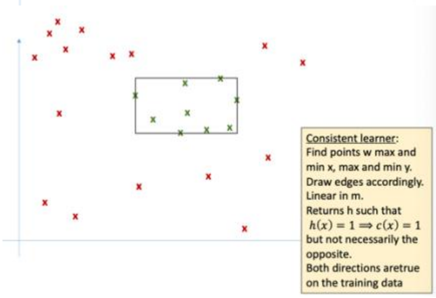
PAC Learnability – 9 המשך מהרצאה



- נניח כי מרחב ההיפותזות הוא המלבן. נגדיר ולאחר מכן נחזור לדוגמה.
- בהינתן קלאס C (מלבנים) של target-concepts אפשריים המוגדרים על דגימות X ממימד n, ואלגוריתם למידה L המשתמש במרחב היפותזות H. (במלבנים מרחב כל הנקודות האפשריות הוא R^2)
- **ההגדרה:** C הוא PAC-Learnable על פי L שמשתמש ב-H (מרחב היפותזות מסוים) אם לכל 0 <ε< ½ , 0 < δ < ½

ולכל התפלגות פאי על מרחב האינסטנסים X ולכל קוסנפט c ששיך למרחב הקונספטים C, מתקיים: עם דאטה שמוגרלת באופן בלתי תלוי לפי ההתפלגות פאי, L יהיה האאוטפוט, בהסתברות שהיא לפחות (1 מינוס דלתא), היפותזה h ממרחב ההיפותזות H כך ש -.
L בסיבוכיות זמן וסיבוכיות דגימות שהיא פולינומיאלית ב-

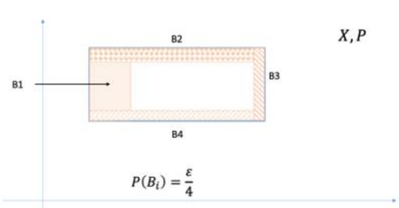


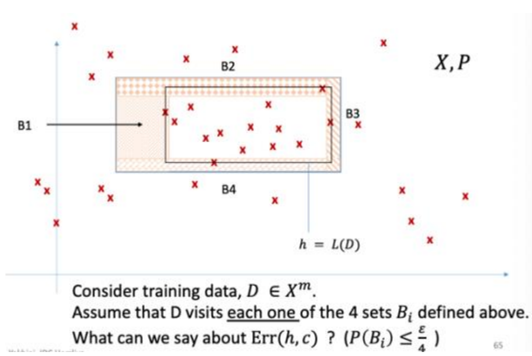


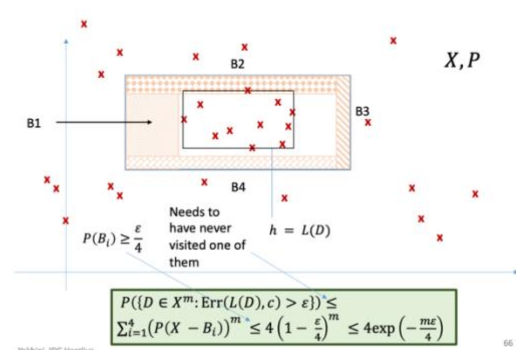
בחזרה לדוגמה: אנחנו לא נראה את המלבן הכחול, אנחנו רואים את הקלאסים השונים. כעת נגדיר אלגוריתם שנטען שהוא קוסיסטנטי: הוא יחזיר את המלבן השחור שמוכל בתוך המלבן הכחול.
לכן הטעות של המלבן השחור על ה-training data היא 0. ההסתברות לטעות בין הקונספט להיפותזה: ההסתברות שעבור נקודה מסוימת המלבן הכחול והשחור לא יסכימו. וידוע לנו שהאלגוריתם בנה את המלבן השחור כך שהוא מוכל בתוך המלבן השחור.
נרצה לחסום את הסיכוי להגיע לטעות שגדולה מאפסילון.

A bound on sample complexity / חסם על סיבוכיות הדגימות

לכל קונספט c במרחב הקונספטים C, נחסום את כל ה-datasets שיכולות להוביל ל – h = L(D)
שמקיים טעות גדולה מאפסילון $\mathrm{Err}(h, c) > \varepsilon$ למספר סופי של קבוצות (תתי קבוצות של X^m).
לאחר מכן נשערך את ההסתברות של כל תת-קבוצה של דגימות ולבסוף את האיחוד שלהן.
מכאן נוכל להסיק כי החסם על סיבוכיות הדגימות כפונקציה של אפסילון ודלתא.







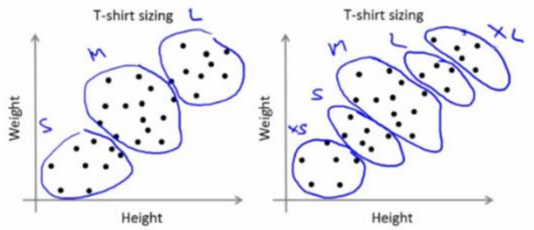
דוגמה: בהינתן שהמלבן הלבן הוא ההיפותזה הנוכחית, והמלבן החיצוני הוא הקונספט. נפלח את השטח שבתוך הקונספט ל-4 פלחים אשר הסיכוי שדאטה "תיפול" שם הוא רבע אפסילון. אם היו נקודות בתוך המלבנים B1-4, אז הטעות תהיה קטנה מאפסילון (ולכן זה לא רע, נחשיב training data כרע אם הטעות שלה גדול מאפסילון). הסיכוי שלא נבקר בכל ה-4 נקודות הוא קטן מאפסילון, לכן נגדיל את היפותזה ש"תגיע" לדגימות אלה (כפי שניתן לראות באיור האמצעי). ההסתברות של קבוצת כל "הרעים" המוכלים באיחוד – קטנה מסכום ההסתברויות של מי שכן נמצא מחוץ לכל אחד מהשטחים Bi, לכן קיבלנו אי-שוויון של טור טיילור (כפי שראינו בהרצאה הקודמת).
ההסתברות של "הרעים" (בעלי טעות יותר מאפסילון) חסומה מלמעלה, אבל היינו רוצים שהחסם יהיה קטן מדלתא – לכן נוכל **להגדיר את m** באופן שיתקיים כי החסם שהתקבל קטן מדלתא. ואז הביטוי שהתקבל בשקף הימני יקיים את הדרישות שאנו צריכים. ניתן להבין כי ה-m הינו תלוי ב"4" הזה שהגדרנו מראש עם השטחים Bi. ומכאן נובע שעבור צורה אחרת שאינה מלבן ה-4 היה משתנה ועל כל מספר הדגימות ישתנה.

UnSupervised Learning and the K-Means Algorithm

הנחה הבסיסית עד כה הייתה כי עבור ה-training data אנחנו מקבלים label-ים (ערך של פונקציה או קלאס). הסרת ההנחה מביאה אותנו לשטח של למידה לא מפוקחת / unsupervised learning. אזי במקום target function יהיה עלינו ללמוד: מבנה structure / רגולריות regularity / דמיון similarities / קיבוץ לקבוצות grouping. ולכן למידה לא מפוקחת לפעמים משתמשת ב-**clustering algorithms**.

Clustering

נשתמש בפיצ'רים שיש בדאטה במטרה לפלח / להפריד את ה-training sets ל-clusters (קבוצות). איברים בכל קבוצה אמורים להיות "יותר דומים" אחד לשני מאשר אלמנטים בקבוצות אחרות. לכן, המפתח ל-clustering הוא דמיון ואיך מודדים אותו.



דוגמאות לשימוש ב-Clustering:

משתמשים בקלאסטרינג עבור רשתות חברתיות, אינטראקציה של פרוטאינים, דמיון בין רצועות השמעה.

פילוח שוק – בניית מוצר שמתאים לצרכים של תתי קבוצות באוכלוסייה.

מדידת דמיון / Similarity Measures

ניתן להסתכל על clustering כחיפוש אחר הקיבוץ "הטבעי" ב-dataset. השאלה איך נדע שדגימות ב-cluster אחד יותר דומות אחת לשניה מאשר דגימות ב-cluster אחר, מערבת שני נושאים עיקריים:

- איך מודדים דמיון בין דגימות? (למשל, מרחק אוקלידי קטן = דומים)
- איך נוכל להעריך את החלוקה / ה-partitioning של set לתוך clusters?

Distance Metric / מטריקה

= במרחב שמוגדרת עליו מטריקה, יש פונקציה שמקבלת שני אינסטנסים במרחב שמקיימת:

- אי שלילי: $d(x_1,x_2) \geq 0$
- זהות בין משתנים: $d(x_1,x_2)=0 \iff x_1=x_2$
- סימטריה: $d(x_1,x_2)=d(x_2,x_1)$
- אי שוויון המשולש: $d(x_1,x_2) \leq d(x_1,x_3)+d(x_3,x_2)$

Minkowski Metric: $L_k(a,b)=\left(\sum_{i=1}^{d}|a_i-b_i|^k\right)^{\frac{1}{k}} \quad k \geq 1$

Manhattan Distance: L_1

Euclidean Distance: $L_2=\left(\sum_{i=1}^{d}|a_i-b_i|^2\right)^{\frac{1}{2}} \qquad \|a-b\|_2^2$

Infinity Norm: $L_\infty=\max(|a_i-b_i|)$

Algorithmics – constructing clusters

האלגוריתם הכי פשוט (אך שימושי) הוא Naïve cluster growing:

- היפר-פרמטר: מרחק threshold, T
- כל עוד יש עדיין איברים שעוד לא חולקו לקבוצות בדאטה תבצע:

-- תבחר אלמנט seed, s, ותייצר קלאסטר Cs

-- סמן את s כ-clustered (חולק לקבוצה)

-- כל עוד אין אלמנטים e שלא חולקו עם מרחק d(e,Cs) < T תבצע:

---- הכנס את כל האיברים e המקיימים את האי שוויון לעיל ל-Cs וסמן אותם כ-clustered

הבעיות באלגוריתם הזה:

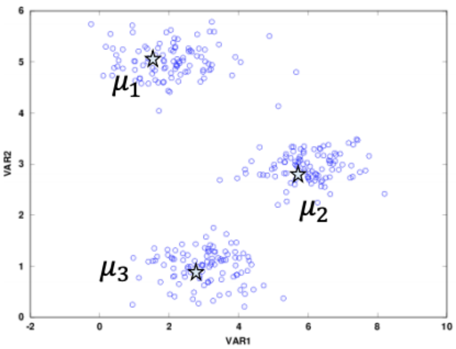
- החיפוש / מבנה הנתונים – חישוב גרף השכנויות יעלה O(n^2), לכן אנחנו צריכים לחשוב על שיטות יעילות יותר להוספת כל האלמנטים עם מרחק קטן ממש מ-T. ראינו כבר שיטות כאלה עבור אלגוריתם KNN.
- תלות בסדר רנדומי של הבחירה
- מרחק ה-threshold T חייב להיות קבוע – T-ים שונים יכולים להוביל לתוצאות שונות.
- איך משערכים את התוצאה?

Criterion Function

משימת ה-Clustering: נקבל דאטה של דגימות $D = \{x1,\dots,xn\}$ ונרצה לחלק אותן ל-c תתי-קבוצות זרות: C1,…,Cc.

האתגר: לחפש נוסחה ולבצע את משימת הקלסטרינג כאופטימיזציה של פונקציית קריטריון.

הפונקציה צריכה לקבל את הדאטה-סט D ולהגדיר את הקלסטרים כאינפוט ולהפיק מספר ממשי.



לאיכות הקלאסטרינג יש 2 אספקטים:

- מדידת הקומפקטיות של כל ענן דגימות, המייצג קלאסטר.
- מדידת כמה רחוקים העננים אחד מהשני (הקלאסטרים).

Cenroid Base clustering / מרכז הכובד

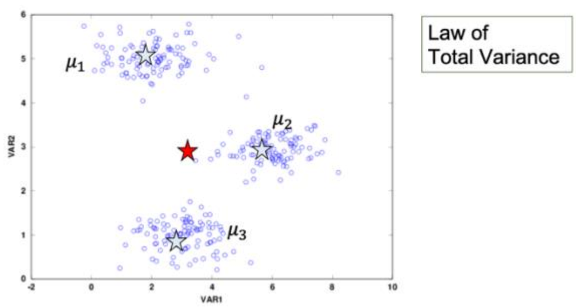
כל קלאסטר מיוצג על ידי הצנטרויד שלה = מרכז הכובד שלה: $\mu_i = \frac{1}{|C_i|}\sum_{x\in C_i} x$

הפונקציה האובייקטיבית שלנו תחת מרחק אוקלידי הינה: $G(D,\{C_i\}_{i=1}^k) = \frac{1}{m}\sum_{i=1}^{k}\sum_{x\in C_i}\|x-\mu_{C_i}\|^2$

הממוצע של מרחקי הנקודות מהמייצג שלהן.

זוהי פונקציית המטרה שמודדת כמה ענן הוא קומפקטי בפני עצמו – נרצה להביא אותה למינימום כי ככל שהיא קטנה יותר כך העננים קומפקטיים יותר, והיא פותרת רק חצי מהבעיה.

$$S_T(D) = \frac{1}{m}\sum_{x\in D}\|x-\mu\|^2 = G(D,\{C_i\}_{i=1}^k) + \sum_{i=1}^{k}\frac{|C_i|}{m}\cdot\|\mu_{C_i}-\mu\|^2$$



אבל, כאשר ממזערים את הקומפקטיות של העננים אנחנו למעשה מגדילים את המרחקים בין העננים.

ה-total variance של קבוצת נקודות D (או scatter) מוגדרת כ:

ה-total scatter איננו תלוי ב-clustering. ניתן להראות כי קיים clear monotone tradeoff בין הפיזור של שני הגורמים: כשאחד גדל השני קטן.

לכן, כאשר ממזערים את המרחקים בתוך הקלאסטרים, הדבר ממקסם את הפיזור של הקלאסטרים (המרחקים בין העננים).

קריטריון הפיזור / The Scatter Criterion

לכן המיזעור של: $\boxed{G\big((x\in D)(\mu_1,\mu_2,\dots,\mu_k)\big) = \sum_{i=1}^{k}\sum_{x\in C_i}\|x-\mu_i\|_2^2}$

גם יביא למקסימום את הסקאטר של מרכזי הכובד / ה-centroids: $\sum_{i=1}^{k}\frac{|C_i|}{m}\cdot\|\mu_{C_i}-\mu\|^2$

- לכן, יש לנו ייצוג של הטעות הכוללת של ייצוג m הדגימות ע״י קבוצה של k קלאסטרים ספציפיים.

הפתרון: ברגע שיש לנו את פונקציית הקריטריון, בעיית הקלאסטרינג נהפכת מוגדרת-היטב. באופן תיאורטי, שימוש בחיפוש ארוך יכול למצוא פתרון אופטימלי. יש בערך k^m / k! דרכים לחלק m אלמנטים ל-k קלאסטרים (הפתרונות המדוייקים נקראים Stirling numbers). וכן, יש אפילו יותר דרכים אם אנחנו מחפשים אחר ה-k הטוב ביותר. גישה פרקטית אפשרית לפתרון תהיה: הגדלת ה-cluster / region או שימוש בשיטות חיפוש יוריסטיות אחרות.

K-Means Algorithm

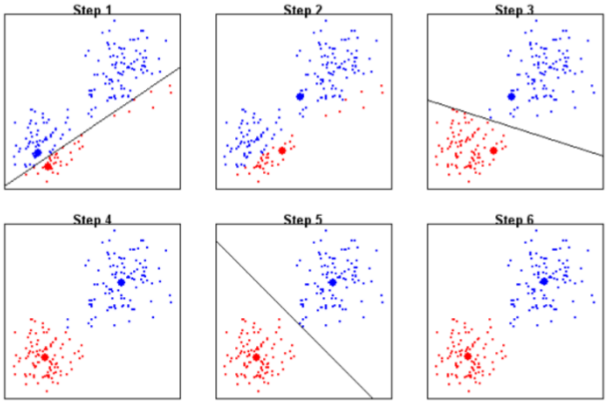
k-Means Clustering

Initialize $\mu_1, \dots, \mu_k$ (randomly)
Loop:
  Assign all n samples to their nearest μ_i
  Re-compute $\mu_1, \dots, \mu_k$ using their cluster members
Until no change in $\mu_1, \dots, \mu_k$
Return $\mu_1, \dots, \mu_k$

אחד אלגוריתמי ה-clustering היותר פופולריים והשימושיים. נניח כי נקבע את מספר הקלסטרים שבו אנו מעוניינים מראש להיות k (זהו מודל היפר-פרמטרי). נחפש אחר חלוקת הדאטה ל-k קבוצות (זרות), שהאיחוד שלה היא כל הדאטה, שמביאה למינימום את ה-Euclidean norm error criterion שהוא: $G\big((x\in D)(\mu_1,\mu_2,\dots,\mu_k)\big)$.

אלגוריתם זה יכול לעבור על כל מטריקה, אבל אז הדואליות יכולה לא להתקיים.

באלגוריתם: אנחנו מאתחלים באופן רנדומלי k ערכי מיו. ובללואה אנו מבצעים: נכניס את כל n הדגימות ל-מיוו הקרוב ביותר אליהן, נחשב מחדש את k ערכי מיו על פי החלוקה שהתבצעה. נבצע את הלולאה עד איטרציה בה לא יהיה שינוי בערכי המיו ונחזיר אותם.



<u>שתי לולאות פנימיות:</u>

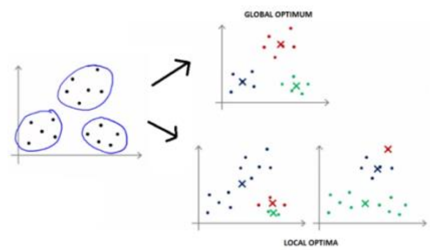
- <u>השמה:</u> נרוץ בלולאה על כל הדגימות ונכניס אותם ל"מייצגים" הקרובים ביותר אליהם.
  - בדרך כלל על ידי בדיקת מרחק אוקלידי.
- <u>חישוב מחדש עבור המייצגים:</u> נרוץ בלולאה על כל הקלאסטרים ונחשב נציגים חדשים.
  - בדרך כלל על ידי חישוב מרכז הכובד, הצנטרויד, התוחלת.

התכנסות מובטחת

בכל לולאה פנימית הפונקציה $\frac{1}{m}\sum_{i=1}^{m}\left\|x_i-\mu_{c_i}\right\|^2$ היא (באופן חלש מאוד) ממוזערת. בשלב ההשמה: אם דגימה קרובה יותר לנציג של קלאסטר אחר, אזי היא מקבלת השמה חדשה והפונקציה מתמזערת בשלב החישוב מחדש של הנציגים: הצנטרויד של תת-קבוצה ממזער את הממוצע של המרחקים בין כל קבוצה.

יש מספר סופי של השמות אפשרויות (אמנם סופי גדול מאוד אך עדיין סופי), אזי הפונקציה חסומה מלמטה (על ידי המינימום של ההשמות האפשריות). מכאן שהיא בהכרח מתכנסת, אבל לא בהכרח למינימום גלובלי, אלא לוקאלי.

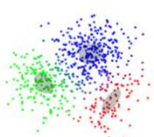
דוגמה להתכנסות שלא לאופטימום:



<u>הרחבות יוריסטיות להימנע מהתכנסות לא רצויה:</u>

- הרצת k-means מספר פעמים עם איתחולים שונים ובחירת המינימום הלוקאלי הטוב ביותר.
- כאשר הנציג איננו מייצג אף דגימה:
  - נוותר עליו (ונהפוך לאלגוריתם k-1 means clustering)
  - נבחר נציג רנדומלי חדש במקומו
  - נבחר את הקלאסטר בעל השגיאה הגדולה ביותר ונפצל אותו ל-2

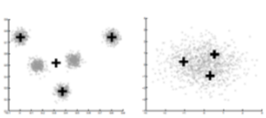
Fuzzy Clusters



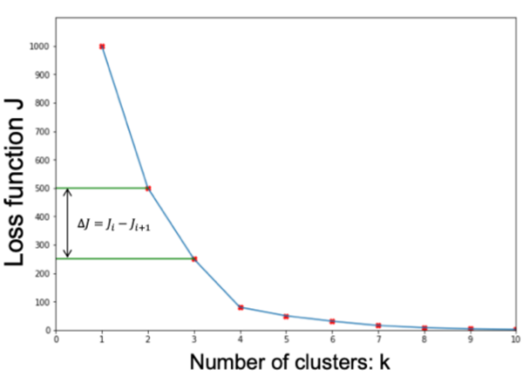
<u>איך נדע באיזה k לבחור?</u>

- אין תשובה שלמה/מוצדקת תיאורטית לשאלה זו.
- בפעמים רבות הבחירה תלויה בדאטה / במטרת העסק (למשל כמה סוגים שונים אמורים להיות למוצר?).
  בפרקטיקה – נשתמש בקריטריון ספציפי כדי להניע לכיוונו את הבחירה של k)
- באופן כללי, ככל שה-k גדול יותר, כך ערך המינימום של הפונקציה קטן: $G=\frac{1}{m}\sum_{i=1}^{m}\left\|x_i-\mu_{C(x_i)}\right\|^2$
- זאת מכיוון שאין "עונש" ב-G עבור k שהולך וגדל (וכן עבור k=m נקבל G=0)

Wrong k ...



נחפש אחר "מפרק / elbow" בביצועים של הגרף – מקום בו הגדילה של k איננה משפרת כל כך את הביצועים.



Semi-supervised clustering