APML
שיעור 7

Clustering

I מבוא ואלגוריתמים בסיסיים:

מוטיבציה: הדאטא בעולם האמיתי הוא הרבה פעמים מבולגן ולא מתויג

$\Leftarrow$ Unsupervised

נרצה לחקור את הדאטא שלנו ולמצוא אליו משהו - נרצה מודל שמסביר את הדאטא אבל לא תמיד אפשר להסתכל על כל הדאטא.

שאלות שנרצה לשאול: כמה "סוגים" שונים של דאטא יש לנו?
מה הצירים המרכזיים בדאטא?

נפרמל את הבעיה:

יהי $X=\{x_i\}$ נקודות

$d=$ פונקציית מרחק $d: X\times X \to \mathbb{R}$ ונדרוש שהיא תהיה מטריקה.

$C=$ קלסטרינג ל-K קבוצות כך שמתקיים:

$\forall i,j \quad C_i \cap C_j = \emptyset$ החיתוך ריק

$\bigcup_{i=1}^{K} C_i = X$ כל הנקודות נמצאות.

פונקציית מטרה אפשרית:

להביא למינימום את סכום המרחקים בין הנקודות בתוך הקלאסטרים

$$\arg\min_{C} \sum_{k=1}^{K} \sum_{x_i,x_j \in C_k} d(x_i,x_j)$$

הנחה: אנחנו יודעים את K מראש.

אבל זו לא פונקציה אידיאלית, מיד נשתכנע בהמשך למה...

מאמר של John Kleinberg מ-NIPS 2002 עוסק בתכונות של קלאסטרים ומראה ש-3 התכונות הבאות לא תמיד מתקיימות במקביל:

- Scale Invariance ← אפשר לשנות את יחידת המידה של המרחק ולשמור על הקלסטר
- Richness ← אפשר לסדר את הנקודות במרחב כך שהקלאסטרינג הטבעי יהיה נכון
- Consistency ← אם נקרב את הנקודות בתוך כל קלאסטר (בלי לשנות את המרחק היחסי בין קלאסטרים), הקלסטור לא ישתנה.

Clustering: המנק

I מבוא ואלגוריתמים בסיסיים: המנק

שני גישות שימושיות: — Bottom-Up → מתחילים מקלאסטר לכל נקודה ומחברים

— Top-Down → מתחילים מקלאסטר אחד ומפרידים.

*Hierarchical Agglomerative Clustering: → Bottom-Up

איך נמדוד מרחק בין קלאסטרים?

Single Linkage: עפ' המינימום:

$$d(A,B)=\min_{\substack{a\in A\\ b\in B}} d(a,b)$$

Average Linkage: עפ' הממוצע

$$d(A,B)=\frac{\sum_{a\in A}\sum_{b\in B} d(a,b)}{|A|\cdot|B|}$$

זמן הריצה של זה יהיה בערך $O(n^3)$

*Lloyd's Algorithm (k-means): → Top-Down

המטרה שלנו:

$$C=\operatorname{argmin}_C \sum_{k=1}^{K}\sum_{x_i,x_j\in C_k}\|x_i-x_j\|_2=\operatorname{argmin}_C \sum_{k=1}^{K}\sum_{x_i\in C_k}\|x_i-\mu_k\|_2$$

הצנטרואידים.

האלגוריתם הוא למעשה בדיוק EM אבל עם השמה hard-assignment במקום הסתברויות.

האלגוריתם: - נאתחל לכל k, את הצנטרואיד μ_k

למשל ע"י בחירת k נקודות באקראי.

- לכל k:

* נכניס לקלאסטר C_k את כל הנקודות ש-μ_k הכי קרוב אליהן. כלומר: $C_k=\{x: k=\operatorname{argmin}\|x-\mu_k\|\}$

* נחשב מחדש את μ_k

שיפורים: ① נריץ הרבה פעמים ובכל פעם נבחר את האופטימום שמצאנו.

② נדגום את הדאטא (subsample)

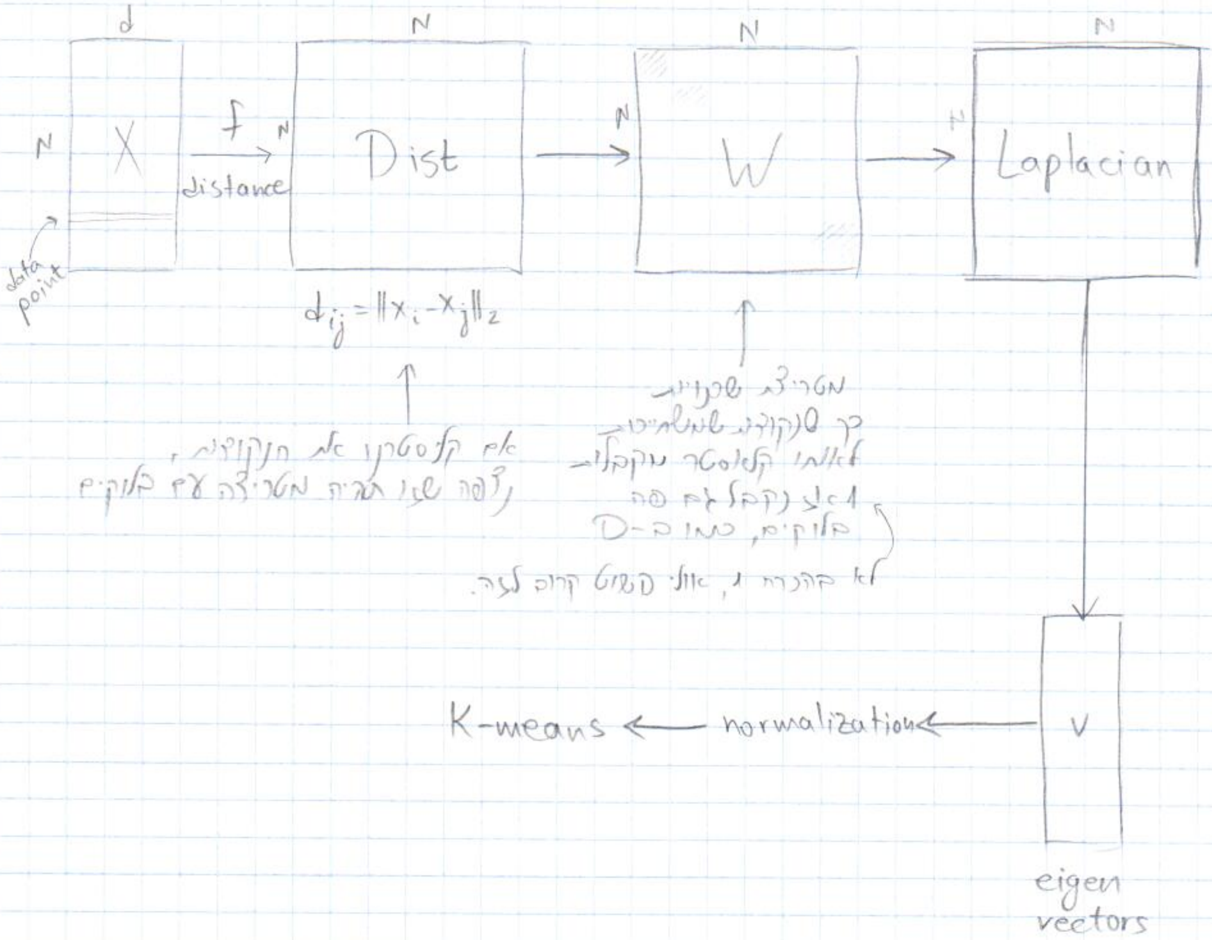
③ נבחר נקודות התחלה טובות יותר עם K-means ++

בחירת k: נקלסטר את הדאטא לכל k ואז נבחר את k שהוא הקפיצה הכי גדולה בציון

ותמיד כדאי לעשות ויזואליזציה על הדאטא עוד.

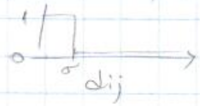
Clustering - המשך

Spectral Clustering II

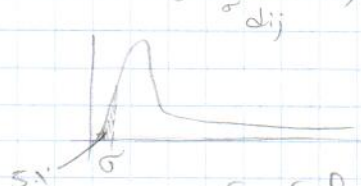


איך עוברים בין Dist לבין W מטריצת השכנויות?

<u>הצעה 1</u>: נעשה <u>thresholding</u> על Dist: הנקודות שערכן קטן מ-σ יקבלו 1 והיתר 0.



איך נבחר את הסף σ?

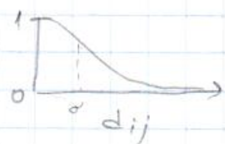


נסתכל על ההיסטוגרמה של המרחקים בין זוגות:

ונבחר σ כך שרק 5% מהזוגות יעברו את הסף למשל.

<u>הצעה 2</u>: נשתמש בקרנל גאוסייני כך ש-

$$w_{ij} = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)$$



ואת σ נבחר באופן דומה למה שאמרנו קודם

(נגדיל את σ => יהיו לנו יותר שכנים)

אז עכשיו כל קלאסטר בדאטא מיוצג ע"י רכיב קשירות בגרף המיוצג ע"י W.

APML
שיעור 8

המשך: Spectral Clustering II

איך נמצא רכיבי קשירות בגרף?

נגדיר מטריצה נוספת:

$$\boxed{\begin{matrix} D \\ \text{degree} \end{matrix}} \leftarrow d_{ii} = \sum_j w_{ij} =$$ סכום השורה ה-i ב-W

אפשר להסתכל על שאלת הקלאסטרינג כמציאת חתך מינימלי בגרף (אולי כמה פעמים)

עבור שני קלאסטרים קודקודים:

נגדיר:

- $|A| = \sum_{i \in A} \mathbb{1}$
- $vol(A) = \sum_{i \in A} d_{ii} = \sum_{i \in A} \sum_j w_{ij}$

ועכשיו את ה- clustering אפשר להגדיר כ:

$$\underset{A,B}{\operatorname{argmin}} \; CUT(A,B) = \sum_{\substack{i \in A \\ j \in B}} w_{ij}$$

- יש לנו אלגוריתמים שפותרים את זה: ← פורד-פולקרסון $O(VE^2)$
עוד אלגוריתמים...

מה הבעיה? outliers!

אם יש נקודה בודדת רחוקה מהשאר, פשוט נקבל אותה כקלאסטר ואת השאר כקלאסטר...

איך נפתור את זה? נוסיף רגולריזציה שתלויה בגדלי הקלאסטרים

$$\underset{A,B}{\operatorname{argmin}} \; CUT(A,B) \cdot \underbrace{\left[\frac{1}{|A|} + \frac{1}{|B|}\right]}$$

קרוב ל-1 עבור חלוקה לא שוויונית של הנקודות
וקרוב ל-0 עבור חלוקה חצי חצי בין הקלאסטרים.

עכשיו זה NP-קשה אפילו לקירוב ☹

Spectral Clustering : המשך II

אפשרות אחרת לרגולריזציה: נשתמש ב- volume:

$$\underset{A,B}{\arg\min}\ CUT(A,B)\cdot\left[\frac{1}{vol(A)}+\frac{1}{vol(B)}\right]$$

ולמרבה הצער גם זה NP-קשה לקירוב.

Graph Laplacian:

נגדיר את מטריצת הלפלסיאן: $L = D - W$

laplacian, degree, adjacency

נשים לב שהסכום של כל שורה ב-L הוא 0. כלומר $\vec{1}$ הוא הוקטור העצמי הראשון המשויך לע"ע $\lambda=0$.

$$\begin{bmatrix} L \end{bmatrix}\cdot\begin{bmatrix}1\\1\\1\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix}$$

בנוסף, L ריבועית וסימטרית.

האם היא גם PSD?

תזכורת: Positive Semi Definite = PSD $\Longleftrightarrow$

כל הערכים העצמיים ממשיים ואי-שליליים $\Longleftrightarrow$

לכל וקטור f, $fLf \geq 0$.

זה אומר שהטרנספורמציה ש-L עושה ל-f משנה את f בפחות מ-90°.

נוכיח שהיא PSD:

$$f^TLf = f^TDf - f^TWf = \sum_i d_i f_i^2 - \sum_{i,j} f_i f_j w_{ij} =$$

$$= \sum_i \sum_j w_{ij} f_i^2 - \sum_{i,j} f_i f_j w_{ij} =$$

$$= \frac{1}{2}\left(\sum_i \sum_j w_{ij} f_i^2 - 2\sum_i \sum_j f_i f_j w_{ij} + \sum_j \sum_i w_{ij} f_j^2\right) =$$

$$= \frac{1}{2}\sum_{i,j} \underbrace{w_{ij}}_{\geq 0} \cdot \underbrace{(f_i - f_j)^2}_{\geq 0} \geq 0$$

$\Leftarrow$ ניתן לסדר $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda$

λ_1: $\mathbb{1}$, λ_2: Fiedler Vector

Spectral Clustering II :המשך

\# מה שאר הערכים העצמיים מלמדים אותנו?

אם הגרף הוא איחוד של רכיבי קשירות שהם מנותקים אחד מהשני,
אז עבור הוקטור המציין שבו אחדות במקומות המתאימים לרכיב קשירות
אחד ואפסים בכל השאר, הוא גם יהיה עם הערך העצמי 0.
במילים אחרות, לכל קלסטר קיים וקטור שמציין לע"ע 0 $\Leftarrow$
מספר הוקטורים העצמיים המתאימים לע"ע 0 הוא מספר הקלסטרים.

זה לא חייב להיות ממש 0
אבל קרוב ל-0 אם הם כמעט קשורים
לא מופרדים באופן מושלם.

\# אבל מה עם הנרמול לפי גודל הקלסטר? Normalized Laplacian

1. Shi & Malik, 2000: $L_n = D^{-1} L$

זה נרמול של כל שורה שהופך את זה להסתברויות.
(מקושר להילוך מקרי = Random Walk)

2. Ng, Jordan & Weiss, 2002:

$$L_{sym} = \underbrace{D^{-\frac{1}{2}} \cdot L}_{\text{מחלקים כל שורה בשורש הדרגה}} \cdot \underbrace{D^{-\frac{1}{2}}}_{\text{מחלקים כל עמודה בשורש הדרגה}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

המטריצה הזו נשארת סימטרית, וגם PSD

$\boxed{\text{לקרוא את ה-notes שלו}}$

\# סיכום, האלגוריתם שהצגנו:

קלט: X, $x_i \in \mathbb{R}^d$

1. חשב $S_{n\times n}$ מרחקים: $\forall ij \; S_{ij} = \|x_i - x_j\|$
2. חשב $W_{n\times n}$ מטריצת שכנויות: $W_{ij} = \exp\left(\frac{-S_{ij}^2}{2\sigma^2}\right)$

   עבור $\sigma \triangleq$ ith percentile of S
3. חשב $D_{n\times n}$ מטריצת דרגות: $D_{ii} = \sum_j W_{ij}$, $D_{ij} = 0, \; i \neq j$
4. חשב $L_{sym\,n\times n}$: $L_{sym} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$
5. חשב את k הוקטורים העצמיים הראשונים: $U = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_k \\ | & | & & | \end{bmatrix}$

   כש-k נבחר לפי שיטת ה-eigen-gap.
6. חשב T שהוא נרמול של השורות של U: $T_i = \frac{U_i}{\|U_i\|}$ $\rightarrow$ כל הנקודות על כדור היחידה אז וקטורים שמצביעים לאותו כיוון יהיו קרובים.
7. הרץ k-means לוקטורי השורות של T.

Clustering: המשך

t-distribution Stochastic Neighbour Embedding t-SNE III

נניח שיש לנו אוסף נקודות X: $X_i \in \mathbb{R}^p$

ונרצה לעשות embedding של X ל-Y: $Y_i \in \mathbb{R}^d$ $\quad d << p$

כך שהמרחקים לא ישתנו הרבה, כלומר $d'_{ij} \approx d_{ij}$

$$d_{ij} = \|x_i - x_j\|$$

אבל לא באמת אכפת לנו לשמור את כל המרחקים אלא רק את הדמיון -

נרצה שהשכנים "ישארו שכנים" אבל פחות אכפת לנו מנקודות רחוקות.

נסתכל על זה כמו הילוך מקרי:

$$P_{j|i} = \frac{\exp(-d_{ij}^2/2\sigma_i^2)}{\sum_{k\neq i}\exp(-d_{ik}^2/2\sigma_i^2)}$$

ההסתברות לעבור מ-i ל-j

ועכשיו נרצה שהמרחק בין ההתפלגויות יהיה קטן, ונמדוד את זה בעזרת מרחק-KL.

$$\underset{Y}{\operatorname{argmin}}\ C = \sum_i KL(P_i \,\|\, Q_i)$$

ההתפלגות של מעבר מ-i ב-X   ההתפלגות של מעבר מ-i ב-Y

$$q_{j|i} = \frac{\exp(-d'^2_{ij}/2\sigma'^2_i)}{\sum_{k\neq i}\exp(-d'^2_{ik}/2\sigma'^2_i)}$$

תזכורת: אנטרופיה: $H(P_i) = \sum_j P_{ij}\log_2(P_{ij})$

perplexity: $2^{H(P_i)}$ ⟵ זה בעצם מייצג את כמות השכנים

KL: $KL(P_i\|Q_i) = \sum_j P_{j|i}\log\frac{P_{j|i}}{Q_{j|i}}$

$$\Rightarrow \underset{Y}{\operatorname{argmin}} \sum_i KL(P_i\|Q_i) = \sum_i\sum_j P_{j|i}\log\frac{P_{j|i}}{Q_{j|i}}$$

נשים לב שזה לא סימטרי: אפילו אם נשנה מאוד את $Q_{j|i}$
אבל $P_{j|i}$ מאוד קטן (כלומר נקודות רחוקות), זה יהיה
מספר קטן בסכימה <= פחות אכפת לנו. <= זה לא סימטרי.

נאתחל Y באופן אקראי ואז נחשב את המרחקים ונסתכל על הגרדיאנט:

$$\frac{\partial C}{\partial y_i} = 2\sum_j \left(P_{j|i} - Q_{j|i} + P_{i|j} - Q_{i|j}\right)(y_i - y_j)$$

כלל: $Y^{(t)} = Y^{(t-1)} + \eta\frac{\partial C}{\partial Y} + \alpha(t)\left[Y^{(t-1)} - Y^{(t-2)}\right]$

מומנטום

SNE

t-SNE : המשך

מה הבעיה ב-SNE שהצגנו בעמוד הקודם?

1. רגיש ל- outliers

2. בהורדה של המימד אנו מקודדים את הדאטא, לא נוכל לשמר את כל המרחקים אז או שיהיה לנו underestimation של נקודות קרובות, או שיהיה over-estimation של נקודות רחוקות.

את הנקודה השנייה פותרים ב- t-SNE בעזרת t-distribution (התפלגות t).
התפלגות t היא כמו התפלגות נורמלית אבל עם זנב כבד.
אז נשתמש בגרעין של התפלגות t:

$$q_{j|i} = \frac{(1+\|y_i - y_j\|^2)^{-1}}{\sum_k (1+\|y_i - y_k\|^2)^{-1}}$$

בנוסף, במקום להסתכל על הסתברות מותנית של j בהינתן i, הם הסתכלו על התפלגויות משותפות של i ו-j:

$$C = \underset{Y}{\operatorname{argmin}}\ KL(P \| Q)$$

התפלגות מולטינומית למעבר מכל i לכל j

$$\frac{\partial C}{\partial y_i} = 2\sum_j (P_{ji} - Q_{ji}) \cdot (Y_i - Y_j) \cdot [\text{term}]$$

וככה גם נותנים פחות משקל ל- outliers.