בינה מלאכותית – תרגיל 3

בכך כהן

[redacted]

1) (a) תחילה נראה כי $\|A^tA - B^tB\|_2^2 = \lambda_1^2$

נסמן: $M = A^tA - B^tB$

$$\|M\|_2^2 = \max_{\|x\|=1} \|Mx\|^2 = \max_{\|x\|=1} (Mx)^t Mx =$$

$$\max_{\|x\|=1} x^t M^t M x = \max_{\|x\|=1} x^t M^2 x$$

המטריצה M^2 סימטרית ממשית ולכן ניתן ללכסן אותה באופן הבא: $M^2 = PDP^{-1}$

כאשר D ו-P מטריצות ממימד $d \times d$

P מטריצה אורתונורמלית

D מטריצה אלכסונית עם ע"ע של M^2

כך $\lambda_1^{M^2} \geq \lambda_2^{M^2} \geq \dots \geq \lambda_{rank(M^2)}^{M^2} - 0$

לכן:

$$\|M\|_2^2 = \max_{\|x\|=1} x^t PDP^{-1} x \underset{\|y\|=1}{=\max} (Py)^t PDy =$$

נסמן: $y = P^{-1}x$

כלומר $x = Py$

$$\max_{\|y\|=1} y^t P^t P D y = \max_{\|y\|=1} y^t D y = \max_{\|y\|=1} \sum_i \lambda_i^{M^2} y_i^2$$

$$\leq \sum_i \lambda_1^{M^2} y_i^2 = \lambda_1^{M^2} \sum_i y_i^2 = \lambda_1^{M^2}$$

לצורך קיימת התכונה הזאת: $\lambda_1^{M^2} = (\lambda_1^M)^2$

ולכן קיבלנו כי $\|M\|_2^2 \le (\lambda_1^M)^2$

כעת נסתכל על V_1, הוקטור העצמי שמתאים ל-λ_1^M:

$$\|MV_1\|^2 = \|\lambda_1^M V_1\|^2 = (\lambda_1^M)^2 \|V_1\|^2 \underset{V_1 \text{ מנורמל}}{=} (\lambda_1^M)^2$$

מצאנו את הוקטור שממקסם את $\|Mx\|^2$ וראינו כי התוצאה במקסימום מגיעה ל-$(\lambda_1^M)^2$ ומכאן השוויון המבוקש.

כעת נראה כי $\max_{\|x\|=1} (x^t Mx)^2 = (\lambda_1^M)^2$

המטריצה M סימטרית וממשית ולכן ניתן ללכסן אותה באופן הבא: $M = PDP^{-1}$

באופן דומה ללכסון הקודם: $\lambda_1^M \ge \lambda_2^M \ge \dots$

ולכן:

$$\max_{\|x\|=1} (x^t Mx)^2 = \max_{\|x\|=1} (x^t PDP^{-1}x)^2 =$$

נסמן: $y = P^t x$, $x = Py$

$$\max_{\|y\|=1} ((Py)^t P D P^t x)^2 =$$

$$\max_{\|y\|=1} (y^t P^t P D y)^2 = \max_{\|y\|=1} (y^t D y)^2 = \max_{\|y\|=1} \left(\sum_i \lambda_i^M y_i^2\right)^2 \le (\lambda_1^M)^2$$

כאשר קודם את המעברים הבינוניים

נסתכל על V_1:

$$((V_1)^t M V_1)^2 = ((V_1)^t \lambda_1^M V_1)^2 = (\lambda_1^M V_1^t V_1)^2 = (\lambda_1^M)^2$$

שוב ראינו כי V_1 נותן את המקסימום ונגיע לשוויון.

(ב) נסתכל על שני מקרים אפשריים:

(i) $\mathrm{rank}(A^tA) \le l$

במקרה כזה:

$$\|A^tA - B_{opt_T}^t B_{opt_T}\| = \|V\Sigma U^t U\Sigma V^t - V'(\Sigma')^2 V'^t\|$$

$$= \|V\Sigma^2 V^t - V'(\Sigma')^2 V'^t\| = 0$$

מאחר ש- $\Sigma = \Sigma'$ ו- $V' = V$, לפי הגדרה של Σ' ו-V'.

(ii) $\mathrm{rank}(A^tA) > l$

בנוסף $\mathrm{rank}(B^tB) \le l$

לכן, לפי המשפט היסודי של האלגברה, המימד של $\ker(B^tB)$ הוא לפחות $d-l$.

נסתכל על אוסף הוקטורים העצמיים של A^tA: $v_1, \dots, v_{l+1}$

הם פורשים תת מרחב ממימד $l+1$.

לפי עקרון שובך יונים, בחתוך המרחבים האלו מתקיים:

$\ker(B^tB) \cap \mathrm{span}\{v_1, \dots, v_{l+1}\}$ הוא תת מרחב ממימד 1 לפחות

נבחר וקטור יחידה כלשהו z מתוכו: $\|z\| = 1$

$$z \in \ker(B^tB) \cap \mathrm{span}\{v_1, \dots, v_{l+1}\}$$

כלומר: $B^tBz = 0$ ו- $z = \sum_{i=1}^{l+1} \alpha_i v_i$

$$\|A^tA-B^tB\| \geq \|(V\Sigma^2V^t-B^tB)z\| \underset{B^tBz=0}{=} \|(V\Sigma^2V^t)z\|$$

$$=\Big\|\Big(\sum_i \sigma_i^2 v_i v_i^t\Big)\Big(\sum_i \alpha_i v_i\Big)\Big\| \underset{v_i \text{ אורתונורמליים}}{=} \Big\|\sum_i \alpha_i \sigma_i^2 v_i v_i^t v_i\Big\|$$

$$=\Big\|\sum_{i=1}^{l+1} \alpha_i \sigma_i^2 v_i\Big\| \geq \sigma_{l+1}^2 \Big\|\sum_{i=1}^{l+1}\alpha_i v_i\Big\| = \sigma_{l+1}^2 \|z\| = \sigma_{l+1}^2$$

קיבלנו כי: $\|A^tA-B^tB\| \geq \sigma_{l+1}^2$

נראה כי השוויון מתקיים עבור B_{OPT}:

$$\|A^tA-B_{OPT}^tB_{OPT}\| = \|V\Sigma^2V^t - V'(\Sigma')^2(V')^t\|$$

$$=\max_{\|x\|=1}\|(V\Sigma^2V^t-V'(\Sigma')^2(V')^t)x\|$$

נסמן ב-y את הוקטור שממקסם את הביטוי, ולכן:

$$=\|(V\Sigma^2V^t-V'(\Sigma')^2(V')^t)y\|$$

$y=\sum_{i=1}^{rank(A^tA)}\alpha_i v_i$ לכן:

$$=\Big\|\Big(\sum_{i=1}^{rank(A^tA)}\sigma_i^2 v_i v_i^t - \sum_{i=1}^{l}\sigma_i^2 v_i v_i^t\Big)\Big(\sum_{i=1}^{rank(A^tA)}\alpha_i v_i\Big)\Big\|=$$

$$\Big\|\Big(\sum_{i=l+1}^{rank(A^tA)}\sigma_i^2 v_i v_i^t\Big)\Big(\sum_{i=1}^{rank(A^tA)}\alpha_i v_i\Big)\Big\| \underset{v_i \text{ אורתונורמליים}}{=}$$

$$\Big\|\sum_{i=l+1}^{rank(A^tA)}\alpha_i\sigma_i^2 v_i v_i^t v_i\Big\| = \Big\|\sum_{i=l+1}^{rank(A^tA)}\alpha_i\sigma_i^2 v_i\Big\| = \sqrt{\sum_{i=l+1}^{rank(A^tA)}\alpha_i^2\sigma_i^4}$$

y שממקסם את הביטוי מוכרח לקיים $\|y\|=1$,
כלומר $\sum_{i=1}^{rank(A^tA)} \alpha_i^2=1$ והדבר מושג רק כאשר

$\alpha_{l+1}=1$ ו-$\alpha_i=0$ ל-$i\neq l+1$, מאחר שאז נעצב את

σ_{l+1} במלואו, שהוא הערך הסינגולרי הגדול יותר מבין σ_i

הנתונים $(l+1\le i\le rank)$.

נציב את ערכי α_i ונקבל:

$$\|A^tA-B^tB\| = \sqrt{1\cdot \sigma_{l+1}^4} = \sigma_{l+1}^2$$

(c) הגבול העליון של $\|A^tA-\hat{B}^t\hat{B}\|$ הוא

$\frac{2}{\ell}\|A\|_F^2$. נסביר את מהות ההוכחה:

תחילה ניעזר בטענה כי:

$$\|\hat{B}\|_F^2 \leqslant \|A\|_F^2 - \frac{\ell}{2}\sum_{i=1}^{n}\delta_i$$

מאחר ש- $\|\hat{B}\|_F^2$ אי שלילי נקבל:

$$0 \leqslant \|A\|_F^2 - \frac{\ell}{2}\sum_{i=1}^{n}\delta_i$$

כלומר: $\sum_{i=1}^{n}\delta_i \leqslant \frac{2}{\ell}\|A\|_F^2$

כעת ניעזר בטענה נוספת:

$$\|A^tA-\hat{B}^t\hat{B}\| \leqslant \sum_{i=1}^{n}\delta_i$$

סך הכל נקבל:

$$\|A^tA-\hat{B}^t\hat{B}\| \leqslant \sum_{i=1}^{n}\delta_i \leqslant \frac{2}{\ell}\|A\|_F^2$$

כלומר: $\|A^tA-\hat{B}^t\hat{B}\| \leqslant \frac{2}{\ell}\|A\|_F^2$

כנדרש

2) נוכיח כי האלגוריתם נכון;

לפי סעיף (א) כל השלישיות שנספרות הן משולשים. לכן, נותר להראות כי כל השלישיות המשולשות נספרות פעם אחת בדיוק.

בהינתן השלישיה (k,i,j), בה"כ $k<j<i$, השלישיה תתווסף לשלשות הפלט כאשר נעבד צלעות i, שם נראה כי j ו-k הם שכנים של i. בשלב (א) נראה ש-(k,j) היא קשת בגרף ולכן נספור את השלישיה.

לכל שלישיה ניתן ליצור שלוש פרמוטציות אבל לכל היותר אחת, כי: $>$ הוא יחס סדר מלא בין הצמתים.

3) נעזר בשלושה סבבים של MapReduce:

בשלב הראשון, נקבל כל קשת בגרף כמפתח $\langle v,u \rangle$ ונעביר הלאה רק קשתות עבורן מתקיים ~~[illegible]~~ $v<u$. הקשת תועבר בצורה הבאה: u ישמש כמפתח ו-v כערך.

ה-Reducer הראשון מקבל צומת v ורשימה של שכניו, ומעביר הלאה משולשים פוטנציאליים כך ש- $w<v<u$ כאשר u ו-w הם שכנים של v.

ה-Mapper השני מקבל שני סוגי קלט; מה-Reducer הראשון הוא יקבל צומת וכל זוגות שכניו. כמו כן, נדאג לשלוח אליו את כל קשתות הגרף, כשבמקרה הראשון נשלח את זוג השכנים ~~[illegible]~~ כמפתח ואת הצומת כערך, ואילו במקרה השני נעביר את הקשת הלאה עם הסימון $, כדי לציין שקשת זו קיימת, ופוטנציאלית מגדירה משולש בגרף.

ה-Reducer השני מקבל כמפתח זוג שכן
לרשימת צמתים המשותפת לשכן. בנוסף, ברשימה
תהיה $ במקרה שקיימת קשת בין זוג זה.
ה-Reducer, במקרה שאכן קיימת קשת כזו, יגדיל
שנוצר משולש סגור, ויפלוט הודעה להגדלת counter.
ה-Mapper הבא יעביר כפלט את ההודעה ל-Reducer
הסופי אשר יסכם את מספר ההודעות הללו.

להלן הפסאודו-קוד:

```
Map1 (key: <u,v> ∈ E, value: null)
    if (u < v)
        Emit(u; v)

Reduce1 (key: v ∈ V, value: S ⊆ V)
    for (u,w) ∈ S^2
        if u < w
            Emit(v; <u,w>)

Map2 (key, value)
    if key of type (v; <u,w>)
        Emit(<u,w>; v)

    if key of type (<u,w>; null)
        Emit(<u,v>; $)
```

Reduce 2 (key: $\langle u,w \rangle$, value: $S \subseteq V \cup \{\$\}$)

if $\$ \in S$

for $v \in S / \{\$\}$

Emit ($\$$, 1)

Map 3 (key: x, value: 1)

Emit(x,1)

Reduce 3 (key: x, value: S)

count:=0

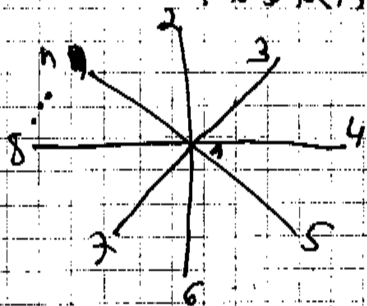
for each item in S

count++

4) (a) עבור גם באלגוריתם הטורי, ככל ש-p גדול
יותר, זמן הריצה וכמות הזיכרון הנדרשים לחישוב
ושמירת המשולשים הפתוחים גדולים יותר.
גם ב-MapReduce, ככל ש-p גדול יותר, סביר שזמני
הריצה של ה-Mappers וה-Reducers יהיו גדולים
יותר, וכן גם כמות הזיכרון הנדרשת. כך למשל,
ב-Reduce1 וב-Map2 נשלחו יותר משולשים פתוחים
והביצוע ב-flow ידרוש לבדוק בגם גם מקרים ולעבור אותם.
עבור M, באלגוריתם הטורי, אין השפעה על
המשתנים, וזאת מאחר שחלוקת המשולשים הפתוחים
בין זוגות הצמתים אין משמעות באלגוריתם הטורי
שיעבור על כל המשולשים הפתוחים בין כל זוג של כל
הצמתים.
לעומת זאת, ב-MapReduce, כאשר M גדול, חלוקת
העבודה בין ה-Reducers תהיה לא יעילה, מאחר
שאז, ב-Reduce1, יהיו key-ים רבים שעבורם
רשימת ה-values יהיו קטנות (ולכן רוב
ה-Reducers ימצאו ריק) ויהיו "מעט" key אחד בעל
רשימת values ארוכה (המספר של הצמתים שקשור ל-M)
ולכן ה-Reducer המתאים יעבוד הרבה.

(ל) נראה הבדלים מהותיים בשתי הדוגמאות הבאות:

דוגמא א':



בדוגמה זו כל המשולשים הפתוחים הם מהצורה:

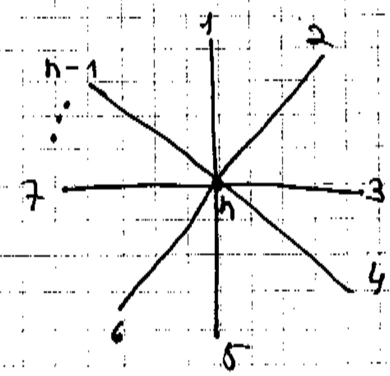
$(i,1,j)$

$1<i<j\le n$

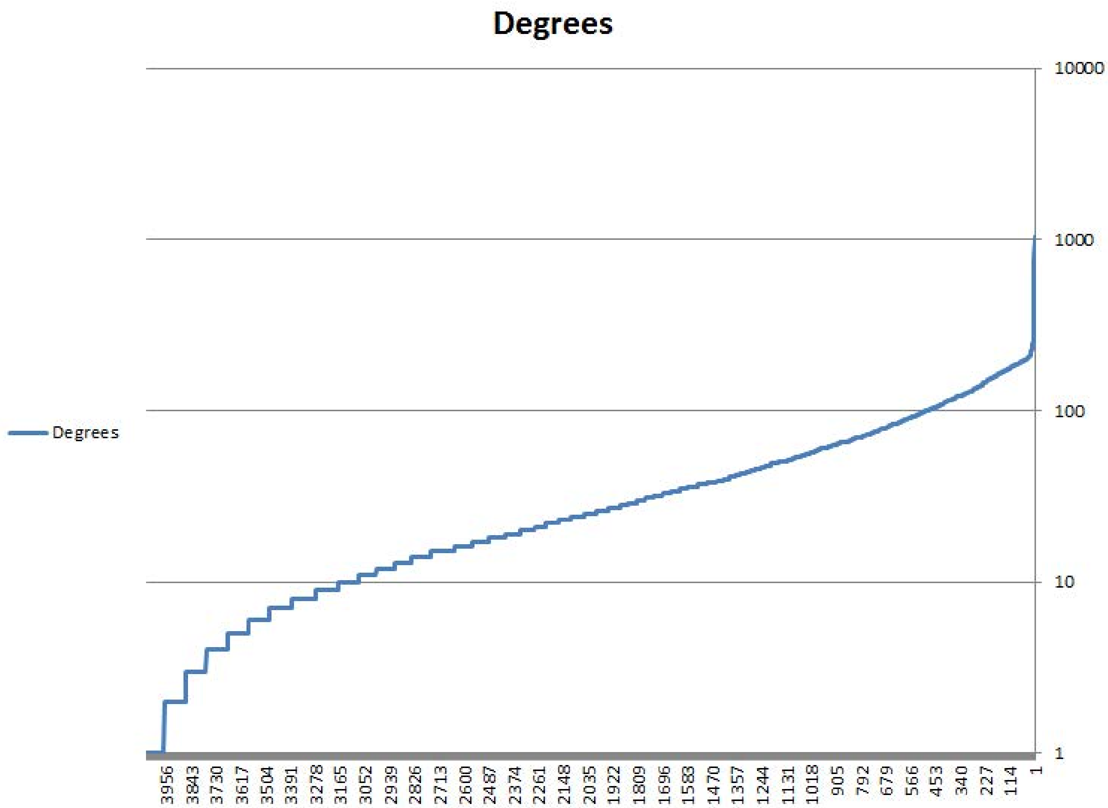
סה"כ נקבל $P2=\binom{n-1}{2}$

מאחר שכל המשולשים הפתוחים כוללים את 1 כצומת אמצעי, אזי $P2=M$ ולכן $M=\binom{n-1}{2}$

דוגמה ב':



בדוגמה זו אין משולשים פתוחים שניתן ליצור, מאחר שכל המסלולים באורך 2 חייבים לעבור דרך n, ולכן כל משולש פתוח חייב את n כצומת אמצעי, אולם אז לא ניתן למצוא שני צמתים שמופיעים אחרי n בסדר שנקבע.



Degrees
Degrees
10000
1000
100
10
1
3956
3843
3730
3617
3504
3391
3278
3165
3052
2939
2826
2713
2600
2487
2374
2261
2148
2035
1922
1809
1696
1583
1470
1357
1244
1131
1018
905
792
679
566
453
340
227
114
1

5) ניתן לראות בגרף כי ישנם מעט צמתים עם דרגה גבוהה ורוב הצמתים הם בעלי דרגה נמוכה.

6) (a) $P2 = 3{,}975{,}462$

$M = 543{,}403$

(b) $P2 = 1{,}922{,}379$

$M = 7{,}750$

(c) $P2 = 4{,}823{,}621$

$M = 546{,}490$

נשים לב להשפעה כמספר l מאחר שבו נקבל את זמן הריצה בקלט רגיל באלגוריתם הסטנדרטי וגם ב-MapReduce. כמו כן, במספר l הגם בקלט M קטן יותר לכן במספר הגבוה הגיע הפערים רבים ב-MapReduce.

7) מספר המשולשים המאותרים: 1,612,010

השגנו מספר זה ע"י השלמת הצלעות על המשולשים הפתוחים שנמצאו בשלב הקודם (ראה מסמכים) ובדיקה האם כל משולש פתוח הוא סגור.

8) להלן הפסאודו-קוד להערכת $|N(i) \cap N(j)|$:

תחילה נחשב את $S(i)$ לכל צומת i :

foreach $i \in V$

  $S(i) \leftarrow$ Find $\min\{k, |N(i)|\}$ smallest

    $h(j)$ s.t. $j \in N(i)$

  sort$(S(i))$ s.t. $S_{i_1} \leq S_{i_2} \leq \ldots \leq S_{ik}$

כעת נחשב את האומד ל- $|N(i) \cap N(j)|$:

Given $S(i)$ and $S(j)$

  if $|N(i)| \leq k$ and $|N(j)| \leq k$

    return $|S(i) \cap S(j)|$

  if $(|N(i)| < k)$

    $S_{ik} \leftarrow 1$

  if $(|N(j)| < k)$

    $S_{jk} \leftarrow 1$

  $T \leftarrow \min\{S_{ik}, S_{ij}\}$

  $S_{union} \leftarrow \emptyset$

  for each $t \in S(i) \cup S(j)$

    if $t \leq T$

      $S_{union} \leftarrow S_{union} \cup \{t\}$

  $k' \leftarrow |S_{union}|$

המשך בעמוד
הבא ←

```
counter ← 0
for each t ∈ S_union
    if t ∈ S(i) ∩ S(j)
        counter++
```

$$\hat{\alpha} \leftarrow \frac{counter}{k'}$$

$$\hat{\beta} \leftarrow \frac{k'-1}{T}$$

return $\hat{\alpha}\hat{\beta}$

9) עבור $k=10$, האומד שקיבלנו הוא 1,965,477 והטעות היחסית היא 0.219.

- ה-tradeoff בין זמן הריצה ודיוק ההערכה נשלט ע"י ערכו של k. ככל ש-k גדל יותר, זמן הריצה גדול יותר אך השגיאה היחסית נמוכה יותר. למשל, נסתכל על התוצאות הבאות:

k	זמן ריצה (ms)	טעות יחסית
5	338	0.581
10	634	0.219
20	1263	0.089

10) בהינתן צומת i, ניתן לחשב את מספר המשולשים שתלוים ב-i באמצעות הביטוי הבא:

$$T_i = \frac{1}{2} \sum_{j \in N(i)} |N(i) \cap N(j)|$$

הסבר: כאשר נעבור על כל השכנים של i, אזי עבור שכן j נספור משולשים כמספר הצמתים שמשותפים בשכנותם ל-i ו-j. עם זאת, כל משולש יספר פעמיים (על ידי כל אחד משני שכניו של i) ולכן נכפיל ב-$\frac{1}{2}$.

על מנת להעריך את T_i ניתן להשתמש ב-sketches על מנת להעריך את $|N(i) \cap N(j)|$ באותו האופן שהתבצע בתשובה לשאלה 8.
$\hat{T}_i$ יהיה מחצית סכום ההערכות ל-$|N(i) \cap N(j)|$ כך ש-j הוא שכן של i.