אלגוריתמים מבוזרים ־ הרצאה 6, בעיית MDS

שרון מלטר, אתגר 17

22 בספטמבר 2024

תוכן עניינים

1 הקדמה

היום נדבר על בעיית MDS ונפתור אותה עם אלגוריתם רנדומי פשוט.

2 הגדרת בעיית MDS

$Dominating\ set$ או "קבוצה שולטת" D היא קבוצה של צמתים בגרף כך שלכל צומת v מתקיים $v \in D$ או שיש לו שכן בקבוצה. יש דמיון לבעיה של כיסוי בצמתים, אך כפי שניתן להבין לא כל קשת חייבת להכיל צומת שנמצא בקבוצה.

$MDS\ -\ Minimal\ Dominating\ Set$

2.1 מה ידוע לנו על הבעיה?

- זוהי בעיה NP־ קשה קלאסית (שנחקרה רבות)
- בעולם של אלגוריתמים סדרתיים קיימים קירובים של $O(log\Delta)$ כאשר Δ היא הדרגה המקסימלית בגרף.
- אלגוריתמים פולינומיים לא יכולים להשיג קירוב טוב יותר, בין היתר תחת ההנחה הסבירה ש־ $P \neq NP$.

3 מטרה

נרצה להראות אלגוריתם מבוזר יעיל שמשיג קירוב $O(log\Delta)$.
(כאשר ב־ 'יעיל' אנו מתכוונים לפולינום של $O(logn)$)
נתחיל דווקא מלהציג אלגוריתם סדרתי קלאסי לפתרון הבעיה.

4 אלגוריתם סדרתי

באלגוריתם נבנה קבוצה D שמאותחלת $D = \varnothing$ ובהדרגה נוסיף אליה צמתים.
נאמר שצומת v **מכוסה** אם $v \in D$ או שיש לו שכן שנמצא ב־ D.
במהלך האלגוריתם, לכל צומת v נסמן ב־ $\rho(v)$ את מספר הצמתים בסביבת 1 של v (השכנים שלו) שאינם מכוסים.

1. בכל שלב, נוסיף ל־ D את הצומת v בעל $\rho(v)$ מקסימלי.
2. נמשיך עד שכל הצמתים מכוסים :)

4.1 נכונות

קודם כל, ברור שבסיום האלגוריתם כל הצמתים מכוסים (ממשיכים להוסיף ל־ D צמתים עד שכולם מכוסים)
נותר להראות ש־ $|D| \leqslant O(log\Delta)|D^*|$ כלומר, שהפתרון שלנו הוא קירוב $O(log\Delta)$ לפתרון.
נניח כעת שאי־השוויון נכון, ונוכיח לאחר מכן משהו דומה.

5 סימולציה מבוזרת לאלגוריתם הסדרתי

ניתן לסמלץ את האלגוריתם הסדרתי באופן מבוזר, אך לא מדובר בפתרון יעיל. בכל שלב מוסיפים רק צומת יחיד ל־ D לכן במקרה הגרוע נצטרך מספר לינארי של איטרציות. כמו כן, מחפשים בכל פעם $\rho(v)$ מקסימלי, לכן נצטרך בין כל הצמתים לכל הצמתים. נצטרך d סיבובים כאשר d הוא קוטר הגרף.

6 פתרון מבוזר

6.1 ניסיון ראשון

- בכל שלב, נוסיף ל־ D את כל הצמתים בעלי $\rho(v)$ מקסימלי ב־ 2־סביבה שלהם (מצריך רק 2 סיבובים)

אלגוריתם זה פותר את שתי הבעיות שהיו לנו מקודם;

1. בכל שלב, יש אפשרות להוסיף הרבה צמתים במקביל.
2. מסתכלים רק על $\rho(v)$ מקסימלי ב־ 2־סביבה, כלומר מדובר בחישוב לוקאלי של $O(1)$ סיבובים.

6.2 בעיה

כביכול הכל מושלם. אבל אנחנו עלולים לקחת יותר מדי צמתים, כך שהקירוב לא מובטח.
למשל: קליקה.
בקליקה כל הצמתים נראים אותו הדבר, כך שנרצה לקחת את כולם, אך מספיק לקחת אחד.

7 ניסיון שני

רעיון: בדוגמה שראינו נרצה לשבור סימטריה, כך שלמרות שכל הצמתים נראים לנו אותו הדבר, רק אחד ייכנס ל־ D. ניתן לעשות זאת לפי המזהים; על כל צומת v נסתכל עליו כ־ $(\rho(v), id(v))$ ולכל צומת u בסביבה־2 שלו נוסיף את v ל־ D רק אם $\rho(v) \geqslant \rho(u)$ וגם $id(v) > id(u)$.
אזי האלגוריתם החדש שלנו הוא:

1. בכל שלב נוסיף ל־ D את כל הצמתים בעלי $(\rho(v), id(v))$ מקסימלי ב־2־סביבה שלהם.

7.1 בעיה

חשבתם שכבר סיימנו?

פתרנו את בעיית הקליקה, אך נסתכל על גרף מסלול (או "שרוך" כפי שאחרים אומרים) עם המזהים $(1, 2, 3, 4, 5, 6, 7)$ בהתחלה כמעט לכל הצמתים בגרף יש ערך $\rho(v) = 3$ אך רק צומת 6 הינו גם בעל מזהה מקסימלי ולכן רק הוא ייבחר באיטרציה הראשונה.

דוגמה נוספת היא קליקה שבה לכל צומת יש מספר שכנים נוספים שלא בקליקה. כאן הפתרון האופטימלי הוא להוסיף את כל צמתי הקליקה, מכיוון שהם מכסים גם את עצמם וגם את כל שכניהם שמחוץ לקליקה (ויש לפחות שני שכנים שאינם בקליקה לכל צומת מהקליקה) כלומר, נרצה פשוט להוסיף מייד את כל צמתי הקליקה.
אך לפי האלגוריתם הנוכחי נוסיף רק צומת אחד בכל שלב.

שימו לב שנתקלנו בשתי בעיות הפוכות. מצד אחד אנחנו לא רוצים להוסיף יותר מדי צמתים ולפגוע בקירוב, אך אנחנו גם רוצים להיות יעילים ולא לבצע יותר מדי סיבובים.
נגדיר מטרה לפתרון הבעיות: אנחנו רוצים להוסיף 'מעט' צמתים שייכסו הרבה צמתים שונים.
נצטרך אלגוריתם מורכב יותר בשביל לפתור את הבעיה.

8 ניסיון שלישי

סימון: נסמן ב־ $\hat{\rho}(v)$ את החזקה הקרובה של 2 שגדולה ממש מ־ $\rho(v)$.

באלגוריתם החדש נסתכל על הצמתים בעלי הערך $\hat{\rho}(v)$ המקסימלי ב־2 סביבה שלו ונוסיף את כולם (ללא התחשבות במזהה)

למה זה טוב?
דוגמה: שרוך שבו לכל צומת יש מספר שכנים עלים.
מאחר שאנו מעגלים לחזקות של 2, יש לכל היותר $O(log\Delta)$ ערכים שונים של $\hat{\rho}(v)$, כך שנצטרך פחות איטרציות (יותר צמתים יוכלו להיות המקיסמליים ב־2־סביבה שלהם)

האלגוריתם הסופי:
(שימו ♡ שהוא רנדומי)

1. נאתחל $D = \varnothing$.
2. בכל איטרציה, כל הצמתים בעלי $\hat{\rho}(v)$ מקסימלי ב־2־סביבה שלהם הם **מועמדים** ל־D.
3. כל מועמד v יבחר מספר אקראי $r_v \in \{1, ..., n\}$.
4. כל צומת u שעדיין לא מכוסה **מצביע** על המועמד בעל r_v מינימלי שמכסה אותו (יכול להיות גם הוא עצמו אם הוא מועמד), אם קיים אחד כזה.
5. מועמד v שקיבל לפחות $\frac{\rho(v)}{8}$ קולות, מצטרף ל־ D.
6. ממשיכים עד שכל הצמתים מכוסים ע"י D.

8.1 נכונות

אינטואיציה:
נתנו לצמתים שאנחנו רוצים לכסות להחליט איזה מועמדים להוסיף ל־ D ומוסיפים צומת ל־ D רק אם קיבל $\frac{1}{8}$ מהקולות של כל הצמתים שהוא מכסה. כיוון שכל צומת מצביע רק למועמד אחד, לא נוסיף מועמדים שונים בשביל לכסות את אותם צמתים.

מקרה קליקה: כל הצמתים מועמדים, אך נוסיף רק את הצומת בעל r_v מינימלי וסיימנו (איטרציה אחת)

ובמקרה של קליקה עם שכנים עלים לכל צומת: נניח שישנם 2 שכנים עלים לכל צומת בקליקה. כל הצמתים של הקליקה מועמדים. באיטרציה 1 נוסיף צומת אחד מהקליקה (שהוא בעל ה־ r_v המינימלי כמובן) שיכסה את הקליקה. באיטרציה 2 נוסיף את כל הצמתים האחרים של הקליקה (מכיוון ש־ $\rho = 2$ עבורם עכשיו והם מקבלים שני קולות מהשכנים החיצוניים שלהם)

נעבור להראות שהאלגוריתם שלנו פותר את שתי הבעיות.

8.1.1 אנליזת קירוב

נראה שהאלגוריתם משיג קירוב $O(log\Delta)$.

רעיון: ניתן משקל $cost(u)$ לכל צומת ונראה ש־

$$|D| \leqslant 8 \cdot \sum_{u \in V} cost(u) \leqslant O(log\Delta) \cdot |D^*|$$

כאשר D^* הוא הפתרון האופטימלי.

הגדרת $cost(u)$: נסתכל על האיטרציה הראשונה i שבה u כוסה באלגוריתם. אם u כוסה ע"י צומת v ש־ u הצביע לו, נגדיר $cost(u) = \frac{1}{\rho(v)}$ ($\frac{1}{\rho(v)}$ של האיטרציה ה־ i)
אחרת, אם הוא כוסה ע"י צומת שהוא לא הצביע עליו, יתקיים $cost(u) = 0$.

אינטואיציה ללמה כל זה טוב: בעצם מחלקים את "המשקל" $\rho(v)$ של v ע"פ מספר הצמתים שהצביעו לו.

טענה 1: $|D| \leqslant 8 \cdot \sum_{u \in V} cost(u)$

הוכחה 1:
עבור צומת $v \in D$ נסמן ב־ $Votes(v)$ את קבוצת כל הצמתים שהצביעו לו באיטרציה בה הוספנו אותו ל־ D.
נשים לב שכל צומת u נמצא לכל היותר בקבוצה $Votes$ אחת, שהיא הקבוצה של הצומת שהוא הצביע לו וכיסה אותו לראשונה.
הוספנו את v ל־ D רק אם הוא קיבל לפחות $\frac{\rho(v)}{8}$ קולות ולכן, כך שזהו גם התנאי לכך ש־ $cost(u) \neq 0$ אם u הצביע ל־ v, ומכאן נקבל ש־

$$\sum_{u \in Votes(v)} cost(u) \geqslant \frac{\rho(v)}{8} \cdot \frac{1}{\rho(v)} = \frac{1}{8}$$

ומכאן נסיק כי;

$$|D| = \sum_{v \in D} 1 \leqslant \sum_{v \in D} 8 \cdot \sum_{u \in Votes(v)} cost(u) \leqslant 8 \cdot \sum_{u \in V} cost(u)$$

טענה 2: $\sum_{u \in V} cost(u) \leqslant O(log\Delta) \cdot |D^*|$

הוכחה 2:
יהי $v \in D^*$. נסתכל על **כל** הצמתים שהוא מכסה (הוא והשכנים שלו) ונסדר אותם לפי הסדר שכוסו באלגוריתם: $u_1, u_2, ..., u_l$
באיטרציה שבה u_1 כוסה, מתקיים $\rho(v) = l$ (מכיוון שכל l לא היו מכוסים בהתחלה)
v מכסה את u_1 ולכן הם שכנים. נניח ש־ u_1 כוסה על ידי שכן v', כך ש־ v, v' הם באותה 2־סביבה (שניהם שכנים של u_1) כאן אנחנו מגיעים ללמה אנחנו הסתכלנו על 2־סביבה. מאחר ש־ v' נכנס ל־ D קודם, ומאחר ש־ $\rho(v) = l$, ומאחר שאנחנו בוחרים מועמדים ל־ D לפי $\hat{\rho}$, מתקיים $\rho(v') \geqslant \frac{l}{2}$ (אחרת $\hat{\rho}(v')$ לא מקסימלי ב־2־סביבה שלו, מכיוון ש־ $\hat{\rho}(v)$ יהיה גדול ממש ממנו)

מהגדרת $cost(u_1)$ נקבל ש־ $cost(u_1) \leqslant \frac{1}{\rho(v')} \leqslant \frac{2}{l}$
באופן דומה, כאשר u_2 מכוסה, מתקיים $\rho(v) \geqslant l-1$ (מכיוון ש־ $u_2, ..., u_l$ לא כוסו עדיין) ומכאן נובע ש־ $cost(u_2) \leqslant \frac{2}{l-1}$
ניתן להמשיך באותו אופן להראות שלכל $1 \leqslant i \leqslant l$:

$$cost(u_i) \leqslant \frac{2}{l-i+1}$$

בסך הכל נקבל ש־

$$\sum_{i=1}^{l} cost(u_i) \leqslant 2 \cdot \sum_{i=1}^{l} \frac{1}{l-i+1} = 2 \cdot \sum_{j=1}^{l} \frac{1}{j}$$

זהו טור הרמוני, כך ש־

$$\sum_{i=1}^{l} cost(u_i) \leqslant O(logl) = O(log\Delta)$$

המעבר האחרון נובע מכך שמספר הצמתים ש־ v מכסה הוא לכל היותר $\Delta + 1$.
עבור $v \in D^*$, נסמן ב־ $Cov(v)$ את הצמתים ש־ v מכסה (שהם v ושכניו)
מכיוון ש־ D^* מכסה את כל הצמתים, נקבל, כנדרש;

$$\sum_{u \in V} cost(u) \leqslant \sum_{v \in D^*} \sum_{u \in Cov(v)} cost(u) \leqslant O(log\Delta) \cdot \sum_{v \in D^*} 1 = O(log\Delta) \cdot |D^*|$$

8.1.2 אנליזת זמן

ניתן לראות שאפשר לממש כל איטרציה ב־ $O(1)$ סיבובים (מתעסקים רק עם 2־סביבות לכל היותר) כך שהמטרה היא לחסום את מספר האיטרציות.

מכיוון שעיגלנו את הערכים $\rho(v)$, יש רק $O(log\Delta)$ ערכים אפשריים ל־ $\hat{\rho}(v)$. אפשר להראות שעבור כל ערך זה יש $O(log^2 n)$ איטרציות, וזוהי מטרתנו.
(אפשר גם להראות $O(logn)$ ע"י אנליזה יותר הדוקה)

לכל צומת לא מכוסה נסמן ב־ $S(v)$ את מספר המועמדים השכנים שלו (שכמובן יכולים לכסות אותו)

למען הבהרה, ניתן לחשוב על זאת כך;
נחשוב על גרף דו־צדדי, בו בצד שמאל אחד המועמדים ל־ D ובצד ימין כל הצמתים שעוד לא כוסו. הדרגה של צומת v מהצד השמאלי היא $\rho(v)$ ושל צומת u מהצד הימני דרגתו היא $S(u)$.

נגיע למטרה שלנו בכך שנוכיח שלאחר $O(logn)$ סיבובים, $max\ S$ קטן פי קבוע. למה זה טוב?־ כי אז $max\ S(v)$ יקטן ל־ 1 לאחר $O(log^2 n)$ איטרציות. זאת מכיוון ש־ $max\ S(V)$ קטן פי קבוע $1 < O(1)$, $O(log)$ פעמים, כאשר $log(max\ S(V)) = O(logn)$.
אזי, באיטרציה הבאה, כל המועמדים שנשארו (בעלי ערך $\hat{\rho}(v)$ מקסימלי) יצטרפו ל־ D.

<u>חסם לזמן האלגוריתם:</u> באלגוריתם לכל אחד מ־ $O(logn)$ האפשרויות של ערך $\hat{\rho}(v)$, נדרשות $O(log\Delta)$ הורדות של הדרגה מימין (במקרה הגרוע נדרשת הרצה לכל מקרה, אם כל $\hat{\rho}(v)$ הוא מקסימלי) והורדת הדרגה הנ"ל פי קבוע מצריכה $O(logn)$ איטרציות, כך שסך הכל $O(log^3 n)$ איטרציות נדרשות לסיום האלגוריתם.