הגדרנו שתתק"ב הי'א של- $G\subseteq H$ תהיה ε-kernel אם לכל $v\in R^{d-1}$, $(1-\varepsilon)\mathcal{E}_H(v)\le\mathcal{E}_G\le\mathcal{E}_H(v)$

והאמת, במקום על מישורים, אפשר לקחת פונקציות אלגבריות בערך אילו שבאו לנו. למשל פולינומים, ע"י לינאריזציה להתאים אליהם בגוף על מישורים.
אפשר יהיה לחשב מעטפת, ולמצוא שנק' נעות בזמן, וה- ε-kernel יוכל לייצג המון דברים שימושיים.

הרצאה

היום ח"ים מלמד על Locality Sensitive Hashing - LSH

עד 2 ממדים מוצאים NN Nearest Neighbor בקלות. זמן לינארי לבנות ו- log ל- query.
עם העליה בממדים זה נהיה אקספ' בממד.

ה- LSH זו השיטה היחידה כיום שאיננה אקספ' בממד, אבל גם ה- query קצת יותר יקר. כמובן query לינארי זה טריוויאלי, פשוט ב- brute force.

הרעיון הוא להשתמש ב- hashing, ונחלק את הנק' לתאים ובגדול נרצה לחפש את ה- NN בתאים.

יש 2 הגדרות ל- LSH מסביבות שנת 2000. ע"י Charikar
או ע"י Indyk & Motwani...

אם כן, <u>הגדרה:</u> (צ'אריקר)
משפחה H של פונקציות היא <u>locality Sensitive</u> כאשר פונק' similarity של $0\le sim(p,q)\le 1$ אם

$$Pr[h(p)=h(q)] = sim(p,q)$$

בכלל, מגדיר את המטריקה. אפשר לחשוב על מרחק אוקלידי למשל.

כלומר, ככל שהנק' יותר "קרובות", יותר "1", ההסתברות שיפלו באותו bucket יותר גדולה, ~~ו~~ ושווה ל- sim שלהם.

<u>דוגמה:</u> Hamming Similarity

אם לנושא הנק' שלנו הם strings באורך m ביטים ואנו שוקלים פונק' דמיון של

$$sim(p,q) = 1 - \frac{ham(p,q)}{m}$$

כאשר $ham(p,q)$ זה על כמה ביט'ים p ו-q מסכימים.
אז המשפחה H היא די מתבקשת

$$H = \{h_i(p) = \text{הביט ה-}i\text{ של } p\}$$

Jaccard?
עוד דוגמה Jaccard. בכמה מקומות p ו-q דומים חלקי כמה מקומות יש סה"כ.

$$\begin{array}{lll} 101110 & \rightarrow 1,3,4,5 & \\ 101010 & \rightarrow 1,3,5 & \rightarrow \frac{3}{4} \end{array}$$

ונניח כל bit שכזה יכול לייצג term כלשהו שמופיע ב-p או ב-q.
פונקציות ה-H כאן: מסתכלים על פרמוטציה אקראית ואם יש סדר על האיברים

$H = \{h_\pi(p) = \text{min in } \pi \text{ of the items in } p\}$ אפשר לראות מה ההסתב' שהם מסכימים או משהו... שלכל אחד מפרמוטציה יש אותה הסתב' להיות ראשון.

אבל זו משפחה עצומה. חיים אומר שלא באמת צריך את כל הפרמוטציות. רק צריך שאם נגריל 2 פרמוטציות לכל איבר יהיה הסתברות שווה להיות הראשון, כלומר הקטן ביותר. אז יש כל מיני מאמרים על בחירת הפרמוטציות...

<u>ועוד דוגמה:</u>
יש נק' ב- R^d והמשפחה היא כל העל מישורים (עובר בראשית). כל h של על מישור ממפה נק' ל-1 אם הנק' מעל.

הסיכוי "שנפריד" בין שתי נק' פרופורציונלי לזווית ביניהם, כלומר הסיכוי שהעל מישור בדיוק עובר ביניהם. אם כן, יפלו ל-2 buckets שונים.

$$Pr[h_r(p) = h_r(q)] = 1 - \frac{\theta}{\pi}$$

על מישור

מה הבעיה? 2 קליים (buckets) זה לא מספיק. אז לוקחים בפרקטיקה פשוט הרבה על מישורים אקראיים ואז פונק' ה-hash היא למעשה שרשור של כמה h מהמשפחה. ואז נקבל עבור כל שרשור 0100101 וכעת זה כאילו עשינו רדוקציה ל-hamming distance ולא חייבים אפילו להסתכל על האם מקבלים בדיוק אותה מחרוזת, אולי מעוניינים גם במחרוזת "קרובה".

<u>משפט:</u>

אם קיימת משפחה H עבור פונק' similarity, אז $\overset{=1-sim}{d(p,q)}$ מקיים את אי שוויון המשולש.

<u>הוכחה:</u>

$$X(p,q)=\begin{cases}1 & \text{bucket אם לא באותו} \\ 0 & \text{אחרת}\end{cases}$$

$$X(p,q)+X(q,r)\geq X(p,r)$$

וכעת יש כמה מקרים של אם q ו-p באותו תא ו- r,q לא אז r,p לא וכן הלאה. זה בוודאי מתקיים בכולם ואז זה נכון גם בתוחלת..

$$\Rightarrow 1-sim(p,q)+1-sim(q,r)\geq 1-sim(p,r)$$

$$d(p,q)+d(q,r)\geq d(p,r)$$

ובאמת מתקיים אי שוויון המשולש ובאמת יש קשר מסוים למטריקה.

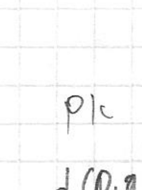
וכעת <u>ההגדרה האמיתית ל- LSH:</u>

משפחה H של פונק' היא $(r_1<r_2,p_1>p_2)$-sensitive אם

$$d(p,q)\leq r_1 \rightarrow Pr[h(p)=h(q)]\geq p_1$$

$$d(p,q)\geq r_2 \rightarrow Pr[h(p)=h(q)]\leq p_2$$

כלומר, נק' קרובות הולכות בהסתב' גבוהה לאותו תא. אם רחוקות, אז בהסתב' נמוכה הולכות לאותו תא.



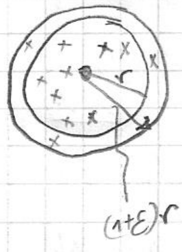
אפשר לראות את הקשר להגדרה הקודמת, בין המרחקים להסתברויות.

למשל, אם המשפחה היא ham, אז ההגדרה השנייה מתקיימת עם $p_1=1-\frac{r_1}{m}$, $p_2=1-\frac{r_2}{m}$.

אז לא יודעים לגמרי איך לפתור את בע"ת ה-NN, אבל נפתור משהו קרוב. ~~אז~~ עושים רדוקציה מבע"ת ה-ANN (Aproximate) לבעיה הבאה: (r,ε)-neighbor problem:

אם קיים שכן p כך ש- $d(p,q)\le r$ אז נרצה להחזיר p' כך ש- $d(p',q)\le(1+\varepsilon)r$

וזו הבעיה שנרצה לפתור.



נבנה מבנה נתונים הסתברותי (בגלל פונק' ה-hash האקראית), כלומר שמצליח בהסתב' קבועה, ואז נוכל להגדיל אותה ע"י הפעלתו כמה פעמים.

בשביל לפתור את בע"ת ה-(r,ε)-Neighbor נצרך H שהיא LS עם הפרמטרים: $(r,(1+\varepsilon)r,p_1,p_2)$ והביצועים שלנו יהיו תלויים ב-Gap בין p_1 ו- p_2.

ב- Hamming dist. - יודעים מהם; $p_1=1-\frac{r}{m}, p_2=1-\frac{(1+\varepsilon)r}{m}$.

אנחנו רוצים הסתב' הצלחה קבועה, איכות, שלא תהיה תלויה ב-r וב-m, וזמן הריצה יהיה תלוי ב-p_1 ו-p_2.

כדי לתפוס נק' צריך לחזור על הניסוי $1/p_1$ פעמים... כי הסיכוי להשלון הוא $1-p$ ואם נחזור על זה הרבה פעמים נקבל $1/e$ או משהו...

(שנתפוס נקודה שגויה)

אם יש המון נק' על שפת הכדור ברדיוס $(1+\varepsilon)r$ אז יש סכנה של $n\cdot p_2\cdot\frac{1}{p_1}$ אם נחזור על הניסוי $1/p_1$ פעמים (כלומר נקח הרבה פונקציות)... וזה בעיה כי זה לינארי...

מה עושים? כל פונק' hash הולכת להיות מכפלה קרטזית של k פונק' hash ועכשיו כל bucket יורכב מ-k ערכים שמייצגים אותו.

בעצם עשינו boosting להסתברויות וקיבלנו משפחה $(r_1<r_2,(p_1)^k>(p_2)^k)$.

אז כמה כדי לתפוס נק' צריך $1/p_1^k$ פונקציות, וגם נקבל $n(\frac{p_2}{p_1})^k$ נק' שהן false positives.

אז מה האלג'?

מגיע query, נסמן ב-q. ממפים את q ע"י כל אחת מה- $\frac{1}{p_1^k}$ פונקציות, אוספים את הנק' עד שיש $4n(\frac{p_2}{p_1})^k$, ומחזירים את הכי קרובה.
בהסתברות $1/4$ נחזיר נק' שהיא false positive, כי יש לנו את החסם על גודל כמות ה-false positives.
וגם, מקבלים שהן קרוב בהסתב' $\geq 1-1/e$.

נשאר רק לבחור את k, ומה שמעניין אותנו זה ה-query time וה-space, כלומר הזכרון.
זמן הריצה מחושב לפי k ביחד עם k פונק'? עושים את האופטימיזציה ורואים מהו ה-k האופטימלי (בסקלה)

$$\Rightarrow \quad k=\log_{1/p_2}(n)-\Theta(\log\log n)$$

נציב ונראה שהזמן של ה-query יוצא: $n^\rho = n^{\frac{\log(1/p_1)}{\log(1/p_2)}}$

משתמשים עוד איזו פונק' hash רגילה שממפה באקטים מלאים לבאק' כי יש מלא באקטים ריקים ולא רוצים המון באקטים כאלו.
כך, מקבלים $O(n\cdot n^\rho)$ מקום. גם הטבלה הנוספת היא בגודל פרופורציונלי ל-n, (מוגבל לכמות ה-buckets המלאים).

לבסוף, עם ה-p_1,p_2 של ה-hamming מקבלים

query $n^{\frac{1}{1+\varepsilon}}$
space $n^{1+\frac{1}{1+\varepsilon}}$

ל-ε קטן זה די נוטו...

ויש מעבר ל-ANN מ-(ε, r)...

כמו איים מכאה שניסו לעשות דברים גם לא רק עם hamming. עבדו עם L_2 ויש כמה שקפים על זה, וגם'ך של Andoni ו-Indyk פירסמו את זה ב-2005, וגם חיים ומיכה השתמשו בזה...
וכתבו לזה קוד ← ELLSH

הראו לא כל כך ונליטית שה-p שהתקבל היה בין $\frac{1}{1+\varepsilon}$ לבין $\frac{1}{(1+\varepsilon)^2}$ והיתה שאלה פתוחה האם אפשר להשיג $(1+\varepsilon)^2$.

התשובה התגלתה כבר, ע"י גריד (או כמה גרידים) של כדורים שמכסים את המרחב. וכל פונק' היא אוסף (ספרה) של גרידים אקראיים שמוזזים עד שהם מכסים את כל המרחב. וכל פונק' hash נותנת. ולכל של הנק' את המינימום של הכדור, בגריד הראשון, שבו היא מופיעה ו' אפשר לעשות Location point ב-d ממדים את הכדורים והגרידים הם רנדומליים וצריך להבין באיזה כדור הנק'. איפשהו 33% פונק' hash בסיבוכיות nlogn בגלל נפח הכדורים, ולכן עושים הטלות ממד כך שמקבלים $t^t \log n$ אבל פוגעים ב-p_1 ו-p_2.

תודה לגיים!

עכשיו מי כהן רוצה לומר עוד כמה מילים על Coresets כהמשך לשבוע שעבר.

כזכור, בגרסה הפרימאלית היו לנו n נק' ב-$\mathbb{R}^d$, ובנינו ε-kernel $Q \subseteq D$, בגודל $\sim \varepsilon^{-\frac{d-1}{2}}$

בגרסה הדואלית - H: קב' של n על מישורים $h_1,\dots,h_n$ פונקציות לינאריות $x_d = h_i(x_1,\dots,x_{d-1})$
$(x_1,\dots,x_{d-1})$ הוא כיוון "קולי" ומסתכלים על ההפרש בין המספר העליונה והתחתונה ולזה קוראים ה extent:

$$\max_i h_i(x_1,\dots,x_{d-1}) - \min_i h_i(\dots)$$

המטרה שלנו היא לקבל תת קבוצה $G \subseteq H$ כך ש-

$$(1-\varepsilon)\,\text{extent}_H(\vec{x}) \le \text{extent}_G(\vec{x}) \le \text{extent}_H(\vec{x}) \qquad \vec{x} \in \mathbb{R}^d$$

נקבל (תכף) שזה בוודאות הדואלי לעובי.

במקרה (הפרימאלי), עובי בכיוון u: $\max_{p\in P} p\cdot u - \min_{p\in P} p\cdot u$

$$\downarrow$$

$$\sum_{i=1} p_i u_i$$

אז הנק' $p=(p_1,...,p_d)$ תעבור להיות העל מישור p^*:

$$x_d = \sum_{i=1}^{d-1} p_i x_i + p_d$$

שזה בדיוק המכפלה $p_i x_i$ במפורש, עד כדי נורמליזציה. למה? כי $p\cdot(1,x_{d-1},...,x_1)$ זה לא בדיוק כיוון על וקטור היחידה אבל אפשר לנרמל את זה יהיה, ולכן ההפרש יהיה אותו דבר כמו קודם עד כדי פקטור כפלי ואז ה-ו' עבורו נשיג מינ' ומקס' יהיה אותו דבר.

לכן, מה שעשינו בפרימאלי אפשר לעשות גם כאן וההפך.

עכשיו אפשר לנו כל מיני פטנטים נחמדים. למשל, הענין יעבוד גם לפונק' לא ליניארית. למשל פונק' ריבועית:

$$h_i(x,y) = a_i x^2 + b_i xy + c_i y^2 + \dots$$

נעשה פנ"ם שהפונק' הן ליניארית ע"י ליניאריזציה. זה מוסיף ממדים באופן הבא:

$$\Leftarrow \quad x_4 = a_i \underset{x^2}{\underset{\downarrow}{x_1}} + b_i \underset{xy}{\underset{\downarrow}{x_2}} + c_i \underset{y^2}{\underset{\downarrow}{x_3}} + \dots$$

וכך אפשר לקבל קירוב ל- extent עד כדי $(1-\varepsilon)$ גם לפונק' הלא ליניארית.

אפשר לעשות דברים גם הרבה יותר מורכבים.

(מ-d ממדים)

למשל, נניח שיש לנו נק' שנעות במרחב כפונק' של הזמן, ונניח שאנו נמצאים במקום מסוים ורוצים לחשב בכל רגע את הנק' הכי קרובה והכי רחוקה. "נכון שנרצה לבנות Coreset, מין נק' "נבחרות" שנעקוב אחריהן והן יהיו המתאימות בכל רגע בלי תלות בזמן! נכתוב את הנק' עצמן כפונק' של הזמן, וע"י ליניאריזציה נוסיף ממדים ונופל עבור G...

$$p_1 x_1 + p_2 x_2 + \dots$$

$$(a_1 + b_1 t)x_1 + (a_2 + b_2 t)x_2 + \dots$$

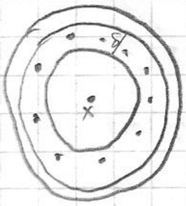
וכמו הקודמ' גבלה פי 2.

ולקינוח, עוד דוגמאות:

$P = n$ נק' ב-d ממדים.

רוצים לחשב קליפה כדורית מינימלית שמכילה את P.
עובי מינימלי

כלומר, מנסים לקרב את P ע"י כדור.



בהנחה שאכן הנקודות הם בערך מייצגות כדור

בעצם רוצים למצוא את מרכז הכדור, ובמילים

אחרות להביא למינימום את:

$$\max_{p\in P}\|x-p\| - \min_{p\in P}\|x-p\|$$

אה-מה-מה, יש כאן שורש ריבועי וזה מבאס. אחרת יכולנו לעשות פה גרסה לינארית.

מה עושים? מעלים בריבוע:

$$f_p = \|x-p\|^2 = \|x\|^2 - 2x\cdot p + \|p\|^2$$

בשביל לינאריזציה, מעתק $\|x\|^2$ נקבל x_{d+1} (עוד קואו' בשביל הערך)

וכעת אנו מגיעים ל-$\mathbb{R}^{d+2}$ ונשים לב שכעת f_p היא לינארית, ומחפשים את

$$\max_{p}\sqrt{f_p(\vec{x})} - \min_{p}\sqrt{f_p(\vec{x})}$$

והרי רוצים להיפטר מהשורשים. יש לנו H שזה על מישורים $(\vec{x})$ $x_{d+2}=f_p(\vec{x})$.

נבנה G שהוא ε-kernel (או Coreset) עבור H.

קורל' ל- $Q\subseteq P$.

ב-$\vec{x}$ ספציפי:

$$\Rightarrow \quad A = \max_{p\in P} f_p$$

$$B = \min_{p\in P} f_p$$

$$C = \max_{p\in Q} f_p$$

$$D = \min_{p\in Q} f_p$$

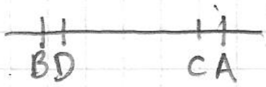
ובגלל שזה Coreset קיבלנו

$$(1-\varepsilon)(A-B) \le C-D \le A-B$$

אבל מה קורה עם השורשים? ובכן $\sqrt{C}-\sqrt{D} \le \sqrt{A}-\sqrt{B}$ גם כן נכון. לצד השני:

ניזכר ש-

$$\sqrt{C}-\sqrt{D} = \frac{C-D}{\sqrt{C}+\sqrt{D}}, \quad \sqrt{A}-\sqrt{B} = \frac{A-B}{\sqrt{A}+\sqrt{B}} \quad \Rightarrow \quad \frac{\sqrt{C}-\sqrt{D}}{\sqrt{A}-\sqrt{B}} = \frac{C-D}{A-B}$$

אנחנו רוצים לטעון ש- $\frac{\sqrt{C}+\sqrt{D}}{\sqrt{A}+\sqrt{B}} \approx 1$ ובציור 

אז זה לא יהיה אותו ε אבל נקבל שבצד השני אכן

יש $(1-c\varepsilon)(\sqrt{A}-\sqrt{B}) \le \sqrt{C}-\sqrt{D}$

נו, נמאס.