محمد امین رامی

تمرین سوم درس تحلیل داده های حجیم

پاییز ۱۴۰۱

سوال اول

الف) زمانی نتیجه minhash برابر don’t know میشود که تمام ستون های انتخابی برابر صفر باشند.

این احتمال برابر است با:

$$P[all\ zero] = \frac{n-k}{n} \times \frac{n-k-1}{n-1} \times \dots \times \frac{n-k-m+1}{n-m+1} \ \leq\ (\frac{n-k}{n})^m$$

پس:

$$P[all\ zeros] \leq\ (\frac{n-k}{n})^m$$

ب) داریم:

$$p[all\ zeros]\ \leq \left(1-\frac{k}{n}\right)^m = \left(1-\frac{km}{n}\frac{1}{m}\right)^m = e^{-\frac{km}{n}}\ \leq\ e^{-10}$$

$$\Rightarrow \frac{km}{n} \geq 10 \quad => \quad k\ \geq\ \frac{10n}{m}$$

ج)

برای اینکه نشان دهیم دهیم جایگشت های تناوبی برای به دست آوردن Jaccard similarity مناسب نیست، مثال زیر را مطرح میکنیم:

$$S_1 = \begin{Bmatrix} 0 \\ 1 \\ 0 \end{Bmatrix}\ , \qquad S_2 = \begin{Bmatrix} 0 \\ 1 \\ 1 \end{Bmatrix}$$

حال داریم:

$$Jaccard\ similarity = \frac{1}{1+1} = 0.5$$

حال سه حالت زیر برای جایگشت های تناوبی را در نظر بگیرید:

۱ – حالتی که شماره ردیف ها به ترتیب برار 1, 2, 3 باشند:

$minhash(S_1) = 2 \qquad minhash(S_2) = 2$

۲ – حالتی که شماره ردیف ها به ترتیب برار 3, 1, 2 باشند:

$minhash(S_1) = 1 \qquad minhash(S_2) = 1$

۳ – حالتی که شماره ردیف ها به ترتیب برار2, 3, 1 باشند:

$minhash(S_1) = 3 \qquad minhash(S_2) = 1$

پس تخمینی که از Jaccard Similarity به دست می آوریم برابر است:

$$\frac{2}{3}$$

تفاوت $\frac{2}{3}$ و $\frac{1}{2}$ زیاد است. پس تکیه بر جایگشت های تناوبی راه مناسبی نیست.

سوال دوم

الف)

برای هر $1 \le j \le L$ و هر نقطه $x \epsilon T$ داریم:

$$P\left[x \epsilon T \cap W_j\right] \le p_2^k = \frac{1}{n} \quad \Rightarrow \qquad E\left[\left|T \cap W_j\right|\right] \le 1$$

باتوجه به خطی بودن expected value داریم:

$$E\left[\sum_{j=1}^{L}\left|T \cap W_j\right|\right] \le L$$

با اعمال نامساوی مارکف داریم:

$$P\left[\sum_{j=1}^{L}\left|T\cap W_j\right|\geq 3L\right]\leq \frac{E\left[\sum_{j=1}^{L}\left|T\cap W_j\right|\right]}{3L}\leq \frac{L}{3L}=\frac{1}{3}$$

پس داریم:

$$P\left[\sum_{j=1}^{L}\left|T\cap W_j\right|\geq 3L\right]\leq \frac{1}{3}$$

ب)

از آنجایی که $d(x^*,\ z)\leq \lambda$ است، برای هر $1\leq j\ \leq L$ داریم: $P\left[g_j(x^*)=g_j(z)\right]\geq p_1^k$

همچنین داریم: $p_1^k = p_1^{\log_{\frac{1}{p_2}}(n)} = n^{-\frac{\log(\frac{1}{p_1})}{\log(\frac{1}{p_2})}} = n^{-\rho} = \frac{1}{L}$

پس بدیهتا داریم: $p\left[g_j(x^*)\neq g_j(z)\right]\leq 1-p_1^k=1-\frac{1}{L}$

بنابر استقلال g_j ها داریم:

$$P\left[\forall: 1\leq j\ \leq L,\ \ g_j(x^*)\neq g_j(z)\right]\leq \left(1-\frac{1}{L}\right)^L=\frac{1}{e}$$

ج)

فرض کنید که U مجموعه نقاط ANN باشد. یعنی : $U=\{x\in A, d(x,z)\leq c\lambda\}$. آنگاه $x^*\in U$ میباشد.

در دوحالت یک نقطه گزارش شده، یک $(c,\lambda)-ANN$ نمیباشد:

- هیچکدام از نقاط ANN به باکت یکسانی با z هش نشده اند. فرض کنید E نشان دهنده این پیشامد باشد. از آنجایی که $x^*\in U$ است، باتوجه به نامساوی قسمت ب داریم:

$$P[E] = P\left[x^* \notin \bigcup_{j=1}^{L} W_j\right] = P\left[\forall: 1 \le j \ \le L, \ \ g_j(x^*) \neq g_j(z)\right] < \frac{1}{e}$$

- حداقل یک نقطه $(c,\lambda) - ANN$ وجود دارد که به یکی باکت هایی که z هش شده است، هش شده ولی بیشتر از 3L نقطه وجود دارد که در فاصله ای بزرگتر از $c\lambda$ در اجتماع آن باکت هایی که z هش شده، قرار دارد. فرض کنید این پیشامد را F بنامیم. آنگاه با اعمال بخش نامساوی به دست آمده در بخش الف به این نتیجه میرسیم که احتمال وقوع F از $\frac{1}{3}$ کمتر است.

پس اگر p_0 را احتمال اینکه "نقطه به دست آمده توسط الگوریتم، یک $(c,\lambda) - ANN$ نباشد" بنامیم، با استفاده از باند اجتماع (union bound) داریم:

$$p_0 = P[E \cup F] \le P[E] + P[F] < \frac{1}{3} + \frac{1}{e}$$

پس احتمال اینکه الگوریتم یک نقطه $(c,\lambda) - ANN$ واقعی گزارش کند بزرگتر از $1 - \frac{1}{3} - \frac{1}{e}$ است:

$$P[algorithm\ reports\ an\ actual\ (c,\lambda) - ANN\ \ point] > 1 - \frac{1}{3} - \frac{1}{e}$$

سوال سوم

الف)

داده های داده شده را به دوصورت ارزیابی میکنیم. ابتدا سطر هایی که در آنها ORIGINE_CAR_KEY و FINAL_CAR_KEY برابر نیستند را حذف میکنیم زیرا داده پرت هستند.

همچنین در برخی موارد این مشکل رخ داده است که وقتی یک ماشین از یک دوربین رد شده است، پلاک آن چندین بار پشت سر هم ثبت شده است. پس در دیتای ثبت شده، مواردی که در کمتر از یک دقیقه چندین تا پلاک یکسان ثبت شده است را حذف میکنیم و فقط یکبار آن پلاک را تاثیر میدهیم.

همچنین برای هر سطر جفت (key, value) را به صورت زیر میسازیم:

Key = (plate, date)   Value = camera code

توجه کنید در date فقط به روز توجه میشود.

سپس یک groupByKey میزنیم تا دیتا به صورت زیر شود:

Key = (plate, date)   Value = [list of camera codes]

ب)

ابتدا یک path دلخواه به صورت رندوم و به طول 3 تولید میکنیم. سپس شبیه ترین path های دیتاست را با استفاده از معیار cosine similarity می یابیم.

نتیجه به صورت زیر میشود:

```
===== Resluts =====
The query:
['1001079', '22009923', '22010119']
Most similar paths:
1- (('95985673', '2021-06-01'), ['22009923', '22010119'], 0.6847192030022828)
2- (('23214739', '2021-06-01'), ['22009923', '22010119'], 0.6847192030022828)
3- (('8075171', '2021-06-01'), ['22009923', '22010119', '100700965'], 0.5880026035475676)
4- (('17974480', '2021-06-01'), ['900236', '22010119', '22009923'], 0.5880026035475676)
5- (('17122136', '2021-06-01'), ['100700824', '22010119', '22009923', '100700824'], 0.5880026035475676)
```

ج)

اینبار مسئله را با استفاده از LSH حل میکنیم. مشاهده میکنیم که نتیجه با قسمت قبل یکی میشود که این موضوع، مورد انتظار ماست.

```
===== Resluts =====
The query:
['1001079', '22009923', '22010119']
Most similar paths:
1- (('95985673', '2021-06-01'), ['22009923', '22010119'], 0.6847192030022828)
2- (('23214739', '2021-06-01'), ['22009923', '22010119'], 0.6847192030022828)
3- (('8075171', '2021-06-01'), ['22009923', '22010119', '100700965'], 0.5880026035475676)
4- (('17974480', '2021-06-01'), ['900236', '22010119', '22009923'], 0.5880026035475676)
5- (('17122136', '2021-06-01'), ['100700824', '22010119', '22009923', '100700824'], 0.5880026035475676)
```