

Amirkabir University of Technology

تكليف اول - تحليل كلان داده ها

استادمربوطه: دکتر حقیر چهرقانی

نام: زهرا اخلاقی

شماره دانشجویی: ۴۰۱۱۳۱۰۶۴

فهرست:

3	بخش اول MapReduce
	سوال اول
	سوال دوم
	بخش دوم Rule Association
10	سوال دوم
16	بخش سوم Hashing Sensitive Locality
	سوال اول
17	سوال دوم
17	and How

بخش اول MapReduce

سوال اول

الف)

ضرب ماتریس ها استفاده از رویکرد MapReduce در دو مرحله به صورت زیر میباشد: مرحله اول:

- فاز Map: برای ماتریس M1 با سطر i و ستون j جفت کلید و مقدار به صورت (j,(M1,i,M1i)) میباشد و برای ماتریس M2 با سطر i و ستون i جفت کلید و مقدار به صورت (i,(M2,i,M2i) میباشد.
- فاز Reduce: در ورودی این مرحله به ازای هر کلید i، دو مقدار وجود دارد، یکی برای Reduce: (M2,k,M2jk) و برای ماتریس دوم که به صورت (M1,i,M1ij) و برای ماتریس دوم که به صورت (M1ij*M2jk) میباشد. که در نتیجه آنها در خروجی به کلید (i, k) و مقدار برابر M1ij*M2jk تبدیل میشوند.

مرحله دوم (این مرحله بر روی خروجی حاصل از مرحله قبل اعمال میشود):

- فاز Map: برای هر عنصر ورودی با کلید (i، k) و مقدار ۷، این جفت کلید-مقدار را تولید کنید.
- فاز Reduce: در ورودی این فاز به ازای کلید (i, k) چند مقدار وجود دارد، خروجی حاصل جمع مقادیر برای کلید یکسان میباشد. که نشان دهنده مقدار سطر i , ستون k در ماتریس نتیجه میباشد

$$M_1:\begin{bmatrix}2&3\\1&2\end{bmatrix}$$
 $M_2:\begin{bmatrix}1&4\\2&3\end{bmatrix}$

$$(1,(M_2,1,1))$$
, $(1,(M_2,2,4))$, $(2,(M_2,1,2))$, $(2,(M_2,2,3))$

CS Scanned with CamScanner

ج)

$$((1,2),9)$$
 $((2,2),6)$

. No la portegn reducer in cold to- No Tion 8

سوال دوم

برای پیاده سازی این مسئله با استفاده از mapReduce از دو مرحله استفاده شده:

در مرحله اول از متد map برای نگاشت هر خط به لیستی از شناسه های کانال، سپس از متد flatMap برای نگاشت هر شناسه کانال به یک جفت کلید-مقدار استفاده شده، که در آن کلید یک جفت کانال است (این کار به صورتی انجام میشود که در هر سطر کانال اول با همه کانال هایی که آن reduce را تبلیغ کرده اند به صورت جفت کلید ایجاد میشود) و مقدار آن 1 است. در قسمت reduce از reduce برای جمع بندی تعداد هر جفت کانال استفاده شده است (برای اینکه هر تبادل دو بار محاسبه شده حاصل جمع بر دو تقسیم میشود).

در مرحله دوم، از متد flatMap برای نگاشت هر کانال در هر جفت به یک جفت کلید-مقدار استفاده شده که در آن کلید کانال و مقدار تعداد آن کانال است. سپس در قسمت reduce از reducByKey برای جمعبندی تعداد هر کانال استفاده می کنیم.

الف)

```
Channel 859 has been advertised 1933.0 times.
Channel 5306 has been advertised 1741.0 times.
Channel 2664 has been advertised 1528.0 times.
Channel 5716 has been advertised 1426.0 times.
Channel 6306 has been advertised 1394.0 times.
```

ب)

Filter for the key-value pair where the key is 1748

```
[8] filtered_counts = total_counts.filter(lambda pair: "1748" in pair[0])

# Sum up the counts for the filtered pair
count_1748 = filtered_counts.map(lambda pair: pair[1]).sum()
count_1748
```

130.0

Filter for the key-value pair where the key is 5633

```
[9] filtered_counts = total_counts.filter(lambda pair: "5633" in pair[0])

# Sum up the counts for the filtered pair
count_5633 = filtered_counts.map(lambda pair: pair[1]).sum()
count_5633
30.0
```

Filter for the key-value pair where the key is 3469

```
filtered_counts = total_counts.filter(lambda pair: "3469" in pair[0])

# Sum up the counts for the filtered pair
count_3469 = filtered_counts.map(lambda pair: pair[1]).sum()
count_3469

119.0
```

بخش دوم Rule Association

سوال اول

برای یافتن آیتم های پرتکرار در ابتدا باید support هر آیتم را محاسبه کنیم. support برای یک آیتم درصد احتمال حضور آن آیتم درون سبد ها میباشد، که به صورت زیر میباشد:

support(i) = probability that item i is in a basket = 1/i

با توجه به اینکه حد آستانه یک درصد در نظر گرفته شده، مجموعه ای شامل یک آیتم پرتکرار است و مجموعه هایی با تعداد بالاتر را باید بررسی کنیم.

برای مجموعه دوتایی داریم:

Pair
$$(1, 2)$$
: support $(1, 2)$ = support (2) = 0.5

Pair (1, 3): support(1, 3) = support(3)
$$\approx$$
 0.33

Pair
$$(1, 4)$$
: support $(1, 4)$ = support (4) = 0.25

Pair
$$(1, 5)$$
: support $(1, 5)$ = support (5) = 0.2

Pair (1, 6): support(1, 6) = support(6)
$$\approx 0.17$$

Pair (1, 7): support(1, 7) = support(7)
$$\approx$$
 0.14

Pair (1, 8): support(1, 8) = support(8)
$$\approx$$
 0.125

Pair (1, 9): support(1, 9) = support(9)
$$\approx$$
 0.11

Pair
$$(1, 10)$$
: support $(1, 10)$ = support (10) = 0.1

Pair (2, 3): support(2, 3)
$$\approx$$
 0.166

Pair (2, 4): support(2, 4)
$$\approx$$
 0.125

Pair
$$(2, 5)$$
: support $(2, 5) = 0.1$

Pair (2, 6): support(2, 6)
$$\approx$$
 0.083

Pair (2, 7): support(2, 7)
$$\approx$$
 0.071

Pair (2, 8): support(2, 8)
$$\approx$$
 0.063

Pair (2, 9): support(2, 9)
$$\approx$$
 0.056

Pair (2, 10): support(2, 10) = 0.05

Pair (3, 4): support (3, 4) \approx 0.083

Pair (3, 5): support(3, 5) \approx 0.066

Pair (3, 6): support (3, 6) \approx 0.055

Pair (3, 7): support(3, 7) \approx 0.048

Pair (3, 8): support (3, 8) \approx 0.041

Pair (3, 9): support(3, 9) \approx 0.037

Pair (3, 10): support(3, 10) \approx 0.033

Pair (4, 5): support(4, 5) = 0.05

Pair (4, 6): support (4, 6) \approx 0.041

Pair (4, 7): support(4, 7) \approx 0.035

Pair (4, 8): support (4, 8) \approx 0.031

Pair (4, 9): support(4, 9) \approx 0.027

Pair (4, 10): support(4, 10) \approx 0.025

.

Pair(10,9): support(10, 9) \approx 0.01

با توجه به اینکه حد آستانه یک درصد در نظر گرفته شده، مجموعه ای شامل دو آیتم پرتکرار هستند، زیرا کمترین مقدار support برای جفت (10,9) میباشد که بزرگتر مساوی ۱ درصد است. استفاده از این حد آستانه برای topport نمیتوند برای سبد های شامل ۱ الی ۲ آیتم برای ما فیلتری ایجاد کند و باید حد آستانه بزرگتر در نظر گرفته میشد.

برای مجموعه سه تایی با حد آستانه کوچکتر از ۱ درصد، داریم:

(1,2,3)-(1,2,4)-(1,2,5)-(1,2,6)-(1,2,7)-(1,2,8)-(1,2,9)-(1,2,10)-(1,3,4)-(1,3,5) (1,3,6)-(1,3,7)-(1,3,8)-(1,3,9)-(1,3,10)-(1,4,5)-(1,4,6)-(1,4,7)-(1,4,8)-(1,4,9) (1,4,10)-(1,5,6)-(1,5,7)-(1,5,8)-(1,5,9)-(1,5,10)-(1,6,7)-(1,6,8)-(1,6,9)-(1,6,10) (1,7,8)-(1,7,9)-(1,7,10)-(1,8,9)-(1,8,10)-(1,9,10)-(2,3,4)-(2,3,5)-(2,3,6)-(2,3,7) (2,3,8)-(2,3,9)-(2,3,10)-(2,4,5)-(2,4,6)-(2,4,7)-(2,4,8)-(2,4,9)-(2,4,10)-(2,5,6) (2,5,7)-(2,5,8)-(2,5,9)-(2,5,10)-(2,6,7)-(2,6,8)-(3,4,5)-(3,4,6)-(3,4,7)-(3,4,8) (3,5,6)

در مجموعه بالا تمام آیتم های سه تایی شامل عنصر ۱ پرتکرار هستند. برای مجموعه چهار تایی با حد آستانه کوچکتر از ۱ درصد، داریم:

(1,2,4,5)-(1,2,3,10)-(1,2,3,9)-(1,2,3,8)-(1,2,3,7)-(1,2,3,6)-(1,2,3,5)-(1,2,3,4) (1,2,5,8)-(1,2,5,7)-(1,2,5,6)-(1,2,4,10)-(1,2,4,9)-(1,2,4,8)-(1,2,4,7)-(1,2,4,6) (1,3,4,8)-(1,3,4,7)-(1,3,4,6)-(1,3,4,5)-(1,2,6,8)-(1,2,6,7)-(1,2,5,10)-(1,2,5,9) (1,3,5,6)

همه سبد ها شامل آیتم ۱ می باشد.

مجموعه پنج تایی و با اندازه بزرگتر با حد آستانه کوچکتر از ۱ درصد، وجود ندارد.

میتوان دریافت که برخی از مجموعه آیتمها بیشتر از بقیه ظاهر میشوند، جزو آیتمهای پرتکرار هستند. پرتکرارترین کالا، آیتم ۱ است که در تعداد زیادی از سبد های سه تایی و همه سبد های چهارتایی وجود دارد.بعد از آیتم ۱، آیتم ۲ بیشترین تعداد تکرار را در سبد ها دارد، به طور کلی، این مجموعه اقلام مکرر بینش هایی را در مورد رایج ترین ترکیبات اقلام در سبد ها ارائه می دهد که می تواند برای کاربردهای مختلف مانند تجزیه و تحلیل سبد بازار و سیستم های توصیه مفید باشد. حد آستانه در نظر گرفته شده بهتر از در مراحل مختلف متفاوت باشد زیرا برای سبد های شامل ۱ و ۲ آیتم فیلتری روی داده ها ایجاد نکرد و در سبد های ۵ تایی به بالاتر با این میزان حد آستانه آیتم های پرتکراری وجود نداشت.

سوال دوم

نتیجه گسسته سازی و پیش پردازش داده ها:

برای انجام این کار ستون Time حذف شده و هر ستون با استفاده از quantile به ۲۰ قسمت تقسیم شده که هر قسمت در ستون مجزا قرار دارد و مقدار آن می تواند ۰ و یا ۱ باشد.

جدول حاصل شامل ۵۸۱ ستون میباشد.

(data.h	ead()																				
C→ /	warni usr/lo	ings. ocal/	warn(-						utureWarni utureWarni											
	Cla	ss A	mount_1	Amount_2	Amount_3	Amount_4	Amount_5	Amount_6	Amount_7	Amount_8	Amount_9	 V28_11	V28_12	V28_13	V28_14	V28_15	V28_16	V28_17	V28_18	V28_19	V28_20	7.
	0	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	 0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	1	0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	 1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	2	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	 0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
	3	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	 0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	
	4	0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	 0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	
	rowe :	v 591	columns																			

قوانین دو تایی به دلیل استفاده از quantile و اینکه توزیع داده ها میان ستون ها یکسان میباشد استخراج نشده، که مشکلی در حل مسئله ایجاد نمیکند.

جدول df برای قوانین سه تایی کاربرد دارد:

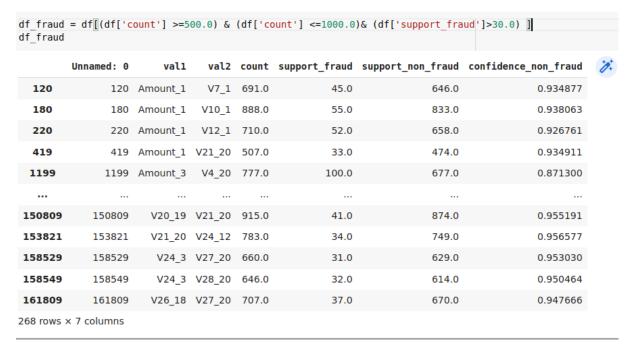
- مقدار count نشان دهنده تعداد دفعاتی است که ستون val1, val2 با یکدیگر ۱ هستند
- مقدار support_fraud تعداد دفعات است که ۱ بودن ستون val1,val2 باعث ایجاد داده fraud شده است
 - مقدار support_non_fraud تعداد دفعات است که ۱ بودن ستون val1,val2 باعث الحاد داده non fraud شده است
 - مقدار confidence_non_fraud حاصل تقسیم support_non_fraud بر support_non_fraud میباشد.



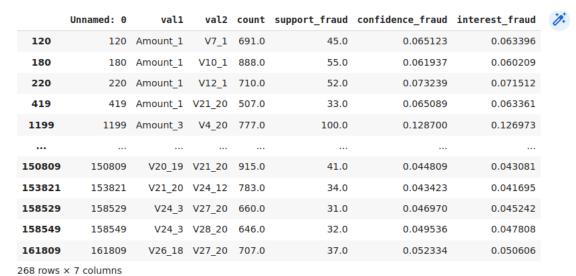
میزان support با مشاهده مقدار describe از داده ها در هر مرحله و با تغییر دادن مقدار آنها به صورت تجربی است.

Fraud

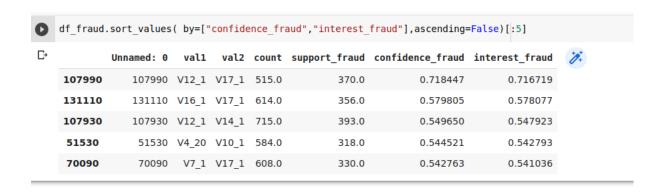
فیلتر به کار گرفته شده برای استخراج داده fraud به صورت زیر میباشد: برای اینکه تعداد داده های fraud نسبت به کل داده ها کم است باید برای جلوگیری از کوچکتر شدن confidence مقدار count را محدود کرد.



قوانین سه تایی استخراج شده برای داده fraud:

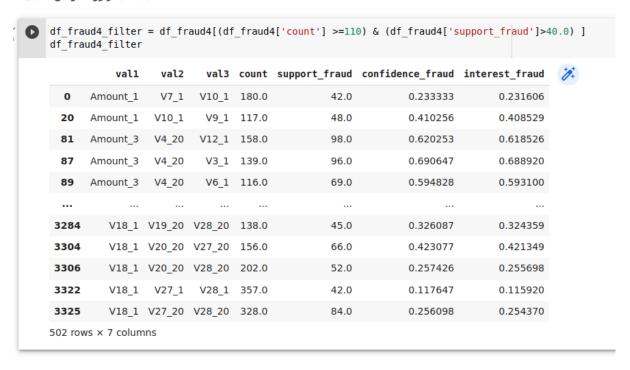


پنج قوانین برتر سه تایی، با معیار confidence , interest برای داده



شرط روی قوانین ۴ تایی:

ایجاد فیلتر روی قوانین ۴ تایی

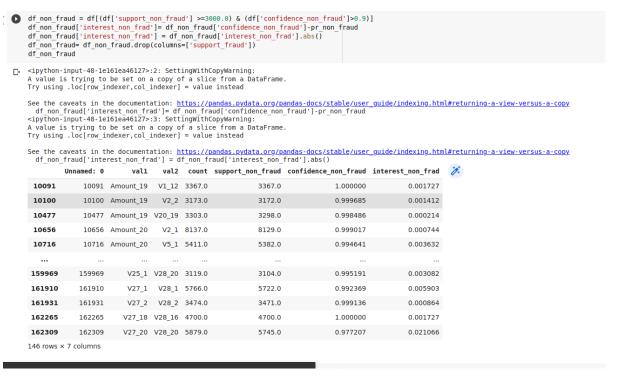


استخراج قوانین برتر روی داده ۴ تایی:

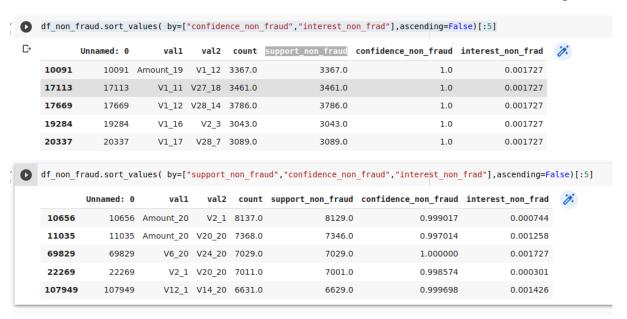


Non fraud

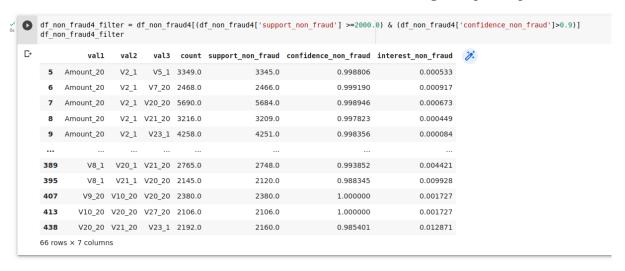
استخراج قوانین ۳ تایی و ایجاد فیلتر روی قوانین سه تایی:



استخراج ۵ قانون برتر روی قوانین سه تایی:



ایجاد فیلتر روی قوانین ۴ تایی:



استخراج قوانین برتر روی داده ۴ تایی:

ینج قانون برتر جهار تایی برای داده non_fraud با معیار confidence و interest



با مشاهده قوانین ۴ تایی برای داده fraud و دقت خوب آن میتوان داده هایی که در این مجموعه قرار میگیرند را fraud و در غیر اینصورت non_fraud در نظر گرفت. صحت روی قوانین دو تایی روی داده fraud فقط دارای ۵ مقدار با confidence بالای ۵.۰ میباشدکه معیار خوبی نیست برای تشخیص داده fraud ولی با قوانین ۴ تایی با توجه به اینکه مقدار confidence ها حدود ۰.۰ نیز وجود دارد، می تواند معیار خوبی باشد.

بخش سوم Hashing Sensitive Locality

سوال اول

الگوریتم LSH تکنیکی برای جستجوی تقریبی نزدیکترین همسایه در فضاهای با ابعاد بالا است که میتواند محدودیت های زیر را داشته باشد:

High False Positive Rate: این الگوریتم می تواند نرخ FP بالایی داشته باشد، به این معنی که ممکن است نقاطی را که در واقع نزدیکترین همسایگان نیستند برگرداند، انجام این کار زمانی که دقت آن بسیار مهم است مشکل ساز باشد.

Parameter Tuning: برای دستیابی به نتایج خوب نیاز به تنظیم دقیق پارامتر دارد، این کار به زمان زیادی نیاز دارد.

High Dimensionality: با افزایش ابعاد داده ها، LSH می تواند کمتر موثر باشد. این به این دلیل است که تعداد جداول هش و توابع هش مورد نیاز برای دستیابی به عملکرد خوب به طور تصاعدی با ابعاد داده ها افزایش می یابد.

Sensitivity to Data Distribution: به توزیع داده ها حساس است. اگر داده ها به طور یکنواخت توزیع نشده باشند، LSH ممکن است نتواند توابع هش خوبی را پیدا کند.

Difficulty with Sparse Data: می تواند با داده های پراکنده، که در آن بیشتر مقادیر صفر هستند، جلوگیری کند. این به این دلیل که توابع هش احتمال کمتری برای ایجاد برخورد برای داده های پراکنده دارند.

در این تکنیک موارد مشابه را با احتمال زیاد به باکت یکسانی نگاشت میکند و امکان جستجوی شباهت کارآمد را فراهم می کند. با این حال، از احتمال نگاشت دو آیتم متفاوت به یک باکت نمی توان به طور کامل اجتناب کرد و در نتیجه منجر به FP میشود. برای کاهش FP، تعداد توابع هش و اندازه جداول هش باید افزایش یابد که پیچیدگی زمانی و مکانی الگوریتم را افزایش می دهد. از طرف دیگر، کاهش تعداد توابع هش و جداول هش، تعداد TP را نیز کاهش می دهد. بنابراین، یافتن

تعادل مناسب بین تعداد توابع هش، جداول هش و تعداد مواردFP ، یک محدودیت کلیدی در LSH است.

سوال دوم

MinHashing به دلایل زیر قابلیت preservingSimilarity را دارد:

- 1. MinHashing مجموعه های مشابه را با احتمال زیاد به یک باکت نگاشت می کند. در نتیجه، اگر دو مجموعه شباهت jaccard بالایی داشته باشند، احتمالاً سیگنیچر MinHash آنها دارای تعداد زیادی از حداقل مقادیر هش منطبق است.
- احتمال اینکه حداقل مقدار هش یک عنصر در مجموعه A برابر با حداقل مقدار هش یک عنصر در مجموعه B باشد،برابر است با شباهت jaccard بین مجموعه ها. که به عنوان ویژگی MinHash شناخته می شود.
- 8. تخمین شباهت jaccard با استفاده از MinHashing با افزایش تعداد توابع هش مورد استفاده برای تولید سیگنیچر، دقیق تر می شود. زیرا با افزایش تعداد توابع هش، احتمال دو مجموعه دارای سیگنیچر یکسان کاهش می یابد. بنابراین، استفاده از MinHashing برای تولید سیگنیچر برای مجموعه ها و مقایسه تعداد حداقل مقادیر هش منطبق، می توان شباهت Jaccard بین مجموعه ها را تخمین زد و شباهت را حفظ کرد.

سوال سوم

```
print('Similar sentences:')
print(get_topk_similar(target_id, candidates, k=5))

Similar sentences:
[['The lazy dog has a quick and brown fox that jumps over it.']
['The lazy dog has a fox that is quick and brown jumping over it.']
['The quick brown fox jumps over the lazy dog.']
['The dog that is lazy has a quick and brown fox jumping over it.']
['A lazy dog has a quick brown fox jumping over it.']]
```