جزوه امتحانی — تحلیل‌ها و سیستم‌های داده‌های حجیم (Big Data Analytics)

استاد: دکتر فرساد زمانی بروجنی
منبع: لکچرهای ترم + نمونه سوالات استاد + Mining of Massive Datasets
نوع امتحان: جزوه‌باز — این جزوه برای پاسخ سریع طراحی شده است

> **نحوه استفاده در امتحان:** ابتدا از «راهنمای صفحه‌بندی» آخر جزوه موضوع را پیدا کن، سپس تعریف حفظی + فرمول + مثال همان بخش را بنویس.

فهرست مطالب

1. مقدمه Big Data و مفاهیم پایه
2. Cluster Computing در برابر RDBMS
3. MapReduce
4. جبر رابطه‌ای با MapReduce
5. ضرب ماتریس با MapReduce
6. Finding Similar Items: Shingle، MinHash، LSH
7. سنجه‌های فاصله و تشابه تاپل‌ها
8. Frequent Itemsets، Apriori، Closed، Maximal
9. خوشه‌بندی Clustering
10. حل کامل نمونه سوالات استاد
11. راهنمای صفحه‌بندی (ایندکس سریع)

---

۱. مقدمه Big Data و مفاهیم پایه

حفظیات

تعریف Big Data (سه V)

- **Volume:** حجم داده آن‌قدر زیاد است که با سیستم‌های معمولی قابل پردازش نیست.
- **Velocity:** داده با نرخ بالا تولید می‌شود.
- **Variety:** داده در شکل‌های مختلف (ساخت‌یافته، نیمه‌ساخت‌یافته، بدون ساختار) وجود دارد.

داده‌کاوی

استخراج الگو/مدل/دانش مفید از داده. دیدگاه‌ها: - آماری → توزیع‌ها - یادگیری ماشین → مدل پیش‌بینی - الگوریتمی → روش‌های مقیاس‌پذیر برای داده حجیم

دو تکنیک محاسباتی مهم

1. **خلاصه‌سازی داده‌ها** (مثل PageRank، خوشه‌بندی)
2. **استخراج ویژگی** (مثل frequent itemsets، شناسایی اقلام مشابه، توصیه‌گر)

PageRank (ایده)

رتبه‌بندی صفحات وب بر اساس ساختار لینک‌ها: اهمیت صفحه تابعی از اهمیت صفحاتی است که به آن لینک داده‌اند.

اصل Bonferroni

در داده عظیم، رویدادهای بسیار نادر هم ممکن است زیاد رخ دهند → بسیاری از یافته‌ها می‌توانند **False Positive** باشند. قبل از داده‌کاوی، تعداد رخدادهای تصادفی مورد انتظار را حساب کن.

عدد e

[e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n \approx 2.718] تقریب مفید: ((a+1)^b \approx e^{ab}) و برای احتمال‌ها: ((1-p)^{pn} \approx e^{-pn})

TF-IDF (خیلی مهم امتحانی)

فرمول‌ها

[TF_{ij}=\frac{f_{ij}}{\max_k f_{kj}}] [IDF_i=\log_2\frac{N}{n_i}] [TF\text{-}IDF = TF_{ij}\times IDF_i]

نماد	معنی
f_{ij}	تعداد تکرار کلمه (i) در سند (j)
$\max_k f_{kj}$	بیشترین تکرار هر کلمه در همان سند (j)
N	تعداد کل اسناد
n_i	تعداد اسنادی که کلمه (i) در آن‌ها آمده

مفهوم

- Stop words پرتکرار در همه اسناد → IDF نزدیک صفر → بی‌اهمیت
- کلمه مفید: در تعداد کمی سند، تکرار زیاد دارد

مثال عددی (نمونه سوال استاد — حل در بخش ۱۰)

$$(N=1024,\ n_w=128,\ f_{wj}=10,\ \max=100)$$
$$(TF=10/100=0.1)$$
$$(IDF=\log_2(1024/128)=\log_2(8)=3)$$
$$(TF\text{-}IDF=0.1\times 3=0.3)$$

Collaborative Filtering / فیلترینگ مشارکتی

تعریف حفظی (سوال امتحانی)

روشی در **سیستم‌های توصیه‌گر (Recommender Systems)** که بر اساس رفتار/امتیاز کاربران مشابه، آیتم‌هایی را به کاربر هدف پیشنهاد می‌دهد.

دو نوع رایج: - **User-based:** کاربران مشابه پیدا کن → آیتم‌های مورد علاقه آن‌ها را پیشنهاد بده - **Item-based:** آیتم‌های مشابه پیدا کن → آیتم‌هایی شبیه آنچه کاربر پسندیده پیشنهاد بده

در درس به Recommendation Engine اشاره شده (Netflix، Amazon) و تحلیل Behavioral Data مشتریان.

---

۲. Cluster Computing در برابر RDBMS

محدودیت‌های RDBMS

1. مقیاس‌پذیری ضعیف/گران در حد ترابایت و پتابایت
2. سرعت
3. تحمل خرابی
4. پردازش‌های پیشرفته
5. پرس‌وجوپذیری روی داده حجیم

ویژگی‌های Cluster Computing (حفظ کن)

1. **High Availability** — دسترسی‌پذیری بالا، کاهش downtime
2. **Fault Tolerance** — مقاومت در برابر خرابی با اجزای تکراری
3. **Data Replication** — تکثیر داده روی چند سرور/مکان
4. **Load Balancing** — توزیع بار بین سرورها
5. **Automated Failover** — جابه‌جایی خودکار به سرور پشتیبان
6. **Backup and Recovery**
7. **Monitoring and Alerting**
8. **Scalability** — مقیاس افقی با افزودن نود

تفاوت‌های کلیدی Cluster/Hadoop در برابر RDBMS سنتی

موضوع	RDBMS سنتی	Cluster / Hadoop
سخت‌افزار	سرورهای گران high-end	commodity servers ارزان
مقیاس	غالباً عمودی (Vertical)	افقی (Horizontal)
داده	ساخت‌یافته، جداول	ساخت‌یافته + بدون ساختار
تحمل خرابی	سخت‌افزاری/گران	نرم‌افزاری (replication)
ایده پردازش	انتقال داده به سمت کد	**انتقال کد به سمت داده**
مدل	SQL / ACID قوی	MapReduce / NoSQL، اغلب eventual consistency

SQL در برابر NoSQL (خلاصه)

- SQL: اسکیمای ثابت، ACID، مقیاس عمودی، مناسب تراکنش و پرس‌وجوی پیچیده
- NoSQL: اسکیما انعطاف‌پذیر، مقیاس افقی، مناسب داده حجیم/متنوع/زمان‌واقعی

Hadoop

- دو ویژگی: **Fault Tolerant** + **Scalable**
- اجزا: HDFS + MapReduce API
- Daemonها: NameNode، DataNode، JobTracker، TaskTracker
- HDFS: فایل‌ها به chunk تقسیم؛ معمولاً replication=3
- حالت‌ها: Standalone / Pseudo-Distributed / Fully Distributed
- JVM مجزا برای هر process → مناسب موازی‌سازی MapReduce

---

۳. MapReduce

حفظیات

تعریف

سبک محاسباتی برای پردازش داده حجیم با **تحمل خرابی**. کاربر فقط دو تابع می‌نویسد: Map و Reduce. هماهنگی، موازی‌سازی و مدیریت خرابی با سیستم است.

سه مرحله

1. **Map:** هر Map Task یک/چند chunk می‌گیرد و زوج‌های ((key,value)) تولید می‌کند.
2. **Shuffle/Sort/Group:** Master Controller خروجی‌ها را بر اساس key مرتب و گروه‌بندی می‌کند → ((k,[v_1,\ldots,v_n]))

3. **Reduce:** برای هر کلید، مقادیر را ترکیب می‌کند → (('k,v))

Combiner

اگر Reduce جابجایی‌پذیر (commutative) و انجمنی (associative) باشد، بخشی از Reduce روی همان نود Map انجام می‌شود تا ترافیک شبکه کم شود.
مثال Word Count: به‌جای چند ((w,1)) مقدار ((w,n_i)) فرستاده می‌شود.

Master Controller

- تخصیص chunk به Map
- hash کلیدها بین Reduceها
- Ping برای تشخیص خرابی
- زمان‌بندی مجدد Taskهای شکست‌خورده

مدیریت خرابی

خرابی	اقدام
Master	معمولاً کل Job باید restart شود
Map Worker	Map Taskها دوباره idle و روی نود سالم اجرا می‌شوند (خروجی محلی از دست رفته)
Reduce Worker	Reduce Task دوباره اجرا می‌شود (ورودی از Map قابل بازیابی است)

Workflow Systems

توسعه MR از دو مرحله ساده به **DAG** از توابع (Clustera، Hyracks). داده از چپ به راست حرکت می‌کند؛ هر تابع با چند Task اجرا می‌شود.

Recursive MR

PageRank و Transitive Closure با تکرار Jobها: ($v^{(t+1)}=Mv^{(t)}$) تا همگرایی.
Pregel: محاسبات گرافی با SuperStep و Checkpoint.

Communication Cost

هزینه ارتباطی ≈ اندازه ورودی Taskها. دو پارامتر: - **Reducer Size (q):** کران بالای مقادیر همراه یک کلید - **Replication Rate (r):** نسبت خروجی Map به ورودی

---

۴. جبر رابطه‌ای با MapReduce

عملیات	Map	Reduce
Selection (sigma_C(R)\)	اگر (C(t)) آنگاه ((t,t))	عبور همان زوج

عملیات	Map	Reduce
Projection (\pi_S(R))	((t',t)) با صفات (S)	حذف تکراری‌ها
Union (R\cup S)	((t,t))	یک بار خروجی
Intersection (R\cap S)	((t,t))	فقط اگر دو مقدار
Difference (R-S)	از ((t,R)) :R؛ از ((t,S)) :S	فقط اگر فقط R
Natural Join	کلید=صفت مشترک B؛ ((b,(R,a))) و ((b, (S,c)))	ترکیب همه جفت‌های R و S
Grouping+Agg	کلید=صفات گروه؛ مقدار=صفت تجمیع	/SUM/COUNT/AVG/MIN MAX

---

۵. ضرب ماتریس با MapReduce

۵.۱ ماتریس × بردار (سوال امتحانی)

فرض (U=MV) و (u_i=\sum_j m_{ij}v_j)

Map: برای هر (m_{ij}) خروجی بده: ((i,\ m_{ij}\cdot v_j))
(اگر (V) در حافظه جا شود، همه Mapها (V) را دارند)

Reduce: برای هر کلید (i): (u_i=\sum) مقادیر

مثال حل‌شده

M=\begin{bmatrix}1&0&5\7&3&0\1&3&5\end{bmatrix},\quad V=\begin{bmatrix}]
[end{bmatrix}\6\4\2

Map: - سطر1: ((1,2)\, (1,0)\, (1,30)) - سطر2: ((2,14)\, (2,12)\, (2,0)) - سطر3: ((3,2)\,
(3,30)\, (3,12))

Reduce: [u_1=32,\ u_2=26,\ u_3=44 \Rightarrow MV=\begin{bmatrix}
[end{bmatrix}\44\26\32

۵.۲ ماتریس × ماتریس (دو Job)

(P=MN)، (p_{ik}=\sum_j m_{ij}n_{jk})

Job1 (Join روی j):
Map: ((j,[M,i,m_{ij}])) و ((j,[N,k,n_{jk}]))
Reduce: همه ترکیب‌ها → ((i,k)\, (m_{ij}n_{jk}))

Job2 (جمع):
Reduce روی کلید ((i,k)): جمع مقادیر → (p_{ik})

مثال

$$N=\begin{bmatrix}2&0\4&8\6&2\end{bmatrix} \Rightarrow P=\begin{bmatrix}34&44\24&26\10&32\end{bmatrix}$$

---

۶. Finding Similar Items

مسئله

پیدا کردن جفت‌های مشابه بین (N) شیء بدون ($O(N^2)$) مقایسه کامل.

Pipeline حفظی

```
Document → Shingling → Sets → MinHash → Signatures → LSH → Candidate pairs → Exact
```

۶.۱ Shingle

تعریف

k-shingle (یا k-gram): دنباله (k) توکن پشت‌سرهم در سند (کاراکتر یا کلمه).

مثال

set: ({ab, bc, ca}) → ($D=abcab,\ k=2$)

نکات

- فقط bag-of-words ترتیب را حفظ نمی‌کند؛ shingle تا حدی ترتیب را نگه می‌دارد.
- (k) کوچک → شباهت کاذب زیاد؛ اسناد کوتاه≈5، اسناد بلند≈10
- معمولاً shingleها hash می‌شوند و مجموعه hashها ذخیره می‌شود.

۶.۲ Jaccard Similarity

$$J(A,B)=\frac{|A\cap B|}{|A\cup B|},\qquad d_J(A,B)=1-J(A,B)$$

مثال

($J=2/5=0.4$) → ($A=\{a,b,c,d\},\ B=\{b,c,e\}$)

۶.۳ Min-Hashing (درهم‌سازی مین — سوال امتحانی)

تعریف

تبدیل مجموعه‌های بزرگ به **امضای کوتاه (signature)** طوری که تشابه تقریباً حفظ شود.

ایده

یک permutation تصادفی π از عناصر؛
$(=h_\pi(C))$ اولین عنصری از (C) که در ترتیب π ظاهر می‌شود.

خاصیت طلایی (حفظ کن)

$$ Pr\big[h_\pi(C_1)=h_\pi(C_2)\big]=J(C_1,C_2)\ $$

Signature

با (K) تابع $[\ sig(C)=[h_1(C),\ldots,h_K(C)]\ ]$ hash/permutation: تخمین تشابه = نسبت خانه‌های برابر دو امضا.

پیاده‌سازی عملی

به‌جای permutation کامل از hash استفاده کن: $[\ h_{a,b}(x)=((ax+b)\bmod p)\bmod N\ ]$
الگوریتم یک‌گذره: برای هر سطر با مقدار 1، $(sig[i]=\min(sig[i], h_i(row)))$

۶.۴ LSH (Locality-Sensitive Hashing)

هدف

فقط جفت‌هایی را candidate کن که احتمالاً مشابه‌اند.

Banding

ماتریس signature با (n) سطر → (b) باند، هر باند (r) سطر، $(n=br)$.
اگر دو ستون در **حداقل یک باند** کاملاً برابر باشند → candidate pair.

فرمول احتمال (خیلی مهم)

اگر تشابه (s) باشد: $[\ P(\text{candidate})=1-(1-s^r)^b\ ]$ آستانه تقریبی: $s^*\approx (1/b)^{1/r}$

مثال عددی ((b=20,\ r=5))

(s)	(P) تقریبی
0.3	0.047≈
0.5	0.47≈
0.8	0.9996≈

اثر پارامترها

- (r)↑ → FN ، FP↓
- (b)↑ → FP ، FN↓

۶.۵ خانواده LSH برای سنجه‌های مختلف

(از chap3 فارسی) برای هر سنجه فاصله، خانواده توابع حساس به محل وجود دارد تا اشیای نزدیک با احتمال بالا به یک bucket بروند.

---

۷. سنجه‌های فاصله و تشابه تاپل‌ها

اصول Metric (حفظ)

1. (d(x,y)\ge 0)
2. (d(x,y)=0 \Leftrightarrow x=y)
3. تقارن (d(x,y)=d(y,x))
4. مثلثی (d(x,y)\le d(x,z)+d(z,y))

فرمول فاصله‌ها

نام	فرمول
اقلیدسی (L_2)	(sqrt{\sum(x_i-y_i)^2}\)
منهتن (L_1)	sum\)
مینکوفسکی (L_r)	sum\))
Chebyshev / (L_\infty)	max_i\)
Jaccard distance	-1)
Cosine similarity	((|x||y|)/(x\cdot y))
Cosine distance	(cos\theta\-1)

نام	فرمول
Hamming	تعداد مؤلفه‌های متفاوت
Edit distance	حداقل درج/حذف/جایگزینی؛ (D=|X|+|Y|-2)

Data Matrix و Dissimilarity Matrix

- Data Matrix: سطر=تاپل، ستون=صفت
- Dissimilarity Matrix: (d_{ij}) بین تاپل‌ها؛ قطر=0؛ متقارن اگر metric باشد

عدم‌تشابه بر اساس نوع صفت

نوع	فرمول
اسمی Nominal	(d=(p-m)/p) ؛ (p)=تعداد صفات، (m)=تطابق‌ها
دودویی متقارن	(d=(r+s)/(q+r+s+t))
دودویی نامتقارن	(d=(r+s)/(q+r+s)) ؛ (J=q/(q+r+s))
عددی	اقلیدسی/منهتن/... بعد از نرمال‌سازی
ترکیبی Mixed	میانگین وزن‌دار عدم‌تشابه صفات معتبر

جدول Contingency دودویی

- (q): هر دو 1
- (r): i=0,j=1
- (s): i=1,j=0
- (t): هر دو 0

مثال اسمی

(x=\text{قرمز},\text{تهران}M,\text{مجرد})
(y=\text{قرمز},\text{تهران}F,\text{متاهل})
(m=2,\ p=4,\ d=(4-2)/4=0.5)

---

۸. Frequent Itemsets

تعاریف

- **Support:** (sup(X)=) کسر تراکنش‌هایی که (X) را دارند
- **Frequent:** (sup(X)\ge minsup)
- **Association Rule:** (A\Rightarrow B):

- $\sup(A\Rightarrow B)=\sup(A\cup B)$
- $\mathrm{conf}(A\Rightarrow B)=\sup(A\cup B)/\sup(A)$

Closed و Maximal (حفظ)

نوع	تعریف
Closed	هیچ ابرمجموعه‌ای با **همان support** ندارد
Maximal	هیچ ابرمجموعه **frequent** ندارد

رابطه طلایی: $M \subseteq C \subseteq F$ (Maximal ⊆ Closed ⊆ Frequent)

- از Closed می‌توان support همه frequentها را بازیابی کرد.
- از Maximal فقط مرز را می‌دانیم؛ support دقیق از دست می‌رود.

CHARM

الگوریتم استخراج closed با tidset intersection؛ pruning با تساوی/زیرمجموعه بودن tidsetها.

Apriori

اصل Apriori

اگر itemset پرتکرار باشد، همه زیرمجموعه‌هایش هم پرتکرارند.
پس کاندید (k)تایی فقط از frequentهای ((k-1))تایی ساخته می‌شود.

مراحل

1. Pass1: شمارش itemset-1ها → L_1
2. Join L_{k-1} برای ساخت C_k
3. Prune با اصل Apriori
4. شمارش support کاندیدها → L_k
5. تکرار تا $L_k=\emptyset$
6. از frequentها قوانین با confidence بساز

PCY

در Pass1 علاوه بر شمارش اقلام، pairها را hash به bucket کن.
در Pass2 جفت فقط کاندید است اگر هر دو item frequent **و** bucket مربوطه frequent باشد.

مثال Apriori کامل → بخش ۱۰ سوال ۵

---

۹. خوشه‌بندی Clustering

تعریف و هدف

گروه‌بندی اشیاء طوری که: - فاصله درون‌خوشه‌ای کمینه - فاصله بین‌خوشه‌ای بیشینه

بدون سرپرست است (برخلاف Classification که برچسب دارد). گاهی «طبقه‌بندی خودکار» نامیده می‌شود.

انواع روش‌ها

1. Partitioning: K-Means، K-Medoids، PAM، CLARA، CLARANS
2. Hierarchical: AGNES، DIANA، BIRCH، CURE
3. Density-based: DBSCAN، OPTICS، DENCLUE
4. Grid-based: STING، CLIQUE

K-Means

1. مرکز اولیه (k)
2. تخصیص هر نقطه به نزدیک‌ترین مرکز
3. به‌روزرسانی مراکز (میانگین)
4. تکرار تا ثبات

ضعف‌ها: نیاز به (k)، حساس به init و نویز، خوشه کروی، بهینه محلی

K-Medoids / PAM / CLARA / CLARANS

- Medoid = شیء واقعی داده (مقاوم‌تر به پرت)
- PAM: تعویض medoid اگر هزینه کم شود
- CLARA: PAM روی نمونه‌های تصادفی (برای داده بزرگ)
- CLARANS: جستجوی تصادفی همسایه‌ها برای بهبود CLARA

سلسله‌مراتبی — Linkage

Linkage	فاصله دو خوشه
Single	فاصله اعضا (min\)
Complete	فاصله اعضا (max\)
Average	میانگین همه زوج‌ها
Centroid	فاصله مراکز
Diameter	قطر خوشه حاصل

Linkage	فاصله دو خوشه
Radius	شعاع خوشه حاصل

- فضای اقلیدسی → centroid
- غیراقلیدسی → clustroid/medoid

BFR (داده بزرگ اقلیدسی)

فرض خوشه‌های تقریباً نرمال. خلاصه هر خوشه: [N,\quad SUM,\quad SUMSQ]
[centroid=\frac{SUM}{N},\quad variance=\frac{SUMSQ}{N}-\left(\frac{SUM}{N} \right)^2\ ] مجموعه‌ها: DS (Discard) / CS (Compressed) / RS (Retained)
فاصله: Mahalanobis

CURE

چند نقطه نماینده در هر خوشه که به سمت مرکز جمع می‌شوند → مناسب شکل‌های غیرکروی.

DBSCAN

پارامترها: `Eps`, `MinPts` - **Core:** $|N_{Eps}(q)|\ge MinPts$ - **Border:** در همسایگی یک core - **Noise:** هیچ‌کدام

مزیت: شکل دلخواه + تشخیص نویز؛ ضعف: حساس به پارامتر و چگالی‌های متفاوت.
OPTICS برای چگالی‌های متفاوت بهتر است.

مثال یک تکرار K-Means

نقاط: A(1,1), B(1,2), C(4,4), D(5,4))
مراکز اولیه: $(mu_1=A,\ \mu_2=C\)$

تخصیص: $(C_1={A,B},\ C_2={C,D})$
مراکز جدید: $(mu_1=(1,1.5),\ \mu_2=(4.5,4)\)$

---

۱۰. حل کامل نمونه سوالات استاد

> امتحان نمونه: داده‌های حجیم — ۹۰ دقیقه — ۱۰ نمره توصیفی

سوال ۱ (۲ نمره) — تعریف مفاهیم

الف) Min-hashing

روشی برای ساخت امضای کوتاه از مجموعه‌های بزرگ طوری که احتمال برابر شدن مقدار hash دو مجموعه برابر **شباهت ژاکارد** آن‌ها باشد: $Pr[h(C_1)=h(C_2)]=J(C_1,C_2)$. برای هر permutation، اولین عنصر مجموعه در ترتیب permutation به‌عنوان مقدار hash گرفته می‌شود.

ب) Collaborative Filtering

روش توصیه‌گر مبتنی بر مشارکت کاربران/آیتم‌ها: با یافتن کاربران مشابه (یا آیتم‌های مشابه) بر اساس امتیازها/رفتار، آیتم‌های مناسب به کاربر هدف پیشنهاد می‌شود (مثل Netflix/Amazon).

ج) Jaccard Similarity

برای دو مجموعه (A,B): $$J(A,B)=\frac{|A\cap B|}{|A\cup B|}$$

د) Shingle

دنباله‌ای از (k) توکن متوالی در سند (k-shingle). مجموعه shingleهای سند برای مقایسه شباهت اسناد (near-duplicate) استفاده می‌شود.

سوال ۲ (۲ نمره) — Cluster Computing در برابر RDBMS

RDBMS سنتی: مقیاس‌پذیری گران/عمودی، مناسب داده ساخت‌یافته، ACID قوی، سرورهای قدرتمند، در حجم ترابایت/پتابایت محدود از نظر سرعت، تحمل خرابی و پردازش پیشرفته.

Cluster Computing: مقیاس افقی با commodity servers، replication و failover خودکار، high availability، load balancing، تحمل خرابی نرم‌افزاری ارزان‌تر، انتقال کد به سمت داده (مثل Hadoop/MapReduce)، مناسب داده حجیم و متنوع.

سوال ۳ (۲ نمره) — ضرب ماتریس×بردار با MapReduce

برای $(U=MV)$: 1. **Map:** هر عنصر (m_{ij}) را با (v_j) ضرب کن و زوج $((i,\ m_{ij}v_j))$ تولید کن. 2. **Shuffle:** همه مقادیر مربوط به سطر (i) را کنار هم بگذار. 3. **Reduce:** برای هر (i) مقادیر را جمع کن تا (u_i) به‌دست آید.

اگر بردار (V) در حافظه جا شود، همه Mapperها کپی (V) را دارند. اگر جا نشود، می‌توان (V) را به تکه‌ها تقسیم کرد و چند پاس انجام داد.

(مثال عددی کامل در بخش ۵)

سوال ۴ (۲ نمره) — TF-IDF

[N=1024,\quad n_w=128,\quad f_{wj}=10,\quad \max_k f_{kj}=100]

[TF=\frac{10}{100}=0.1] [IDF=\log_2\frac{1024}{128}=\log_2 8=3] [TF\text{-}IDF=0.1\times 3=\mathbf{0.3}]

سوال ۵ (۲ نمره) — Apriori با MST=0.3

تراکنش‌ها ((n=8)):

TID	اقلام
1	a,b
2	b,c
3	a,b,c
4	d,e,f
5	a,b,c
6	d,f
7	c,d,e,f
8	a,b,c,d,e

حداقل support count: (0.3\times 8=2.4) → معمولاً آستانه شمارشی **≥3** (چون support باید ≥0.3 باشد؛ count≥3 ⇒ count≥2.4 ⇒ count/8≥0.3).
(اگر استاد floor بگیرد و ۲ را هم قبول کند، در برگه بنویس: «با تفسیر ceiling از 2.4، حداقل شمارش ۳».)

Pass 1 — 1-itemsets

Item	Count	Support	Frequent?
a	4	0.50	✓
b	5	0.625	✓
c	5	0.625	✓
d	4	0.50	✓
e	3	0.375	✓
f	3	0.375	✓

(L_1={a,b,c,d,e,f})

Pass 2 — کاندیدهای 2تایی و شمارش

Pair	Count	Sup	Freq?
ab	4	0.50	✓
ac	3	0.375	✓
ae	1	0.125	✗
ad	1	0.125	✗
bc	4	0.50	✓
bd	1	0.125	✗
be	1	0.125	✗
cd	2	0.25	✗ (<0.3)
ce	2	0.25	✗
de	3	0.375	✓
df	3	0.375	✓
ef	2	0.25	✗
af,bf,cf	≥1 یا 2	<0.3	✗

(L_2={ab, ac, bc, de, df})

Pass 3 — کاندیدهای 3تایی

از (L_2): فقط {a,b,c} قابل ساخت است (چون ab,ac,bc همگی frequent).
{d,e,f} کاندید نیست چون ef frequent نیست (اصل Apriori).

Triple	Count	Sup	Freq?
abc	3	0.375	✓

(L_3={abc})

Pass 4

کاندید 4تایی ساخته نمی‌شود → توقف.

Frequent Itemsets نهایی

- 1: a,b,c,d,e,f
- 2: ab, ac, bc, de, df
- 3: abc

---

۱۱. راهنمای صفحه‌بندی (ایندکس سریع)

در امتحان جزوه‌باز: موضوع را اینجا پیدا کن → برو به بخش مربوطه.

موضوع / کلمه کلیدی	بخش جزوه	یادداشت سریع
Big Data، ۳V، Volume/Velocity/Variety	§۱	تعریف سه‌گانه
TF-IDF، اهمیت کلمه	§۱ و §۱۰-س۴	(TF\times\log_2(N/n))
Collaborative Filtering، توصیه‌گر	§۱ و §۱۰-س۱	User/Item-based
Bonferroni، False Positive	§۱	رخداد تصادفی در داده عظیم
PageRank	§۱ و §۳	اهمیت از لینک‌ها؛ تکرار MR
RDBMS محدودیت‌ها	§۲	مقیاس، سرعت، fault
Cluster Computing ویژگی‌ها	§۲ و §۱۰-س۲	HA, FT, Replication, Failover, LB
SQL vs NoSQL	§۲	schema، ACID، مقیاس
Hadoop، HDFS، NameNode	§۲	commodity + replication
MapReduce تعریف/مراحل	§۳	Map→Shuffle→Reduce
Combiner	§۳	associative+commutative
خرابی Master/Map/Reduce	§۳	جدول مدیریت خرابی
Workflow / DAG / Pregel	§۳	توسعه MR
Selection/Projection/Join	§۴	جدول جبر رابطه‌ای
Matrix×Vector	§۵ و §۱۰-س۳	key=سطر؛ جمع حاصل‌ضرب‌ها
Matrix×Matrix	§۵	دو Job یا یک Job با replication
Shingle / k-gram	§۶ و §۱۰-س۱	دنباله k توکن
Jaccard	§۶ و §۱۰-س۱	|∪|/|∩|
MinHash	§۶ و §۱۰-س۱	Pr=Jaccard
LSH، banding، (1-(1-s^r)^b)	§۶	candidate pairs
S-curve، threshold	§۶	({1/r}^(b/1))
فاصله اقلیدسی/منهتن/(L_\infty)/(L_r)	§۷	اصول metric
Cosine، Hamming، Edit	§۷	متن/بردار/رشته
Nominal ((p-m)/p)	§۷	صفات اسمی
Binary symmetric/asymmetric	§۷	با/بدون t

موضوع / کلمه کلیدی	بخش جزوه	یادداشت سریع
Data/Dissimilarity Matrix	§۷	قطر صفر
Support / Confidence	§۸	قوانین انجمنی
Apriori	§۸ و §۱۰-س۵	اصل زیرمجموعه
PCY	§۸	hash bucket برای pairs
Closed / Maximal	§۸	(M\subseteq C\subseteq F)
CHARM	§۸	closed با tidset
Clustering تعریف	§۹	min intra / max inter
K-Means	§۹	centroid + SSE
K-Medoids، PAM، CLARA، CLARANS	§۹	medoid واقعی
Hierarchical / Linkage	§۹	single/complete/average
BFR: N,SUM,SUMSQ	§۹	داده بزرگ اقلیدسی
CURE	§۹	چند نماینده
DBSCAN: Core/Border/Noise	§۹	Eps, MinPts
OPTICS / DENCLUE	§۹	چگالی
STING / CLIQUE	§۹	grid-based
نمونه سوالات + حل	**§۱۰**	اول اینجا را ببین

چک‌لیست ۳۰ ثانیه‌ای قبل از نوشتن پاسخ

1. تعریف یک‌خطی دقیق؟
2. فرمول با نمادها؟
3. یک مثال عددی کوچک؟
4. تفاوت با مفهوم نزدیک (مثل Closed vs Maximal، K-Means vs K-Medoids، RDBMS vs Cluster)؟

منابع فایل‌های خام استخراج‌شده

متن خام لکچرها در پوشه extracted/ ذخیره شده تا در صورت نیاز جزئیات بیشتر را جستجو کنی: - 08_chap1_intro.txt — مقدمه و TF-IDF - 10_overview.txt — Big Data، RDBMS، Hadoop - 07_chap2_1_mapreduce.txt / 06_chap2_2_... — MapReduce - 03_matrix_mult_mapreduce.txt — مثال ضرب ماتریس - 09_mmds_lsh.txt / 05_chap3_... — Similar Items و فاصله - 02_similarity_dissimilarity.txt — تشابه تاپل‌ها - 04_closed_maximal_itemsets.txt — Closed/Maximal -

- خوشه‌بندی — 11_chap7_clustering_mmds.txt / 12_clustering_han.txt
- مبانی داده‌کاوی — 01_DM_S01_data_mining.txt
- نمونه سوالات — 13_sample_exam.txt + exam_images/

موفق باشی در امتحان. با این جزوه، برای هر سوال نمونه استاد پاسخ آماده داری؛ بقیه سوالات هم از روی ایندکس آخر سریع پیدا می‌شوند.