جزوه امتحانی — داده‌های حجیم (Big Data Analytics)

استاد: دکتر فرساد زمانی بروجنی | **امتحان:** جزوه‌باز — ۹۰ دقیقه | **منبع:** لکچرهای ترم + نمونه سوالات + MMDS

در امتحان: ۱) اول چیت‌شیت و راهنمای صفحات را ببین ۲) موضوع را در جدول پیدا کن و به بخش برو ۳) پاسخ = تعریف یک‌خطی + فرمول + مثال کوتاه ۴) اگر شبیه نمونه استاد بود → بخش ۱۰

چیت‌شیت سریع (صفحه اول امتحان)

تعاریف یک‌خطی

مفهوم	تعریف کوتاه
Shingle	دنبالهٔ k توکن متوالی در سند. مثال k=2 از abcab → {ab, bc, ca}
Jaccard	شباهت دو مجموعه = |A ∩ B| / |A ∪ B|
MinHash	درهم‌سازی طوری که احتمال برابر شدن hash دو مجموعه = Jaccard باشد
LSH	فقط جفت‌های احتمالاً مشابه را candidate می‌کند
Collaborative Filtering	توصیه بر اساس کاربران یا آیتم‌های مشابه
Closed itemset	هیچ ابرمجموعه‌ای با همان support ندارد
Maximal itemset	هیچ ابرمجموعه frequent ندارد
MapReduce	پردازش موازی با دو تابع Map و Reduce + تحمل خرابی

فرمول‌های حیاتی

TF-IDF

```
TF(i,j)  =  f(i,j) / max_k f(k,j)
IDF(i)   =  log2( N / n(i) )
TF-IDF   =  TF(i,j) × IDF(i)
```

Jaccard و MinHash

```
J(A,B) = |A ∩ B| / |A ∪ B|
d_J    = 1 − J(A,B)

Pr[ hπ(C1) = hπ(C2) ] = J(C1, C2)
```

LSH Banding (b باند، هر باند r سطر، شباهت s)

```
P(candidate) = 1 − (1 − s^r)^b
threshold   ≈ (1/b)^(1/r)
```

فاصله‌ها

```
Euclidean L2 :  sqrt( Σ (xi − yi)^2 )
Manhattan L1 :  Σ |xi − yi|
Minkowski Lr :  ( Σ |xi − yi|^r )^(1/r)
Chebyshev L∞ :  max |xi − yi|
Cosine sim   :  (x · y) / ( ||x|| × ||y|| )
Cosine dist  :  1 − cos(θ)
Hamming      :  # differing positions
Edit         :  |X| + |Y| − 2×|LCS|
Nominal      :  d = (p − m) / p
Asym. binary :  d = (r+s)/(q+r+s) ;  J = q/(q+r+s)
```

Association Rule و Closed/Maximal

```
support(A ⇒ B)    = support(A ∪ B)
confidence(A ⇒ B) = support(A ∪ B) / support(A)

M ⊆ C ⊆ F
```

MapReduce — ضرب ماتریس در بردار U = M × V

```
Map:     emit( i ,  m_ij × v_j )     for each m_ij
Reduce:  u_i = Σ_j (m_ij × v_j)     for each i
```

K-Means / DBSCAN / BFR

```
SSE = Σ || x − μ_cluster(x) ||^2

DBSCAN Core:   |N_Eps(q)| ≥ MinPts
       Border: near a Core, but not Core
       Noise:  neither

BFR:  N , SUM , SUMSQ
centroid_i = SUM_i / N
variance_i = SUMSQ_i/N − (SUM_i/N)^2
```

مقایسه سریع

موضوع	نکته طلایی
RDBMS در برابر Cluster	عمودی/گران/ACID در برابر افقی/commodity/replication و انتقال کد به داده
K-Means در برابر K-Medoids	میانگین (حساس به پرت) در برابر نقطه واقعی داده (مقاوم‌تر)
Closed در برابر Maximal	Closed اطلاعات support را نگه می‌دارد؛ Maximal فقط مرز را
Cosine در برابر اقلیدسی برای متن	Cosine برای بردارهای sparse متن بهتر است
Combiner	فقط وقتی Reduce انجمنی و جابجایی‌پذیر باشد

راهنمای صفحات / ایندکس سریع (اول جزوه)

در امتحان: کلمه کلیدی سوال را اینجا پیدا کن → برو به بخش گفته‌شده.

کلمه کلیدی سوال	بخش	چه بنویسی
Shingle	§۶ + چیت‌شیت	تعریف + مثال abcab
Jaccard	§۶ + چیت‌شیت	فرمول |∪|/|∩|
MinHash / درهم‌سازی مین	§۶ + چیت‌شیت	تعریف + Pr = Jaccard
Collaborative Filtering	§۱ + چیت‌شیت	User-based / Item-based
Cluster در برابر RDBMS	§۲	جدول تفاوت‌ها
Matrix-Vector / ضرب ماتریس×بردار	§۵	Map و Reduce کامل
TF-IDF	§۱ و §۱۰-س۴	سه فرمول + عدد
Apriori	§۸ و §۱۰-س۵	مراحل Pass به Pass
LSH / banding	§۶	فرمول b^(s^r−1)−1
Support / Confidence	§۸	دو فرمول قانون
Closed / Maximal	§۸	تعریف + M⊆C⊆F
فاصله اقلیدسی / منهتن / L∞	§۷	فرمول‌ها
تشابه اسمی / دودویی	§۷	p/(p−m) و q,r,s,t
تعریف MapReduce	§۳	سه مرحله + Combiner
خرابی نودها	§۳	جدول Master/Map/Reduce
Join با MapReduce	§۴	کلید = صفت مشترک
تعریف Clustering	§۹	min intra / max inter
K-Means	§۹	مراحل + مثال
DBSCAN	§۹	Core / Border / Noise
BFR / CURE	§۹	خلاصه‌ها / نماینده‌ها
Hadoop / HDFS	§۲	Scalable + Fault Tolerant
نمونه سوالات استاد با حل	**§۱۰**	**اول اینجا را ببین**

شماره بخش‌ها: §۱ مقدمه و §۴ · §۲ Cluster / RDBMS · §۳ MapReduce · TF-IDF جبر رابطه‌ای · §۵ ضرب ماتریس · §۶ Similar Items · §۷ فاصله‌ها · §۸ §۹ Frequent Itemsets خوشه‌بندی · §۱۰ حل نمونه سوالات

---

§۱ — مقدمه Big Data و مفاهیم پایه

تعریف Big Data (سه V)

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

داده‌کاوی

استخراج الگو، مدل یا دانش مفید از داده. دیدگاه‌ها: آماری → توزیع‌ها؛ یادگیری ماشین → مدل پیش‌بینی؛ الگوریتمی → روش مقیاس‌پذیر.

دو تکنیک محاسباتی مهم

1. خلاصه‌سازی داده‌ها (مثل PageRank و خوشه‌بندی)

2. استخراج ویژگی (مثل frequent itemsets، شناسایی مشابه، توصیه‌گر)

PageRank (ایده)

رتبه‌بندی صفحات وب بر اساس لینک‌ها: اهمیت یک 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؛ پس بسیاری از یافته‌ها می‌توانند False Positive باشند.

عدد e

```
e ≈ 2.718
e = lim (1 + 1/n)^n          as n → ∞
(1 + a)^b ≈ e^(a×b)
(1 − p)^n ≈ e^(−p×n)
```

TF-IDF (خیلی مهم)

کلمه مفید: در تعداد کمی سند، تکرار زیاد دارد. Stop wordها در همه اسناد پرتکرارند → IDF نزدیک صفر.

فرمول

```
TF(i,j) = f(i,j) / max_k f(k,j)
IDF(i)  = log2( N / n(i) )
TF-IDF  = TF(i,j) × IDF(i)

f(i,j)     = count of word i in doc j
max_k f(k,j) = max word count in doc j
N          = number of documents
n(i)       = docs containing word i
```

مثال نمونه سوال استاد

```
N = 1024 ,  n(w) = 128 ,  f(w,j) = 10 ,  max = 100

TF     = 10 / 100 = 0.1
IDF    = log2(1024 / 128) = log2(8) = 3
TF-IDF = 0.1 × 3 = 0.3
```

Collaborative Filtering (فیلترینگ مشارکتی)

روشی در سیستم‌های توصیه‌گر: بر اساس رفتار یا امتیاز کاربران مشابه، آیتم‌هایی به کاربر هدف پیشنهاد می‌شود.

- **User-based:** کاربران شبیه کاربر هدف را پیدا کن → آیتم‌های مورد علاقه آن‌ها را پیشنهاد بده
- **Item-based:** آیتم‌های شبیه آیتم‌های پسندیده‌شده را پیشنهاد بده

در درس به Recommendation Engine (مثل Netflix و Amazon) و Behavioral Data اشاره شده است.

§۲ — Cluster Computing در برابر RDBMS

محدودیت‌های RDBMS

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

ویژگی‌های Cluster Computing

1. High Availability
2. Fault Tolerance
3. Data Replication
4. Load Balancing
5. Automated Failover
6. Backup and Recovery
7. Monitoring and Alerting
8. Scalability (افقی)

موضوع	RDBMS سنتی	Cluster / Hadoop
سخت‌افزار	سرور گران high-end	commodity server ارزان
مقیاس	غالباً عمودی	افقی
داده	ساخت‌یافته، جدولی	ساخت‌یافته + بدون ساختار
تحمل خرابی	سخت‌افزاری/گران	نرم‌افزاری با replication
ایده پردازش	انتقال داده به سمت کد	**انتقال کد به سمت داده**
مدل	SQL و ACID قوی	MapReduce / NoSQL

SQL در برابر NoSQL

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

Hadoop

- دو ویژگی: **Fault Tolerant** + **Scalable**
- اجزا: HDFS + MapReduce API
- Daemonها: TaskTracker ، JobTracker ، DataNode ، NameNode
- HDFS فایل را به chunk تقسیم می‌کند؛ معمولاً replication = 3
- حالت‌ها: Standalone / Pseudo-Distributed / Fully Distributed

MapReduce — §۳

تعریف: سبک محاسباتی برای پردازش داده حجیم با تحمل خرابی. کاربر فقط دو تابع Map و Reduce می‌نویسد.

سه مرحله

1. **Map:** هر Map Task یک یا چند chunk می‌گیرد و زوج‌های (key, value) تولید می‌کند.
2. **Shuffle / Sort / Group:** Master خروجی‌ها را بر اساس key گروه‌بندی می‌کند.
3. **Reduce:** برای هر کلید، مقادیر را ترکیب می‌کند.

شکل کلی

```
(k, v1), (k, v2), ...  →  (k, [v1, v2, ..., vn])
(k, [v1, ..., vn])     →  (k, v')
```

Combiner

اگر Reduce جابجایی‌پذیر و انجمنی باشد، بخشی از Reduce روی همان نود Map انجام می‌شود تا ترافیک شبکه کم شود.

مثال Word Count

```
بدون Combiner:  چند بار (w, 1)
با Combiner:     (w, n_i)  در و Reduce جمع نهایی
```

مدیریت خرابی

خرابی	اقدام
Master	معمولاً کل Job باید از نو شروع شود
Map Worker	Map Taskها دوباره روی نود سالم اجرا می‌شوند
Reduce Worker	Reduce Task دوباره اجرا می‌شود

Workflow Systems: توسعه MR از دو مرحله ساده به DAG از توابع (Hyracks ، Clustera).

Recursive MR: PageRank و Transitive Closure با تکرار Job تا همگرایی.

PageRank iteration

```
v^(t+1) = M × v^(t)
```

§۴ — جبر رابطه‌ای با MapReduce

عملیات	Map	Reduce
Selection	اگر شرط برقرار بود: (t,t)	عبور همان زوج
Projection	('t',t) با صفات انتخاب‌شده	حذف تکراری‌ها
Union	(t,t) از R و S	یک‌بار خروجی
Intersection	(t,t)	فقط اگر دو مقدار باشد
Difference R−S	از R: (t,R) ؛ از S: (t,S)	فقط اگر فقط R آمده باشد
Natural Join	کلید = صفت مشترک B	ترکیب همه جفت‌های R و S
Grouping + Agg	کلید = صفات گروه	SUM / COUNT / AVG / MIN / MAX

Join ساده R(A,B) ⋈ S(B,C)

```
Map from R:  emit( b , (R, a) )
Map from S:  emit( b , (S, c) )
Reduce:      for each b, combine all a with all c → (a, b, c)
```

§۵ — ضرب ماتریس با MapReduce

۵.۱ ماتریس × بردار (سوال امتحانی)

هدف و الگوریتم

```
U = M × V
u_i = Σ_j ( m_ij × v_j )

Map:     emit( i , m_ij × v_j )   for each m_ij
Reduce:  u_i = sum of values      for each i
```

مثال حل‌شده

```
M = | 1  0  5 |      V = | 2 |
    | 7  3  0 |          | 4 |
    | 1  3  5 |          | 6 |

Map row1: (1,2) , (1,0) , (1,30)
Map row2: (2,14), (2,12), (2,0)
Map row3: (3,2) , (3,12), (3,30)

Reduce:
u1 = 2+0+30 = 32
u2 = 14+12+0 = 26
u3 = 2+12+30 = 44

U = [32, 26, 44]^T
```

۵.۲ ماتریس × ماتریس (دو Job)

الگوریتم

```
P = M × N
p_ik = Σ_j ( m_ij × n_jk )

Job1 Map:    emit( j , [M, i, m_ij] )
             emit( j , [N, k, n_jk] )
Job1 Reduce: emit( (i,k) , m_ij × n_jk )

Job2 Reduce: p_ik = sum of values for key (i,k)
```

نتیجه مثال

```
N = | 2  0 |      P = | 32  10 |
    | 4  8 |          | 26  24 |
    | 6  2 |          | 44  34 |
```

§۶ — Finding Similar Items

Pipeline

```
Document → Shingling → Sets → MinHash → Signatures → LSH → Candidates → Exact check
```

۶.۱ Shingle

تعریف: k-shingle یعنی دنباله k توکن پشت‌سرهم در سند.

مثال

```
D = abcab ,  k = 2
S(D) = {ab, bc, ca}
```

- bag-of-words ترتیب را از دست می‌دهد؛ shingle تا حدی ترتیب را نگه می‌دارد.
- اسناد کوتاه ≈ k=5 ؛ اسناد بلند ≈ k=10

Jaccard ۶.۲

فرمول و مثال

```
J(A,B) = |A ∩ B| / |A ∪ B|
d_J    = 1 − J(A,B)

A = {a,b,c,d} , B = {b,c,e}
|A ∩ B| = 2 , |A ∪ B| = 5
J = 2/5 = 0.4 ,  d = 0.6
```

MinHash ۶.۳

تعریف: تبدیل مجموعه بزرگ به امضای کوتاه طوری که تشابه تقریباً حفظ شود.

ایده: permutation تصادفی π ؛ hπ(C) = اولین عنصر C در ترتیب π.

خاصیت طلایی و signature

```
Pr[ hπ(C1) = hπ(C2) ] = J(C1, C2)

sig(C) = [h1(C), h2(C), ..., hK(C)]
estimated similarity = (# equal rows) / K

h(a,b)(x) = ((a×x + b) mod p) mod N
if row has 1:  sig[i] = min(sig[i], hi(row))
```

LSH ۶.۴

ماتریس signature با n سطر را به b باند تقسیم کن؛ هر باند r سطر دارد (n = b×r). اگر دو ستون در حداقل یک باند برابر باشند → candidate.

فرمول و مثال

```
P(candidate) = 1 − (1 − s^r)^b
threshold    ≈ (1/b)^(1/r)

b=20 , r=5 , s=0.8 → P ≈ 0.99965
b=20 , r=5 , s=0.3 → P ≈ 0.047

r↑ → FP↓ , FN↑
b↑ → FN↓ , FP↑
```

§۷ — سنجه‌های فاصله و تشابه تاپل‌ها

اصول Metric

1. $d(x,y) \geq 0$
2. $d(x,y) = 0$ اگر و فقط اگر $x = y$
3. $d(x,y) = d(y,x)$
4. $d(x,y) \leq d(x,z) + d(z,y)$

فرمول فاصله‌ها و مثال

```
Euclidean:  sqrt( Σ (xi−yi)^2 )
Manhattan:  Σ |xi−yi|
Minkowski:  ( Σ |xi−yi|^r )^(1/r)
L∞:         max |xi−yi|
Cosine:     (x·y) / (||x|| ||y||)
Hamming:    # differing positions
Edit:       |X| + |Y| − 2×|LCS|

Example Euclidean:
x=(1,2), y=(4,6)
d = sqrt(9+16) = 5

Example Hamming:
X=10101 , Y=11110 → d = 3
```

ماتریس داده و عدم‌تشابه

- **Data Matrix:** سطر = تاپل ، ستون = صفت
- **Dissimilarity Matrix:** فاصله بین تاپل‌ها؛ قطر = 0

بر اساس نوع صفت

```
Nominal:        d = (p − m) / p
Sym. binary:    d = (r+s) / (q+r+s+t)
Asym. binary:   d = (r+s) / (q+r+s)
Jaccard binary: J = q / (q+r+s)

q=both 1, r=(0,1), s=(1,0), t=both 0

Example nominal:
x=(قرمز,M,مجرد,تهران)
y=(قرمز,F,متاهل,تهران)
m=2, p=4 → d=(4−2)/4 = 0.5
```

§۸ — Maximal ، Closed ، Apriori ، Frequent Itemsets

تعاریف

```
support(X) = fraction of transactions containing X
Frequent:  support(X) ≥ minsup

support(A ⇒ B)    = support(A ∪ B)
confidence(A ⇒ B) = support(A ∪ B) / support(A)
```

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

رابطه طلایی

```
M ⊆ C ⊆ F
```

اصل Apriori: اگر itemset پرتکرار باشد، همه زیرمجموعه‌هایش هم پرتکرارند.

1. Pass1: شمارش 1-itemsetها → L1
2. از L(k−1) کاندیدهای Ck بساز
3. با اصل Apriori هرس کن
4. support را بشمار → Lk
5. تکرار تا Lk تهی شود

PCY: در Pass1 جفت‌ها را hash به bucket کن. در Pass2 جفت فقط وقتی کاندید است که هر دو item frequent باشند و bucket هم frequent باشد.

CHARM: استخراج Closed با tidset intersection.

§۹ — خوشه‌بندی Clustering

هدف: فاصله درون‌خوشه‌ای کمینه، فاصله بین‌خوشه‌ای بیشینه. بدون سرپرست است.

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

K-Means

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

ضعف‌ها: نیاز به k ، حساس به init و نویز ، خوشه تقریباً 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
- CLARANS: جستجوی تصادفی همسایه‌ها

Linkage	فاصله دو خوشه
Single	کمینه فاصله اعضا
Complete	بیشینه فاصله اعضا
Average	میانگین همه زوج‌ها
Centroid	فاصله مراکز

DBSCAN / BFR / CURE

```
DBSCAN: Eps , MinPts
  Core / Border / Noise

BFR (large Euclidean):
  N , SUM , SUMSQ
  DS / CS / RS

CURE: several representative points per cluster
      (good for non-spherical shapes)
```

مثال یک تکرار K-Means

```
Points: A(1,1), B(1,2), C(4,4), D(5,4)
Init:   μ1=A(1,1) , μ2=C(4,4)

Assign: C1={A,B} , C2={C,D}
Update: μ1=(1, 1.5) , μ2=(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: ساخت امضای کوتاه از مجموعه‌های بزرگ طوری که احتمال برابر شدن hash دو مجموعه برابر Jaccard باشد.

```
Pr[ h(C1) = h(C2) ] = J(C1, C2)
```

ب) Collaborative Filtering: روش توصیه‌گر مبتنی بر مشارکت؛ با یافتن کاربران/آیتم‌های مشابه، آیتم مناسب پیشنهاد می‌شود (مثل Netflix و Amazon).

ج) Jaccard Similarity:

```
J(A,B) = |A ∩ B| / |A ∪ B|
```

د) Shingle: دنباله k توکن متوالی در سند؛ برای مقایسه near-duplicate استفاده می‌شود.

سوال ۲ (۲ نمره) — Cluster در برابر RDBMS

RDBMS سنتی: مقیاس غالباً عمودی و گران، داده ساخت‌یافته، ACID قوی، سرورهای قدرتمند؛ در حجم خیلی بالا محدود است.

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

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

```
U = M × V

1) Map:    emit(i, m_ij × v_j) for each m_ij
2) Shuffle: group by i
3) Reduce: u_i = Σ values for key i
```

اگر V در حافظه جا شود همه Mapperها آن را دارند. مثال عددی در §۵.

سوال ۴ (۲ نمره) — TF-IDF

```
N=1024 , n(w)=128 , f(w,j)=10 , max=100

TF     = 10/100 = 0.1
IDF    = log2(1024/128) = log2(8) = 3
TF-IDF = 0.1 × 3 = 0.3
```

پاسخ نهایی: 0.3

سوال ۵ (۲ نمره) — Apriori با MST = 0.3

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

آستانه و Passها

```
minsup = 0.3 × 8 = 2.4  ⇒  need count ≥ 3
(because 2/8 = 0.25 < 0.3)

Pass1 counts:
a=4, b=5, c=5, d=4, e=3, f=3
L1 = {a,b,c,d,e,f}

Pass2 frequent pairs:
ab=4, ac=3, bc=4, de=3, df=3
L2 = {ab, ac, bc, de, df}
(cd=2, ce=2, ef=2 are NOT frequent)

Pass3:
abc = 3  → frequent
{def} not candidate (ef not in L2)
L3 = {abc}

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

چک‌لیست ۳۰ ثانیه‌ای: ۱) تعریف یک‌خطی؟ ۲) فرمول؟ ۳) مثال عددی؟ ۴) تفاوت با مفهوم نزدیک؟