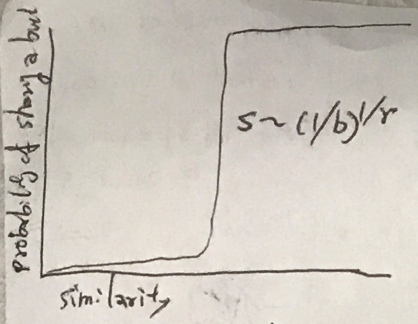
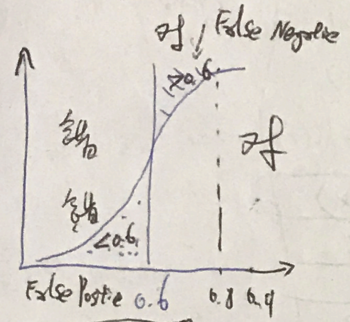


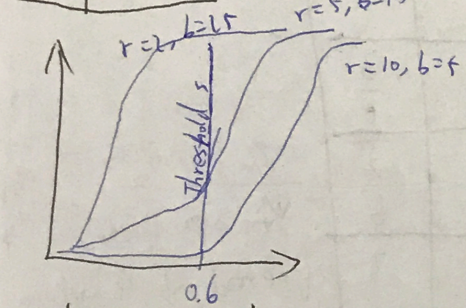
$$1-(1-t^r)^b$$

- at least one band identical → $1-(\ldots)^b$
- no band identical → $(1-t^r)^b$
- some row of a band unequal → $1-t^r$
- all rows of a band are equal → t^r



s	$1-(1-s^r)^b$
0.2	0.006
0.3	0.47
4	0.186
5	0.470
6	0.802
7	0.975
8	0.9996

→ at least one band is identical



LSH ⇒ Tune M, b, r to get
① almost all pairs with similar signatures
② eliminate most pairs that do not have similarity

Data Stream

Sampling a fixed proportion

固定大小的 sample

$\frac{s}{n}$ ⇒ s sample size, n items

↓ 存储比率

Reservoir Sampling

Elements $X_1 \ldots X_i \ldots$

1. Store all first elements $X_1 \ldots X_s$
2. Suppose element X_n arrives
   - with probability $(1-s/n)$, ignore this element
   - with probability s/n:
     - → Discard a random element from the reservoir
     - → Insert element X_n into the reservoir

X_{n+1} is sampled with probability $s/n+1$

过去在 sample 的 → s/n stay in the sample with probability $n/(n+1)$ 已在储里的 rate

Hence, in the sample with probability $\frac{s}{n+1}$ 新rate

element X_{n+1} arrives what is the probability that an already-sampled element X_i stays in the sample

$$\left(1-\frac{s}{n+1}\right)+\left(\frac{s}{n+1}\right)\cdot\left(\frac{s-1}{s}\right)=\frac{n}{n+1}$$

X_{n+1} not sampled; X_{n+1} sampled; X_i not evicted

① 数据流很长，无法存入内存
② O(N)
③ 随机 m 个数，每个数选中概率为 m/N

Morris Algorithm

$X \leftarrow 0$

for each of the n events

$X \leftarrow X+1$ with $(1/2)^X$

Return $n' = 2^X + 1$

Counter X needs only $\log_2(n)$ bits

Let $X(n)$ denote count after arrival n. Transition $X \to X+1$ with probability $2^{-X}(1-1/2^X)$

$0 \xrightarrow{1} 1 \xrightarrow{1/2} 2 \xrightarrow{1/4} \cdots \xrightarrow{1/2^{X-1}} X \xrightarrow{1/2^X}$

Unbiasedness

initialize $x=0$;
increment w.p. $p=2^{-X}$;
estimate $n'=2^X-1$

n=1
before: $X=0$, $P_0=1$
Prob. 1: $X \to 1$
estimation $n'=2^1-1=1=n$

n=2
before: $X=1$, $P_1=1/2$
prob. 1/2: X stays at 1, $n'=2^1-1=1$
Prob. 1/2: $X \to 2$; $n'=2^2-1=3$

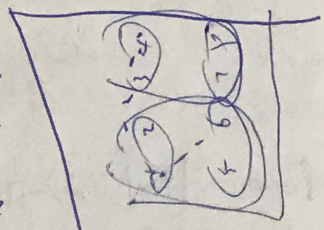
Hierarchical Clustering (Top-Down) Bisecting k-Means

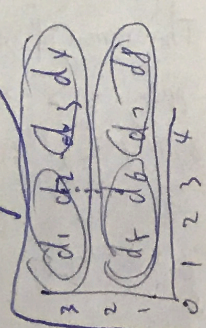
For I=1 to k-1 do
  Pick a leaf cluster C to split
    For J=1 to ITER do
      use k-means to split C into two sub-clusters, C_1 and C_2
    Choose the best of the above splits and make it permanent

Bottom Up Clustering

key operation: Repeatedly combine two nearest clusters

3 important questions:
① How to represent a cluster of more than one point
② How do you determine the "nearness" of clusters
③ When to stop combining clusters





Single

$$sim(c_i,c_j)=\max_{x\in c_i,\, y\in c_j} sim(x,y)$$

$$sim((c_i\cup c_j),c_k)=\max(sim(c_i,c_k),sim(c_j,c_k))$$

Complete link

$$sim(c_i,c_j)=\min_{x\in c_i,\, y\in c_j} sim(x,y)$$

$$sim((c_i\cup c_j),c_k)=\min(sim(c_i,c_k),sim(c_j,c_k))$$

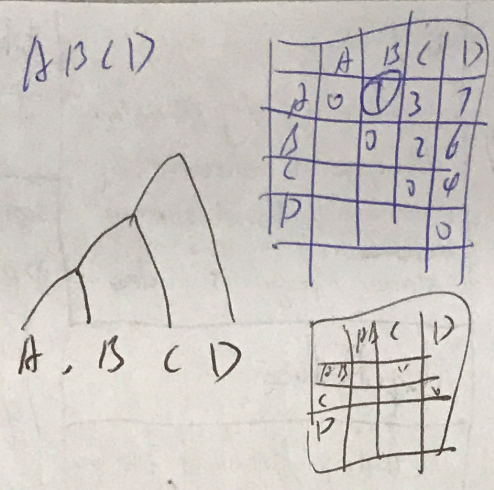
Mid-term

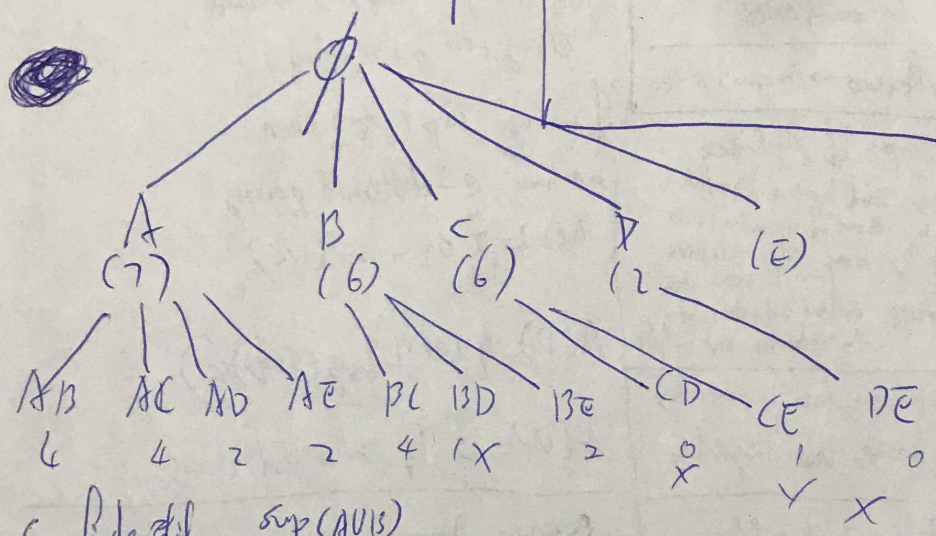
① Rule with 100% confidence are not interesting people want to have no thing

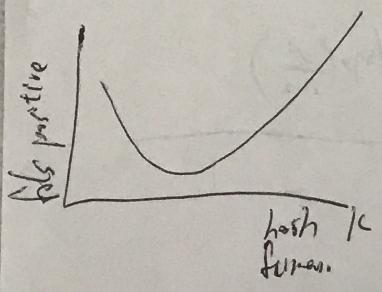
②

	Movie ID
1	ABE
2	AD
3	BC
4	ABD
5	AC
6	BC
7	AC
8	ABCE
9	ABC

	A	B	C	D	E
1	1	1			1
2	1			1	
3		1	1		
4	1	1		1	
5	1		1		
6		1	1		
7	1		1		
8	1	1			1
9	1	1	1		







handle deletion

use counters instead of 0/1s

when adding an element, increment the counters

when deleting an element, decrement the counters

counters must be large enough to avoid overflow/underflow

$$\underbrace{1-(1-1/n)^{n(m/n)}}_{\text{some target X not hit by a dart}} < \underbrace{1-e^{-m/n}}_{\text{prob. at least one dart hit X}}$$

false positive probability $\approx 1-e^{-km/n}$

Bloom filter operations

```
Insert(k)
    for i from 1 to r
        B[h_i(k)] <- 1

IsMember(k)
    for i from 1 to r
        if B[h_i(k)] == 0
            return false
    return true
```

why Visualization

① Record

② Analyze data to support reasoning

③ Communicate

Complex ideas communicated with clarity, precision, and efficiency

substance, statistics, Design

Tufte's Principles

Class Design Principles:

- no cute, on fours data
- Have a good explanatory_clear title
- Data integrity: maintain scale, right axes
- know your goal
- Balance high-level with detail

Lie Factor = $\frac{\text{size of effect shown in graphic}}{\text{size of effect in data}}$

① Maxize data-ink ratio

② Erase non-data-ink/redundant

③ Revise and edit

Data-ink ratio = $\frac{\text{data-ink}}{\text{Total ink used to print graphic}}$

argue for every pixel

starting point: erase non-data ink
erase redundant ink

encoding data with shapes, color, and size. which cues to chose depends on your data and your goals

logarithmic → log → percent change

correlation	distribution
clustering	outliers

Good story

introduce interesting characters

challenge is believable

There are hurdles to overcome

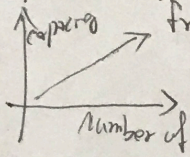
~~there are~~

outcome or prognosis is clear

MapReduce

The Hadoop Distributed File System (HDFS)

MapReduce → A distributed processing framework



MapReduce → Programing Model

3 steps of MapReduce

① Map: Read input and produce a set of key-value pairs

② Group by key: collect all pairs with same key

③ Reduce: collect all values belong to the key and output

spark process data in-memory

spark better than mapReduce

MapReduce 2 limitation

- Difficulty of Programing directly in MR
- performance bottleneck, or batch not fitly.

Spark → RDD

Resilient Distributed Dataset (RDD)

- across the cluster. Read-only
- caching database in memory

RDD → part of core Sparle

Amazon Dynamo

一个物理节点 → 多个虚拟节点

replica

sensitivity

PDF: $f(x|\mu,b)=\frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right)$

mean 0 variance $2b^2$

$|q(D)-q(D')| \le s(q)$

Example. Count query

Number of people having disease

sensitivity = 1

"differing in one row" → could change one by max of 1)

if we use $\text{lap}(1/\varepsilon)$ then we have ε-differential privacy

$A(D)=\sum_{i=1}^{n} d_i + \text{lap}(1/\varepsilon)$

$A(D)=q(D)+\text{lap}(s(q)/\varepsilon)$

$f(D)=\sum_{i=1}^{n} d_i$, where each $d_i \in \{0,1\}$

example. Average query

Sensitivity = M/n

$A(D)=\frac{1}{n}\sum_{i=1}^{n} d_i + \text{lap}\left(\frac{M}{n\varepsilon}\right)$

Statistical Data Privacy

目标：个人信息被保护

使用目标：统计数据有用

① Anonymization → linkage attacks

② Query auditing → 拒绝query就是泄密了

③ Summary statistics → differencing attack

可以offline/online来检测

回答request ← 是否回答request → 是否可信

DIFFERENTIAL PRIVACY

不论是否加入了db，统计结果无法查到个人

(Perturbed with differential privacy)

$$\log\left(\frac{Pr[A(D_1)=0]}{Pr[A(D_2)=0]}\right) < \varepsilon \quad (\varepsilon > 0)$$

不论什么record加入，都不变

$$Pr[A(D_1)=0] \le e^{\varepsilon} Pr[A(D_2)=0]$$

控制 D_1，D_2 的分别程度

ε 越小，越 privacy

$$\frac{Pr[A(D_1)\in Y]}{Pr[A(D_2)\in Y]} \le \exp(\varepsilon)$$

$\ge \exp(-\varepsilon)$

对于 ε 很小的情况

$1-\varepsilon \le \sim \le 1+\varepsilon$

Laplace: $f(x \mid \mu, b) = \frac{1}{2b}\exp\left(-\frac{|x-\mu|}{b}\right)$

Theorem: if sensitivity of the query is S, then the algorithm $A(D) = q(D) + Lap(S(q)/\varepsilon)$ 确保 ε-differential privacy

结果为 COUNT → 加一个值，可：

$$A(D) = \underbrace{\sum_{i=1}^{n} d_i}_{f(D)} + Lap(1/\varepsilon)$$

Sensitivity $= M/n$，$A(D) = \frac{1}{n}\sum_{i=1}^{n} d_i + Lap\left(\frac{M}{n\varepsilon}\right)$

$M_1 \ldots M_k$ 都是对 D 的算法，满足 ε_i-differential privacy

那么 $\varepsilon = \varepsilon_1 + \varepsilon_2 + \ldots + \varepsilon_k$

结果结合

Sequential Composition

$M_1 \ldots M_k$ 是 $D_1 \ldots D_k$ 的算法，为 $\varepsilon_1 \ldots \varepsilon_k$，则加起结果为 $\max\{\varepsilon_1 \ldots \varepsilon_k\}$

$M_1 \to D$ 为 ε differentially private

$M_2(M_1(D))$ 还是为 ε

K-Mean with Differential Privacy

$\varepsilon/T \Rightarrow \varepsilon$ 如果T iteration

1. 分配点

2. Noisily 计算 cluster 大小

3. 计算 noisy sum

$$sim(C_1, C_2) = \frac{|C_1 \cap C_2|}{|C_1 \cup C_2|}$$

Jaccard distance $= 1 - sim$

无法简单比较：① 很多照片，很难相似 ② 太多文件 ③ 内存不够

documents as sets of shingles

→ Why shingling ① set of terms 没有顺序 ② 要表达单词太多

↓

为什么要 shingling? 10个也比较 $N(N-1)/2 \approx 5 \times 10^{11}$

把 docs 塞进 bucket，相似的 docs 会塞进相似的 bucket

random permutation π

$h_\pi(C) = \min_\pi \pi(C)$

Similarity

LSH

Single-pass method

```
for each row r
    for each hash function hi
        compute hi(r)
    for each column c
        if c has 1 in row r
            for each hash function hi
                if hi(r) < M(i, c)
                then M(i, c) <- hi(r)
```

Row	C_1	C_2	$h(x)$	$g(x)$
1	△1	0	1	3
2	0	△1	2	0
3	△1	1	3	2
4	△1	0	4	4
5	0	△1	0	1

$h(x) = x \bmod 5$

$g(x) = (2x+1) \bmod 5$

initial	$M(i, C_1)$ Int. Max / Int. Max	$M(i, C_2)$ int. Max / int. Max
$h(1)=1$, $g(1)=3$	1, 3	
$h(2)=2$, $g(2)=0$	1, 3	2, 0
$h(3)=3$, $g(3)=2$	1, 2	2, 0
$h(4)=4$, $g(4)=4$	1, 2	2, 0
$h(5)=0$, $g(5)=1$	1, 2	0, 0

LHS

Candidate pair

Similarity threshold $(0 < s < 1)$

1. Hash columns of signatures matrix M 多次

把M 横切 b份 每份 r行

→ 每份，hash 去 hash table with k bucket. k 越大越好

调 b 和 r，让 找到 pair

Same bucket means "identical in the band"

① Find Pair of $\ge s = 0.8$, $b = 20$, $r = 5$

② Probability C_1, C_2 identical in one particular band: $(0.8)^5$

一个band 完全一样 $= 0.328$

③ all 20 band not similar $(1-0.328)^{20} = 0.00035$

完全没机会的 rate ↗

1/3000th 是 false negative

find 99.965% pair 真相似

④ → ~~~~

if sim$(C_1, C_2) = 0.3$

$0.3 < 0.8$

小于s，所以希望不要映射到 common bucket

$(0.3)^5 = 0.00243$

$(1-0.00243)^{20} = 0.047$ 一个都不一样的概率

4.7% 成为 candidate pairs

↓ false positives

if we had only 15 bands/5 rows，false positive ↓，false negative ↗

① all rows in band equal $= t^r$

② some row in band unequal $= 1-t^r$

③ no band identical $= (1-t^r)^b$

④ 1 band identical $1-(1-t^r)^b$