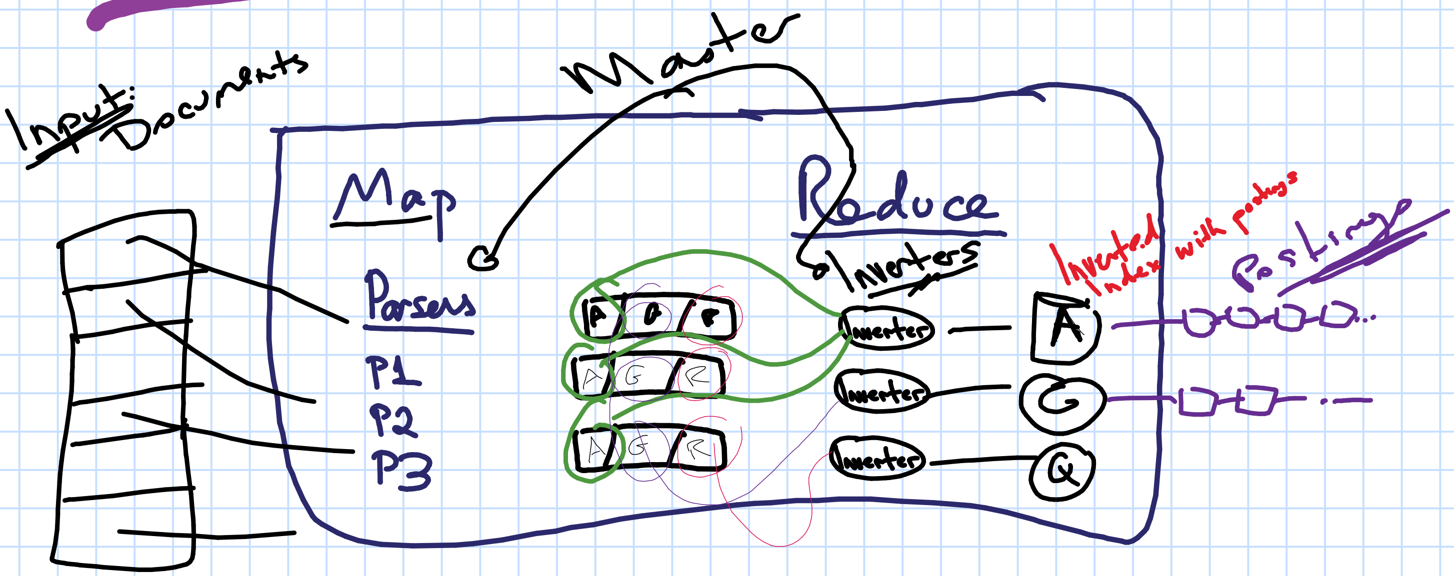
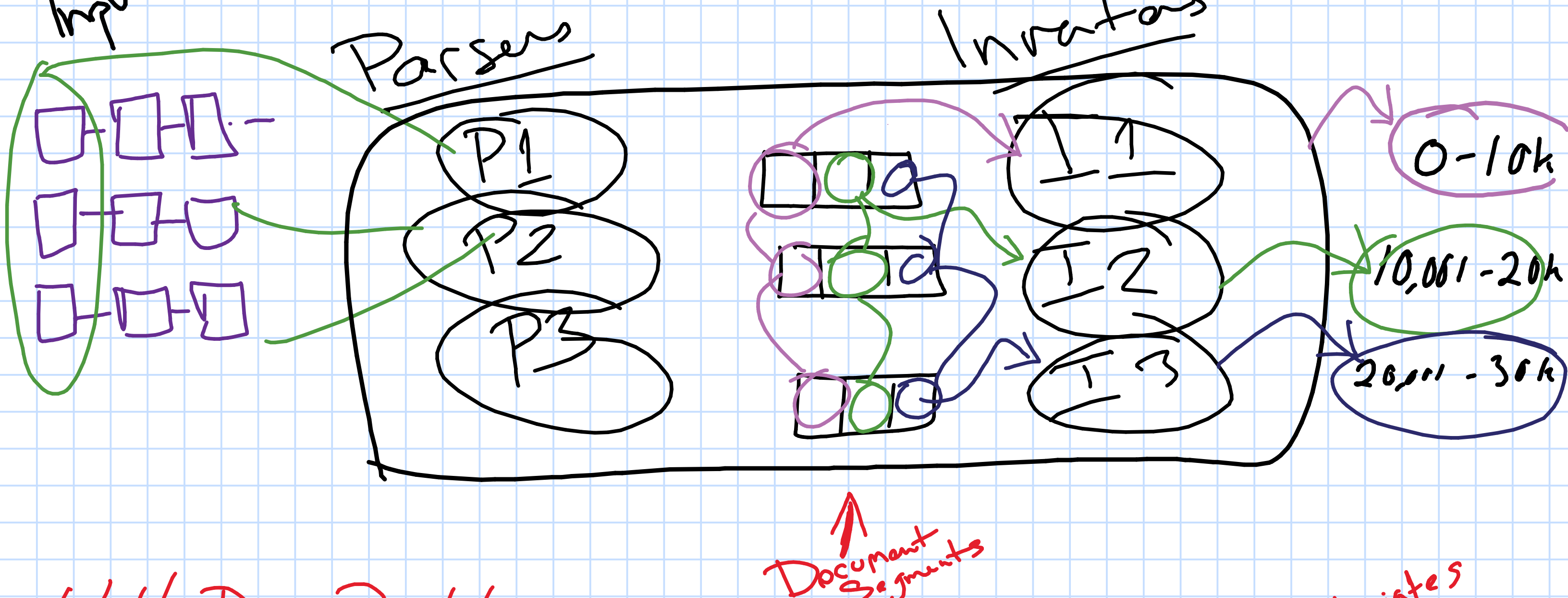
Distributed Index Generation (MAP-REDUCE)

Phase 1: Build the Term-Partition Index



Phase 2: Build Document Partion Index

Input: Term-Partitioned Postings (Output from Phase 1)

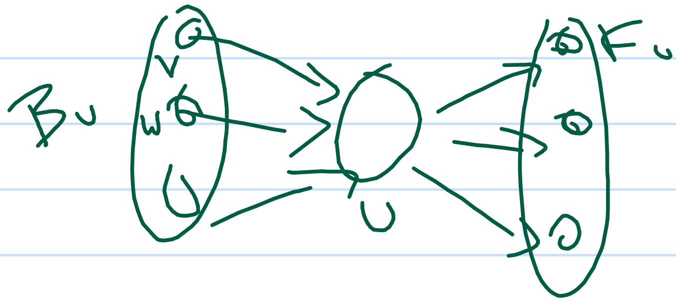


With Doc Partitioning, the entire query is sent to every node which ~~reduces~~ alleviates the doc list merging that takes place with Term-Partition indexes.

Nov 1 EXAM Result

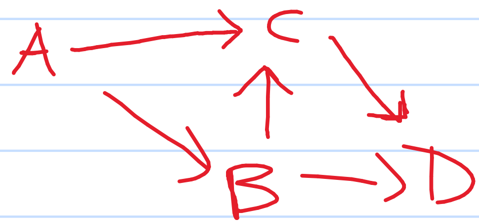
Use IDF for Query Calculation

Nov 5 — Talked to Dr. D., the quantum indicated which fmla to use
∴ "do not normalize the query" means to leave it out of the Cosine Similarity fmla altogether.



$$\frac{R(v)}{4} + \frac{R(w)}{2}$$

Build an Adjacency Matrix



1

$$\begin{array}{c|cccc} & A & B & C & D \\ \hline A & 0 & 1 & 1 & 0 \\ B & 0 & 0 & 1 & 1 \\ C & 0 & 0 & 0 & 1 \\ D & 0 & 0 & 0 & 0 \end{array}$$

Since D has no out links just assign the Prob. that they jump to any node.

$$P_r = \frac{1}{N} = \frac{1}{4} = 0.25$$

$$P \quad \begin{pmatrix} & & & \\ & & & \\ & & & \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$$

Then fill in the other rows based on link probabilities

$$\begin{pmatrix} 0 & .5 & .5 & 0 \\ 0 & 0 & .5 & .5 \\ 0 & 0 & 0 & 1 \\ .25 & .25 & .25 & .25 \end{pmatrix}$$

Prob of user following a link

then... $P = (1-\alpha)P$ (for all but last row)

$+$

2

$(\alpha)\left(\frac{1}{N}\right)$ (except last row b/c already calculated

③ P.R.

$\vec{x}_0 = (1,0,0,0)$ ← can choose any starting vector.

$\vec{x}_1 = (\vec{x}_0)(P)$

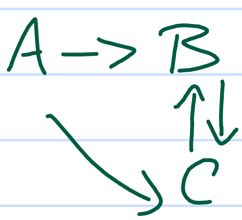
$= (.05, 0.45, .45, .05)$

$\vec{x}_2 = \vec{x}_1 \cdot P$

$\vdots$

Until the Δ is very small. -- then we have P.R.

Nov 15



① Adj Matrix

$$\begin{array}{c} A \\ B \\ C \end{array} \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

(A) PageRank

- all have outlinks!, unlike last class's example

$Pr = 1/\text{sum}(1\text{'s})$

$$\begin{pmatrix} 0 & 1/2 & 1/2 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$ ← Pr of user following a link

α, say, 0.1
So P: 0.9

$$\Rightarrow \begin{pmatrix} 0 & .45 & .45 \\ 0 & 0 & .9 \\ 0 & .9 & 0 \end{pmatrix}$$

Cont'd...

Pr of teleport: → $1/3$

$$Pr(Link) + (1-\alpha)\,1/3$$

$$= \begin{pmatrix} .03 & .48 & .48 \\ .03 & .03 & .93 \\ .03 & .93 & .03 \end{pmatrix}$$

Finally, choose $\vec{X}_0$

$\vec{X}_0 = (1, 0, 0)$

$\vec{X}_1 = \vec{X}_0 \cdot P$

$\vec{X}_2 = \vec{X}_1 \cdot P$

⋮

Until convergence

⋮

or after so many steps.

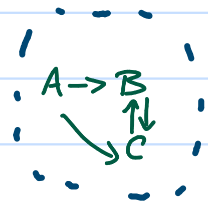
Hits Algorithm

Need Hub score & auth score!

1. Initial vectors:

$\vec{h}_0 = (1, 1, 1)$

$\vec{a}_0 = (1, 1, 1)$



2. Check in-links. — B+C have 2 in-Links

A has none ∴ →

$\vec{a}_1 = (0, (1+1), (1+1))$

$= (0, 2, 2)$

Normalize: → Make so that the MAX value is 1

$a_1 = (0, 1, 1)$ $[0, 2/2, 2/2]$. divide by largest

Hub Score: Sum of Auth. Score:

$h_1 = (2, 1, 1)$

Sum of outbound nodes' auth scores

$h_{1_{NORMAL}} = (1, .5, .5)$

$a_2 = (0, 1, 1)$ – converged, so stop.

* authority score subsequently is sum of inbound hub scores.

Nov 19/13

Recommender Sys Cont'd

Movies

	M1	M2	M3	...
Alice	3	2	5	
Bob	4	5	2	
C	∅	3	4	
⋮				

empty set → hasn't seen

Recommend movies to users
or users to movies } diff results!

Content-based

Utility $u(c,s)$;

$u(c, s_i) \mid s_i \in S$

similar

User $\vec{w_c}$ - of w_{ck} ... keywords .. Need to determine these!

avg rating of...

Content $\vec{w_s}$

directors
animation
drama

Need to do Normalization

(profile)
Bob: $(.9, .4, .2, 0, .3, .9)$

Predict Rating for "UP"

MDD: $(4+5)/2 = 4.5 \rightarrow$ Normalize $\rightarrow 0.9$

$2/5 \rightarrow 0.4$ } [1]
$1/5 \rightarrow 0.2$ } [2]

Up: $(0,0,0,1,1,0)$

Sim(Bob, Up)

$= 0.3 \times 1 = 0.3$

(1.8) ✓

Dark Knight $= (1,0,0,0,0,1)$

User-based CF

Alice avg(3,2,5,4) = 3.5
Bob avg(4 5 2 1) = 3

} only commonly rated } Pearson Coefficient fmla

m1 m2 m3

$(3-3.5)(4-3) + (2-3.5)(5-3) + \cdots = -6$

A rating A's avg

$$\frac{-6}{\sqrt{\sum \text{component}^2\text{s}}} \rightarrow \text{always } -1 \text{ to } 1$$

?

$\therefore = -0.85$

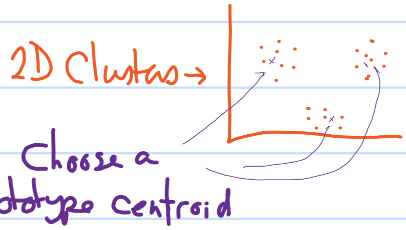
highest is best!

$$\text{Rating}_{\text{BOB}_{\text{UP}}} = 1/\left(\sum (\text{highly similar People})\right) * \left(\sum (\text{SimPerson})(\text{SimPerson Rating})\right)$$

Item-based

Nov 22/13

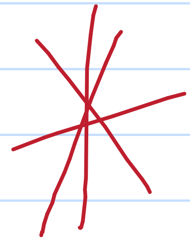
Clustering



K-mean

$$\vec{\mu}(c) = \frac{1}{|c|}\sum_{\vec{x}\in c}\vec{x}$$

- Select k docs randomly as centroids

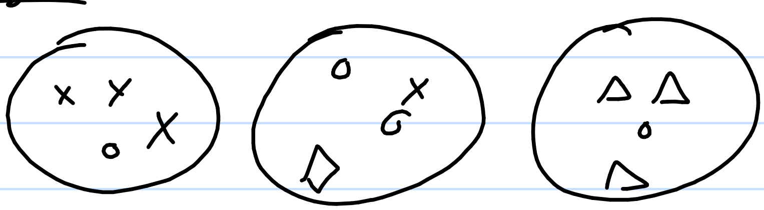


Hierarchical

- at the bottom level, every object in its own cluster; same at the top.

Initial centroid selection affects the resulting clusters.

Purity Ex

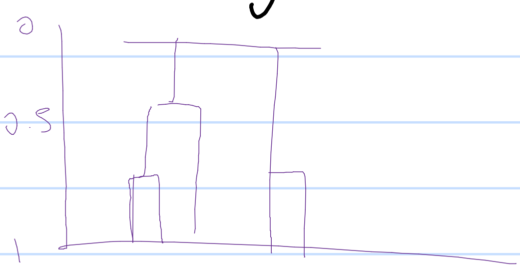


$$\text{Purity} = \frac{\sum(\text{majority items in each circle})}{N}$$

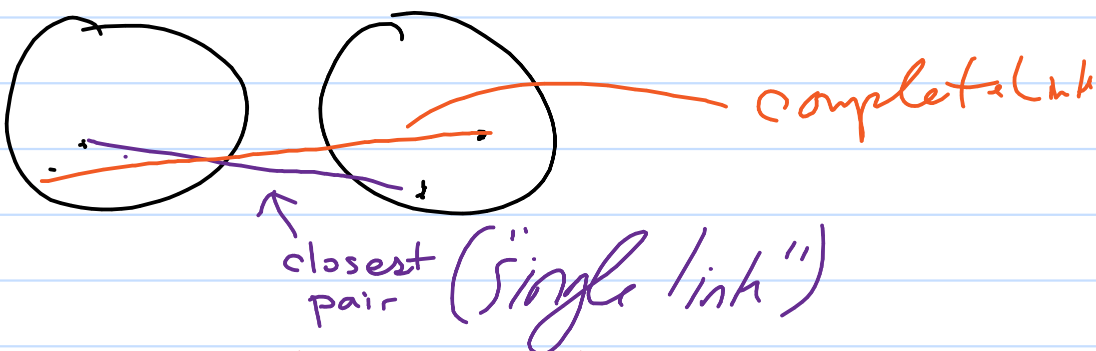
Rand Index = $\frac{TP+TN}{TP+FP+FN+TN}$

HAC

- Repeatedly joins two clusters until there is only 1.
- A Dendogram



Sim



3 - centroid. (4)

4. Avg (6 links)

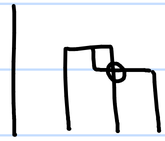
Single Link Clustering — find closest cluster by measuring from inner side of clusters.

- dist betw. clusters is closest pairs
- group closest things repeatedly
- results in undesirable long chains

Complete Link
- Longest dist

- measure from outer edge
- better balance; instead of having lots of clusters with only 1 item
- downside: outliers screw up the grouping

Inversion in Dendogram



don't use centroid HAC for this reason

Q3 HW

	T1	T2	T3
D1	.	.	.
⋮	⋮	⋮	⋮
D5	.	.	.

K-Means first

1) k=2; centroids d1, d2

decide the rest; which cluster....

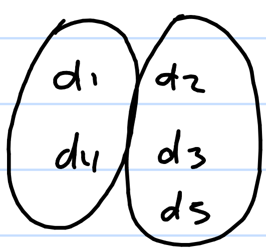
d3 : $d_{1,3}$ = [inner product] 1.0 + 0.9 + 0.4 = ∅

$d_{2,3}$ = 0.8 d2 ✓

d4 : $d_{1,4}$ = .9 ✓ d1

$d_{2,4}$ = .3

d5 : $d_{1,5}$ = 0.6 ; $d_{2,5}$ = 0.98 d2



1st Iteration

2) Now calc new centroids

C_1 = avg(d1, d4) = (0, 0.95, 0.2)

C_2 = ... (0.83, 0.23, 0.37)

Calc distances again:

$d1_{c_1}$ = ((.9)(.95) + 0.08) = .935 ?

$d1_{c_2}$ = (.9)(.23) + (.4)(.3)) = 0.3 ? (smaller)

do calcs for the rest of the docs to put in either C_1 or C_2

don't include the centroid in subsequent iterations.

HAC EX!

	d_1	2	3	4	5
d_1	.				
2		.		.98	
3			.		
4				.	
5					.

← similarity scores
pairwise similarity "inner prod"
← Same on diagonal

$d_{25} = 0.98$ — just do 1 corner of Matrix
↑
1st cluster

$d_2 \ d_5$

$d_{25} \ d_1 \ d_3 \ d_4$

Find → after Midterm, focus on WebSearch.

$sim(d_1, d_{25}) = ?$ → use Complete Link
→ farthest
→ Least similar!
→ smaller → 0.47

$d_3, d_{25} = 0.7$
$d_4, d_{25} = 0.3$

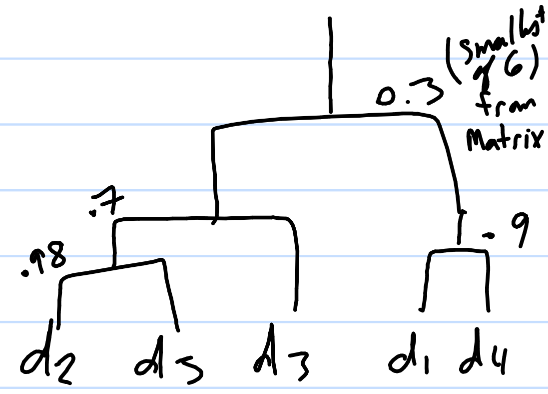
Then find most similar to cluster → d_1 & d_4 (b/c 0.9)

Now have: $d_2 \ d_5 \quad d_1 \ d_4$

$d_{25} \ d_{14} \ d_3$ ← .7 ← 0

	d_1	d_4
d_2	.47	.3
d_5	.6	.4

pick lowest → 0.3



Can calc top-most score too → 0.3

For Single Link; still start at d_{25} (0.98), but now choose highest scores.

Exam

"Item-Based" CF; same as assignment (the values anyway)
- avg of items rather than users.
- basically the vector will be in columns instead of rows
- Finds most similar ITEMs instead of people.

- Know K-means & hierarchical

Review

- Lectures after MID.
- UI -> qry in; qry results; qry reformulation
- Docs & Queries -> Zipf  ; Word freq.

Compression:
- Decompression speed is most important!

Query Intent (Broder): navigational, informational, transactional

NB Web Search ("Focus on this")

NB

1. W.search Basics

Key differences: links, queries/content, users/docs, spam, advertisements

· how to estimate Index size

· " " detect near duplicates

· ranking signals -> 1 Content 2 Link (PR) 3 Usage [clicks]

2. Crawler

· must have features: robust, politeness | Should-have: efficient, etc

- Crawl process: seed set -> fetch -> parse -> extract links & text, dupl. check -> URL frontier
- Scheduling

Architecture ("Mercator")

3. Link Analysis => PR, Hits

Recommender

- Content-based, collab. filtering
    - memorize fmla's
    - Know limitations of each
    -

Clustering

K-means: ① Initial seeds ② Recalc centroids ③ Iterate till objective function is optimized

Hierarchical Aggl. Clust. (HAC)

- group repeatedly till only 1 cluster.
    - Single link (measure from near side)
    - complete (measure from far side)

- Evaluation - ext criteria → purity, rand. index.