

Dimensionality Reduction, Clustering and Trends in NLP

Kan Min-Yen
Day 3 / Afternoon

Outline

Dimensionality Reduction

- **Synonymy and Polysemy**
- **Bit of Linear Algebra**
- **Latent Semantic Indexing**

Clustering

- **Partitional**
- **Hierarchical**

Wrapping Up: Trends in NLP

- **Genre Treatments**
- **Multilingual Treatments**

References



Problems with Lexical Semantics

- **Ambiguity and association in natural language**
 - **Polysemy**: Words often have a **multitude of meanings** and different types of usage (*more severe in very heterogeneous collections*).
 - The vector space model is unable to discriminate between different meanings of the same word.

$$\text{sim}_{\text{true}}(d, q) < \cos(\angle(\vec{d}, \vec{q}))$$

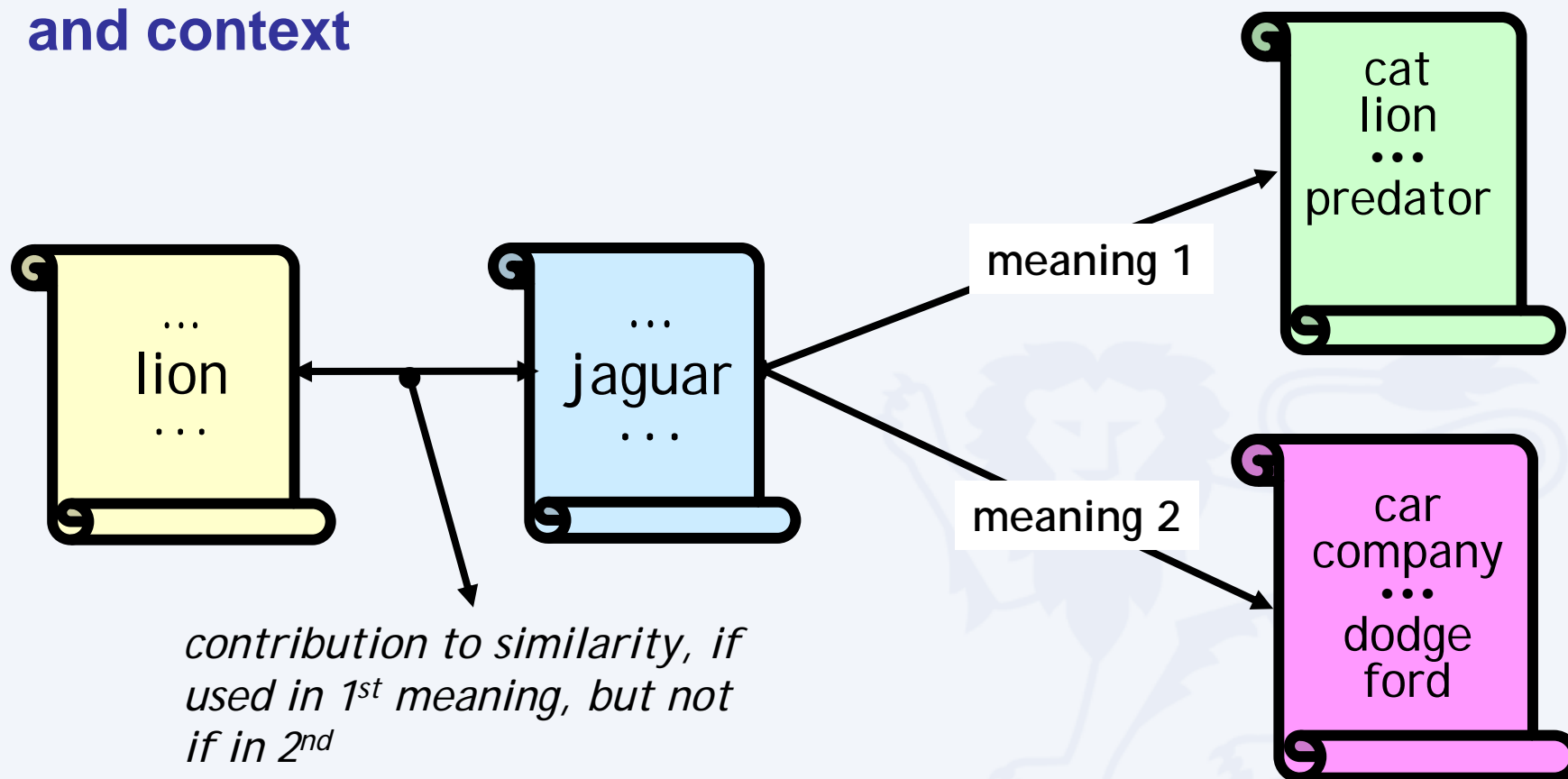
Problems with Lexical Semantics

- Synonymy**: Different terms may have an **identical or a similar meaning** (weaker: words indicating the same topic).
- No associations between words are made in the vector space representation.

$$\text{sim}_{\text{true}}(d, q) > \cos(\angle(\vec{d}, \vec{q}))$$

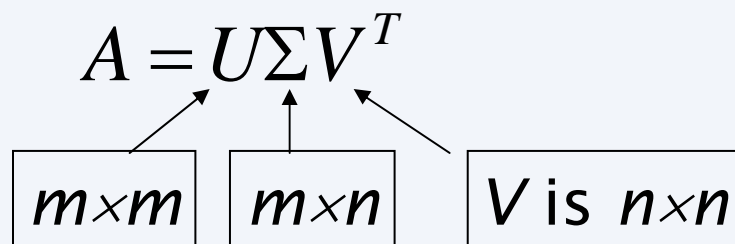
Polysemy and Context

- Document similarity on single word level: polysemy and context



Singular Value Decomposition

For an $m \times n$ matrix \mathbf{A} of rank r there exists a factorization (Singular Value Decomposition = **SVD**) as follows:

$$A = U \Sigma V^T$$


$m \times m$

$m \times n$

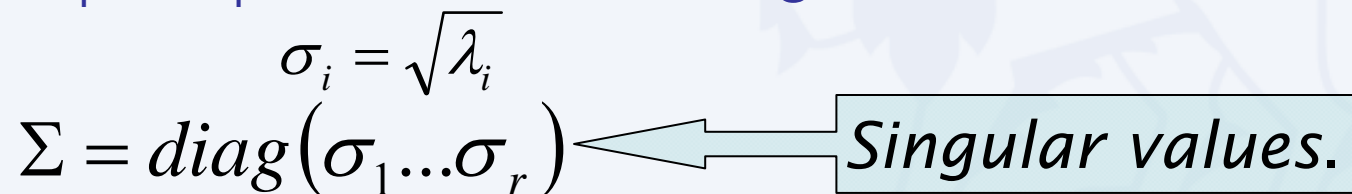
$V \text{ is } n \times n$

The columns of \mathbf{U} are orthogonal eigenvectors of $\mathbf{A}\mathbf{A}^T$.

The columns of \mathbf{V} are orthogonal eigenvectors of $\mathbf{A}^T\mathbf{A}$.

Eigenvalues $\lambda_1 \dots \lambda_r$ of $\mathbf{A}\mathbf{A}^T$ are the eigenvalues of $\mathbf{A}^T\mathbf{A}$.

$$\sigma_i = \sqrt{\lambda_i}$$

$$\Sigma = \text{diag}(\sigma_1 \dots \sigma_r)$$


Singular values.

Singular Value Decomposition

- Illustration of SVD dimensions and sparseness

$$\underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_A = \underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_U \underbrace{\begin{bmatrix} \bullet & & & & \\ & \bullet & & & \\ & & \bullet & & \\ & & & \bullet & \\ & & & & \bullet \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{V^T}$$

(Note: In the original image, the last two columns of U and the last three rows of Σ are highlighted in yellow.)

$$\underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_A = \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_U \underbrace{\begin{bmatrix} \bullet & & & & \\ & \bullet & & & \\ & & \bullet & & \\ & & & \bullet & \\ & & & & \bullet \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}}_{V^T}$$

(Note: In the original image, the last three rows of Σ and the last five rows of V^T are highlighted in yellow.)

SVD example

$$\text{Let } A = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Thus $m=3$, $n=2$. Its SVD is

$$\begin{bmatrix} 0 & 2/\sqrt{6} & 1/\sqrt{3} \\ 1/\sqrt{2} & -1/\sqrt{6} & 1/\sqrt{3} \\ 1/\sqrt{2} & 1/\sqrt{6} & -1/\sqrt{3} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{3} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

Typically, the singular values arranged in decreasing order.

Low-rank Approximation

- SVD can be used to compute optimal low-rank approximations.
- Approximation problem: Find A_k of rank k such that

$$A_k = \min_{X: \text{rank}(X)=k} \|A - X\|_F \longleftarrow \text{Frobenius norm}$$

- A_k and X are both $m \times n$ matrices.
Typically, want $k \ll r$.

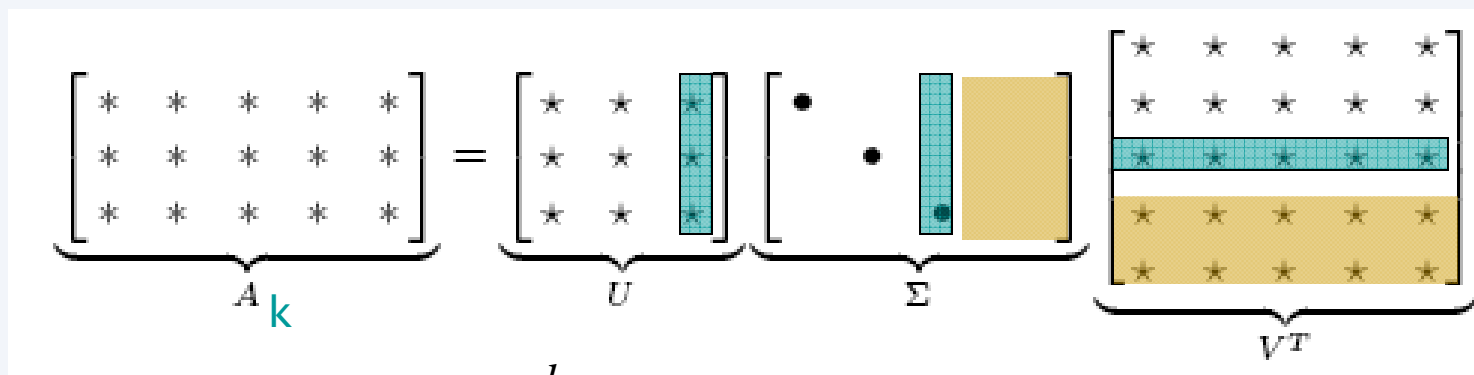
$$\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2}$$

Low-rank Approximation

- Solution via SVD

set smallest $r-k$ singular values to zero

$$A_k = U \text{diag}(\sigma_1, \dots, \sigma_k, \underbrace{0, \dots, 0}_{r-k}) V^T$$



$$A_k = \sum_{i=1}^k \sigma_i u_i v_i^T$$

column notation: *sum of rank 1 matrices*

Approximation error

- How good (bad) is this approximation?
- It's the best possible, measured by the Frobenius norm of the error:

$$\min_{X: \text{rank}(X)=k} \|A - X\|_F = \|A - A_k\|_F = \sigma_{k+1}$$

where the σ_i are ordered such that $\sigma_i \geq \sigma_{i+1}$.

Suggests why Frobenius error drops as k is increased.

SVD Low-rank approximation

- Whereas the term-doc matrix A may have $m=50000$, $n=10$ million (and rank close to 50000)
- We can construct an approximation A_{100} with rank 100.
 - Of all rank 100 matrices, it would have the lowest Frobenius error.
- Great ... but why would we??
- Answer: *Latent Semantic Indexing*



Latent Semantic Analysis (LSA)

- **LSA aims to discover something about the meaning behind the words; about the topics in the documents.**
- **What is the difference between topics and words?**
 - Words are observable
 - Topics are not. They are latent.
- **How to find out topics from the words in an automatic way?**
 - We can imagine them as a combination of words

Goals of LSI

- **Similar terms map to similar location in low dimensional space**
- **Noise reduction by dimension reduction**

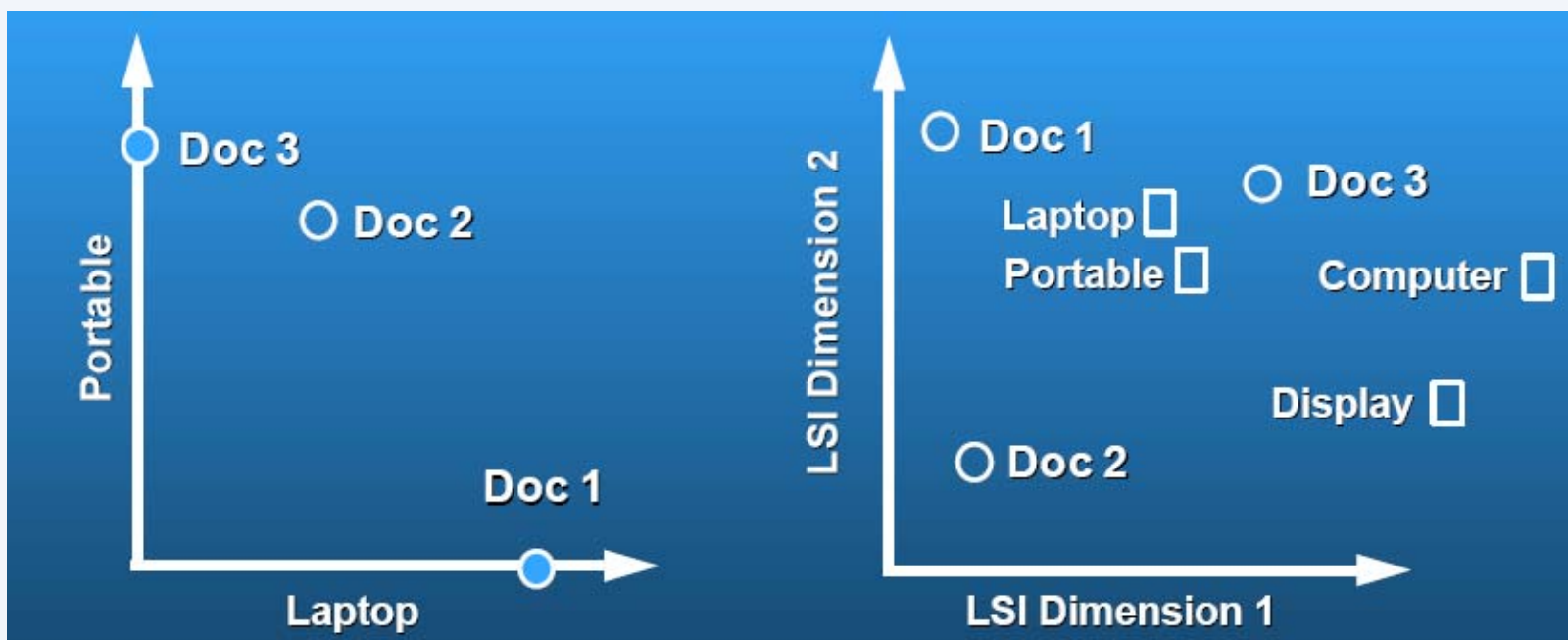


Latent Semantic Indexing (LSI)

- Perform a **low-rank approximation of document-term matrix** (typical rank 100-300)
- **General idea**
 - Map documents (*and* terms) to a **low-dimensional representation**.
 - Design a mapping such that the low-dimensional space reflects **semantic associations** (latent semantic space).
 - Compute document similarity based on the **inner product** in this **latent semantic space**

Latent Semantic Analysis

- Latent semantic space: illustrating example



courtesy of Susan Dumais

Performing the maps

- Each row and column of A gets mapped into the k -dimensional LSI space, by the SVD.
- Claim – this is not only the mapping with the best (Frobenius error) approximation to A , but in fact *improves* retrieval.
- A query q is also mapped into this space, by

$$q_k = q^T U_k \Sigma_k^{-1}$$

–Query NOT a sparse vector.



LSI Example

$m=5$ (interface, library, Java, Kona, blend), $n=7$

$$A = \begin{pmatrix} 1 & 2 & 1 & 5 & 0 & 0 & 0 \\ 1 & 2 & 1 & 5 & 0 & 0 & 0 \\ 1 & 2 & 1 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 3 & 1 \\ 0 & 0 & 0 & 0 & 2 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 0.58 & 0.00 \\ 0.58 & 0.00 \\ 0.58 & 0.00 \\ 0.00 & 0.71 \\ 0.00 & 0.71 \end{pmatrix} \times \begin{pmatrix} 9.64 & 0.00 \\ 0.00 & 5.29 \end{pmatrix} \times \begin{pmatrix} 0.18 & 0.36 & 0.18 & 0.90 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.53 & 0.80 & 0.27 \end{pmatrix}$$

U Δ V^T

- a query $q = (0 \ 0 \ 1 \ 0 \ 0)^T$ is transformed into $q' = U^T \times q = (0.58 \ 0.00)^T$ and evaluated on V^T
- a new document $d_8 = (1 \ 1 \ 0 \ 0 \ 0)^T$ is transformed into $d_8' = U^T \times d_8 = (1.16 \ 0.00)^T$ and appended to V^T

Empirical evidence

- **Precision at or above median TREC precision**
 - Top scorer on almost 20% of TREC topics
- **Slightly better on average than straight vector spaces**
- **Effect of dimensionality:**

Dimensions	Precision
250	0.367
300	0.371
346	0.374



“Arts”

“Budgets”

“Children”

“Education”

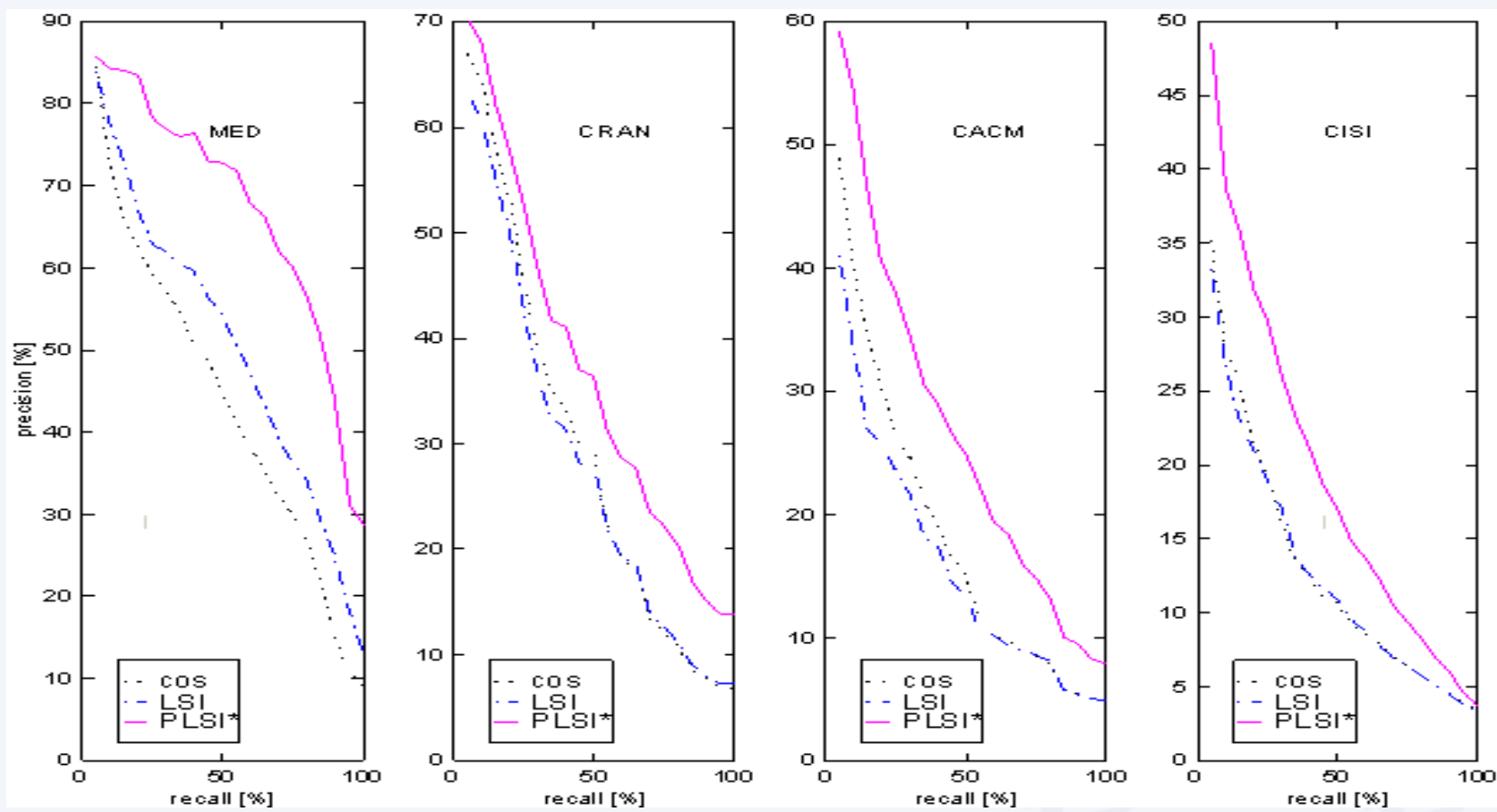
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give \$1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. “Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services,” Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center’s share will be \$200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive \$400,000 each. The Juilliard School, where music and the performing arts are taught, will get \$250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual \$100,000 donation, too.

Example of topics found from a Science Magazine papers collection

universe	0.0439	drug	0.0672	cells	0.0675	sequence	0.0818	years	0.156
galaxies	0.0375	patients	0.0493	stem	0.0478	sequences	0.0493	million	0.0556
clusters	0.0279	drugs	0.0444	human	0.0421	genome	0.033	ago	0.045
matter	0.0233	clinical	0.0346	cell	0.0309	dna	0.0257	time	0.0317
galaxy	0.0232	treatment	0.028	gene	0.025	sequencing	0.0172	age	0.0243
cluster	0.0214	trials	0.0277	tissue	0.0185	map	0.0123	year	0.024
cosmic	0.0137	therapy	0.0213	cloning	0.0169	genes	0.0122	record	0.0238
dark	0.0131	trial	0.0164	transfer	0.0155	chromosome	0.0119	early	0.0233
light	0.0109	disease	0.0157	blood	0.0113	regions	0.0119	billion	0.0177
density	0.01	medical	0.00997	embryos	0.0111	human	0.0111	history	0.0148
bacteria	0.0983	male	0.0558	theory	0.0811	immune	0.0909	stars	0.0524
bacterial	0.0561	females	0.0541	physics	0.0782	response	0.0375	star	0.0458
resistance	0.0431	female	0.0529	physicists	0.0146	system	0.0358	astrophys	0.0237
coli	0.0381	males	0.0477	einstein	0.0142	responses	0.0322	mass	0.021
strains	0.025	sex	0.0339	university	0.013	antigen	0.0263	disk	0.0173
microbiol	0.0214	reproductive	0.0172	gravity	0.013	antigens	0.0184	black	0.0161
microbial	0.0196	offspring	0.0168	black	0.0127	immunity	0.0176	gas	0.0149
strain	0.0165	sexual	0.0166	theories	0.01	immunology	0.0145	stellar	0.0127
salmonella	0.0163	reproduction	0.0143	aps	0.00987	antibody	0.014	astron	0.0125
resistant	0.0145	eggs	0.0138	matter	0.00954	autoimmune	0.0128	hole	0.00824

The performance of a retrieval system based on this model (PLSI) was found superior to that of both the vector space based similarity (cos) and a non-probabilistic latent semantic indexing (LSI) method.



Summary

- **Synonymy and Polysemy affect all standard IR models – not just limited to VSM**
- **We want to instead model latent (unobserved) topics**
 - SVD factors the term-document matrix into orthogonal eigenvectors (“topics”), automatically ranked by salience (“eigenvalue magnitude”).
 - LSA does SVD and then drops low order topics to create approximation

Related resources

- Lost on Linear Algebra wrt SVD? Try:
<http://www.uwlax.edu/faculty/will/svd/> (great stuff!)
- The BOW toolkit for creating term by doc matrices and other text processing and analysis utilities:
<http://www.cs.cmu.edu/~mccallum/bow>
- SVD is implemented in the SVDPACK software library
<http://www.netlib.org/svdpack>
- Latent Dirichlet Allocation LDA – more powerful version of pLSA
 - Uses a Dirichlet **prior** instead of making a uniform assumption
 - Hence, replace ML with MAP for inference

Clustering



Partitioning Algorithms

- **Partitioning method:** Construct a partition of n documents into a set of K clusters
- **Given:** a set of documents and the number K
- **Find:** a partition of K clusters that optimizes the chosen partitioning criterion
 - Globally optimal: exhaustively enumerate all partitions
 - Effective heuristic methods: K -means and K -medoids algorithms

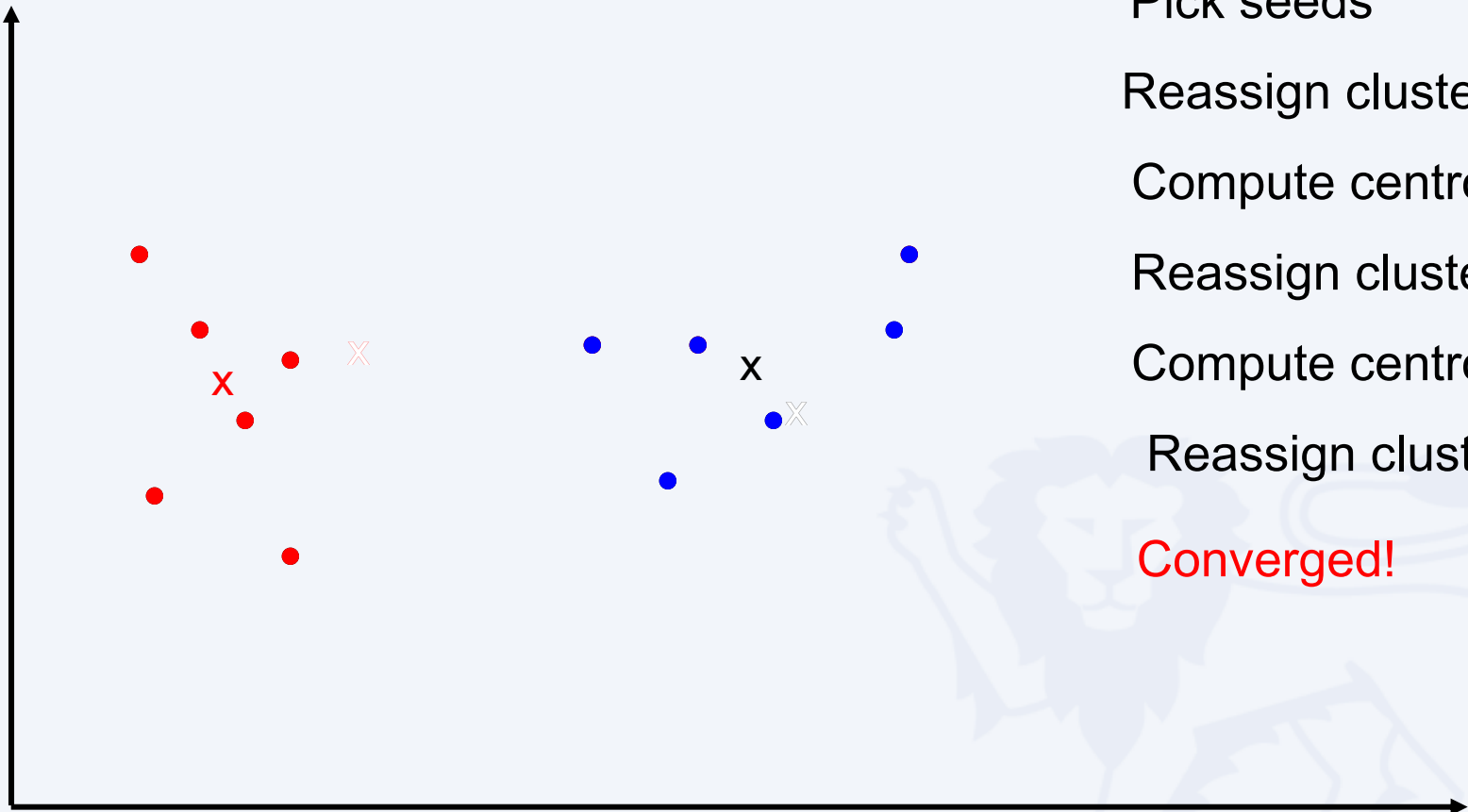
K-Means

- Assumes documents are real-valued vectors.
- Clusters based on **centroids** (aka the **center of gravity** or mean) of points in a cluster, c :

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

- Reassignment of instances to clusters is based on distance to the current cluster centroids.
 - (Or one can equivalently phrase it in terms of similarities)

K Means Example ($K=2$)



Pick seeds

Reassign clusters

Compute centroids

Reassign clusters

Compute centroids

Reassign clusters

Converged!

Seed Choice

- Results can vary based on seed selection.
- Some seeds can result in poor convergence rate, or convergence to sub-optimal clusterings.
 - Select good seeds using a heuristic (e.g., doc least similar to any existing mean)
 - **Try out multiple starting points**
 - Initialize with the results of another method

Example showing sensitivity to seeds

A	B	C
○	○	○
○	○	○
D	E	F

Select B and E as centroids:
Converge to {A,B,C}
and {D,E,F}

Select D and F, converge to
{A,B,D,E} {C,F}

How Many Clusters?

- **Number of clusters K is given**
 - Partition n docs into predetermined number of clusters
- **Finding the “right” number of clusters is part of the problem**
 - Given docs, partition into an “appropriate” number of subsets.
 - E.g., for query results - ideal value of K not known up front - though UI may impose limits.



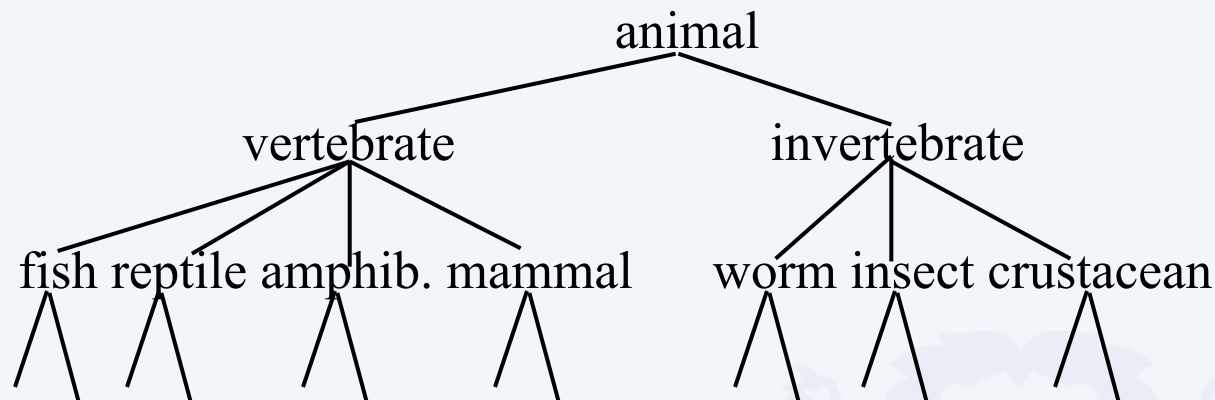
K not specified in advance

- Grade clustering versus a metric.
- Metric must have at least two parts:
Total Benefit - Total Cost
- **Benefit** (of a doc) = cosine sim to its centroid
- **Cost** (constant cost c) in creating a new cluster

What happens if one of these criterion is missing?

Hierarchical Clustering

- Build a tree-based hierarchical taxonomy (*dendrogram*) from a set of unlabeled examples.



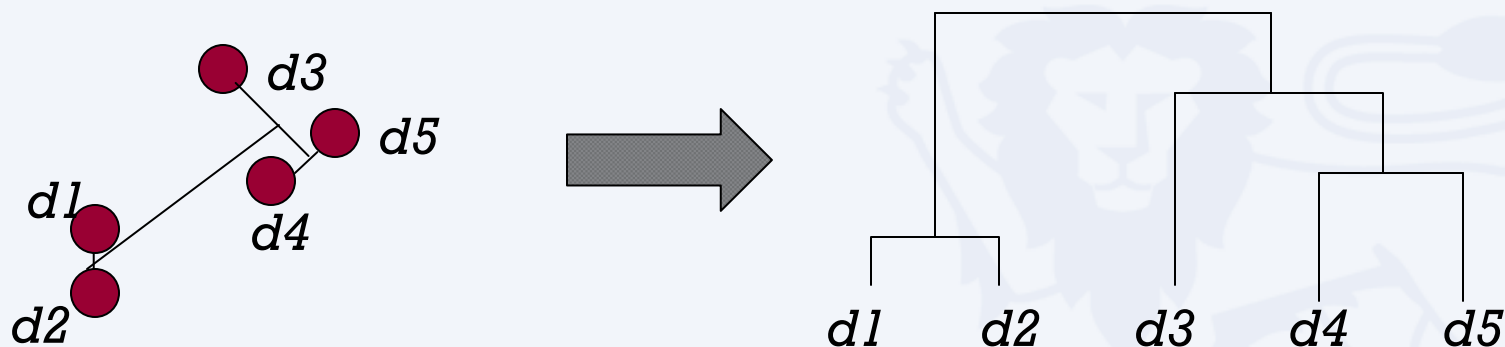
- One option to produce a hierarchical clustering is to recursively apply partitional clustering.
- What are other ways?

Hierarchical Agglomerative Clustering (HAC)

- **Agglomerative (bottom-up):**
 - Start with each document being a single cluster.
 - Eventually all documents belong to the same cluster.
- **Divisive (top-down):**
 - Start with all documents belong to the same cluster.
 - Eventually each node forms a cluster on its own.
- Does not require the number of clusters k in advance
- Merging/splitting history yields the binary hierarchy
- Assumes a binary symmetric distance function.
- Needs a termination condition - why?
 - The final state in both agglomerative and divisive clustering is no use.

Dendrogram: Document Example

- As clusters *agglomerate*, docs likely to fall into a hierarchy of “topics” or concepts.



Bisecting K-means

Almost identical to X-means as in Nomoto and Matsumoto's summarization approach. How is it different?

- Divisive hierarchical clustering method using K-means

For $l=1$ 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}

 }

}

- Steinbach *et al.* suggest HAC is better than k-means but Bisecting K-means is better than HAC for their text experiments

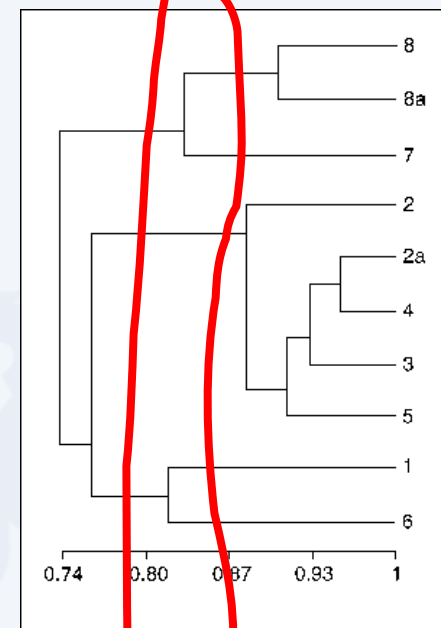
Complexity

- In the first iteration, all HAC methods need to compute similarity of all pairs of n individual instances which is $O(n^2)$.
- In each of the subsequent $n-2$ merging iterations, it must compute the distance between the most recently created cluster and all other existing clusters.
 - Since we can just store unchanged similarities
- In order to maintain an overall $O(n^2)$ performance, computing similarity to each other cluster must be done in constant time.
 - Else $O(n^2 \log n)$ or $O(n^3)$ if done naively

Buckshot Algorithm

- Another way to an efficient implementation:
 - Cluster a sample, then assign the entire set
- Buckshot combines HAC and K-Means clustering.
- First randomly take a sample of instances of size \sqrt{n}
- Run group-average HAC on this sample, which takes only $O(n)$ time.
- Use the results of HAC as initial seeds for K-means.
- Overall algorithm is $O(n)$ and avoids problems of bad seed selection.

**Cut where
You have k
clusters**

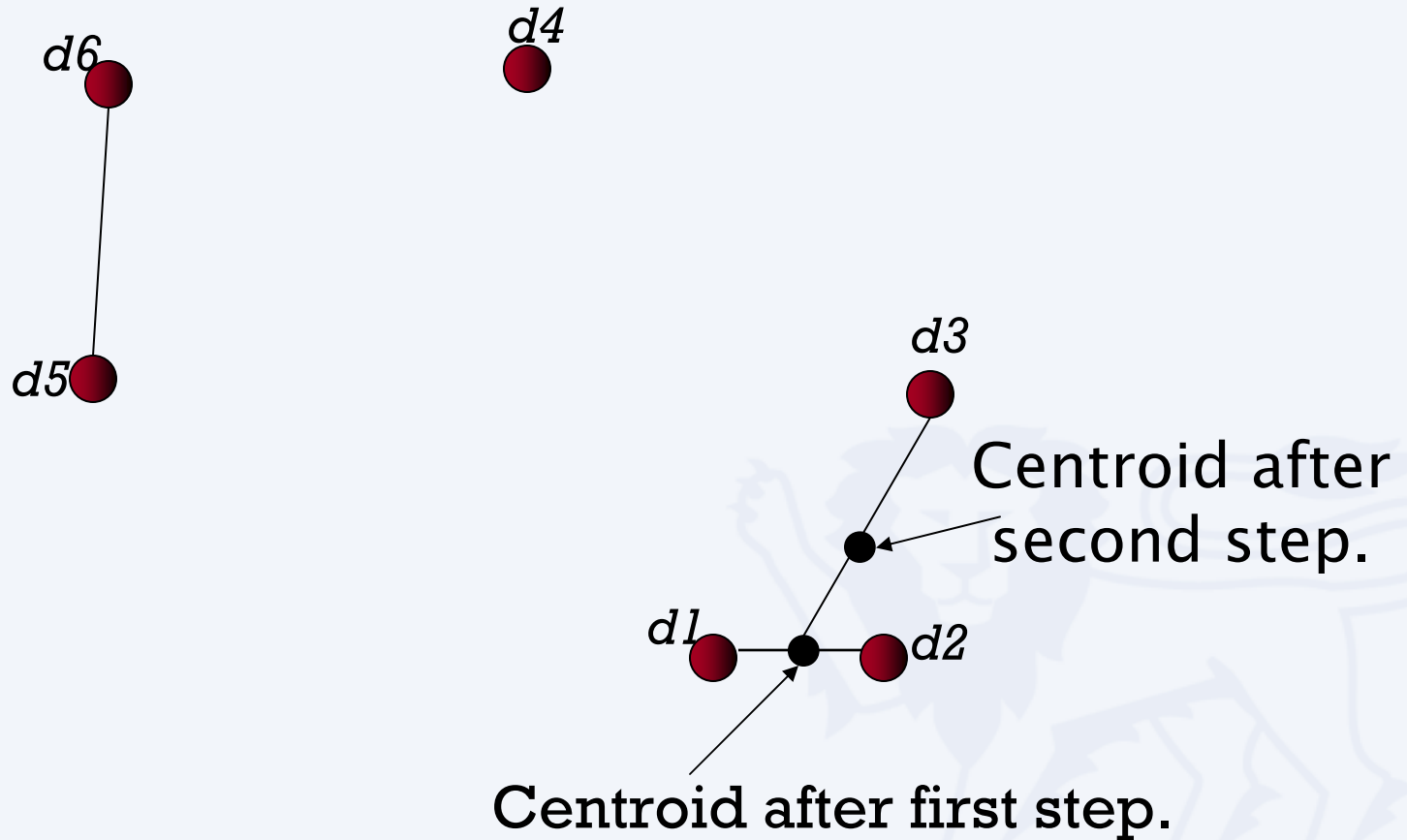


Uses HAC to bootstrap K-means

Cluster representative

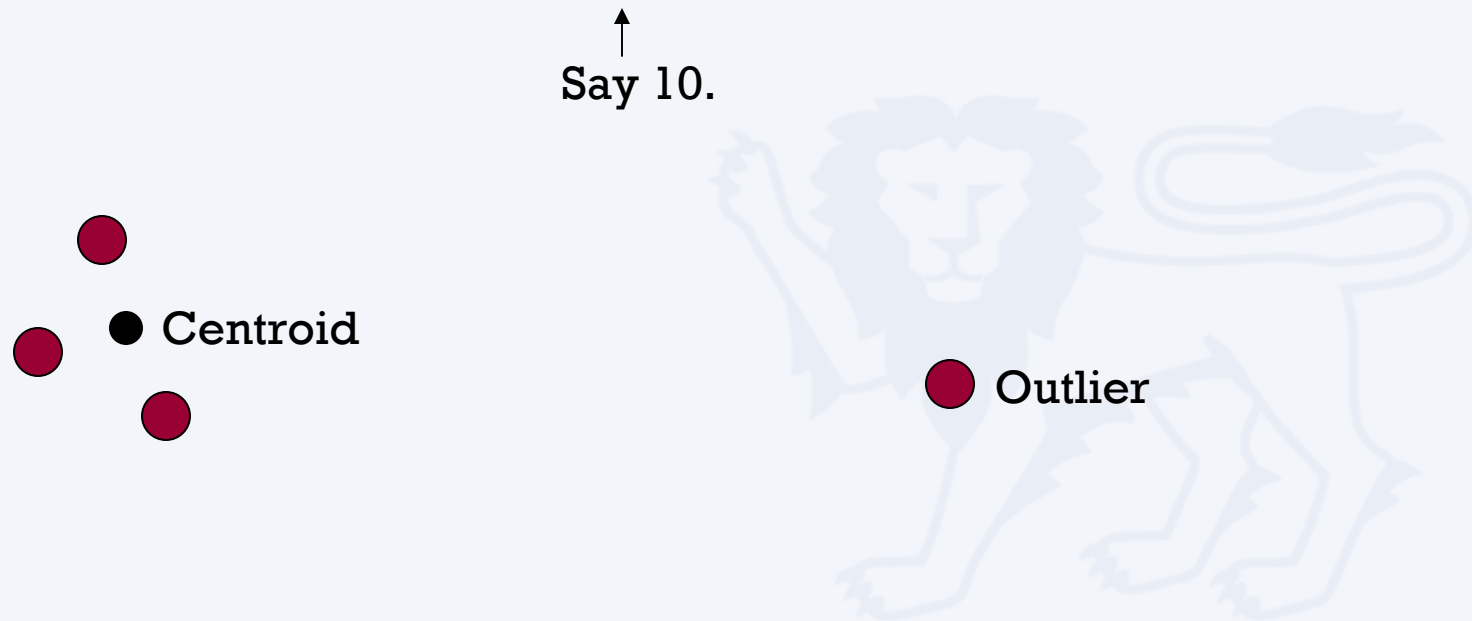
- We want a notion of a representative point in a cluster
- Representative should be some sort of “typical” or central point in the cluster, e.g.,
 - point inducing smallest radii to docs in cluster
 - smallest squared distances, etc.
 - point that is the “average” of all docs in the cluster
Centroid or center of gravity

Example: $n=6$, $k=3$, closest pair of centroids



Outliers in centroid computation

- Can ignore outliers when computing centroid.
- What is an outlier?
 - Lots of statistical definitions, e.g.
 - moment* of point to centroid $> M \times$ some cluster *moment*.



Common similarity functions

Many variants to define closest pair of clusters

- **“Center of gravity”**
 - Clusters whose centroids (centers of gravity) are the most cosine-similar
- **Average-link**
 - Average cosine between pairs of elements
- **Single-link**
 - Similarity of the most similar (single-link)
- **Complete-link**
 - Similarity of the “furthest” points, the least similar

Single vs. Complete Link

- Use max sim pairs:

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

- Can result in long and thin clusters due to chaining effect.
–When is it appropriate?

- Use min. sim of pairs:

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

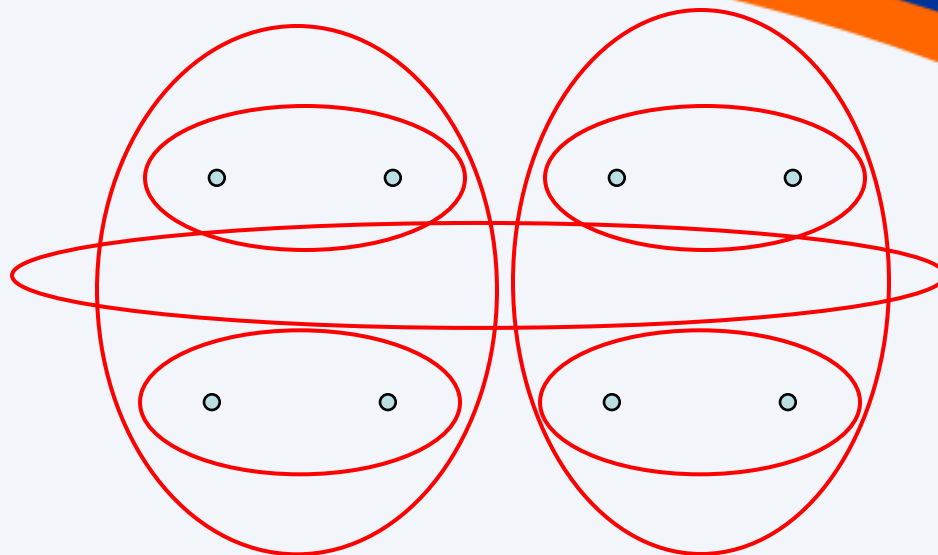
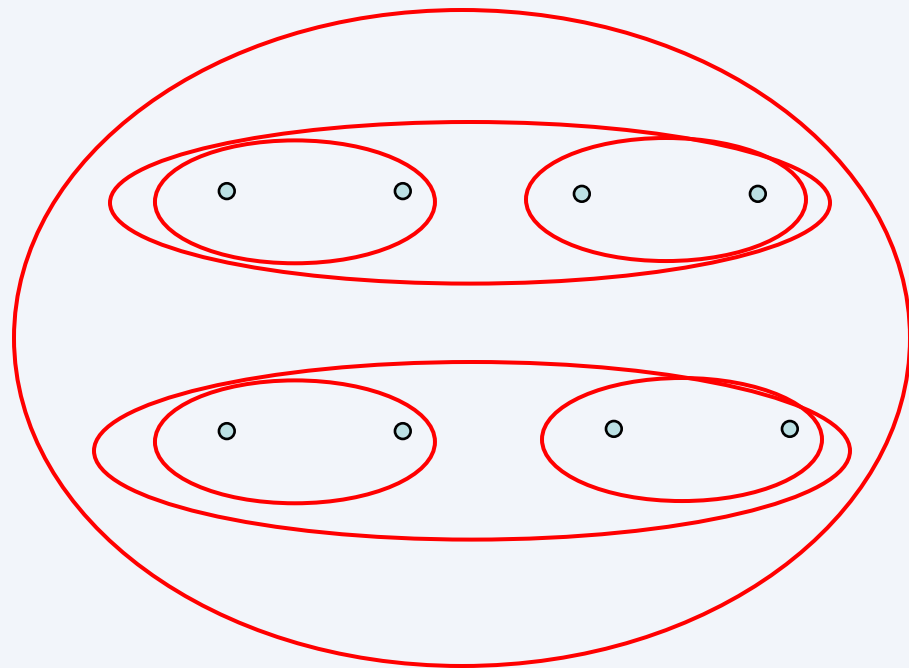
- Makes “tighter,” spherical clusters that are typically preferable.

- After merging c_i and c_j , the similarity of the resulting cluster to another cluster, c_k , is:

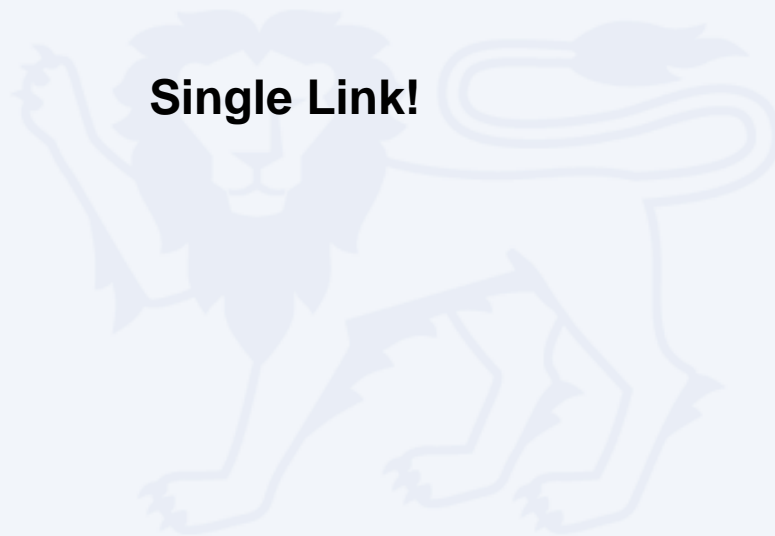
$$\begin{aligned} &sim((c_i \cup c_j), c_k) \\ &= \max(sim(c_i, c_k), sim(c_j, c_k)) \end{aligned}$$

$$\begin{aligned} &sim((c_i \cup c_j), c_k) \\ &= \min(sim(c_i, c_k), sim(c_j, c_k)) \end{aligned}$$

Complete Link!



Single Link!



Group(wise) Average

- Use average similarity across all pairs within the merged cluster to measure the similarity of two clusters.

$$sim(c_i, c_j) = \frac{1}{|c_i \cup c_j|(|c_i \cup c_j| - 1)} \sum_{\vec{x} \in (c_i \cup c_j)} \sum_{\vec{y} \in (c_i \cup c_j): \vec{y} \neq \vec{x}} sim(\vec{x}, \vec{y})$$

- **Compromise between single and complete link.**
- **Two options:**
 - Averaged across all ordered pairs in the merged cluster
 - Averaged over all pairs *between* the two original clusters
- Some previous work has used one of these options; some the other. No clear difference in efficacy

Computing Group Average Similarity

- Assume cosine similarity and normalized vectors with unit length.
- Always maintain sum of vectors in each cluster.

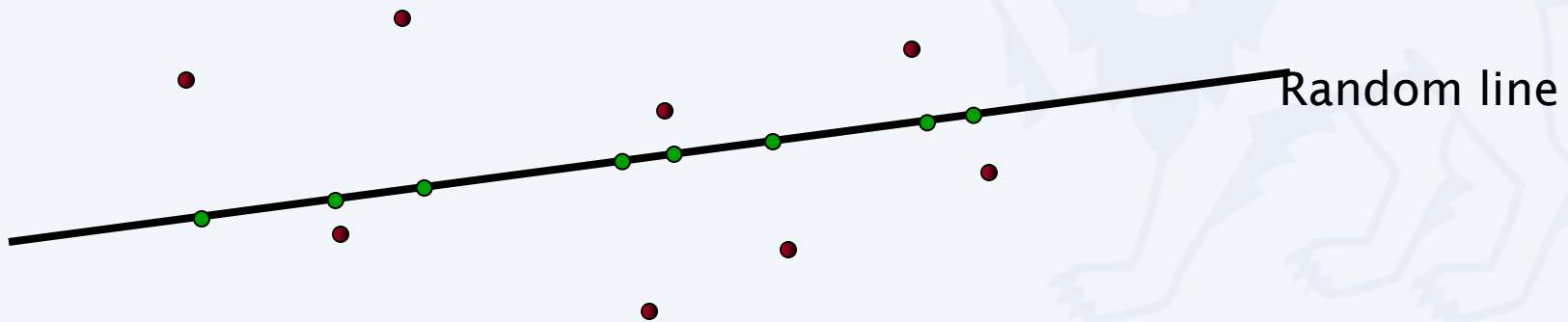
$$\vec{s}(c_j) = \sum_{\vec{x} \in c_j} \vec{x}$$

- Compute similarity of clusters in constant time:

$$\text{sim}(c_i, c_j) = \frac{(\vec{s}(c_i) + \vec{s}(c_j)) \bullet (\vec{s}(c_i) + \vec{s}(c_j)) - (|c_i| + |c_j|)}{(|c_i| + |c_j|)(|c_i| + |c_j| - 1)}$$

Efficiency by approximation

- In standard algorithm, must find closest pair of centroids at each step
- **Approximation: instead, find nearly closest pair**
 - Use some data structure that makes this approximation easier to maintain
 - Simple example: maintain closest pair based on distances in projection on a random line



Trends in NLP

Genre Treatments
Multilingual Treatments

Email – Indexing

Preprocessing

- **Threading of messages**
 - Remove stopwords (re:, fwd:)
- **Identifying earlier messages**
 - Also heuristic: “>” tokens, lines after “original message”, “On DATE PERSON writes:”, etc.
- **Indexing differently:**
 - By message
 - By thread
 - By threads with forwarded messages removed

Email – Indexing Results

- **Retrieval results on three indices not very different**
 - Noted that thread indices bias to retrieve long threads
 - Near duplicate emails common → perhaps need work here



Email: understanding retrieval needs

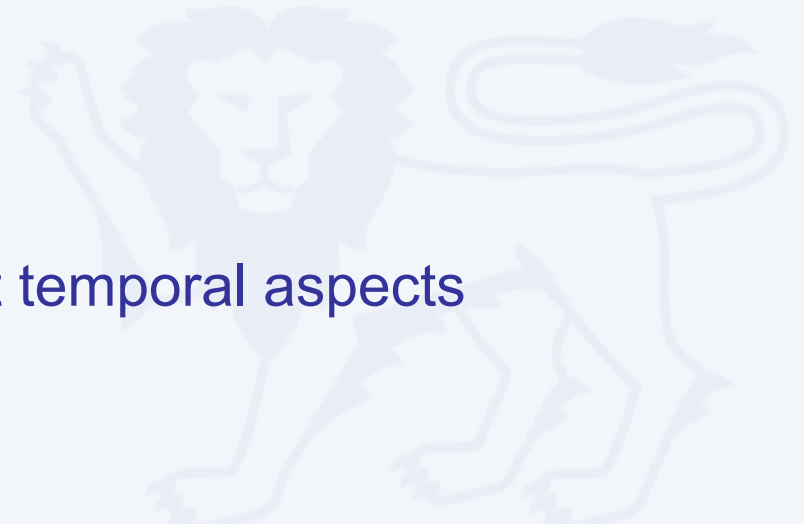
- **Towards a understanding of email retrieval:**
 - Who was involved in activity X?
 - Who made decision Y and what was the decision?
 - How often and in what way did person A and B interact?
Temporal rhythm of email also plays a role
- **Necessitates name disambiguation if given a large collection**
 - Multiple “John Doe”s within a single collection
 - Context of sender’s and recipient’s social network influences how ambiguous entities are referred to

SMS / Instant Messaging: Language

- **Keypad limits input**
 - Corrupted and shortened version of true message on output
- **Seen as problem where actual message undergoes a summarization / noise transform**
 - Recovery model needs to account for transliteration, shortening (e.g., “ur pc 2?”)
 - Emoticons easy to build dictionary, serve as punctuation
 - Also need to recover correct case
 - Noisy channel methods

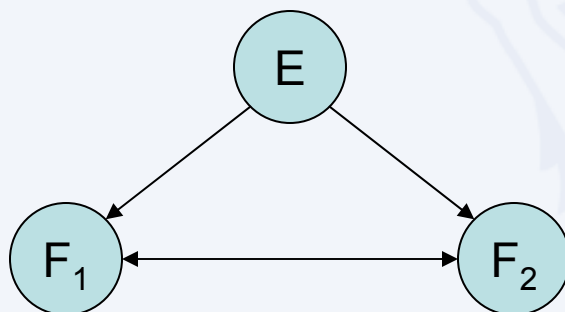
SMS / Instant Messaging: Threading

- **Exacerbated problem (in comparison to e-mail)**
- **Basic architecture:**
 - Use cosine vector similarity between turns
 - Use telltale discourse cues (e.g., “Ok,” , Q/A pairs)
- **Results:**
 - Discourse markers very helpful.
 - No works yet to deal with explicit temporal aspects



Multilinguality with scarce resources

- Traditionally, need bilingual aligned data to train MT systems
- Work done to build these resources automatically from monolingual data sources
- **Triangulation** where links to multiple rich resource languages help



Multilinguality with scarce resources

- **Cross Language IR problems**
 - untranslatable terms, inflectional forms, phrase identification and translation, and translation ambiguity
- **Leverage simple bilingual wordlists**
- **Take advantage of cognates (words with common origin; English night and German nacht), loanwords (e.g. sushi), and transliterations (Malay kopi, coffee)**
 - Align sentences and docs
 - Run “spelling correction” on cognates in target language

Wrapping up...

References
Final Review

References - NLP

- ACL Anthology
- ACL, EACL, NAACL
- EMNLP
- COLING
- IJCNLP, LREC, RANLP

- **Shared Tasks**

- SensEval, Semeval
- DUC, TAC

- **MALINDO**: workshop on Malay / Indonesian language
- **SIGHAN**: workshops and other activities around Chinese
- **TDIL**: centralized resources for Indian languages by the government of India

References - Speech

- Eurospeech
- Interspeech
- ICASSP

- **Shared Tasks**
 - NIST Benchmarking

References – Information Retrieval / Extraction

- **SIGIR**
- **CIKM, ECIR, KDD**
- **Web Intelligence**
- **Shared Tasks**
 - TREC, CLEF, NTCIR, FIRE, INEX
 - MUC, TAC



Reprise

Day 1

AM

- Applications' Input / Output
- Resources

PM

- Selected Toolkits
- Python Intro
- NLTK Hands-on

Day 2

AM

- Evaluation
- Annotation
- Information Retrieval
- ML Intro

PM

- Machine Learning
- SVM Hands-on

Day 3

AM

- Sequence Labeling
- CRF++ Hands-on

PM

- Dimensionality Reduction
- Trends & Issues

Summary

- **NLP – Ambiguity**
 - Finite state automata, sequence models: HMMs, CRFs
 - Standard machine learning: feature engineering
- **IR – largely token based, vector space model**
- **Interface between the two in several areas**
 - Stemming, question answering / passage retrieval
 - Controlling ambiguity: dimensionality reduction
- **Trends:**
 - Multilingual systems using common languages as bridge