Course Review
Primarily Focused on Exam 2

CSCI 572 Review of Concepts/Algs/Etc. Presented The Entire Semester

- **Algorithms**
  - PageRank
  - HITS (ranking)
  - Porters (stemming)
  - Soundex (phonemes)
  - *Levenshtein (spell correction)
  - *K-Means Clustering
  - *Agglomerative Clustering
  - *Rocchio (classification)
  - *K-Nearest Neighbor (classification)
  - Block Sort-Based Indexing (**not covered**)
  - Single-Pass In-Memory Indexing **(not covered)**
  - Logarithmic Merge Indexing
  - Computing Cosine Scores
- **Data Structures**
  - B-trees (holding the dictionary)
  - Tries (spelling correction)
  - *Suffix trees (autocomplete)
  - *Priority Queues (ranking)
- **Concepts (cont'd)**
  - *Centroids (classification)
  - *Dendrogram (clustering)
  - *Pay-Per-Click Auctions (advertising)
  - Botnets (click fraud)
  - *Ontologies: WordNet, FreeBase, KnowledgeGraph
  - Precision (ranking)
  - Recall (ranking)
  - TP, FP, FN, TN formulations
  - Harmonic Mean (ranking)
  - F Measure (ranking)
  - Mean Average Precision
  - Discounted Cumulative Gain

*starred items are for exam 2

CSCI 572 Review of Concepts/Algs/Etc. Presented The Entire Semester

- **Techniques**
  - Crawling (distributed, techniques)
  - Crawling: robots.txt, depth vs. breadth
  - Crawling: URL representation, threading
  - Stemming, Stop words, Lemmatization
  - Inverted index, positional inverted index
  - *Champion lists (results)
  - *quality scoring
  - *N-grams
  - Cryptographic hashing (de-duplication)
  - Shingling (de-duplication)
  - Jaccard Similarity
  - Document Boolean Model
  - Document Vector Model
  - Cosine similarity
  - Euclidean distance
- **Techniques (cont'd)**
  - *Purity Index (clustering)
  - *Rand Index (clustering)
  - *Microformats (snippets)
  - *TF-IDF (matching)
  - ***Mean Reciprocal Rank
- **Other**
  - Zipf's Law
  - Heaps Law
- **Software**
  - Crawler4j
  - *Lucene
  - *Solr
  - ***Norvig Spelling Program
  - *NetworkX library

*starred items are for exam 2

ALGORITHMS

Basic Spelling Correction Algorithm

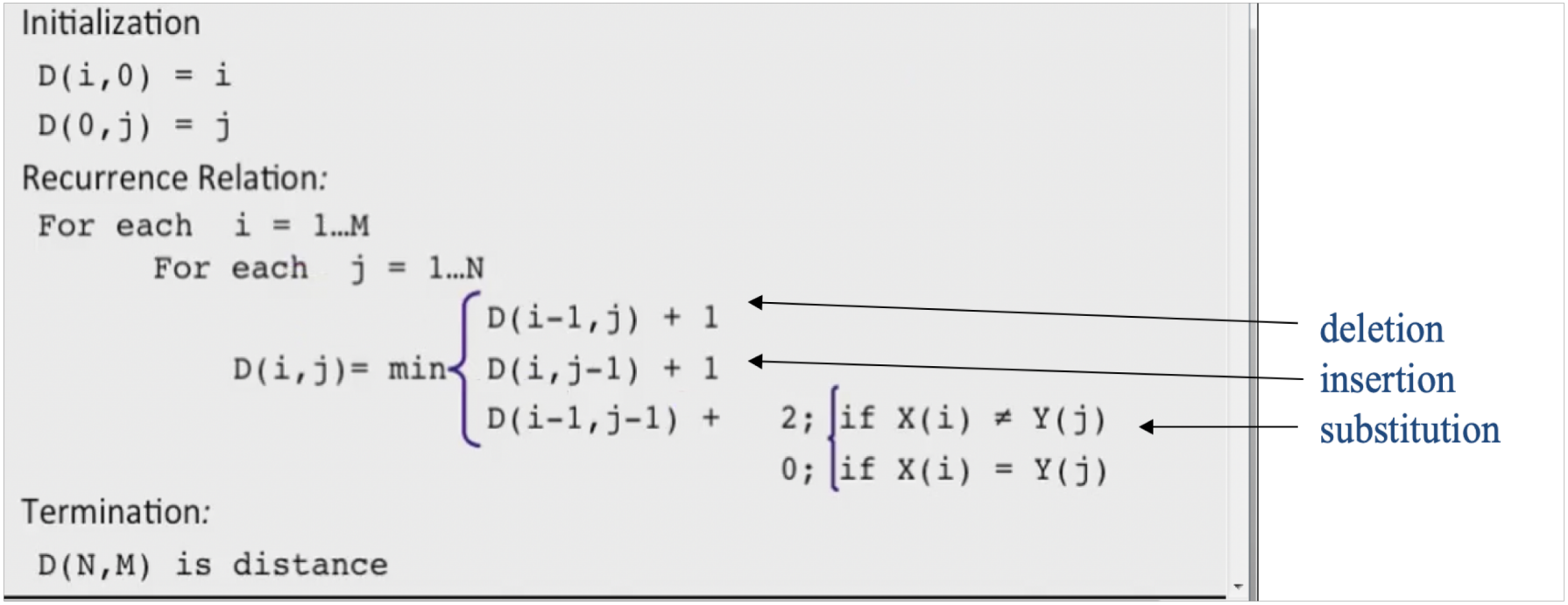
1. Initial step: Create a dictionary and encode it for fast retrieval
2. When a query is submitted, the spell checker examines each word and looks for possible character edits, namely
   - insertions,
   - deletions,
   - substitutions, and occasionally
   - transpositions
   - Observation:
     - 80% of errors are within edit distance 1
     - Almost all errors within edit distance 2
3. Take the output of step 2 and compute probabilities for the candidates using previously identified n-grams
   - The Stupid Backoff Algorithm finds the appropriate value of N
4. Select the result with highest probability
   - Confusion matrix

LEVENSHTEIN ALGORITHM

- For computing minimum edit distance
- Dynamic programming: a tabular computation of *D(n,m)*
- Solving problems by combining solutions to subproblems
- bottom-up
  - we compute *D(i,j)* for small *i, j*
  - and compute larger *D(i,j)* based on previously computed smaller values
  - i.e. compute *D(i,j)* for *all i (0 < i < n)* and *j (0 < j < m)*

  - **(for details see the Processing Text slides)**

Levenshtein Algorithm



- **the distance in any cell is the minimum distance from one of three possible cells**
- **Using backtrace**
- **Weighted minimum edit distance**

Centroid

- Recall that we represent documents as points in a high-dimensional space
- The centroid is the center of mass of a set of points.
- *Definition*: Centroid

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

where C is a set of documents, |C| is the size of the set and $\vec{v}(d) = \vec{d}$ is the vector representing document d

(for details see the Clustering slides, see also the Rocchio Algorithm)

The K-Means Clustering Algorithm

A method for determining a **set of clusters** for a collection of documents

1. Select K points as initial centroids
2. **repeat**
   - form K clusters by assigning each point to its closest centroid
   - re-compute the centroid of each cluster
3. **until** centroids do not change

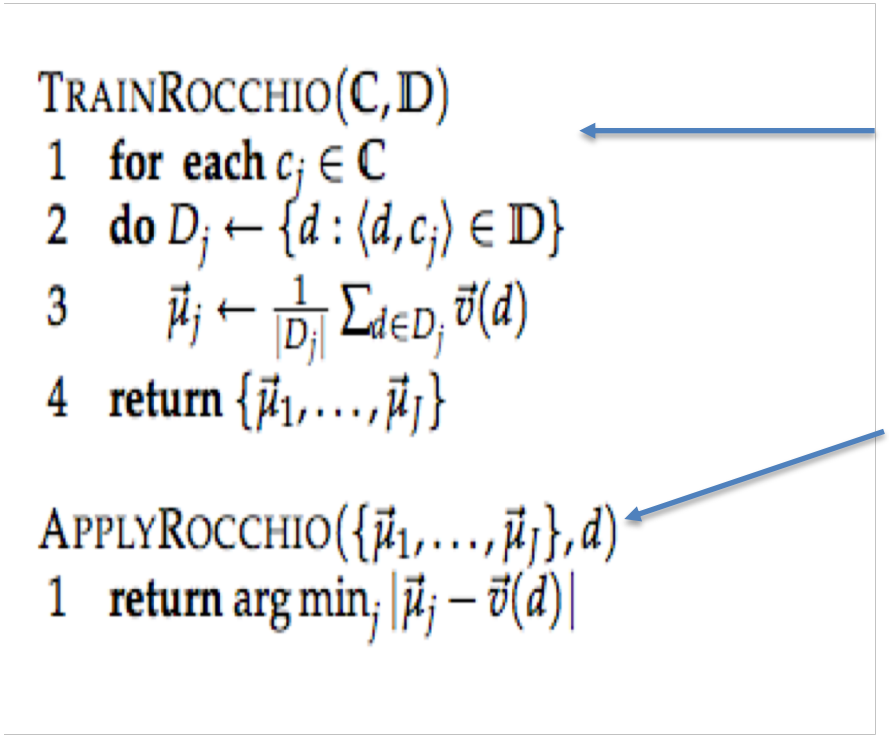
Summary

- the algorithm will always terminate, however it does not always find the optimal solution
- this is an example of a greedy algorithm
- A standard way to pick the initial set of k means is to choose a random selection
- The algorithm has converged when the assignments no longer change.
- The algorithm will converge to a (local) optimum.

Agglomerative Clustering Algorithm

- The Algorithm
  1. Compute the distance matrix between the input data points (i.e. the distance between all pairs of points)
  2. Let each data point be a cluster unto itself
  3. Repeat
  4. Merge the two closest clusters
  5. Update the distance matrix
  6. Until only a single cluster remains
- Key operation is the computation of the distance between two clusters
  - Different definitions of the distance between clusters lead to somewhat different algorithms

Rocchio Algorithm for Classification



TRAINROCCHIO($\mathbb{C}, \mathbb{D}$)
1 **for each** $c_j \in \mathbb{C}$
2 **do** $D_j \leftarrow \{d : \langle d, c_j \rangle \in \mathbb{D}\}$
3 $\quad \vec{\mu}_j \leftarrow \frac{1}{|D_j|} \sum_{d \in D_j} \vec{v}(d)$
4 **return** $\{\vec{\mu}_1, \ldots, \vec{\mu}_J\}$

APPLYROCCHIO($\{\vec{\mu}_1, \ldots, \vec{\mu}_J\}, d$)
1 **return** $\arg\min_j |\vec{\mu}_j - \vec{v}(d)|$

- Two phases

1. *TRAINROCCHIO* takes a set of initial class ids in C and a set of documents in ID and *computes* the set of documents belonging to D_j; $u_1, \ldots u_j$ are the centroids determined by TRAINROCCHIO;
2. *APPLYROCCHIO* takes the centroids and a new document d and returns the centroid with minimum distance;

used for classification and
primarily used for relevance feedback

(for details see the Vector Space Classification slides)

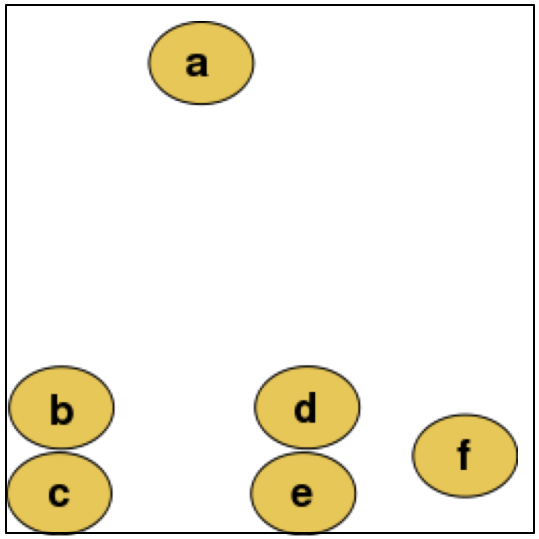
The K Nearest Neighbor Algorithm for Classification

- Given a set of points each of which has a class indicator and given an integer *k*, and a new point, then
  1. compute the distance between the new point and all other points
  2. determine the *k* closest points to the new point
  3. examine the class indicators of the *k* closest points and choose the class indicator that occurs most often
  - The contiguity hypothesis holds
  - Best choice of *k*, not too large; *k=1* is a special case
  - For binary choice problems choose *k* odd, e.g. *k=3*

CONCEPTS

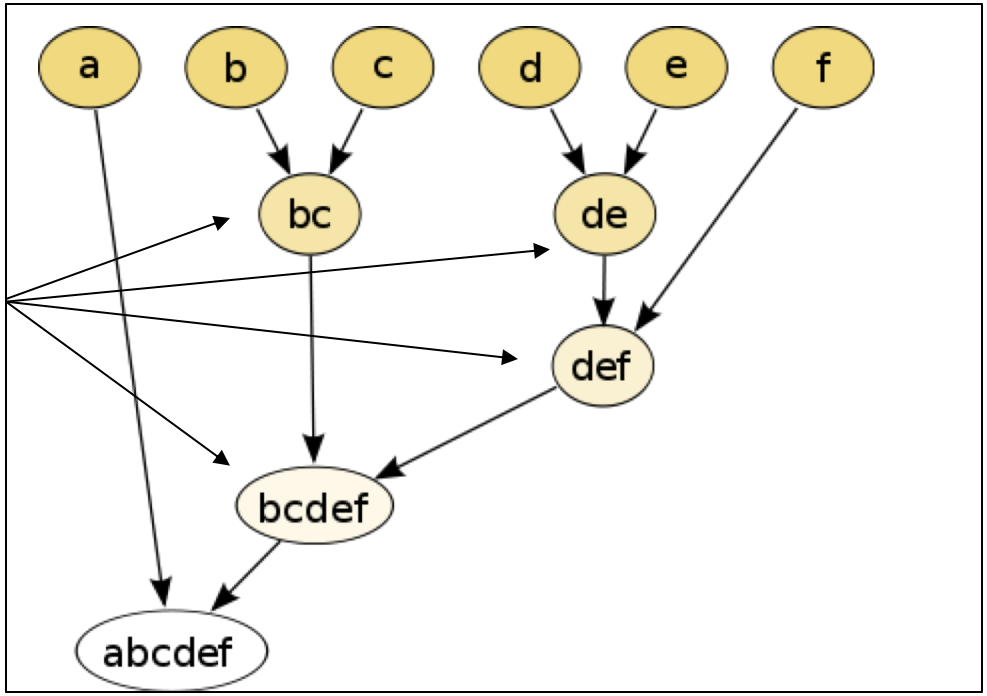
Dendrogram are Used to Display Clusters

- A dendrogram is a tree diagram frequently used to illustrate the arrangement of the clusters produced by hierarchical clustering



original input

clusters



corresponding dendrogram

second row clusters are: {a}, {b c}, {d e} {f}
third row clusters are: {a}, {b c} {d e f}

Important for search engines that focus on clustering

Cosine Similarity

- Cosine similarity
  - Cosine similarity is a measure of similarity between two vectors of an inner product space that measures the cosine of the angle between them. Range from -1 (dissimilar) to 1 exactly similar

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

  - Euclidean distance is a close alternative that is also popular

  - **(For details see Clustering Slides)**

Speeding Up Inverted Index Retrieval Using Champion Lists Heuristic

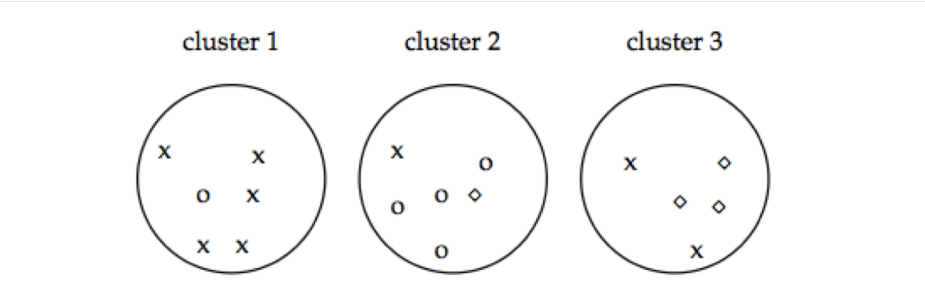
- Pre-compute for each dictionary term *t,* the *r* docs of highest weight (tf-idf) in *t*'s postings
  - Call this the champion list for *t*
  - (aka fancy list or top docs for *t*)
- Note that *r* has to be chosen at index build time
  - Thus, it's possible that $r < K$
- At query time, only compute scores for docs in the champion list of some query term
  - champion lists that include the terms of the query
  - Pick the *K* top-scoring docs from amongst these

Authority Measures

- Assign to each document a query-independent quality score in [0,1] to each document d
  - Denote this by g(d), g stands for goodness •
- Thus, a quantity like the number of citations is scaled into the range of [0,1]
- The PageRank can also be scaled to the same range
- Heavily curated content, e.g. Wikipedia can be given a high quality score
- Consider a simple total score combining cosine relevance and authority
  - *net-score(q,d) = g(d) + cosine(q,d) = g(d) + tf-idf*$_{td}$
- Maintain champion lists in g(d) order

Purity and Rand Index for Cluster Evaluation

consider the three clusters,
cluster 1 has 5 related items;
cluster 2 has 4 related items and
cluster 3 has 3 related items;



- Purity Measure - accuracy is measured by the number of correctly assigned documents divided by the total number of documents;
- Purity = (1/17) * (5 + 4 + 3) ~= 0.71
- Rand index: measures the percentage of decisions that are correct.
- RI = (TP + TN) / (TP + FP + FN + TN)

(For details see Clustering Slides)

Legal Concepts

- Intellectual property protections fall into four categories:
  1. copyright (for literary works, art, and music),
  2. patents (for inventions and processes),
  3. trademarks (for company and product names and logos), and
  4. trade secrets (for recipes, code, and processes).

—No Questions on Legal Concepts

Types of Click Fraud

1. Individuals deploying automated clicking programs or software applications (called bots) specifically designed to click on ads
2. An individual employing low-cost workers or incentivizing others to click on the advertising links
3. Website publishers manually clicking on the ads on their pages
4. Website publishers manipulating web pages in such a way that user interactions with the web site result in inadvertent clicks, e.g iframes
5. Website publishers subscribing to paid traffic websites that artificially bring extra traffic to the site, including extra clicking on the ads
6. Advertisers manually clicking on the ads of their competitors
7. Website publishers being sabotaged by their competitors or other ill-wishers

No Questions on Click Fraud

Videos to Review

- **Knowledge Graph (2 min 30 sec)
- **The History of Wikipedia (2 min)
- **Trolling the Wikipedia Trolls (4.5 min)
- **Text search with Lucene (1 of 2) (13 min)
- **Text search with Lucene (2 of 2) (20 min, see first 9 min)
- **Solr Basics (13 min)
- **Solr in 5 minutes
- **WDM40:Context Sensitive Spell Correction (12 min)
- **WDM38:Spelling Correction Using N-Gram Overlap Technique (21 min)
- **Generating Snippets - Jurafsky & Manning, Stanford U. (7 min)
- **K-Means Algorithm (7 min)
- **Hierarchical Agglomerative Clustering (12 min)
- **What is Question Answering - Jurafsky & Manning, Stanford U. (7 min)
- **Answer Types and Query Formulation - Jurafsky & Manning, Stanford U. (8min)
- **Passage Retrieval and Answer Extraction - Jurafsky & Manning, Stanford U. (6 min)
- **Using Knowledge in QA - Jurafsky & Manning, Stanford U. (4 min)
- **Relevance Feedback Rocchio (2 min)
- **Rocchio Algorithm Illustration (11 min)
- **How KNN Algorithm Works (5 min)
- **KNN-10: pros and cons (2 min)

Sample Test Questions

Study Plan

1. Review classnotes
2. Review old exam
3. Review videos
4. Review textbook

Three previous exams

Exam starts at 8:00

SGM123/THH101

				• WDM123: Discussion of K Nearest Neighbor (25 min)
Week 13 Nov 14	Battling Click Fraud and Legal Issues	PPT, PDF	• Google issues Bad Ads Report, 2015 • Botnet Discovered Costing Advertisers $6 million --- • Google's Response to the EU AntiCompetitive Accusations (includes video 3 min) • Improving Quality Isn't Anti-Competitive Part II • **Click Fraud: Anecdotes from the Front Line CMU talk 2007 (1 hr total; start 15 minutes in for 30 min)	• 60 Minutes Your Data, start at 2 min, end at 12 min • Google's Response to European Delisting Requirement (19 min) • More on Google's Response to European Delisting Requirement (15 min) • Google accused of favoring their own results (2 min) **Other useful videos** • Google accused of Avoiding Taxes (2 min)
Week 14 Nov 19	Course Review	PPT, PDF		
Week 14 Nov. 21 Holiday	**Thanksgiving Holiday**			
Week 15 Nov 26	**Exam 2 Starts at 8:00am in** • **SGM 123 if your last name starts with: A-L), and** • **THH 101 if your last name starts with: M-Z)** **1 hour exam**		Spring 2017 Exam Fall 2017 Exam Spring 2018 Exam **Exam 2 Starts at 8:00am in** • **SGM 123 if your last name starts with: A-L), and** • **THH 101 if your last name starts with: M-Z)** **1 hour exam**	**Exam 2 Starts at 8:00am in** • **SGM 123 if your last name starts with: A-L), and** • **THH 101 if your last name starts with: M-Z)** **1 hour exam**
Week 15 Nov 28	**Live Demos of HW #5, TODAY**	**Demos will start at 7:00 am in SGM 123**	**Check below for your most likely time to present** Schedule	**Homework #5: Demos will start at 7:00 am in SGM 123** **Check the Presentation Schedule for your most likely time**