Week 09: Similarity-Based Search
DS-GA 1004: Big Data
Detailed Notes for Final Exams

Introduction to Similarity Search

The challenge of finding similar items in large datasets efficiently. Traditional brute-force approaches are computationally infeasible at scale. Key techniques include hashing, approximation, and locality-sensitive methods.

Core Concepts

Jaccard Similarity

For sets A and B, Jaccard similarity $J(A, B)$ measures overlap:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Distance Metric: $D(A, B) = 1 - J(A, B)$.

MinHash (Broder, 1997)

Approximates Jaccard similarity using hash functions. For a permutation π of all elements:

$$h(S|\pi) = \min\{\pi(k) \mid \pi(k) \in S\}$$

Key Property:

$$P[h(S_1) = h(S_2)] = J(S_1, S_2)$$

Locality-Sensitive Hashing (LSH)

Improves MinHash by grouping signatures into blocks. Parameters b (bands) and r (rows per band) control precision/recall:

$$P[\text{Collision in LSH}] = 1 - (1 - J^r)^b$$

Algorithms and Implementation

MinHash Signature Calculation

1. Generate m hash functions $H_1, H_2, \ldots, H_m$.
2. For each set S, compute $h_i(S) = \min_{x \in S} H_i(x)$.
3. Similarity is approximated by collision frequency.

LSH Workflow

1. Divide MinHash signatures into b bands of r rows.
2. Hash each band separately.
3. Candidate pairs are those sharing at least one band hash.

Extensions Beyond Sets

Ruzicka Similarity for Bags

Extends Jaccard to multisets by treating duplicates as unique elements:

$$R(A, B) = \frac{\sum \min(A_i, B_i)}{\sum \max(A_i, B_i)}$$

Cosine Similarity for Vectors

For vectors $\mathbf{u}, \mathbf{v}$:

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \cos\theta = \frac{\mathbf{u}^T\mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}$$

LSH via random hyperplanes: $h_w(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x})$.

Tradeoffs and Optimization

Method	**Advantages**	**Limitations**
Brute Force	Exact results	$O(N^2)$ complexity
MinHash	Sub-linear time, scalable	Approximate, sensitive to hash collisions
LSH	Reduces candidate set size	Requires tuning b and r

Case Study: Plagiarism Detection

- Represent documents as sets of shingles (word n-grams).
- Compute MinHash signatures for all documents.
- Use LSH to identify candidate pairs.
- Verify with exact Jaccard similarity on candidates.

Failure Modes and Mitigations

- **Stop Words:** Common words (e.g., "the") cause spurious collisions. Mitigation: Remove stop words before hashing.
- **High Dimensionality:** Curse of dimensionality affects spatial methods. Mitigation: Dimensionality reduction (e.g., PCA).

Exam Tips

- Understand the relationship between Jaccard similarity and MinHash collision probability.
- Know how LSH parameters b and r affect precision/recall tradeoffs.
- Be able to contrast MinHash, LSH, and cosine similarity techniques.
- Practice calculating Jaccard/Ruzicka similarities and interpreting hash collisions.

Appendix: Key Formulas

$$\begin{aligned}
\text{Jaccard:} \quad & J(A,B) = \frac{|A \cap B|}{|A \cup B|} \\
\text{MinHash Collision:} \quad & P[h(S_1) = h(S_2)] = J(S_1, S_2) \\
\text{LSH Collision:} \quad & P = 1 - (1 - J^r)^b \\
\text{Ruzicka:} \quad & R(A,B) = \frac{\sum \min(A_i, B_i)}{\sum \max(A_i, B_i)} \\
\text{Cosine:} \quad & \cos\theta = \frac{\mathbf{u}^T \mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}
\end{aligned}$$