Methods for High Degrees of Similarity

Index-Based Methods
Exploiting Prefixes and Suffixes
Exploiting Length

Mining of Massive Datasets
Leskovec, Rajaraman, and Ullman
Stanford University

Setting: Sets as Strings

- We'll again talk about Jaccard similarity and distance of sets.
- However, now represent sets by strings (lists of symbols):
  1. Order the universal set.
  2. Represent a set by the string of its elements in sorted order.

Example: Shingles

- If the universal set is k-shingles, there is a natural lexicographic order.
- Think of each shingle as a single symbol.
- Then the 2-shingling of abcad, which is the set {ab, bc, ca, ad}, is represented by the list [ab, ad, bc, ca] of length 4.

Example: Words

- If we treat a document as a set of words, we could order the words lexicographically.
- Better: Order words lowest-frequency-first.
- Why? We shall index documents based on the early words in their lists.
  - Documents spread over more buckets.

Jaccard and Edit Distances

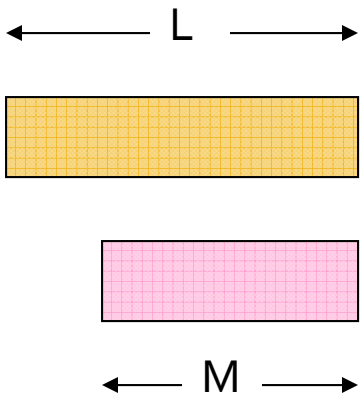
- Suppose two sets have Jaccard distance J and are represented by strings s_1 and s_2. Let the LCS of s_1 and s_2 have length C and the (insert/delete) edit distance of s_1 and s_2 be E. Then:
  - 1-J = Jaccard similarity = C/(C+E).
  - J = E/(C+E).

Works because these strings never repeat a symbol, and symbols appear in the same order.

Length-Based Indexes

- The simplest thing to do is create an index on the length of strings.
- A set whose string has length L can be Jaccard distance J from a set whose string has length M only if $L \times (1-J) \leq M \leq L/(1-J)$.
- Example: if 1-J = 90% (Jaccard similarity), then M is between 90% and 111% of L.

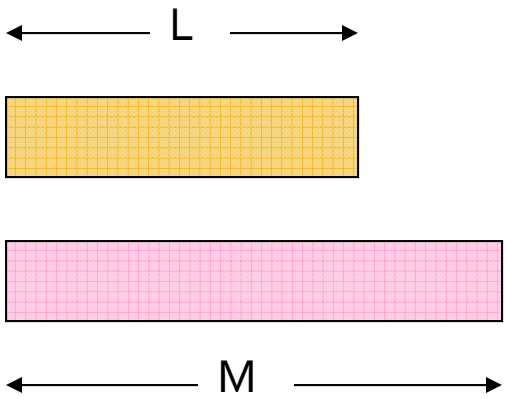
Why the Limit on Lengths?



1-J = M/L

M = L×(1-J)

A shortest candidate



1-J = L/M

M = L/(1-J)

A longest candidate

B-Tree Indexes

- The B-tree is a perfect index structure for a length-based index.
- Given a string of length L, we can find strings with length in the range L×(1-J) to L/(1-J) without looking at any candidates outside that range.
- But just because strings are similar in length, doesn't mean they are similar.