Fingerprint Matching

Minutiae
A New Way of Bucketing

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

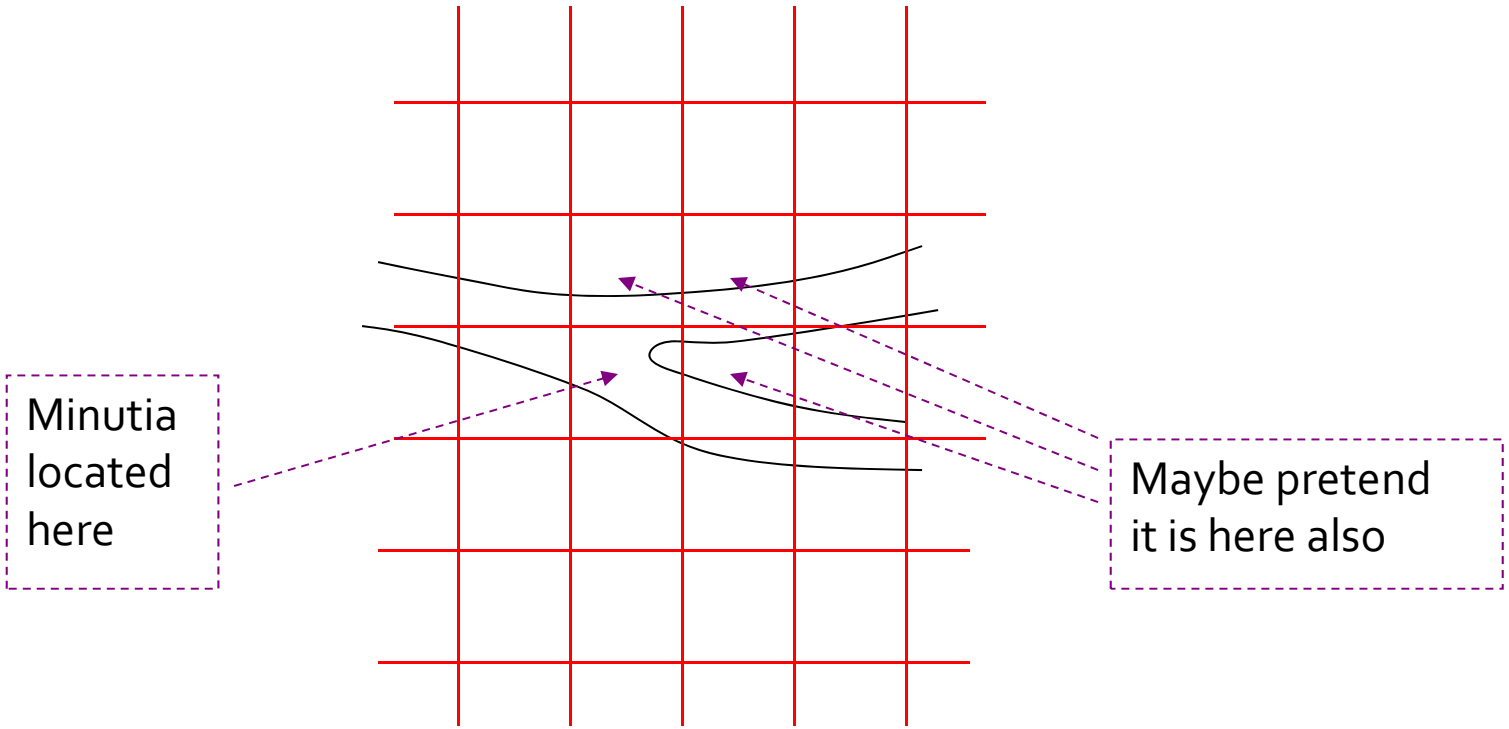
Fingerprint Comparison

- Represent a fingerprint by the set of positions of *minutiae*.
  - These are features of a fingerprint, e.g., points where two ridges come together or a ridge ends.

LSH for Fingerprints

- Place a grid on a fingerprint.
  - Normalize so identical prints will overlap.
- Set of grid squares where minutiae are located represents the fingerprint.
- Possibly, treat minutiae near a grid boundary as if also present in adjacent grid points.

Discretizing Minutiae



Applying LSH to Fingerprints

- Fingerprint = set of grid squares.
- No need to minhash, since the number of grid squares is not too large.
- Represent each fingerprint by a bit-vector with one position for each square.
  - 1 in only those positions whose squares have minutiae.

LSH/Fingerprints – (2)

- Pick 1024 (?) sets of 3 (?) grid squares (components of the bit vectors), randomly.
- For each set of three squares, two prints that each have 1 for all three squares are candidate pairs.
- Funny sort of ‘bucketization.”
  - Each set of three squares creates one bucket.
  - Prints can be in many buckets.

Example: LSH/Fingerprints

- Suppose typical fingerprints have minutiae in 20% of the grid squares.
- Suppose fingerprints from the same finger agree in at least 80% of their squares.
- Probability two random fingerprints each have minutiae in all three squares = $(0.2)^6$ = .000064.

Example: Continued

First print has has minutia in this square

Second print of the same finger also has minutia in that square

- Probability two fingerprints from the same finger each have 1's in three given squares = $((0.2)(0.8))^3 = .004096$.
- Prob. for at least one of 1024 sets of three points = $1-(1-.004096)^{1024} = .985$.
- But for random fingerprints: $1-(1-.000064)^{1024} = .063$.

1.5% false negatives

6.3% false positives