Similarity Search
Vision Example

Have we met?
Similarity Search
Formal Definition

- **Query Space** - on-line
- **Database Space** - dynamic
- **Similarity Model**
- **Transformation Model**
- **Similarity Measure**
- **Similarity Threshold**

$$Q \subseteq \mathbb{R}$$
$$\forall q \in Q$$
$$DB \subseteq Q$$
$$\forall v \in DB$$

$$s(q, v) \rightarrow \{0, 1\}$$

$$T : Q \rightarrow Q$$

$$f : Q \times Q \rightarrow [0, 1] \subseteq \mathbb{R}$$

$$s(q, v) = \left[ f(q, T(v)) < \alpha \right]$$
Similarity Model

- Coordinates System Translation
- Amplitude Translation and Scaling
- Additive Noise
- Zero Mean
- Euclidean Unit Hypersphere
- Euclidean Norm
- Angle Threshold (Similarity variance)
Geometry

Inner Product

\[ \langle \mathbf{v} , \mathbf{u} \rangle = \sum_{i=1}^{d} v_i \cdot u_i = \cos(\angle(\mathbf{v}, \mathbf{u})) \]

Euclidean Norm

\[ \|\mathbf{v} - \mathbf{u}\|_2^2 = \langle \mathbf{v} - \mathbf{u} , \mathbf{v} - \mathbf{u} \rangle = \|\mathbf{v}\|_2^2 + \|\mathbf{v}\|_2^2 - 2 \cdot \langle \mathbf{v} , \mathbf{u} \rangle \]

Cross-Correlation

\[ E(\mathbf{u} \cdot \mathbf{v}) - E\mathbf{u} \cdot E\mathbf{v} \]

\[ \sqrt{Var(\mathbf{u}) \cdot Var(\mathbf{v})} \]

Unit Vector

\[ \|\mathbf{v} - \mathbf{u}\|_2 = \sqrt{2 \cdot (1 - \cos(\angle(\mathbf{v}, \mathbf{u})))} \]
Inner Product using
Fast Fourier Transform

\[ x_j = \langle v, u^j \rangle = \sum_{i=1}^{d} v_i \cdot q_{j+i} \]

\[ x = (x_1, \ldots, x_{d'-d+1}) \]

\[ x = \overline{v} \ast \overline{q}^* = \text{ifft} \left( \text{fft} (v) \ast \text{conj} (\text{ifft} (q)) \right) \]
Exhaustive Exact Search

- $n$ database elements.
- $d$ possible shifts in a query.
- Compare all database elements for every shift.
- Every inner-product costs $O(d)$ operations.
- Time: $O(n \, d^2)$
- I/O: $n$ - as a sequential scan.
kd-Tree

Figure 25: The k-d-B-tree.

Voronoi Diag: best for exact search when $n=\exp(d)$
Curse of Dimensionality
Exhaustive Win

- General Metric
- Discrete Metric
- Histogram of Distances
- Vector Space
- Cube volume grows exponentially
- Points are sparse
- The variance of the distances becomes small
Approximate Nearest Neighbor

- Very close to the most similar element (NN)
- Feature Extraction – Domain Specific & no FFT
- Indexing (inner products)
  - Randomized $kd$-tree - Yianilos 2000
  - Locally Sensitive Hashing – Indyk 2004
  - Sum-Synopsis – Cohen 2005
Randomized *kd*-Tree
Yianilos 2000

- Vector coordinates: i.i.d. random variables
- Uniform distribution (unit vector)
- Binary search tree based on projections
- Orthogonalized vectors as external pivots
- Redundancy: *l*-trees
- Inner-products $\sim N(0, d^{1/2})$
Locally Sensitive Hashing
Indyk 2004

- No assumptions on the input.
- External pivots from a $p$-Stable distribution.
- $N(0,1)$ is a 2-stable distribution.
- Hash function or
- Multi way search tree: projections and $r$ bins.
- Redundancy: $l$-trees of depth $k$
**p–Stable Distributions**

- **p–stable distribution** \((p, 0)\): A distribution \(D\) over \(\mathbb{R}\)
  - \(n\) real numbers \(v_1, \ldots, v_n\)
  - i.i.d. variables \(X_1, \ldots, X_n\) with distribution \(D\),ting \(\Sigma_i v_i X_i \sim (\Sigma_i |v_i|^p)^{1/p}X = l_p(v)X\)
  - \(X\) is a r.v. with distribution \(D\)

- **Cauchy distr is a** 1–Stable distribution

- **Gaussian distr is a** 2–Stable distribution

- for \(0 < p < 2\) there is a way to sample from a \(p–\)stable distribution given two uniform r.v.’s over \([0, 1]\)
p-Stable Distribution App.
taken from Indyk

- Using multiple independent $X$'s
- $a \cdot X - b \cdot X$ can be used to estimate $l_p(a - b)$
- Divide the real line into segments of width $w$
- Each segment defines a hash bucket, i.e. vectors that project onto the same segment belong to the same bucket
Sum-Synopsis

- Vector coordinates: i.i.d. random variables.
- Synopsis as the sum of annuli subsets.
- Synopses as external pivots.
- Binary search tree based on projections.
Spherical Collars

- Toroidal
- Annuli
- Annulus
- Ring
Empirical Evaluation

- No standard cost model.
- Counting Time, I/Os, Inner-products, FFTs.
- Uniform distribution
  - Maximized entropy
  - The example for the curse of dim.
  - Unrealistics.
- Sparsity and Homogeneity.
LSH & \textit{kd}-Tree

Time vs. $n$ ($d=100$)

$|q| = d$
LSH & $kd$-Tree

Time vs. Dimension ($n=10^5$)

$|q| = d$
Bless of Dimensionality?

\[ |q| = 2 \cdot d \]

SNR = 8db, Database size = \( 10^4 \), Accuracy level = 99.0

Methods parameters were selected to optimized speed.
Bless of Dimensionality?

Dimension = 128, Database size = $10^4$, Accuracy level = 99.0

$\text{SNR} = 20.0$ ($\alpha=0.1$), 14, 10.5, 8.0, 6.1, 4.5, 3.2 and 2.1 ($\alpha=0.8$) [db]

Methods parameters were selected to optimized speed.
Future Research

- Low-level operations count.
- Time vs. Database size.
- Time vs. Space.
- Insertion phase analysis.
- Change noise with respect to dimension.
- Time vs. Noise for other dimensions.
- Theoretical Analysis.
Var Proof

\[ \text{Var} (v_i) = \frac{1}{3} \]

\[ \text{Var} \left( \frac{v_i}{\|v\|} \right) = \frac{1}{d} \]

\[ \text{Var} (v_i \cdot u_i) = \frac{1}{9} \]

\[ \text{Var} \left( \frac{v_i}{\|v\|} \cdot u_i \right) = \frac{1}{3 \cdot d} \]

\[ \text{Var} \left( \frac{v_i}{\|v\|} \cdot \frac{u_i}{\|u\|} \right) = \frac{1}{d^2} \]

\[ \text{Var} (\langle v, u \rangle) = \frac{d}{9} \]

\[ \text{Var} (\langle \frac{v}{\|v\|}, u \rangle) = \frac{1}{3} \]

\[ \text{Var} (\langle \frac{v}{\|v\|}, \frac{u}{\|u\|} \rangle) = \frac{1}{d} \]

\[ \text{Var} \left( \sum_{i=1}^{\left|G \right|} \langle \frac{v}{\|v\|}, \frac{u}{\|u\|} \rangle \right) = \frac{\left|G \right|}{d} \]