Similarity Search in Multimedia Data

Databases and Information Systems

Fabian Panse

panse@informatik.uni-hamburg.de

University of Hamburg

Acknowledgements

These slides are based on slides provided by

- Prof. Dr. Thomas Seidl
  Ludwig-Maximilians-Universität München
  http://www.dbs.ifi.lmu.de/cms/

- Dr. Christian Beecks
  RWTH Aachen University
  http://dme.rwth-aachen.de/de

- Dipl.-Inform. Merih Seran Uysal
  RWTH Aachen University
  http://dme.rwth-aachen.de/de

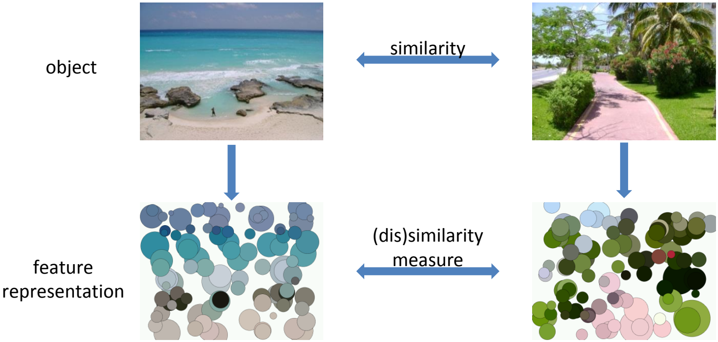
Introduction

- **Motivation:**
  - Explosive growth of multimedia data
  - Rapid spread of multimedia data (nowadays, almost all (mobile) devices allow to generate and share multimedia data)
- **How to search for multimedia data objects?**
  - A query is a description of the desired content and/or additional meta data (e.g. format, size, quality, location, time)
  - Most frequent query type: keyword(s)
- **Content-based querying:**
  - Keywords of multimedia data objects can be wrong, incomplete, ambiguous, or missing

  $\Rightarrow$ In addition to keywords, content-based access in terms of features is often desired (i.e. find objects which are similar to a given one)

Content-based Access

- **Similarity model:**
  - Feature representation describing the characteristic properties
  - (Dis)similarity measure comparing two feature representations

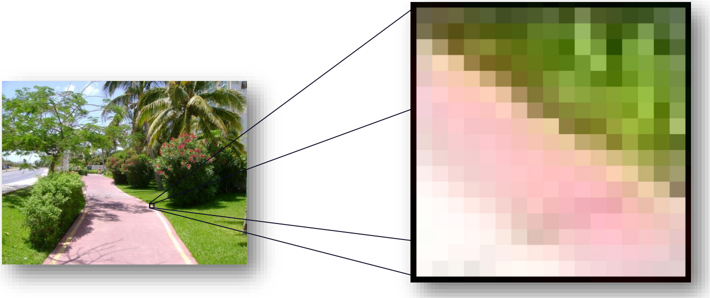


Feature Extraction

- **Feature of a multimedia data object:**
  - Mathematical description of an inherent property
  - Usually in the Euclidean space $\mathbb{R}^d$
- **Different types of features:**
  - Global features describe a multimedia data object as a whole
  - Local features describe parts of a multimedia data object
- **Different semantics of features:**
  - High-level features such as concepts, tags, etc.
  - Low-level features such as
    - color, texture, shape, etc. (images)
    - pitch, loudness, etc. (audio objects)
    - key-frame features, motion features, etc. (videos)

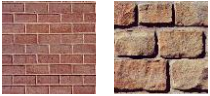
Example: Image Features

- An image is a matrix of pixels
- A pixel is an atomic element which has a certain color
- An image $\mathcal{I}$ of width $w \in \mathbb{N}$ and height $h \in \mathbb{N}$ is modeled as $\mathcal{I}(x, y) \rightarrow \mathbb{R}^d$ for $x \in \{1, \ldots, w\}$ and $y \in \{1, \ldots, h\}$
- Value d depends on color model (e.g. CMYK $d = 4$, RGB $d = 3$)



Tamura Features

- Six textural features corresponding to human visual perception proposed by Hideyuki Tamura et al. in 1978
- **Coarseness** is the most fundamental textural feature and reflects the size and the repetition of the texture elements
  - It increases with bigger element sizes and/or less element repetitions



- **Contrast** reflects the picture quality
  - Dynamic range of gray-levels,
  - Sharpness of edges
  - Period of repeating patterns

- **Directionality** measures the total degree of the direction of the patterns
  - It involves both element shape and placement

- **Line-likeness**, **regularity**, **roughness**

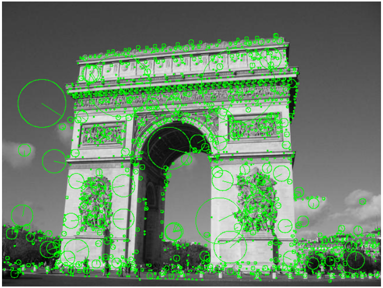
SIFT: Scale Invariant Feature Transform

- One of the most prominent local feature description method for images
- Proposed by David Lowe in 1999
- The SIFT method includes two parts:
  - Keypoint detection
  - Keypoint description
- A SIFT descriptor is a 128-dimensional vector that is invariant to
  - scale
  - translation
  - rotation

- A detailed analysis and implementation can be found at: `http://demo.ipol.im/demo/82/`

SIFT: Scale Invariant Feature Transform

- One of the most prominent local feature description method for images
- Proposed by David Lowe in 1999
- The SIFT method includes two parts:
  - Keypoint detection
  - Keypoint description
- A SIFT descriptor is a 128-dimensional vector that is invariant to
  - scale
  - translation
  - rotation



- A detailed analysis and implementation can be found at: `http://demo.ipol.im/demo/82/`

Advanced Feature Descriptors

- Current research aims at improving or approximating SIFT descriptors
- A multitude of local feature descriptors have been proposed recently:
  - **PCA-SIFT:** A more distinctive representation for local image descriptors
  - **CSIFT:** A SIFT descriptor with color invariant characteristics
  - **SURF:** Speeded-Up Robust Features
  - **ORB:** An efficient alternative to SIFT or SURF
  - **BRISK:** Binary Robust Invariant Scalable Keypoints
  - **BRIEF:** Computing a local binary descriptor very fast
  - **CHoG:** Compressed Histogram Of Gradients: A low-bitrate descriptor

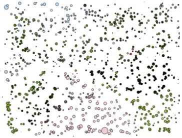
Software

- Many feature extraction and processing tools are available online:
  - **OpenCV:** Open source Computer Vision
    `http://opencv.org/`
  - **VLFeat:** a cross-platform open source collection of vision algorithms
    `http://www.vlfeat.org/`
  - **ImageJ:** Image Processing and Analysis in Java
    `http://rsbweb.nih.gov/ij/`
  - **OpenIMAJ:** Open Intelligent Multimedia Analysis toolkit for Java
    `http://www.openimaj.org/`
  - **Lire:** An Open Source Java Content Based Image Retrieval Library
    `http://www.semanticmetadata.net/lire/`
  - **Color Descriptor Software:** Binary for local feature extraction
    `http://koen.me/research/colordescriptors/`

Feature Representation

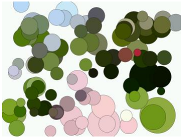
multimedia data object

feature extraction



features $f_1, \ldots, f_n \in \mathbb{F}$

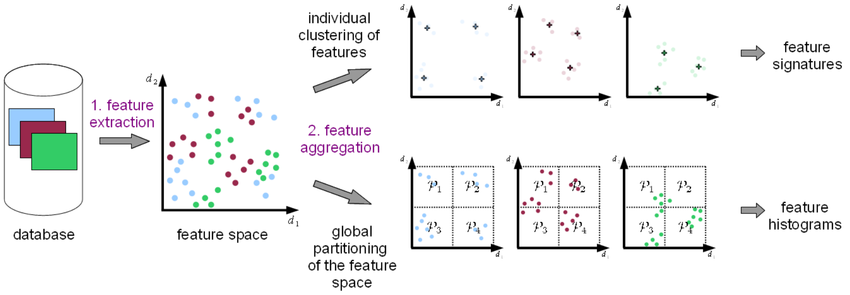
feature aggregation



feature representation

- **Feature extraction:** A multimedia data object is represented by means of features $f_1, \ldots, f_n \in \mathbb{F}$ in a feature space $\mathbb{F}$
  - SIFT features: $\mathbb{F} = \mathbb{R}^{128}$
- **Feature aggregation:** The features $f_1, \ldots, f_n$ are aggregated into a compact feature representation
  - clustering algorithms: k-means, expectation maximization, ...
- A feature representation is defined as a function $F \colon \mathbb{F} \to \mathbb{R}$

Feature Extraction and Aggregation



- **Different means of feature aggregation:**
  - **Feature Histogram:** features are summarized according to a global partitioning which is fixed for all multimedia data objects
  - **Feature Signature:** features are summarized individually (per object)

Feature Representation

- Given a feature space $\mathbb{F}$, a feature representation F is defined as:

$$F : \mathbb{F} \to \mathbb{R}$$

- The value of zero is designated for features that are not relevant for a certain multimedia data object
- The representatives $\mathrm{R}_F \subseteq \mathbb{F}$ of a feature representation F are defined as:

$$\mathrm{R}_F = \{f \in \mathbb{F} \mid F(f) \neq 0\}$$

- The weight of a single feature $f \in \mathbb{F}$ is defined as $F(f) \in \mathbb{R}$

Feature Signature

- A feature signature S is defined as:

$$S\colon \mathbb{F} \to \mathbb{R} \quad \text{subject to} \quad |\mathrm{R}_S| < \infty$$

- A multimedia data object is described by a finite number of features
- These features are the representatives $\mathrm{R}_S = \{f \in \mathbb{F} \mid S(f) \neq 0\}$
- Two feature signatures S_1 and S_2 may differ in their representatives and weights

Feature Histogram

- Let $\mathbb{F}$ be a feature space and $\mathrm{R} \subseteq \mathbb{F} \wedge |\mathrm{R}| < \infty$ be shared representatives
- A feature histogram H_R w.r.t. the shared representatives R is defined as:

$$H_\mathrm{R} \colon \mathbb{F} \to \mathbb{R} \quad \text{subject to} \quad H_\mathrm{R}(\mathbb{F} \setminus \mathrm{R}) = \{0\}$$

- Every multimedia data object is described by the same finite number of features, i.e. the shared representatives R
- Two feature histograms H_R^1 and H_R^2 can only differ in their weights

Relations of Feature Representations

- Class of feature representations:

$$\mathbb{R}^{\mathbb{F}} = \{F \mid F \colon \mathbb{F} \to \mathbb{R}\}$$

- Class of feature signatures:

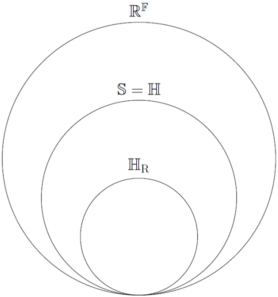
$$\mathbb{S} = \{S \mid S \in \mathbb{R}^{\mathbb{F}} \wedge |\mathsf{R}_S| < \infty\}$$

- Class of feature histograms w.r.t. $\mathsf{R} \subseteq \mathbb{F}, |\mathsf{R}| < \infty$:

$$\mathbb{H}_{\mathsf{R}} = \{H \mid H \in \mathbb{R}^{\mathbb{F}} \wedge H_{\mathsf{R}}(\mathbb{F} \setminus \mathsf{R}) = \{0\}\}$$

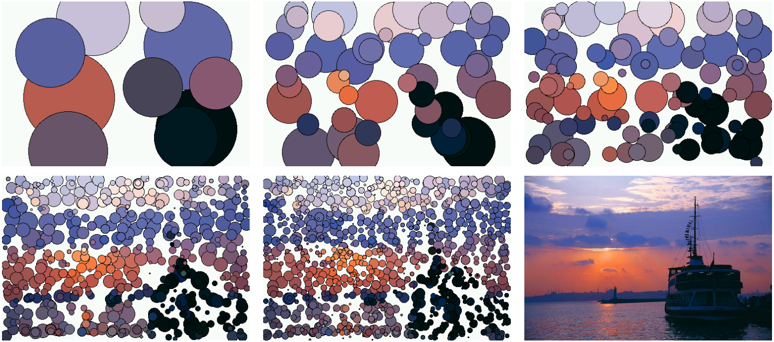
- Union of all feature histograms:

$$\mathbb{H} = \bigcup_{\mathsf{R} \subseteq \mathbb{F}, |\mathsf{R}| < \infty} \mathbb{H}_{\mathsf{R}} = \mathbb{S}$$



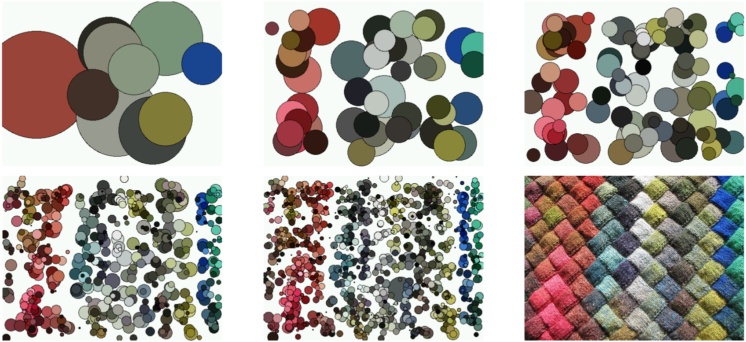
Example: Feature Signatures

- 7-dimensional features: position, color, coarseness, and contrast
- Random sampling of 40.000 image pixels
- Increasing the number of representatives from 10 to 1000:



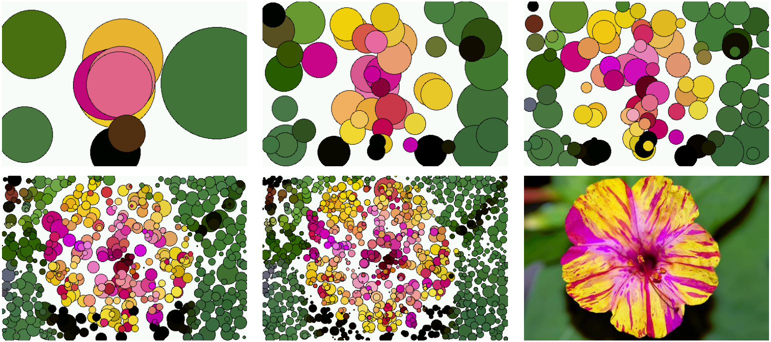
Example: Feature Signatures

- 7-dimensional features: position, color, coarseness, and contrast
- Random sampling of 40.000 image pixels
- Increasing the number of representatives from 10 to 1000:



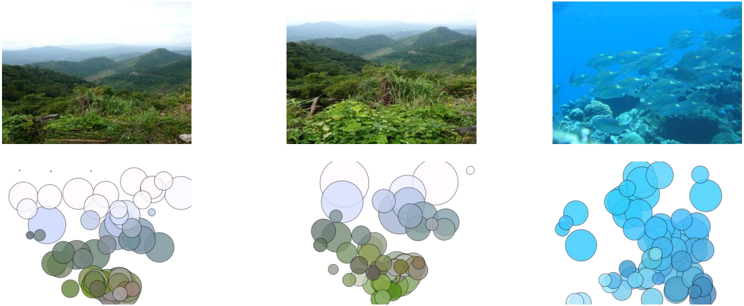
Example: Feature Signatures

- 7-dimensional features: position, color, coarseness, and contrast
- Random sampling of 40.000 image pixels
- Increasing the number of representatives from 10 to 1000:



Similarity vs. Dissimilarity

object o_1 object o_2 object o_3



- A similarity measure *sim* assigns high values to similar objects:
  - $sim(o_1, o_2) \geq sim(o_1, o_3)$
- A dissimilarity measure δ assigns low values to similar objects:
  - $\delta(o_1, o_2) \leq \delta(o_1, o_3)$

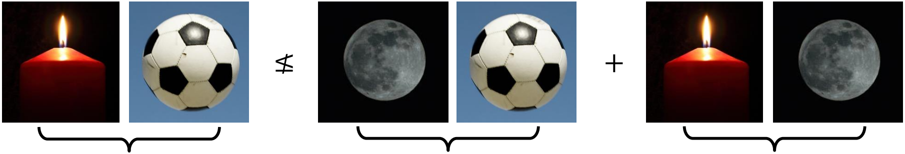
Similarity Function & Metric Distance Function

- A **similarity function** $s\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ quantifies the similarity between two elements from a set $\mathbb{X}$ and satisfies the following properties:
  - **Symmetry:** $\forall x, y \in \mathbb{X}\colon s(x, y) = s(y, x)$
  - **Maximum self-similarity:** $\forall x, y \in \mathbb{X}\colon s(x, x) \geq s(x, y)$

- Geometric distance between the feature representations defines dissimilarity of multimedia objects
- A function $\delta\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}^{\geq 0}$ is called a **metric distance function** if it satisfies the following properties:
  - **Identity of indiscernibles:** $\forall x, y \in \mathbb{X}\colon \delta(x, y) = 0 \Leftrightarrow x = y$
  - **Non-negativity:** $\forall x, y \in \mathbb{X}\colon \delta(x, y) \geq 0$
  - **Symmetry:** $\forall x, y \in \mathbb{X}\colon \delta(x, y) = \delta(y, x)$
  - **Triangle inequality:** $\forall x, y, z \in \mathbb{X}\colon \delta(x, y) \leq \delta(x, z) + \delta(z, y)$

From a Psychological Perspective . . .

- The distance-based approach has the advantage of a rigorous mathematical interpretation
- There is a long-lasting discussion of whether the distance properties and in particular the metric properties reflect the perceived dissimilarity correctly
- Consider the following example, where it holds that $\delta(\mathit{flame}, \mathit{ball}) \nleq \delta(\mathit{moon}, \mathit{ball}) + \delta(\mathit{flame}, \mathit{moon})$:



no properties shared alike

similar w.r.t. roundness

similar w.r.t. luminosity

Taking a closer look at this example . . .

- Validity clearly depends on the (dis)similarity model
- Consider the following two-dimensional "binary" feature representations

luminosity: roundness:	$\begin{pmatrix}1\\0\end{pmatrix}$	$\begin{pmatrix}0\\1\end{pmatrix}$	$\begin{pmatrix}1\\1\end{pmatrix}$

- Applying the Euclidean distance $L_2(x,y) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$ yields:
  - $L_2(\mathit{flame}, \mathit{ball}) = L_2\left(\binom{1}{0}, \binom{0}{1}\right) = \sqrt{2} \simeq 1.41$
  - $L_2(\mathit{ball}, \mathit{moon}) = L_2\left(\binom{0}{1}, \binom{1}{1}\right) = 1$
  - $L_2(\mathit{flame}, \mathit{moon}) = L_2\left(\binom{1}{0}, \binom{1}{1}\right) = 1$

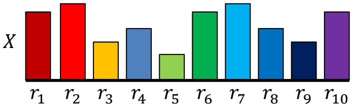
$$\Rightarrow \quad \delta(\text{flame}, \text{ball}) \leq \delta(\text{moon}, \text{ball}) + \delta(\text{flame}, \text{moon}) \quad \checkmark$$

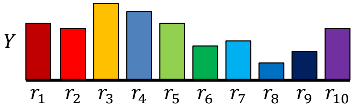
From a Database Perspective . . .

- The distance-based approach provides a powerful tool
- Metric distance functions allow
  - domain experts to model their notion of dissimilarity
  - database experts to design efficient query processing approaches (particularly the utilization of the triangle inequality)
- Thus, indexing approaches can be investigated without knowing the inner-workings of a metric distance function

Distance Functions for Feature Histograms

- Given two feature histograms $X, Y \in \mathbb{H}_{\mathrm{R}}$, how can we define a distance between them?
- Consider the following color histograms for $\mathrm{R} = \{r_1, r_2, \dots, r_{10}\}$



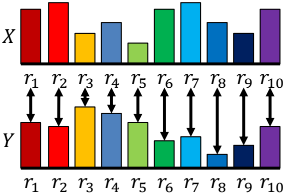


Minkowski Distance

- **Idea:** Measure the dissimilarity by adding up the differences in all dimensions, i.e. for all representatives $f \in \mathrm{R} \subseteq \mathbb{F}$
- Given two feature histograms $X, Y \in \mathbb{H}_{\mathrm{R}}$, the Minkowski Distance is defined for $p \in \mathbb{R}^{\geq 0} \cup \{\infty\}$ as:

$$L_p(X, Y) = \left(\sum_{f \in \mathrm{R}} |X(f) - Y(f)|^p\right)^{\frac{1}{p}}$$

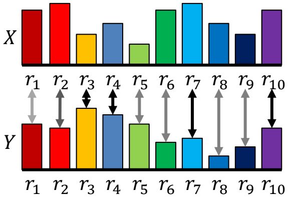
- This corresponds to taking into account all pairwise differences:



Weighted Minkowski Distance

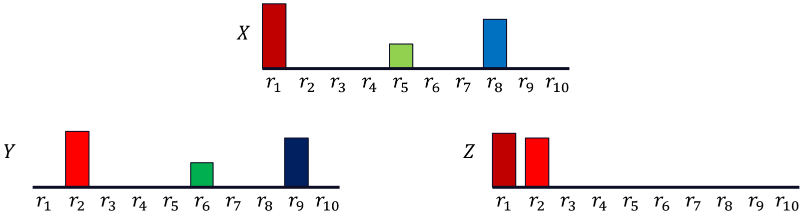
- **Idea:** Model the influence of the shared representatives $\mathrm{R} \subseteq \mathbb{F}$ by a weighting function $w\colon \mathbb{F} \to \mathbb{R}^{\geq 0}$
- Given two feature histograms $X, Y \in \mathbb{H}_{\mathrm{R}}$, the Weighted Minkowski Distance is defined for $p \in \mathbb{R}^{\geq 0} \cup \{\infty\}$ and a weighting function w as:

$$L_p(X, Y) = \left(\sum_{f \in \mathrm{R}} w(f) \times |X(f) - Y(f)|^p \right)^{\frac{1}{p}}$$



Issues of Bin-by-bin Distance Functions

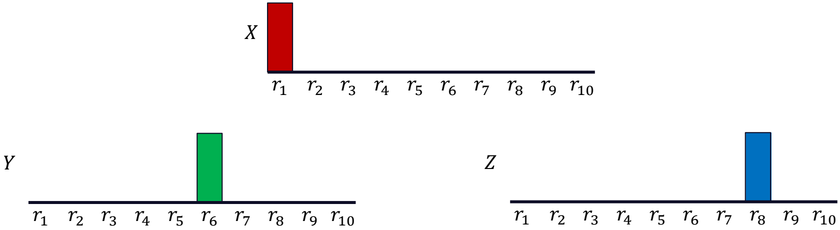
- Bin-by-bin distance functions define a distance value by taking into account single representatives (dimensions)
- Neighboring representatives (dimensions) are neglected
- Consider the following color histograms $X, Y, Z \in \mathbb{H}_\mathrm{R}$ with $\mathrm{R} = \{r_1, \ldots, r_{10}\}$



- In this example, it holds that $L_p(X, Y) \geq L_p(X, Z)$

Issues of Bin-by-bin Distance Functions

- Consider the following color histograms $X, Y, Z \in \mathbb{H}_R$



- All color histograms $X, Y, Z \in \mathbb{H}_R$ result in the same Minkowsi Distance: $L_p(X, Y) = L_p(X, Z) = L_p(Y, Z)$
- The fact that the color green is more similar to the color blue than to the color red is not taken into account

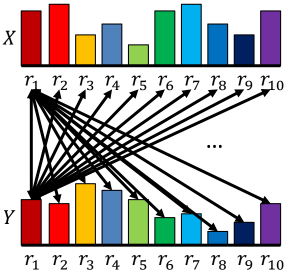
Cross-bin Distance Functions

- More flexible than bin-by-bin distance functions
- **Basic Ideas:**
  - Replace the weighting of single representatives by a weighting of pairs of representatives
  - Model the influence not only for each single representative, but also among different representatives
  - This influence is often defined in terms of a similarity relation
  - Thus, we can utilize a similarity function $s\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ in order to define the influence for all pairs of features

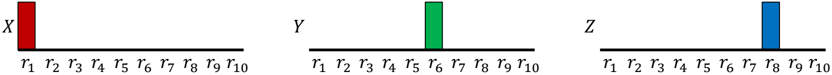
Quadratic Form Distance

- The Quadratic Form Distance is a cross-bin distance function that takes into account all pair-wise similarities
- Given two feature histograms $X, Y \in \mathbb{H}_{\mathrm{R}}$, the Quadratic Form Distance w.r.t. a similarity function $s\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ is defined as:

$$QFD_s(X, Y) = \sqrt{\sum_{f \in \mathrm{R}} \sum_{g \in \mathrm{R}} (X(f) - Y(f)) \times s(f, g) \times (X(g) - Y(g))}$$



Quadratic Form Distance: Example



$\boldsymbol{x} = (1{,}0, \ldots, 0)$ $\quad$ $\boldsymbol{y} = (0, \ldots, 0{,}1{,}0, \ldots, 0)$ $\quad$ $\boldsymbol{z} = (0, \ldots, 0{,}1{,}0{,}0)$

- Let $s(r_i, r_i) = 1$, $s(r_1, r_6) = s(r_1, r_8) = 0.2$ and $s(r_6, r_8) = 0.6$
- The Quadratic Form Distance is as follows:
  - $QFD_s(X, Y) = \sqrt{1.6} \simeq 1.265$
  - $QFD_s(X, Z) = \sqrt{1.6} \simeq 1.265$
  - $QFD_s(Y, Z) = \sqrt{0.8} \simeq 0.894$
- Better fits our intuition of dissimilarity ✓

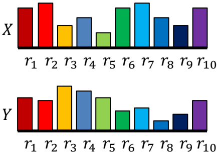
Distance Functions for Feature Histograms

- Distance functions are defined for feature histograms w.r.t. the same shared representatives
- Weighted Minkowski Distances are limited w.r.t. adaptability but show linear computation time complexity
- Quadratic Form Distances are very adaptable but show quadratic computation time complexity
- Other distance functions
  - Geometric measures such as cosine distance
  - Information theoretic measures such as Kullback-Leibler
  - Statistic measures such as χ^2-statistics

Conceptual Differences of Feature Representations

Feature histograms $\mathbb{H}_\mathbf{R}$

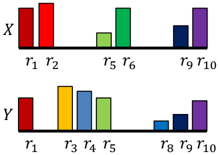
- Multimedia data objects use the same shared representatives:
  - Sufficient to store the weights
  - Feature histograms have the same cardinality
  - Can be thought of as vectors (representatives = dimensions)



  - Distance computation by means of differences in each dimension

Feature signatures $\mathbb{S}$

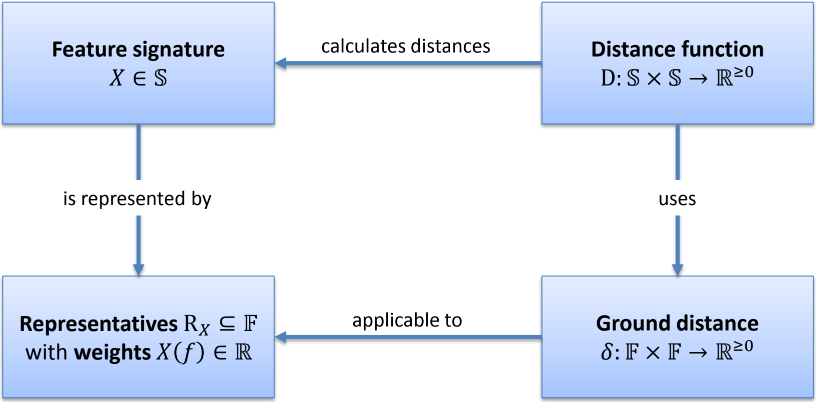
- Multimedia data objects use individual representatives:
  - Weights and representatives have to be stored
  - Feature signatures have different cardinalities



  - Distance computation along single dimensions not meaningful

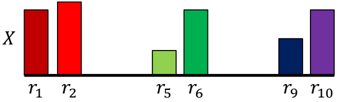
Concept of Using Ground Distance

- **Idea:** Utilization of a ground distance $\delta\colon \mathbb{F} \times \mathbb{F} \rightarrow \mathbb{R}^{\geq 0}$ on the representatives $\mathrm{R}_X, \mathrm{R}_Y \subseteq \mathbb{F}$ of two feature signatures $X, Y \in \mathbb{S}$

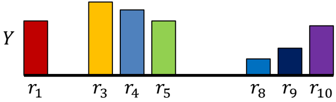


Earth Mover's Distance: Principle

- Given two color signatures $X, Y \in \mathbb{S}$



$\mathrm{R}_X = \{r_1, r_2, r_5, r_6, r_9, r_{10}\}$



$\mathrm{R}_Y = \{r_1, r_3, r_4, r_5, r_8, r_9, r_{10}\}$

- The transportation (earth moving) problem is formalized by:
  - Earth hills R_X with capacities $X(r_i)$ for $r_i \in \mathrm{R}_X$
  - Earth holes R_Y with capacities $Y(r_i)$ for $r_i \in \mathrm{R}_Y$
  - Cost (ground distance) $\delta : \mathbb{F} \times \mathbb{F} \to \mathbb{R}^{\geq 0}$ for moving a unit of earth
  - All possible flows $F = \{f \mid f : \mathrm{R}_X \times \mathrm{R}_Y \to \mathbb{R}^{\geq 0}\}$
- **Solution:** flow $f_{min} \in F$ that minimizes $\sum_{g \in \mathrm{R}_X, h \in \mathrm{R}_Y} f_{min}(g, h) \times \delta(g, h)$

Earth Mover's Distance: Definition

- Given two feature signatures $X, Y \in \mathbb{S}$ over a feature space $\mathbb{F}$, the Earth Mover's Distance $\mathrm{EMD}_\delta : \mathbb{S} \times \mathbb{S} \to \mathbb{R}$ between X and Y is defined as:

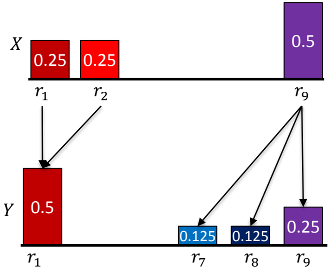
$$\mathrm{EMD}_\delta(X, Y) = \min_{\{f \mid f : \mathrm{R}_X \times \mathrm{R}_Y \to \mathbb{R}^{\geq 0}\}} \left(\frac{\sum_{g \in \mathrm{R}_X} \sum_{h \in \mathrm{R}_Y} f(g, h) \times \delta(g, h)}{\min\left(\sum_{g \in \mathrm{R}_X} X(g), \sum_{h \in \mathrm{R}_Y} Y(h)\right)} \right)$$

subject to the constraints:

- CNNeg: $\forall g \in \mathrm{R}_X, \forall h \in \mathrm{R}_Y : f(g, h) \geq 0$
- CSource: $\forall g \in \mathrm{R}_X : \sum_{h \in \mathrm{R}_Y} f(g, h) \leq X(g)$
- CTarget: $\forall h \in \mathrm{R}_Y : \sum_{g \in \mathrm{R}_X} f(g, h) \leq Y(h)$
- CMaxFlow: $\sum_{g \in \mathrm{R}_X, h \in \mathrm{R}_Y} f(g, h) = \min\left(\sum_{g \in \mathrm{R}_X} X(g), \sum_{h \in \mathrm{R}_Y} Y(h)\right)$

Earth Mover's Distance: Example

- Consider the following two color signatures $X, Y \in \mathbb{S}$



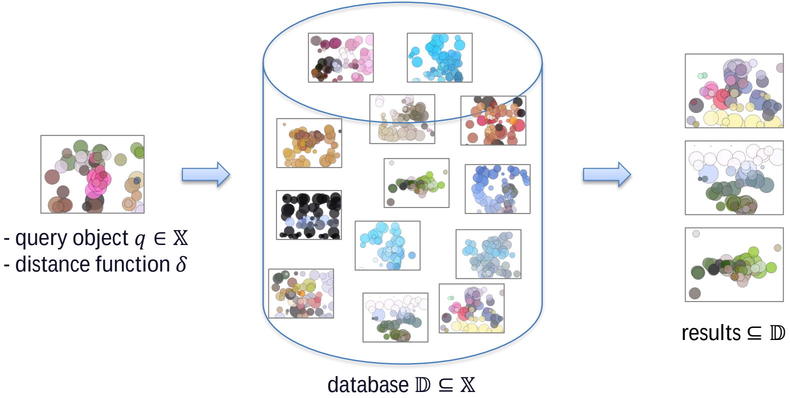
- Given the ground distance $\delta(r_i, r_j) = |i - j|$, we obtain the following distance value:

$$\begin{aligned} \mathrm{EMD}_\delta(X, Y) &= f(r_2, r_1) \times \delta(r_2, r_1) + f(r_9, r_7) \times \delta(r_9, r_7) + f(r_9, r_8) \times \delta(r_9, r_8) \\ &= 0.25 \times 1 + 0.125 \times 2 + 0.125 \times 1 \\ &= 0.625 \end{aligned}$$

Earth Mover's Distance: Properties

- The Earth Mover's Distance is defined as a linear optimization problem
- Finding an optimal solution can be computed based on a specific variant of the simplex algorithm
- Exponential computation time complexity in the worst case
- Average empirical computation time complexity between $\mathcal{O}(|\mathrm{R}_X|^3)$ and $\mathcal{O}(|\mathrm{R}_X|^4)$ for $|\mathrm{R}_X| \geq |\mathrm{R}_Y|$
- More efficient algorithms for specific classes of δ
- Earth Mover's Distance is a metric if and only if
    - feature signatures are normalized, i.e. $\sum_{f \in \mathrm{R}_X} X(f) = \sum_{f \in \mathrm{R}_Y} Y(f)$
    - ground distance δ is a metric

Distance-based Similarity Query

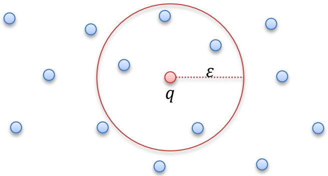


- Different query types:
  - Range Query
  - K-Nearest-Neighbor Query
  - Ranking Query
  - (Top-k Query)
  - (Skyline Query)
  - (Reverse Nearest-Neighbor Query)

Range Query

- Range query includes database objects whose distances to a query object lie within a specific threshold
- Let $\mathbb{X}$ be a set, $\delta\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ be a distance function, $\mathbb{D} \subseteq \mathbb{X}$ be a finite database, and $q \in \mathbb{X}$ be a query object
- The query $range_\epsilon(q, \delta, \mathbb{D})$ is defined w.r.t. the range $\epsilon \in \mathbb{R}^{\geq 0}$ as:

$$range_\epsilon(q, \delta, \mathbb{D}) = \{x \in \mathbb{D} \mid \delta(q, x) \leq \epsilon\}$$

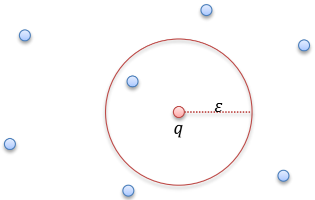


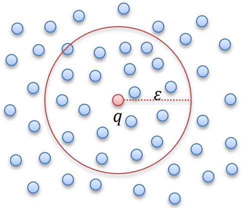
Range Query: Properties

- By performing a sequential scan (also linear scan or naïve scan) of the entire database $\mathbb{D}$, the computation time complexity lies in $\mathcal{O}(|\mathbb{D}|)$
- The result size is bounded by the database size, i.e. it holds that

$$|range_{\epsilon}(q, \delta, \mathbb{D})| \leq |\mathbb{D}|$$

- **Problem:** How to choose an appropriate range $\epsilon \in \mathbb{R}^{\geq 0}$?
  - Different data scales can result in very small or very large result sets





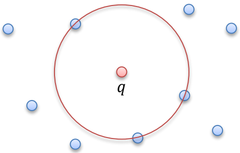
K-Nearest-Neighbor Query

- K-nearest-neighbor query includes database objects up to the k^{th}-smallest distance to a query object
- Let $\mathbb{X}$ be a set, $\delta\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ be a distance function, $\mathbb{D} \subseteq \mathbb{X}$ be a finite database, and $q \in \mathbb{X}$ be a query object
- The query $NN_k(q, \delta, \mathbb{D})$ is defined w.r.t. the number of nearest neighbors $k \in \mathbb{N}$ as the smallest set $NN_k(q, \delta, \mathbb{D}) \subseteq \mathbb{D}$ with $|NN_k(q, \delta, \mathbb{D})| \geq k$ such that

$$\forall x \in NN_k(q, \delta, \mathbb{D}), \forall x' \in \big(\mathbb{D} \setminus NN_k(q, \delta, \mathbb{D})\big)\colon \delta(q, x) \leq \delta(q, x')$$

K-Nearest-Neighbor Query: Properties

- By performing a sequential scan of the entire database $\mathbb{D}$, the computation time complexity lies in $\mathcal{O}(|\mathbb{D}|)$
- If the distances between the query object and the data objects are unique, i.e. if it holds that $\forall x, x' \in \mathbb{D}\colon \delta(q,x) \neq \delta(q,x')$, the result size is bounded by the minimum of database size and parameter k, i.e. it holds that
$$|NN_k(q,\delta,\mathbb{D})| \leq \min(k, |\mathbb{D}|)$$
- If two or more objects have the same distance to the query object, $NN_k(q,\delta,\mathbb{D})$ can comprise more than k objects



In this example:
$|\mathrm{NN}_1(q,\delta,\mathbb{D})| = 3$

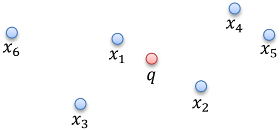
Ranking Query

- Ranking query sorts a database in ascending order w.r.t. the distances to a query object
- Let $\mathbb{X}$ be a set, $\delta\colon \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$ be a distance function, $\mathbb{D} \subseteq \mathbb{X}$ be a finite database, and $q \in \mathbb{X}$ be a query object
- The query $\mathit{ranking}(q, \delta, \mathbb{D})$ is a sequence of $\mathbb{D}$ that is defined as:

$$\mathit{ranking}(q, \delta, \mathbb{D}) = x_1, \ldots, x_{|\mathbb{D}|}$$

where it holds that

$$\delta(q, x_i) \leq \delta(q, x_j) \text{ for all } x_i, x_j \in \mathbb{D} \text{ and } 1 \leq i \leq j \leq |\mathbb{D}|$$



Ranking Query: Properties

- The computation time complexity of this ranking algorithm depends on the computation time complexity of the sorting algorithm
  - In general: $\mathcal{O}(|\mathbb{D}| \times \log(|\mathbb{D}|))$
- The cardinality of $ranking(q, \delta, \mathbb{D})$ can be restricted by nesting the ranking query with other query types
  - Sorted sequence of the k^{th}-nearest neighbors:

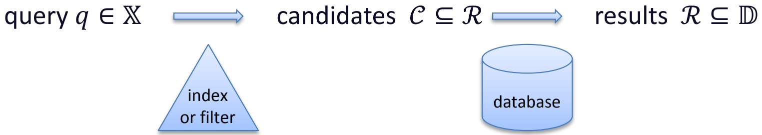
$$ranking(q, \delta, NN_k(q, \delta, \mathbb{D}))$$

  - Sorted sequence of data objects within range ϵ:

$$ranking(q, \delta, range_\epsilon(q, \delta, \mathbb{D}))$$

Multi-Step Query Architecture

- Processing of distance-based similarity queries in multiple steps
- Filter step is applied to all database objects
  - Efficient generation of candidates
  - Use of approximations
- Refinement step only necessary on candidates
  - Use of exact distances
  - Correctness: do not return wrong objects
  - Completeness: do not discard correct objects
  - Efficiency: short response times



Complexity of Distance-based Similarity Queries

- **Problem:** Quality determines complexity
  - High dimensionality $\Rightarrow$ better quality
  - Complex distance measure (e.g. Earth Mover's) $\Rightarrow$ better quality
  - But: both require much computing time
- **Solution:** Filter step for reduction of expensive computations
  - Consider a range query $range_\epsilon(q, \delta, \mathbb{D})$
  - Choose a filter distance δ_{filter} with small computational effort
  - Discard all objects with $\delta_{filter} > \epsilon$
  - Necessary condition: filter distance is a lower bound of the exact distance, i.e.

$$\forall x, y \in \mathbb{X}:$$

$$\delta_{filter}(x,y) \leq \delta(x,y)$$

$$\delta_{filter}(x,y) > \varepsilon \Rightarrow \delta(x,y) > \varepsilon$$

Lower Bound

- Let $\mathbb{X}$ be a set and $\delta\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ be a distance function. A function $\delta_{LB}\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ is a lower bound of δ if it holds that:

$$\forall x, y \in \mathbb{X}\colon \delta_{LB}(x, y) \leq \delta(x, y)$$

- Two approaches of deriving a lower bound:
  - Model-specific approaches which exploit the inner workings of a distance function
  - Generic approaches which exploit the properties of the corresponding metric distance space $(\mathbb{X}, \delta)$
- δ_{LB} is also denoted as the filter (distance) of δ, denoted by $\delta_{LB} \leq \delta$
- Quality of a lower bound depends on the ICES criteria

ICES Criteria for Lower Bounds

- **Indexable:**
  - Filter function should be indexable in order to be applied with an index structure
- **Complete:**
  - No correct answers are dismissed in the filter step
  - There are approximate systems with limited completeness and correctness, e.g. PAC-NN (probably approximate correct)
- **Efficient:**
  - Fast computation of filter distance, e.g., linear complexity w.r.t. dimensionality
- **Selective:**
  - Small candidate set generated in the filter step
  - The larger the filter distance δ_{filter}, the better the filter selectivity

Multi-Step Range Query

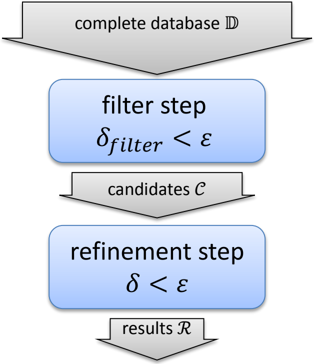
- Given a set $\mathbb{X}$, a database $\mathbb{D} \subseteq \mathbb{X}$, and a distance function $\delta\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$
- Given a lower bound $\delta_{LB}\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ of δ, how to process a query $range_\epsilon(q, \delta, \mathbb{D}) = \{x \in \mathbb{D} \mid \delta(q, x) \leq \epsilon\}$ efficiently?
- **Process:**
  - **Filter step:** evaluate range query with the same $\epsilon \in \mathbb{R}$ but cheaper filter distance δ_{LB} to generate the candidates

$$\mathcal{C} = \{x \in \mathbb{D} \mid \delta_{LB}(q, x) \leq \epsilon\}$$

  - **Refinement step:** refine candidates with the exact distance δ to obtain the results

$$\mathcal{R} = \{x \in \mathcal{C} \mid \delta(q, x) \leq \epsilon\}$$

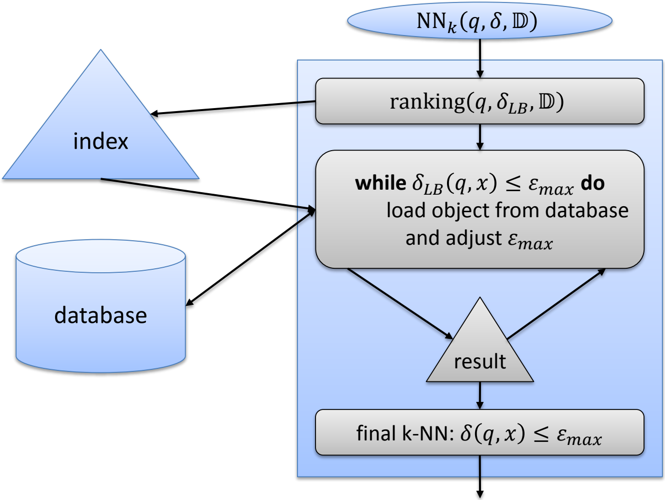
- It holds that $\mathcal{R} = range_\epsilon(q, \delta, \mathbb{D})$ iff $\delta_{LB} \leq \delta$



Optimal Multi-Step k-NN Query

- Given a set $\mathbb{X}$, a database $\mathbb{D} \subseteq \mathbb{X}$, and a distance function $\delta\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$
- How to process a query $NN_k(q, \delta, \mathbb{D})$ efficiently by means of a lower bound $\delta_{LB}\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ and an optimal number of candidates?
- **Idea:**
  - Utilization of a ranking query
  - Adaptation of ϵ_{max} after each object
- **Properties:**
  - It can be shown that the resulting algorithm is complete
  - It can be shown that the number of candidates is optimal (minimal)
- **Note:**
  - $\delta_{LB}(q, x) > \delta_{LB}(q, y) \not\Rightarrow \delta(q, x) > \delta(q, y)$

Optimal Multi-Step k-NN Query



Optimal Multi-Step k-NN Query: Pseudo Code

procedure $\mathrm{NN}_k(q, \delta, \mathbb{D})$:

$results\ \mathcal{R} \leftarrow \emptyset$

$filterRanking \leftarrow \mathrm{ranking}(q, \delta_{LB}, \mathbb{D})$

$x \leftarrow filterRanking.getnext()$

$\varepsilon_{max} \leftarrow \infty$

while $\delta_{LB}(q, x) \leq \varepsilon_{max}$ **do**

  if $|\mathcal{R}| < k$ **then**

    $\mathcal{R} \leftarrow \mathcal{R} \cup \{x\}$

  else if $\delta(q, x) \leq \varepsilon_{max}$ **then**

    $\mathcal{R} \leftarrow \mathcal{R} \cup \{x\}$

    $\mathcal{R} \leftarrow \mathcal{R} - \left\{\underset{r \in \mathcal{R}}{\mathrm{argmax}}\, \delta(q, y)\right\}$

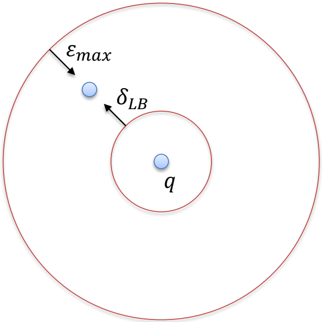
    $\varepsilon_{max} \leftarrow \max_{y \in \mathcal{R}} \delta(q, y)$

  $x \leftarrow filterRanking.getnext()$

return $\mathcal{R}$

Optimal Multi-Step k-NN Query: Properties

- **Observation:**
  - Pruning distance ϵ_{max} decreases
  - Filter distance δ_{LB} increases
  - Algorithm terminates when $\delta_{LB} \geq \epsilon_{max}$



Optimal Multi-Step k-NN Query: Example

Given:

- objects $o_1 - o_7$
- distance function δ, lower bound function δ_{LB}
- $k = 3$

	δ_{LB}	δ	k-NN	ϵ_{max}	remark
o_1	0.01	0.01	$\{o_1\}$	∞	
o_2	0.2	0.25	$\{o_1, o_2\}$	∞	
o_3	0.25	0.35	$\{o_1, o_2, o_3\}$	0.35	
o_4	0.27	0.3	$\{o_1, o_2, o_4\}$	0.3	
o_5	0.28	0.4	$\{o_1, o_2, o_4\}$	0.3	$\epsilon_{max} < \delta_{LB} \Rightarrow$ stop
o_6	0.4	-	-	-	
o_7	0.42	-	-	-	

Optimal Multi-Step k-NN Query: Example

Given:

- objects $o_1 - o_7$
- distance function δ, lower bound function δ_{LB}
- $k = 3$

	δ_{LB}	δ	k-NN	ϵ_{max}	remark
o_1	0.01	0.01	$\{o_1\}$	∞	
o_2	0.2	0.25	$\{o_1, o_2\}$	∞	
o_3	0.25	0.35	$\{o_1, o_2, o_3\}$	0.35	
o_4	0.27	0.3	$\{o_1, o_2, o_4\}$	0.3	
o_5	0.28	0.4	$\{o_1, o_2, o_4\}$	0.3	$\epsilon_{max} < \delta_{LB} \Rightarrow$ stop
o_6	0.4	0.5	$\{o_1, o_2, o_4\}$	0.3	saved computations
o_7	0.42	0.45	$\{o_1, o_2, o_4\}$	0.3	

Lower Bound of Minkowski Distance

- Given two feature histograms $X, Y \in \mathbb{H}_{\mathrm{R}}$ and the Minkowski Distance

$$L_p(X,Y) = \left(\sum_{f \in \mathrm{R}} |X(f) - Y(f)|^p \right)^{\frac{1}{p}}$$

- Any subset R' $\subseteq$ R defines a lower bound, i.e. it holds for all $X, Y \in \mathbb{H}_{\mathrm{R}}$

$$\begin{aligned} L_p(X|_{\mathrm{R'}}, Y|_{\mathrm{R'}}) &= \left(\sum_{f \in \mathrm{R'}} |X(f) - Y(f)|^p \right)^{\frac{1}{p}} \\ &\leq \left(\sum_{f \in \mathrm{R}} |X(f) - Y(f)|^p \right)^{\frac{1}{p}} = L_p(X,Y) \end{aligned}$$

Generic Lower Bound of EMD

- Given two ground distance functions $\delta, \delta_{LB} \colon \mathbb{F} \times \mathbb{F} \to \mathbb{R}^{\geq 0}$ with $\delta_{LB} \leq \delta$ it holds for all feature signatures $X, Y \in \mathbb{S}$ that:

$$\mathrm{EMD}_{\delta_{LB}}(X, Y) \leq \mathrm{EMD}_{\delta}(X, Y)$$

- **Proof:**
  - Let $m = \min\left(\sum_{g \in \mathrm{R}_X} X(g), \sum_{h \in \mathrm{R}_Y} Y(h)\right)$ be the minimum total weight of X and Y
  - Let the flow $f_{min} \in \mathbb{R}^{\mathbb{F} \times \mathbb{F}}$ define a minimum solution:

$$\begin{aligned}
\mathrm{EMD}_{\delta}(X, Y) &= \tfrac{1}{m}\left(\sum_{g,h \in \mathbb{F}} f_{min}(g, h) \times \delta(g, h)\right) \\
&\geq \tfrac{1}{m}\left(\sum_{g,h \in \mathbb{F}} f_{min}(g, h) \times \delta_{LB}(g, h)\right) \\
&\geq \mathrm{EMD}_{\delta_{LB}}(X, Y)
\end{aligned}$$

Centroid-based Lower Bound

- Given a ground distance function $\delta \colon \mathbb{F} \times \mathbb{F} \rightarrow \mathbb{R}^{\geq 0}$, it holds for all feature signatures $X, Y \in \mathbb{S}$ that:

$$LB_{Rubner}(X, Y) = \delta(\overline{x}, \overline{y}) \leq \mathrm{EMD}_{\delta}(X, Y)$$

where $\overline{x}, \overline{y} \in \mathbb{F}$ are defined as the centroids (mean representatives) of X and Y, i.e. $\overline{x} = \sum_{g \in \mathrm{R}_X} g \times X(g)$ and $\overline{y} = \sum_{h \in \mathrm{R}_Y} h \times Y(h)$ if the total weight for every signature is 1.

- **Properties:**
  - LB_{Rubner} is applicable to feature signatures and histograms
  - Centroids can be computed prior to query processing
  - Computation time complexity of LB_{Rubner} solely depends on the dimensionality of the feature space and not on the size of the feature representations

Independent Minimization Lower Bound of EMD

- **Idea:** Approximation of the EMD through constraint relaxation
- **Approach:** Given two feature signatures $X, Y \in \mathbb{S}_m$, the Earth Mover's Distance is defined w.r.t. a metric ground distance δ as:

$$\mathrm{EMD}_\delta(X, Y) = \min_{\{f \mid f\colon \mathrm{R}_X \times \mathrm{R}_Y \to \mathbb{R}^{\geq 0}\}} \frac{1}{m}\Big(\sum_{g \in \mathrm{R}_X} \sum_{h \in \mathrm{R}_Y} f(g,h) \times \delta(g,h) \Big)$$

subject to the constraints:

- CNNeg: $\forall g \in \mathrm{R}_X, \forall h \in \mathrm{R}_Y: f(g,h) \geq 0$
- CSource: $\forall g \in \mathrm{R}_X: \sum_{h \in \mathrm{R}_Y} f(g,h) \leq X(g)$
- CTarget: $\forall h \in \mathrm{R}_Y: \sum_{g \in \mathrm{R}_X} f(g,h) \leq Y(h)$
- CMaxFlow: $\sum_{g \in \mathrm{R}_X, h \in \mathrm{R}_Y} f(g,h) = m$

Independent Minimization Lower Bound of EMD

- **Idea:** Approximation of the EMD through constraint relaxation
- **Approach:** Given two feature signatures $X, Y \in \mathbb{S}_m$, the LB_{IM} Distance is defined w.r.t. a metric ground distance δ as:

$$LB_{IM}(X,Y) = \min_{\{f \mid f:\, \mathrm{R}_X \times \mathrm{R}_Y \to \mathbb{R}^{\geq 0}\}} \frac{1}{m}\Big(\sum_{g \in \mathrm{R}_X} \sum_{h \in \mathrm{R}_Y} f(g,h) \times \delta(g,h) \Big)$$

subject to the constraints:

- CNNeg: $\forall g \in \mathrm{R}_X, \forall h \in \mathrm{R}_Y : f(g,h) \geq 0$
- CSource: $\forall g \in \mathrm{R}_X : \sum_{h \in \mathrm{R}_Y} f(g,h) \leq X(g)$
- CTargetIM: $\forall g \in \mathrm{R}_X,\ \forall h \in \mathrm{R}_Y : f(g,h) \leq Y(h)$
- CMaxFlow: $\sum_{g \in \mathrm{R}_X, h \in \mathrm{R}_Y} f(g,h) = m$

Replace CTarget with the relaxed constraint CTargetIM

- Lower bound LB_{IM} results from replacing CTarget with CTargetIM

Independent Minimization Lower Bound of EMD

- **Idea:** Approximation of the EMD through constraint relaxation
- **Approach:** Given two feature signatures $X, Y \in \mathbb{S}_m$, the LB_{IM} Distance is defined w.r.t. a metric ground distance δ as:

$$LB_{IM}(X,Y) = \sum_{g \in \mathrm{R}_X} \min_{\{f \mid f:\, \mathrm{R}_X \times \mathrm{R}_Y \to \mathbb{R}^{\geq 0}\}} \frac{1}{m} \Big(\sum_{h \in \mathrm{R}_Y} f(g,h) \times \delta(g,h) \Big)$$

subject to the constraints:

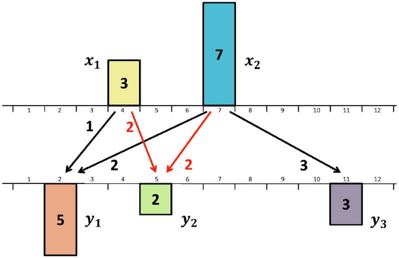
- CNNeg: $\forall g \in \mathrm{R}_X, \forall h \in \mathrm{R}_Y: f(g,h) \geq 0$
- CSource: $\forall g \in \mathrm{R}_X: \sum_{h \in \mathrm{R}_Y} f(g,h) \leq X(g)$
- CTargetIM: $\forall g \in \mathrm{R}_X, \forall h \in \mathrm{R}_Y: f(g,h) \leq Y(h)$
- CMaxFlow: $\sum_{g \in \mathrm{R}_X, h \in \mathrm{R}_Y} f(g,h) = m$

Replace CTarget with the relaxed constraint CTargetIM

- Lower bound LB_{IM} results from replacing CTarget with CTargetIM
- The minimization within LB_{IM} can be computed individually for each representative $g \in \mathrm{R}_X$

Flow computation

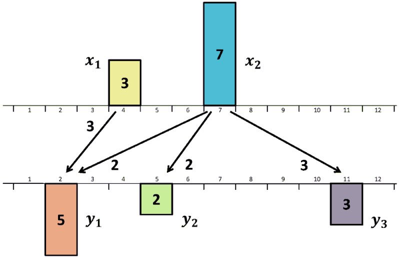
- **Idea:**
  - For each representative $g \in \mathrm{R}_X$, define $S_{X,Y}$ of nearest neighbor representatives $h \in \mathrm{R}_Y$ according to $\delta(g,h)$ in ascending order
  - Capacity of g may not exceed total weight of elements in $S_{X,Y}$
- **Example:**
  - Representative sets: $\mathrm{R}_X = \{x_1, x_2\}$ $\mathrm{R}_Y = \{y_1, y_2, y_3\}$
  - Weights: $X(x_1) = 3,\ X(x_2) = 7,$ $Y(y_1) = 5,\ Y(y_2) = 2,\ Y(y_3) = 3$
  - $S_{X,Y}(x_1) = (y_2, y_1)$ and $S_{X,Y}(x_2) = (y_2, y_3, y_1)$



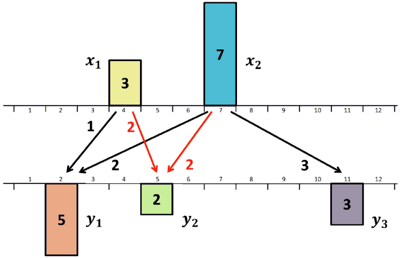
$$LB_{IM}(X,Y) = \frac{1}{10}(1 \times 2 + 2 \times 1 + 2 \times 5 + 2 \times 2 + 3 \times 4) = 3.0$$

Flow computation

- **Idea:**
  - For each representative $g \in \mathrm{R}_X$, define $S_{X,Y}$ of nearest neighbor representatives $h \in \mathrm{R}_Y$ according to $\delta(g, h)$ in ascending order
  - Capacity of g may not exceed total weight of elements in $S_{X,Y}$
- **Example:**



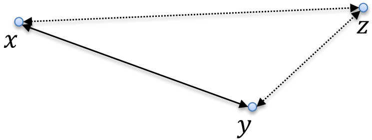
$EMD(X, Y) = 3.2$



$LB_{IM}(X, Y) = 3.0$

Metric Space Properties

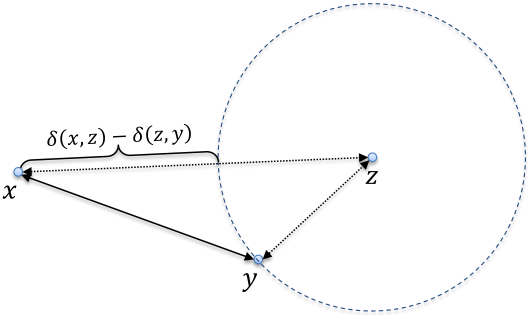
- Given a metric space $(\mathbb{X}, \delta)$ how to estimate the distance $\delta\colon \mathbb{X} \times \mathbb{X} \to \mathbb{R}^{\geq 0}$ between two distinct objects $x, y \in \mathbb{X}$?
  - identity of indiscernibles: $\delta(x, y) \neq 0$
  - non-negativity: $\delta(x, y) \geq 0$
  - symmetry: $\delta(x, y) = \delta(y, x)$
  - triangle inequality: $\delta(x, y) \leq \delta(x, z) + \delta(z, y)$ for any $z \in \mathbb{X}$



- Triangle inequality puts into relation three objects
- Triangle inequality is the only means that allows to estimate the distance between two objects by using another additional object

Geometric Derivation of Triangle Lower Bound

- **Goal:** Lower bounding $\delta(x, y)$ w.r.t. an object z by using the triangle inequality
- Suppose $\delta(x, z) \geq \delta(z, y)$:



- We then have: $\delta(x, z) - \delta(z, y) \leq \delta(x, y)$

Algebraic Derivation of Triangle Lower Bound

- Given a metric space $(\mathbb{X}, \delta)$, it holds for all objects $x, y, z \in \mathbb{X}$ that:

$$\begin{aligned}
\delta(x,z) \leq \delta(x,y) + \delta(y,z) &\Rightarrow \delta(x,z) - \delta(y,z) \leq \delta(x,y) \\
\delta(y,z) \leq \delta(y,x) + \delta(x,z) &\Rightarrow \delta(y,z) - \delta(x,z) \leq \delta(y,x) \\
&\Rightarrow -(\delta(x,z) - \delta(y,z)) \leq \delta(y,x) \\
&\Rightarrow \delta(x,z) - \delta(y,z) \geq -\delta(y,x)
\end{aligned}$$

- Combining both inequalities yields:

$$-\delta(x,y) \leq \delta(x,z) - \delta(y,z) \leq \delta(x,y)$$

- This leads to the **reverse** or **inverse triangle inequality**:

$$\delta_z^{\triangle}(x,y) = |\delta(x,z) - \delta(y,z)| \leq \delta(x,y)$$

- $\delta_z^{\triangle}(x,y)$ is a lower bound of $\delta(x,y)$ w.r.t. any object $z \in \mathbb{X}$

Algebraic Derivation of Triangle Lower Bound

- Multiple lower bounds $\delta^{\triangle}_{z_1}, \dots, \delta^{\triangle}_{z_k}$ w.r.t. objects $\{z_1, \dots, z_k\} \subseteq \mathbb{X}$ are combined to a single lower bound by using their maximum
- Let $(\mathbb{X}, \delta)$ be a metric space and $\mathbb{P} \subseteq \mathbb{X}$ be a finite set of pivot elements, the triangle lower bound $\delta^{\triangle}_{\mathbb{P}} : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ w.r.t. $\mathbb{P}$ is defined for all $x, y \in \mathbb{X}$ as follows:

$$\delta^{\triangle}_{\mathbb{P}}(x, y) = \max_{p \in \mathbb{P}} |\delta(x, p) - \delta(p, y)|$$

- Triangle lower bound $\delta^{\triangle}_{\mathbb{P}}$ can be utilized directly in the multi-step query processing algorithm
- **Problem:** Direct utilization not meaningful since a single lower bound computation requires $2 \times |\mathbb{P}|$ distance evaluations
- **Solution:** Precomputation of distances (i.e. indexing)

Pivot Table

- The idea of a pivot table consists of storing the distances between each database object and each pivot element
- **Approach:**
  - Given a database $\mathbb{D} = \{o_1, \ldots, o_n\}$ and a set of pivot elements $\mathbb{P} = \{p_1, \ldots, p_k\}$
  - Pivot table $\mathcal{T} \in \mathbb{R}^{n \times k}$ stores distances between all pairs of database objects $o_i \in \mathbb{D}$ and pivot elements $p_i \in \mathbb{P}$:

$\mathcal{T}$	$\delta(\cdot, p_1)$	$\ldots$	$\delta(\cdot, p_k)$
o_1			
$\vdots$			
o_n			

  - $|\mathbb{D}| \times |\mathbb{P}| = n \times k$ distance computations necessary prior to query processing

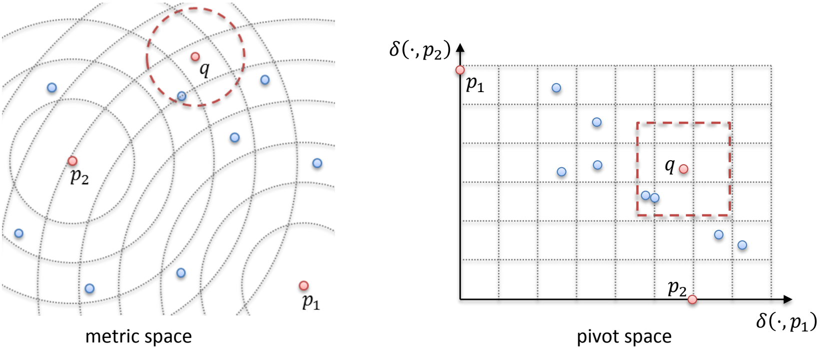
Pivot Table: Query Processing & Properties

- A query $q \in \mathbb{X}$ is processed as follows:
  1. Distances $\delta(q, p_i)$ are computed for all $p_i \in \mathbb{P}$
  2. Linear scan of the pivot table $\mathcal{T}$ with $\delta_{\mathbb{P}}^{\triangle}$ to generate candidates
  3. Refinement of candidates with original distance δ

- **Properties:**
  - Pivot table is regarded as one of the most simplistic yet effective metric access method
  - It applies caching of distances
  - Due to the linear behavior, a pivot table scales for small-to-moderate size databases

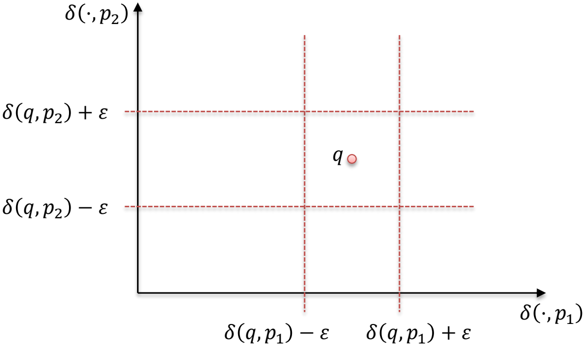
Pivot Space

- A pivot table can be understood as an embedding of objects from a metric space into a multidimensional Euclidean space
- This Euclidean space $\mathbb{R}^k$ whose dimensions are given by the distances to the pivot elements $\mathbb{P} = \{p_1, \ldots, p_k\}$ is denoted as pivot space



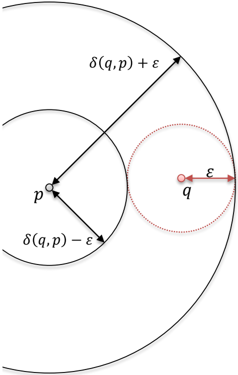
Pivot Space and Triangle Inequality

- Consider a query *range*$_\epsilon(q, \delta, \mathbb{D})$ with range $\epsilon \in \mathbb{R}^{\geq 0}$
- The triangle inequality implies the following bounds in the pivot space:



Pivoting

- Searching by means of precomputed distances to pivot elements $\mathbb{P}$ and the triangle lower bound $\delta_{\mathbb{P}}^{\triangle}$
- Filtering Principle for $\mathbb{P} = \{p\}$ and range query with $\epsilon \in \mathbb{R}^{\geq 0}$:
  - Objects o inside the inner ball around p are filtered out because it holds that $\delta(q,p) - \delta(p,o) > \epsilon$
  - Objects o outside the outer ball around p are filtered out because it holds that $\delta(p,o) - \delta(q,p) > \epsilon$
  - Thus only objects o inside the shell between the two balls are candidates because it holds that $\delta_{\mathbb{P}}^{\triangle} = |\delta(q,p) - \delta(p,o)| \leq \epsilon$



Pivoting - Example

- **Given:** Pivot table $\mathcal{T}$ with pivot objects $\mathbb{P}$, database $\mathbb{D} = \{o_1, o_2, o_3\}$, query object $q = (1, 2, 2)$, range $\epsilon = 1$, $\delta = L_1$ (Manhattan Distance)

$\mathcal{T}$	$p_1 = (0, 0, 2)$ $\delta(\cdot, p_1)$	$p_2 = (1, 3, 0)$ $\delta(\cdot, p_2)$	$p_3 = (1, 1, 1)$ $\delta(\cdot, p_3)$	
o_1	3	1	4	
o_2	5	2	1	
o_3	4	4	2	
$\delta(q, \cdot)$				

Pivoting - Example

- **Given:** Pivot table $\mathcal{T}$ with pivot objects $\mathbb{P}$, database $\mathbb{D} = \{o_1, o_2, o_3\}$, query object $q = (1, 2, 2)$, range $\epsilon = 1$, $\delta = L_1$ (Manhattan Distance)

$\mathcal{T}$	$p_1 = (0,0,2)$ $\delta(\cdot, p_1)$	$p_2 = (1,3,0)$ $\delta(\cdot, p_2)$	$p_3 = (1,1,1)$ $\delta(\cdot, p_3)$	
o_1	3	1	4	
o_2	5	2	1	
o_3	4	4	2	
$\delta(q, \cdot)$	3	3	2	

- Compute distances between q and pivot objects

Pivoting - Example

- **Given:** Pivot table $\mathcal{T}$ with pivot objects $\mathbb{P}$, database $\mathbb{D} = \{o_1, o_2, o_3\}$, query object $q = (1, 2, 2)$, range $\epsilon = 1$, $\delta = L_1$ (Manhattan Distance)

$\mathcal{T}$	$p_1 = (0,0,2)$ $\delta^{\triangle}_{p_1}(q,\cdot)$	$p_2 = (1,3,0)$ $\delta^{\triangle}_{p_2}(q,\cdot)$	$p_3 = (1,1,1)$ $\delta^{\triangle}_{p_3}(q,\cdot)$	
o_1	0	2	2	
o_2	2	1	1	
o_3	1	1	0	
$\delta(q,\cdot)$	3	3	2	

- Compute distances between q and pivot objects
- Compute $\delta^{\triangle}_{p_i}(q, o_i)$ for every object $o_i \in \mathbb{D}$ and pivot object $p_i \in \mathbb{P}$

Pivoting - Example

- **Given:** Pivot table $\mathcal{T}$ with pivot objects $\mathbb{P}$, database $\mathbb{D} = \{o_1, o_2, o_3\}$, query object $q = (1, 2, 2)$, range $\epsilon = 1$, $\delta = L_1$ (Manhattan Distance)

$\mathcal{T}$	$p_1 = (0,0,2)$ $\delta^{\triangle}_{p_1}(q,\cdot)$	$p_2 = (1,3,0)$ $\delta^{\triangle}_{p_2}(q,\cdot)$	$p_3 = (1,1,1)$ $\delta^{\triangle}_{p_3}(q,\cdot)$	$\delta^{\triangle}_{\mathbb{P}}(q,\cdot)$
o_1	0	2	2	2
o_2	2	1	1	2
o_3	1	1	0	1
$\delta(q,\cdot)$	3	3	2	

- Compute distances between q and pivot objects
- Compute $\delta^{\triangle}_{p_i}(q, o_i)$ for every object $o_i \in \mathbb{D}$ and pivot object $p_i \in \mathbb{P}$
- Compute $\delta^{\triangle}_{\mathbb{P}}(q, o_i) = \max_{p_i \in \mathbb{P}} \left(\delta^{\triangle}_{p_i}(q, o_i)\right)$ for every object $o_i \in \mathbb{D}$

Pivoting - Example

- **Given:** Pivot table $\mathcal{T}$ with pivot objects $\mathbb{P}$, database $\mathbb{D} = \{o_1, o_2, o_3\}$, query object $q = (1, 2, 2)$, range $\epsilon = 1$, $\delta = L_1$ (Manhattan Distance)

$\mathcal{T}$	$p_1 = (0,0,2)$ $\delta^{\triangle}_{p_1}(q,\cdot)$	$p_2 = (1,3,0)$ $\delta^{\triangle}_{p_2}(q,\cdot)$	$p_3 = (1,1,1)$ $\delta^{\triangle}_{p_3}(q,\cdot)$	$\delta^{\triangle}_{\mathbb{P}}(q,\cdot)$
o_1	0	2	2	2
o_2	2	1	1	2
o_3	1	1	0	1
$\delta(q,\cdot)$	3	3	2	

- Compute distances between q and pivot objects
- Compute $\delta^{\triangle}_{p_i}(q, o_i)$ for every object $o_i \in \mathbb{D}$ and pivot object $p_i \in \mathbb{P}$
- Compute $\delta^{\triangle}_{\mathbb{P}}(q, o_i) = \max_{p_i \in \mathbb{P}} \left(\delta^{\triangle}_{p_i}(q, o_i)\right)$ for every object $o_i \in \mathbb{D}$
- Select every object o_i where $\delta^{\triangle}_{\mathbb{P}}(q, o_i) \leq \epsilon$ as candidate

Summary

- **Object representations**
  - How to model and represent multimedia data?
- **Fundamental similarity models for multimedia data**
  - What is a distance-based similarity model?
  - What metric distance functions can be used for histograms and signatures?
- **Efficient query processing**
  - What types of distance-based similarity queries exist?
  - How to process such queries efficiently?
- **Indexing**
  - How to index high-dimensional multimedia data?
  - What are the principles behind the metric indexing approach?

Excercise

Consider the query $range_\epsilon = (q, \mathbb{D}, \delta)$ with the query object $q = (1, 2)$, the database $\mathbb{D} = \{o_1, o_2, o_3\}$, the range $\epsilon = 2$ and the Euclidean distance function δ. Moreover, consider the following pivot table with the pivot objects $\mathbb{P} = \{p_1, p_2\}$:

	$p_1 = (1, 4)$	$p_2 = (3, 2 - \sqrt{5})$
o_1	1	3
o_2	4	0
o_3	2	1

Compute the distances between q and the two pivot objects.

- $\delta(q, p_1) =$
- $\delta(q, p_2) =$

Excercise

Consider the query $range_\epsilon = (q, \mathbb{D}, \delta)$ with the query object $q = (1, 2)$, the database $\mathbb{D} = \{o_1, o_2, o_3\}$, the range $\epsilon = 2$ and the Euclidean distance function δ. Moreover, consider the following pivot table with the pivot objects $\mathbb{P} = \{p_1, p_2\}$:

	$p_1 = (1, 4)$	$p_2 = (3, 2 - \sqrt{5})$
o_1	1	3
o_2	4	0
o_3	2	1

Compute the distances between q and the two pivot objects.

- $\delta(q, p_1) = \sqrt{(1-1)^2 + (4-2)^2} = \sqrt{4} = 2$
- $\delta(q, p_2) = \sqrt{(3-1)^2 + ((2-\sqrt{5}) - 2)^2} = \sqrt{4+5} = 3$

Excercise

Consider the query $range_\epsilon = (q, \mathbb{D}, \delta)$ with the query object $q = (1, 2)$, the database $\mathbb{D} = \{o_1, o_2, o_3\}$, the range $\epsilon = 2$ and the Euclidean distance function δ. Moreover, consider the following pivot table with the pivot objects $\mathbb{P} = \{p_1, p_2\}$:

	$p_1 = (1, 4)$	$p_2 = (3, 2 - \sqrt{5})$
o_1	1	3
o_2	4	0
o_3	2	1

Compute the distance $\delta^{\triangle}_{p_i}(q, o_i)$ for every pair of database object $o_i \in \mathbb{D}$ and pivot object $p_i \in \mathbb{P}$.

- $\delta^{\triangle}_{p_1}(q, o_1) =$
- $\delta^{\triangle}_{p_1}(q, o_2) =$
- ...

Excercise

Consider the query $range_\epsilon = (q, \mathbb{D}, \delta)$ with the query object $q = (1, 2)$, the database $\mathbb{D} = \{o_1, o_2, o_3\}$, the range $\epsilon = 2$ and the Euclidean distance function δ. Moreover, consider the following pivot table with the pivot objects $\mathbb{P} = \{p_1, p_2\}$:

	$p_1 = (1, 4)$	$p_2 = (3, 2 - \sqrt{5})$
o_1	1	3
o_2	4	0
o_3	2	1

Compute the distance $\delta^{\triangle}_{p_i}(q, o_i)$ for every pair of database object $o_i \in \mathbb{D}$ and pivot object $p_i \in \mathbb{P}$.

- $\delta^{\triangle}_{p_1}(q, o_1) = |\delta(o_1, p_1) - \delta(q, p_1)| = |1 - 2| = 1$
- $\delta^{\triangle}_{p_1}(q, o_2) = |\delta(o_2, p_1) - \delta(q, p_1)| = |4 - 2| = 2$
- $\delta^{\triangle}_{p_1}(q, o_3) = |\delta(o_3, p_1) - \delta(q, p_1)| = |2 - 2| = 0$

Excercise

Consider the query $range_\epsilon = (q, \mathbb{D}, \delta)$ with the query object $q = (1, 2)$, the database $\mathbb{D} = \{o_1, o_2, o_3\}$, the range $\epsilon = 2$ and the Euclidean distance function δ. Moreover, consider the following pivot table with the pivot objects $\mathbb{P} = \{p_1, p_2\}$:

	$p_1 = (1, 4)$	$p_2 = (3, 2 - \sqrt{5})$
o_1	1	3
o_2	4	0
o_3	2	1

Compute the distance $\delta^{\triangle}_{p_i}(q, o_i)$ for every pair of database object $o_i \in \mathbb{D}$ and pivot object $p_i \in \mathbb{P}$.

- $\delta^{\triangle}_{p_2}(q, o_1) = |\delta(o_1, p_2) - \delta(q, p_2)| = |3 - 3| = 0$
- $\delta^{\triangle}_{p_2}(q, o_2) = |\delta(o_2, p_2) - \delta(q, p_2)| = |0 - 3| = 3$
- $\delta^{\triangle}_{p_2}(q, o_3) = |\delta(o_3, p_2) - \delta(q, p_2)| = |1 - 3| = 2$

Excercise

Consider the query $range_\epsilon = (q, \mathbb{D}, \delta)$ with the query object $q = (1, 2)$, the database $\mathbb{D} = \{o_1, o_2, o_3\}$, the range $\epsilon = 2$ and the Euclidean distance function δ. Moreover, consider the following pivot table with the pivot objects $\mathbb{P} = \{p_1, p_2\}$:

	$p_1 = (1, 4)$	$p_2 = (3, 2 - \sqrt{5})$
o_1	1	3
o_2	4	0
o_3	2	1

Compute the distance $\delta^{\triangle}_{\mathbb{P}}(q, o_i)$ for every database object $o_i \in \mathbb{D}$.

- $\delta^{\triangle}_{\mathbb{P}}(q, o_1) =$
- $\delta^{\triangle}_{\mathbb{P}}(q, o_2) =$
- $\delta^{\triangle}_{\mathbb{P}}(q, o_3) =$

Excercise

Consider the query $range_\epsilon = (q, \mathbb{D}, \delta)$ with the query object $q = (1, 2)$, the database $\mathbb{D} = \{o_1, o_2, o_3\}$, the range $\epsilon = 2$ and the Euclidean distance function δ. Moreover, consider the following pivot table with the pivot objects $\mathbb{P} = \{p_1, p_2\}$:

	$p_1 = (1, 4)$	$p_2 = (3, 2 - \sqrt{5})$
o_1	1	3
o_2	4	0
o_3	2	1

Compute the distance $\delta^{\triangle}_{\mathbb{P}}(q, o_i)$ for every database object $o_i \in \mathbb{D}$.

- $\delta^{\triangle}_{\mathbb{P}}(q, o_1) = \max\left(\delta^{\triangle}_{p_1}(q, o_1), \delta^{\triangle}_{p_2}(q, o_1)\right) = 1$
- $\delta^{\triangle}_{\mathbb{P}}(q, o_2) = \max\left(\delta^{\triangle}_{p_1}(q, o_2), \delta^{\triangle}_{p_2}(q, o_2)\right) = 3$
- $\delta^{\triangle}_{\mathbb{P}}(q, o_3) = \max\left(\delta^{\triangle}_{p_1}(q, o_3), \delta^{\triangle}_{p_2}(q, o_3)\right) = 2$

Excercise

Consider the query $range_\epsilon = (q, \mathbb{D}, \delta)$ with the query object $q = (1, 2)$, the database $\mathbb{D} = \{o_1, o_2, o_3\}$, the range $\epsilon = 2$ and the Euclidean distance function δ. Moreover, consider the following pivot table with the pivot objects $\mathbb{P} = \{p_1, p_2\}$:

	$p_1 = (1, 4)$	$p_2 = (3, 2 - \sqrt{5})$
o_1	1	3
o_2	4	0
o_3	2	1

Determine which database objects $o_i \in \mathbb{D}$ are candidates for a correct query answer based on the previously computed distances (mark each correct answer with a cross). Briefly justify your answer.

- o_1 is ◯ a candidate / ◯ not a candidate because
- ...

Excercise

Consider the query $range_\epsilon = (q, \mathbb{D}, \delta)$ with the query object $q = (1, 2)$, the database $\mathbb{D} = \{o_1, o_2, o_3\}$, the range $\epsilon = 2$ and the Euclidean distance function δ. Moreover, consider the following pivot table with the pivot objects $\mathbb{P} = \{p_1, p_2\}$:

	$p_1 = (1, 4)$	$p_2 = (3, 2 - \sqrt{5})$
o_1	1	3
o_2	4	0
o_3	2	1

Determine which database objects $o_i \in \mathbb{D}$ are candidates for a correct query answer based on the previously computed distances (mark each correct answer with a cross). Briefly justify your answer.

- o_1 is a candidate (YES) because $\delta^{\triangle}_{\mathbb{P}}(q, o_1) = 1 \leq 2 = \epsilon$
- o_2 is a candidate (NO) because $\delta^{\triangle}_{\mathbb{P}}(q, o_2) = 3 \not\leq 2 = \epsilon$
- o_3 is a candidate (YES) because $\delta^{\triangle}_{\mathbb{P}}(q, o_3) = 2 \leq 2 = \epsilon$