http://prisma.dcc.uchile.cl

http://siret.ms.mff.cuni.cz

Non-Metric Similarity Search Problems in Very Large Collections

Benjamin Bustos, University of Chile
Tomáš Skopal, Charles University in Prague

ICDE 2011, Hannover, Germany

Outline of the tutorial

- Benjamin
  - Introduction
  - The non-metric case of similarity
  - Case study 1 – image retrieval
  - Case study 2 – time series retrieval
- Tomáš
  - Case study 3 – protein retrieval
  - Indexing non-metric spaces
  - Challenges

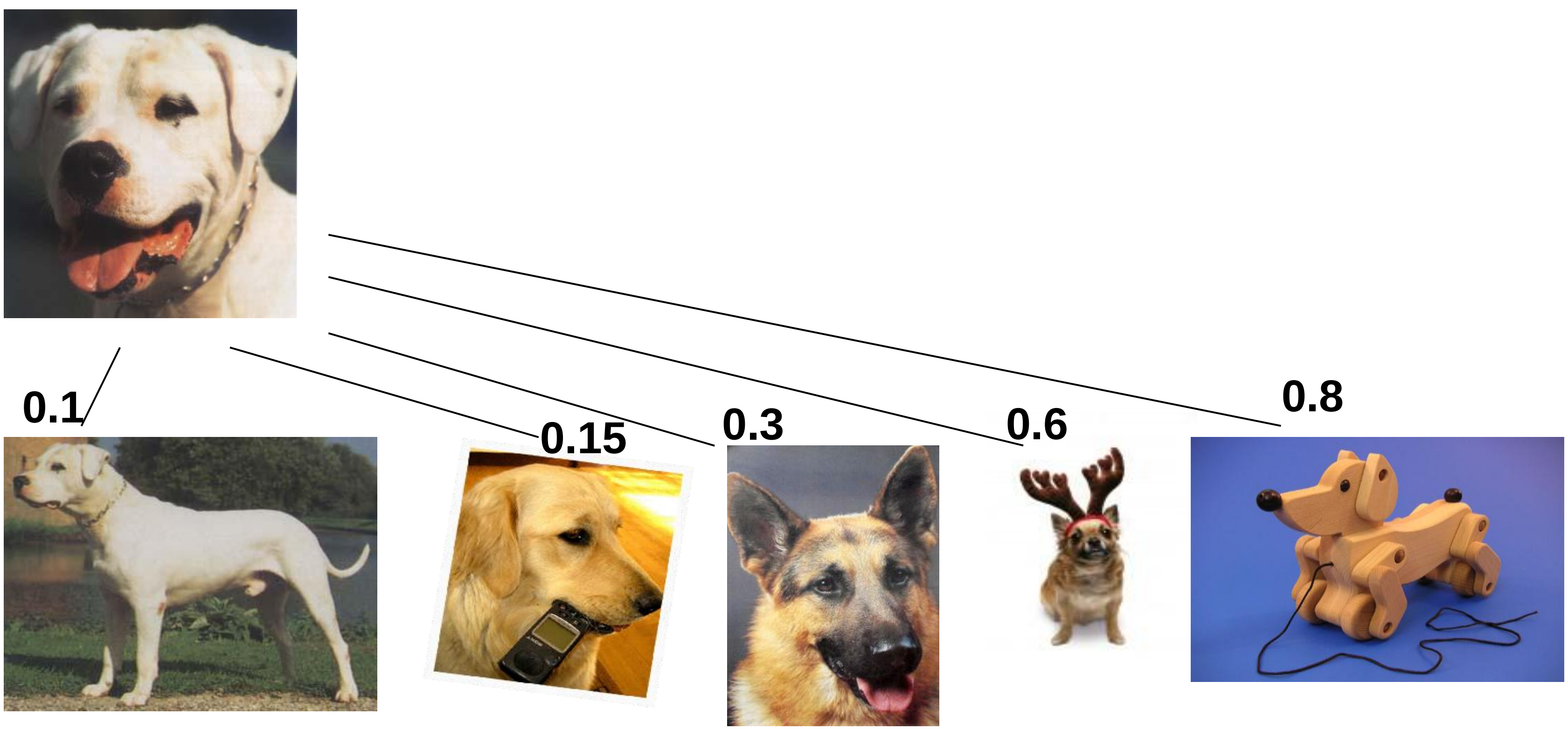
also see the survey [Skopal & Bustos, 2011]

Introduction

- Similarity search
  - Search for “similar objects” (subjective)
  - Content-based similarity search: query by example:

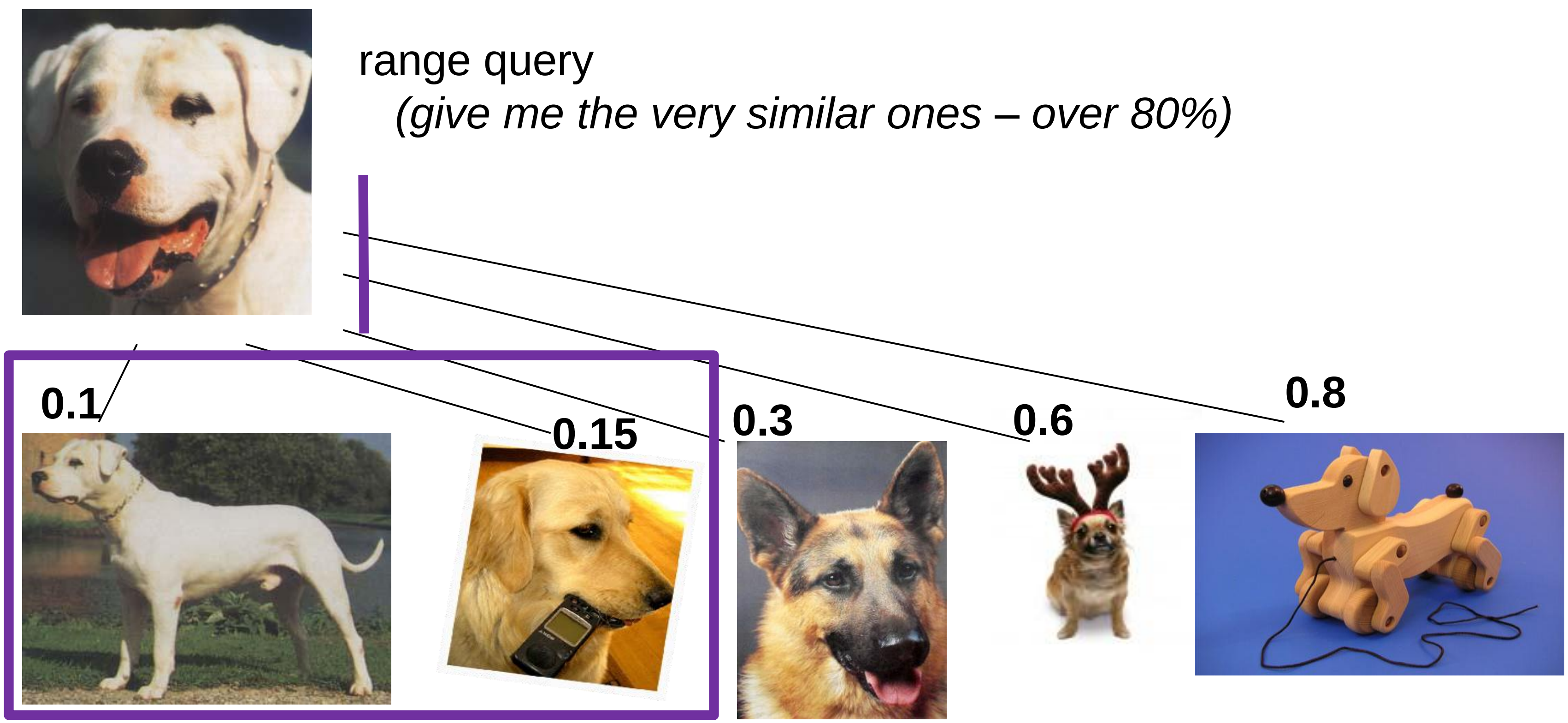
Introduction

- Similarity search
  - Search for “similar objects” (subjective)
  - Content-based similarity search: query by example:



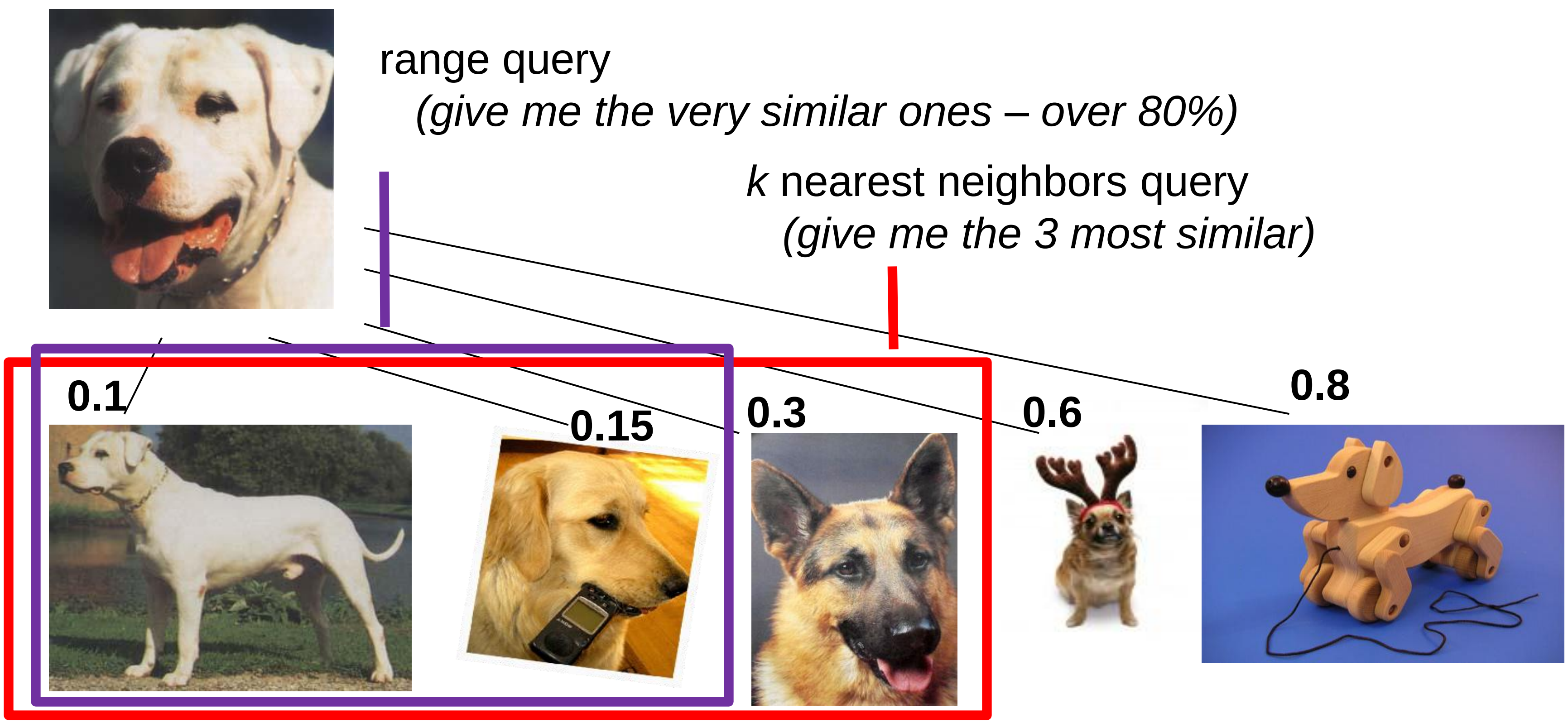
Introduction

- Similarity search
  - Search for “similar objects” (subjective)
  - Content-based similarity search: query by example:



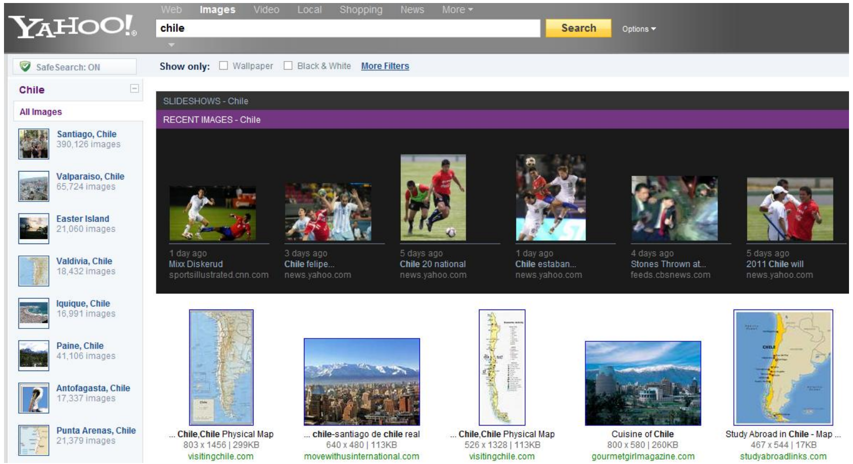
Introduction

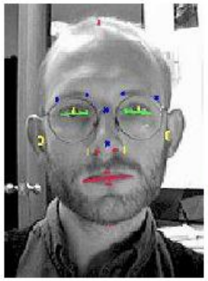
- Similarity search
  - Search for “similar objects” (subjective)
  - Content-based similarity search: query by example:

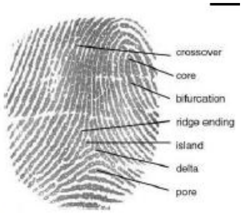


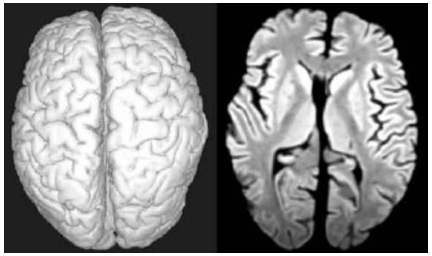
Introduction

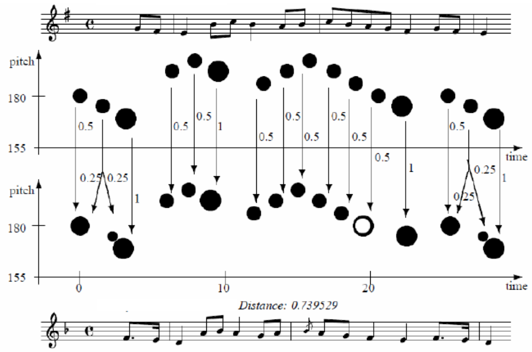
- Application examples of similarity search
  - Multimedia retrieval
  - Scientific databases
  - Biometry
  - Pattern recognition
  - Manufacturing industry
  - Cultural heritage
  - Etc.











Introduction

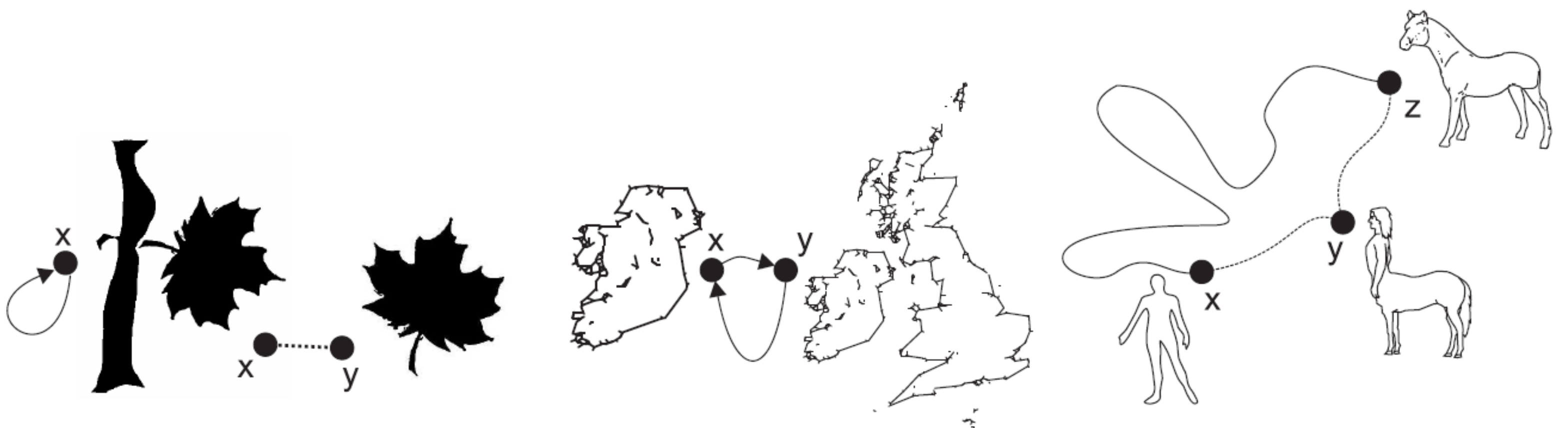
- Metric similarity
  - Dissimilarity function δ (the distance), universe **U**, database **S** ⊂ **U**, objects x,y,z ∈ **U**
  - The higher δ(x,y), the more dissimilar objects x,y are
- Topological properties

$$\begin{aligned} \delta(x,y) = 0 \Leftrightarrow x = y & \qquad \text{identity} \\ \delta(x,y) \geq 0 & \qquad \text{non-negativity} \\ \delta(x,y) = \delta(y,x) & \qquad \text{symmetry} \\ \delta(x,y) + \delta(y,z) \geq \delta(x,z) & \qquad \text{triangle inequality} \end{aligned}$$

- Pros of metric approach
  - Well-studied in mathematics (many known metrics)
  - Postulates support common assumptions on similarity
  - Allows efficient indexing and search (metric indexing)

Introduction

- Cons of metric approach:
  - It may not correctly model the "human" notion of similarity



(a) δ(x,x) > 0 (b) δ(x,y) = 0 (c) δ(x,y) < δ(y,x) (d) δ(x,y) + δ(y,x) < δ(x,z)

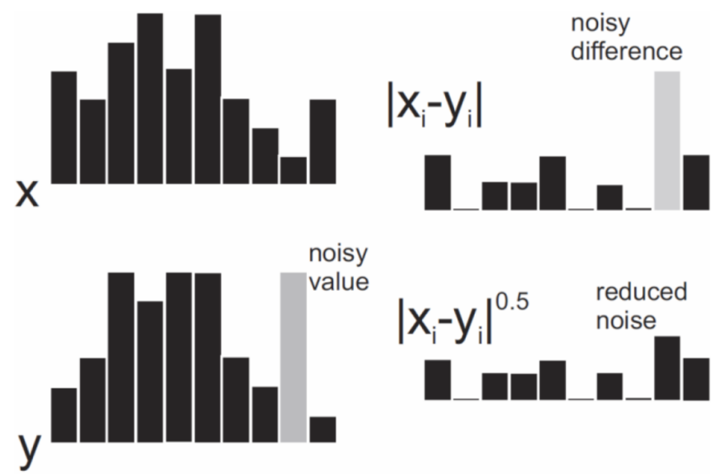
- Identity and non-negativity:
  - single object could be viewed as self-dissimilar
  - two distinct object could be viewed as identical
- Symmetry – comparison direction could be important
- Triangle inequality – similarity is not transitive

The non-metric case of similarity

- What is non-metric?
  - Generally: a distance function that does not satisfy some (or all) properties of a metric
- This could include:
  - Context-dependent similarity functions
  - Dynamic similarity functions
- For this tutorial: similarity functions that are “context-free and static“
  - Similarity between two objects is constant whatever the context is, i.e., regardless of time, user, query, other objects in database, etc.

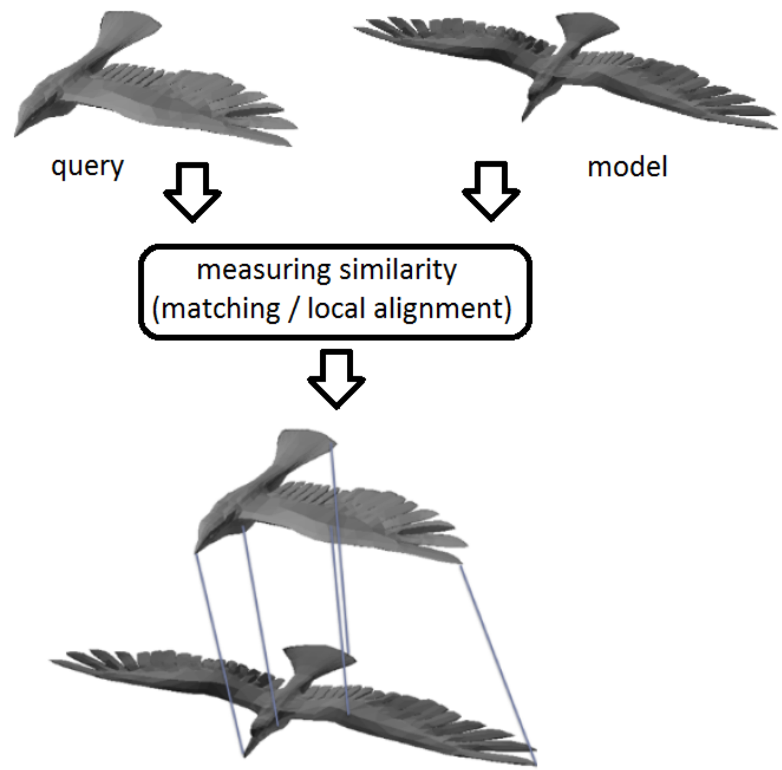
The non-metric case of similarity

- Motivation
  - Robustness
    - A robust function is resistant to outliers (noise or deformed objects), that would otherwise distort the similarity distribution within a given set of objects
    - Having objects x and y and a robust function δ, an extreme change in a small part of x's descriptor should not imply an extreme change of δ(x,y).



The non-metric case of similarity

- Motivation
  - Locality
    - A locally sensitive function is able to ignore some portions of the compared objects
    - The locality is usually used to privilege similarity before dissimilarity, hence, we rather search for similar parts in two objects than for dissimilar parts



The non-metric case of similarity

- Motivation
  - Comfort/freedom of modeling
    - The task of similarity search should serve just as a computer based tool in various professions
    - Domain experts should not be bothered by some “artificial” constraints (metric postulates)
      - Enforcement of metric may represent an unpleasant obstacle
    - Freedom of modeling
      - Complex heuristic algorithms
      - Black-box similarity

The non-metric case of similarity

- Examples of general non-metric functions
  - Fractional *Lp* distances (p<1)

$$L_p(x,y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p}$$

  - Sequence alignment distance

$$\delta_{SAD}(x,y,i,j) = \min \begin{cases} c(x_i, y_j) + \delta_{SAD}(x,y,i+1,j+1) \\ c(-, y_j) + \delta_{SAD}(x,y,i,j+1) \\ c(x_i, -) + \delta_{SAD}(x,y,i+1,j) \end{cases}$$

  - Cosine similarity

$$s_{\cos}(x,y) = \frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum_{i=1}^{d} {x_i}^2 \cdot \sum_{i=1}^{d} {y_i}^2}}$$

  - Earth Mover's distance

$$\begin{aligned} \delta_{EMD}(x,y) &= \min \left\{ \sum_{i=1}^{d} \sum_{j=1}^{d} c_{ij} f_{ij} \right\} \\ \text{subject to} & \\ f_{ij} &\geq 0 \\ \sum_{i=1}^{d} f_{ij} &= y_j \quad \forall j = 1, \ldots, d \\ \sum_{j=1}^{d} f_{ij} &= x_i \quad \forall i = 1, \ldots, d \end{aligned}$$

Case study 1 – image retrieval

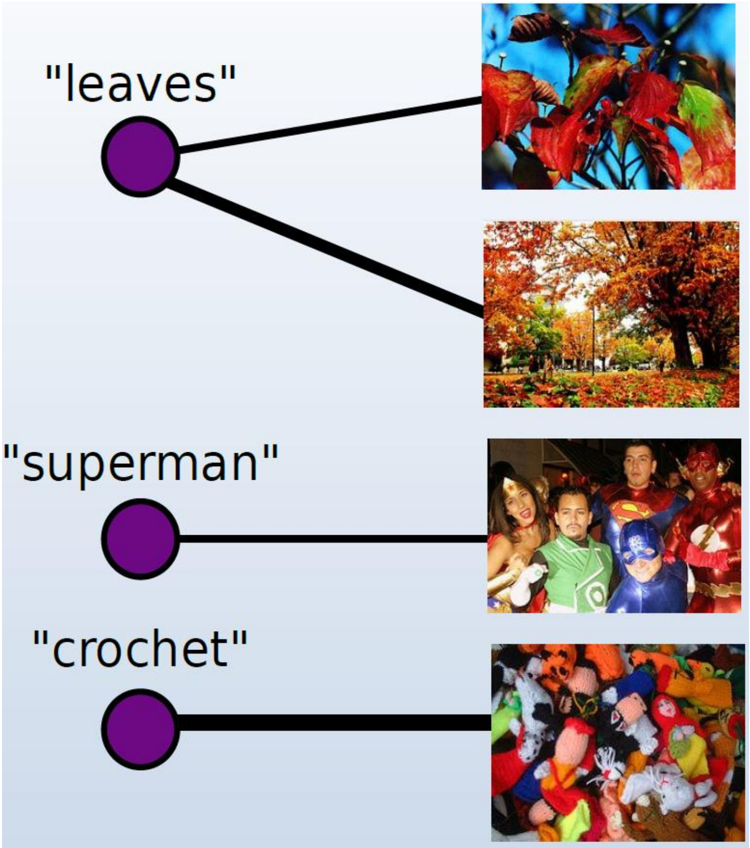
- The problem: find similar images to a given one

- Query specification: Text (metadata), Content-based, Sketch-based, combination

PRISMA Image Search:
http://prisma.dcc.uchile.cl/ImageSearch/

Case study 1 – image retrieval

- Image descriptors
  - High-level features: concepts
    - Metadata
      - Title, tags, etc.
    - Click information
      - Web-logs
      - Also carries semantic information



Case study 1 – image retrieval

- Image descriptors
  - Low-level features: visual attributes
    - Color, texture, shape, edges
    - Global vs. local descriptors

Case study 1 – image retrieval

- Big problem: semantic gap
  - Bridge between high and low features

(credit: Google)

Case study 1 – image retrieval

- Non-metric functions for image retrieval
  - χ^2, Kullback-Leibler (KLD), Jeffrey divergence (JD)

$$\delta_{\chi^2}(x,y) = \sum_{i=1}^{d} \frac{x_i - m(i)}{m(i)} \qquad m(i) = \tfrac{x_i + y_i}{2}$$

$$\delta_{KLD}(x,y) = \sum_{i=1}^{d} x_i \cdot \log\left(\frac{x_i}{y_i}\right)$$

$$\delta_{JD}(x,y) = \sum_{i=1}^{d} x_i \cdot \log\left(\frac{x_i}{\frac{x_i+y_i}{2}}\right) + y_i \cdot \log\left(\frac{y_i}{\frac{x_i+y_i}{2}}\right)$$

  - Better suited for image retrieval and classification than metric distances

Case study 1 – image retrieval

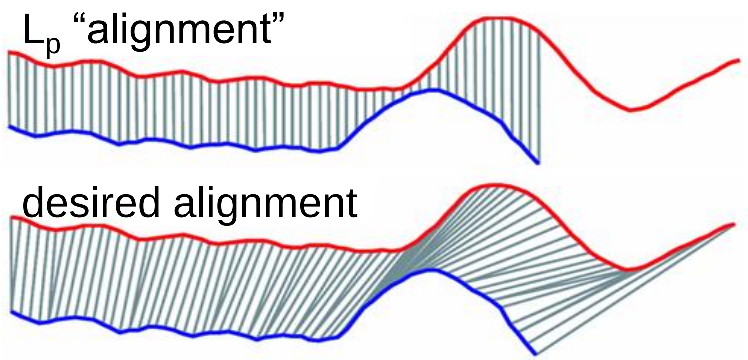
- Non-metric functions for image retrieval
  - Dynamic Partial Function [Goh et al., 2002]

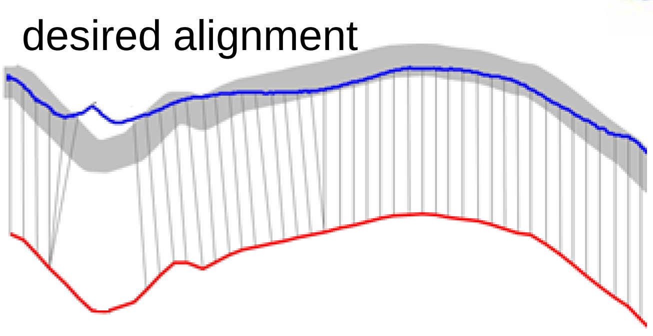
$$\delta_{DPF}(x,y) = \left(\sum_{c_i \in \Delta_m} |x_i - y_i|^p \right)^{1/p}, \ p \geq 1$$

    - Δ_m: set of *m* smallest coordinate differences
    - Better for image classification than Euclidean distance
  - Fractional Lp distances
    - Robust for image matching and retrieval
  - Jeffrey divergence
    - Better than Euclidean distance for retrieval of tomographies

Case study 2 – time series retrieval

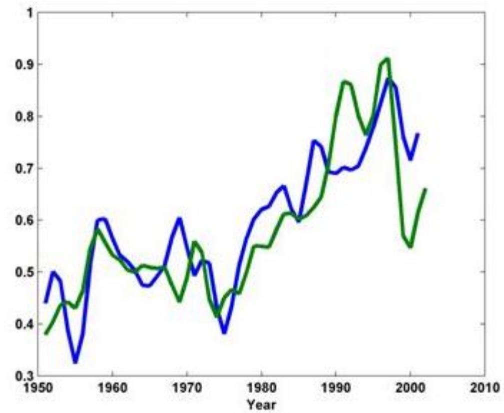
- The problem
  - Time series = ordered set of values
  - Given a set of time series, find similar ones
    - Find the optimal alignment
- L_p distance could be used, but:
  - Scaling/different dimensionality
  - Shift in time
  - Missing values
  - Outliers
  - Locality

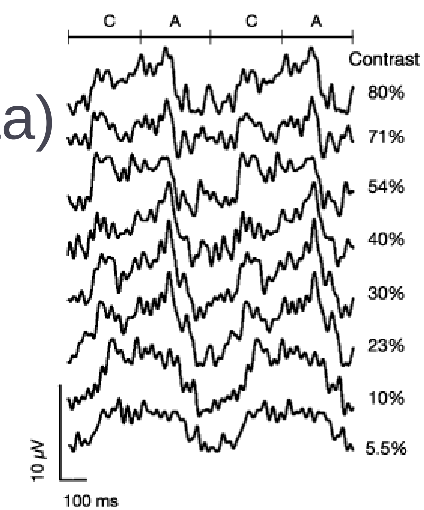


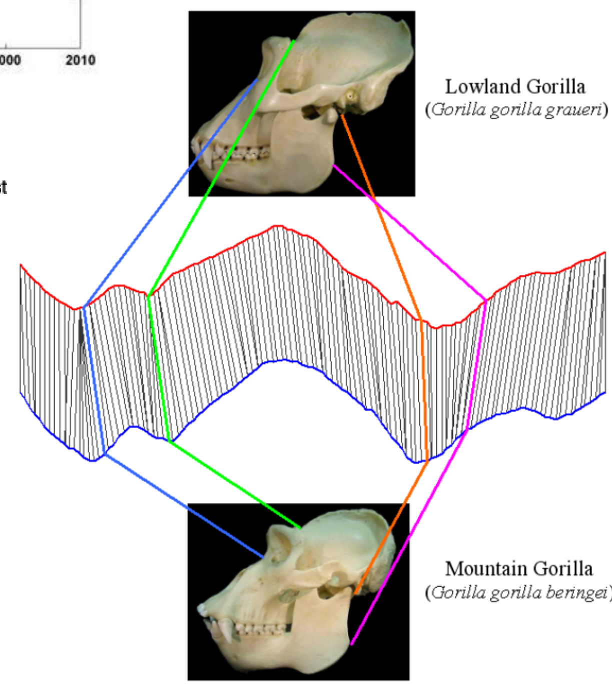


Case study 2 – time series retrieval

- Applications
  - Financial analysis (e.g., stock prices)
  - Medicine (e.g.,ECG, EEG)
  - Scientific data (e.g., seismological analysis, climate data)
  - Shape retrieval
  - Many others…





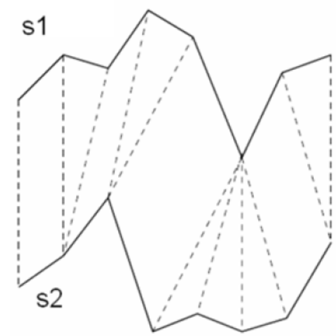


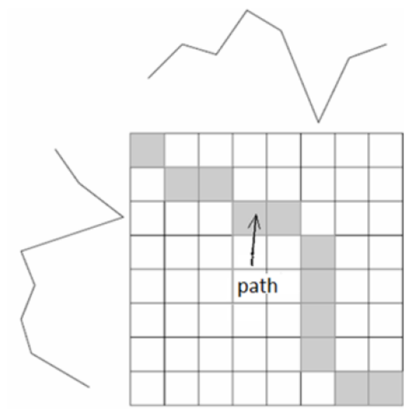
(image © Eamonn Keogh, eamonn@cs.ucr.edu)

Case study 2 – time series retrieval

- Dynamic Time Warping (DTW) [Berndt and Clifford, 1994]
  - Sequences s_1, s_2
  - m x n matrix M, where $m = |s_1|$, $n = |s_2|$
  - Matrix cell $M_{i,j}$ is partial distance $d(s_{1i}, s_{2j})$
  - Warping path $W = \{w_1, \ldots, w_t\}$, $\max\{m, n\} \le t \le m + n - 1$, is a set of cells from M that are contiguous
    - $w_1 = M_{1,1}$, $w_t = M_{m,n}$ *(boundary condition)*
    - if $w_k = M_{a,b}$ and $w_{k-1} = M_{a',b'}$, then
      - $a - a' \le 1$ $b - b' \le 1$ *(continuity)*
      - $a - a' \ge 0$ $b - b' \ge 0$ *(monotonicity)*
  - DTW = L_2 distance on optimally aligned sequences (optimal warping path)
  - non-metric distance

$$\delta_{DTW}(x, y) = \min_{W} \left\{ \sqrt{\sum_{k=1}^{t} w_k} \right\}$$

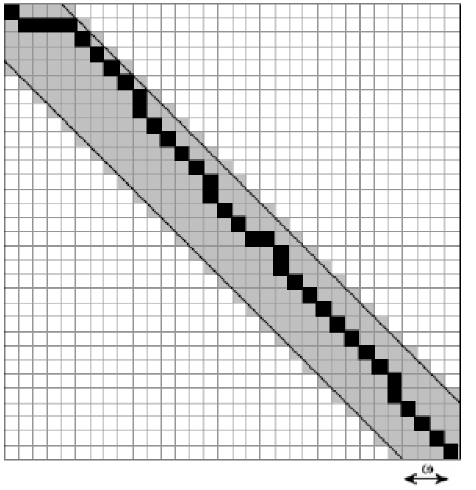




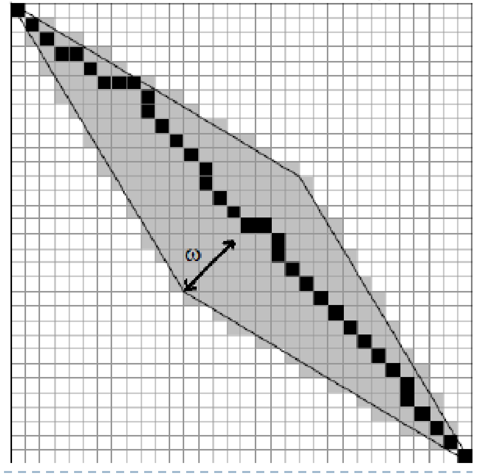
Case study 2 – time series retrieval

- Dynamic Time Warping (DTW)
  - Exponentially many warping paths, but can be computed in O(mn)*O(ground distance) time by dynamic programming
  - Constrained versions of DTW
    - Avoiding pathological paths
      - A range parameter ω
      - By $\omega = 0$, m=n, d(x,y) = |x-y| we get the Euclidean distance (just the diagonal warping path allowed)
    - DTW reduced complexity to $O((m+n)\omega)$
    - Sakoe-Chiba band – warping paths are only allowed near the diagonal
    - Itakura Parallelogram – "time warping" in the middle of sequences is allowed, but not at the ends

Sakoe-Chiba band

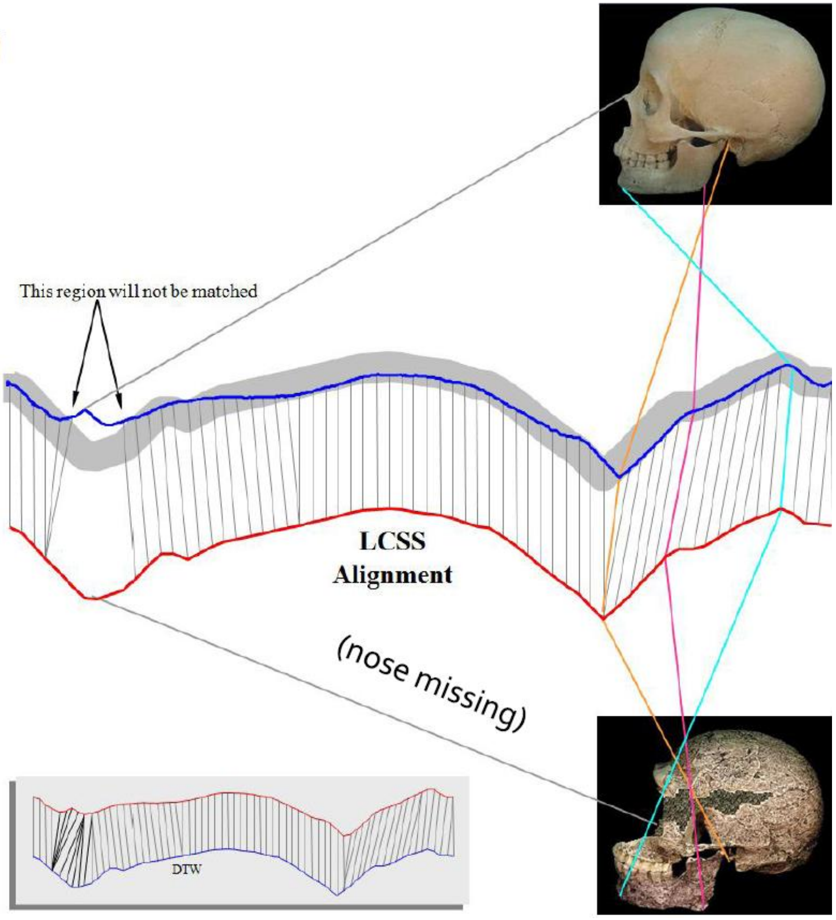


Itakura Parallelogram



Case study 2 – time series retrieval

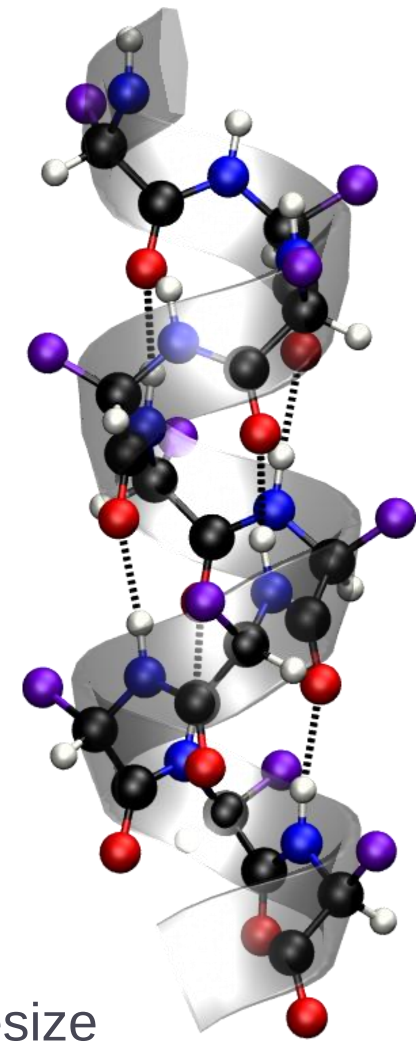
- Longest Common Subsequence (LCS)
  - x is subsequence of y if there is a strictly increasing sequence of indices such that there is a match between symbols in x and y (not necessarily adjacent)
  - z is a common subsequence of x and y if it is a subsequence of both x and y
  - The longest common subsequence (LCS) is the maximum length common subsequence of x and y
  - non-metric (also similarity)



(image © Eamonn Keogh, eamonn@cs.ucr.edu)

Case study 3 – protein retrieval

- Similar proteins → similar biological function
  - Many applications, like protein function/structure prediction (leading to, e.g., drug discovery)
- Protein sequences (primary structure)
  - Strings over 20-letter alphabet, i.e., symbolic chains of amino acids (AA)
  - Biologically augmented **string similarity**
  - Well-established model
- Protein structures (tertiary structure)
  - 3D geometry (polyline + local chemical properties)
  - Biologically augmented **shape similarity**
  - Closer to function than sequence, harder to synthesize



Case study 3 – protein retrieval

- Protein sequences
- String similarity (like edit distance) enhanced by scoring matrices (e.g., PAM, BLOSUM)
  - Score between two letters models the probability of mutating one amino acid into the other
- Needleman-Wunch (NW)
  - Global alignment – a nonmetric measure if scoring matrix is nonmetric and/or sequences are of different lengths
  - Usually used for solving subtasks (e.g., when sequences are split into q-grams which are then indexed/searched)
- Smith-Waterman (SW)
  - Local alignment (nonmetric), more applicable than global alignment
  - BLAST – approximate SW + an access method in one algorithm
  - Used for, e.g., function discovery, phylogenetic analysis, etc.

Case study 3 – protein retrieval

- Example
  - Global alignment (Needlemann-Wunch)

N	P	H	G	I	I	M	G	L	A	E	
-7	-7	+8	+6	-7	-7	+2	+6	+4	-7	-7	→ -16
-	-	H	G	-	-	L	G	L	-	-	

  - Local alignment (Smith-Waterman)

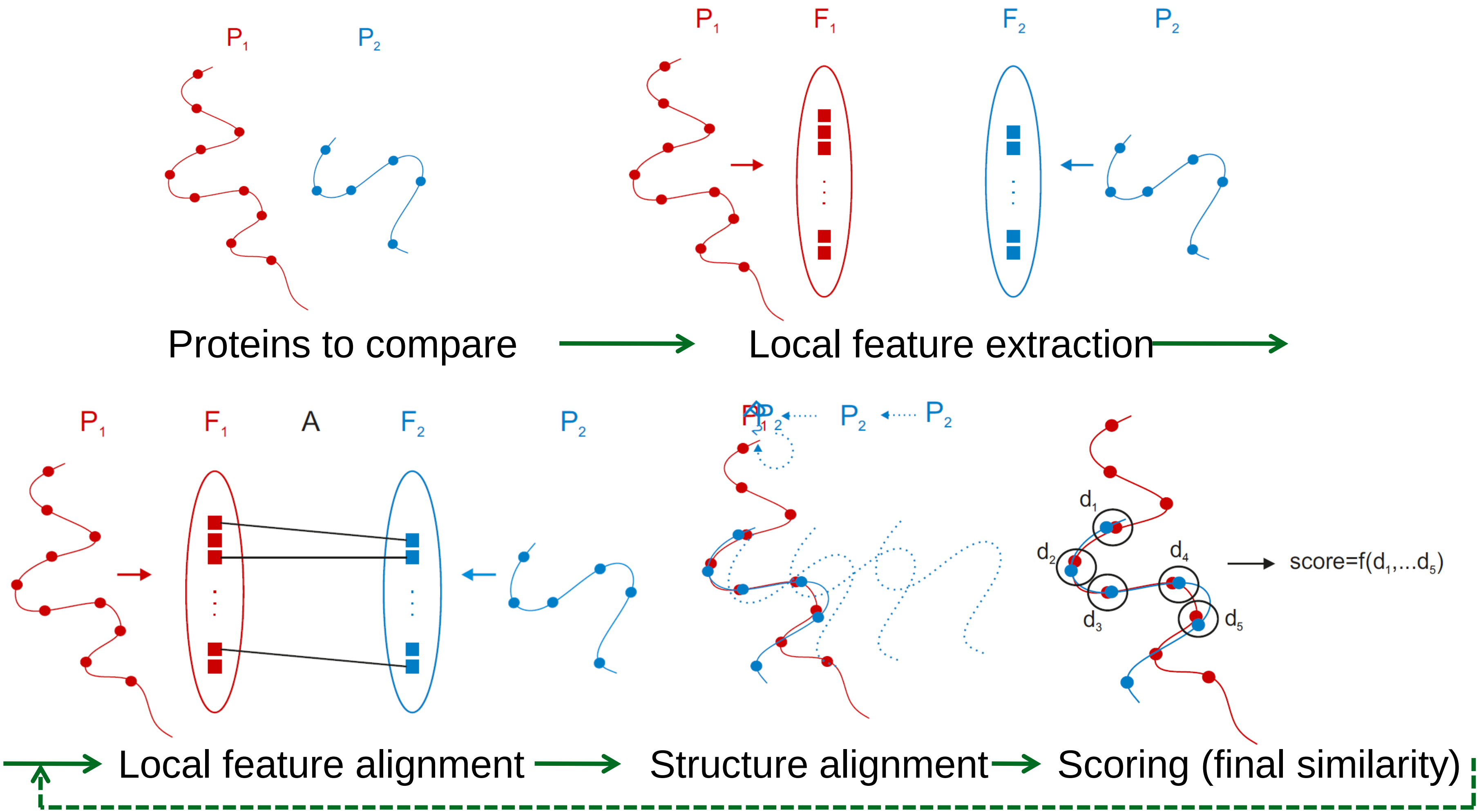
N	P	H	G	I	I	M	G	L	A	E	
		+8	+6	+2							→ 16
		H	G	L							

Case study 3 – protein retrieval

- Protein structure
  - Structure is more correlated to biological function than sequence (but harder to obtain)
- Similarity – two-step optimization process
  1) Alignment of structures based on local properties/features
     - Shape properties (torsion angles between AAs, density of AAs, curvature, surface area)
     - Physico-chemical properties (hydrophobicity, AA volume)
  2) Aggregation measure on top of the alignment
     - RMSD, TM-score
- Existing top algorithms for function assessment
  - DDPIn+iTM, PPM, Vorometric, TM-align, CE

[Hoksza & Galgonek, 2010]

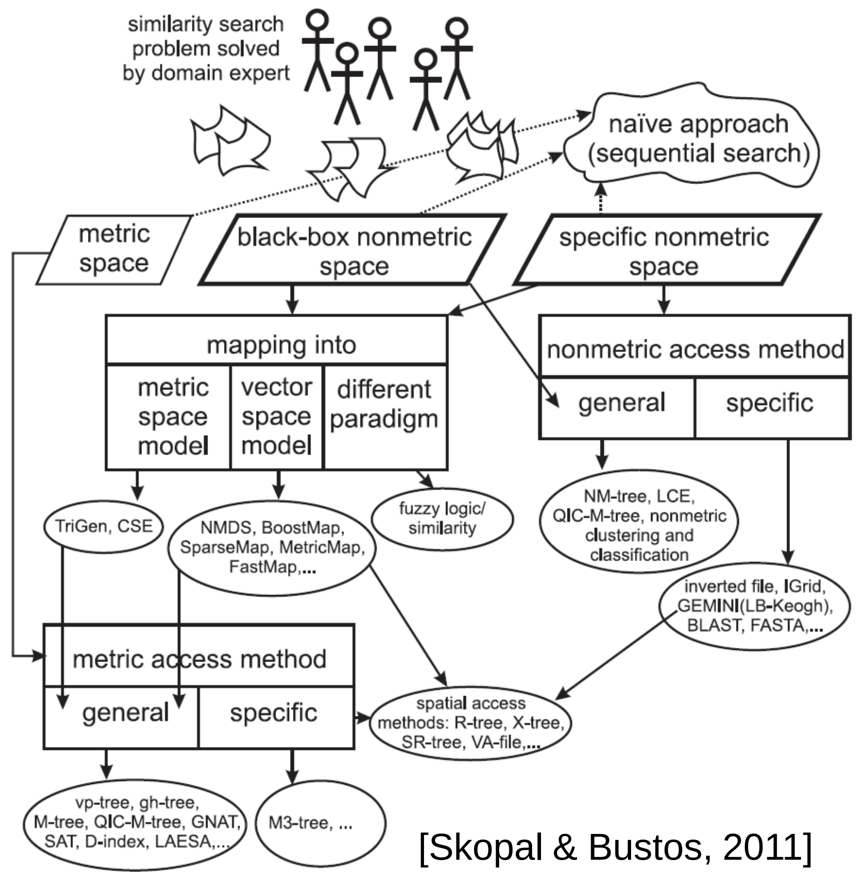
Case study 3 – protein retrieval



Indexing non-metric spaces – framework

- Need to **search efficiently** (fast query processing)
  - Access methods / indexes for similarity search

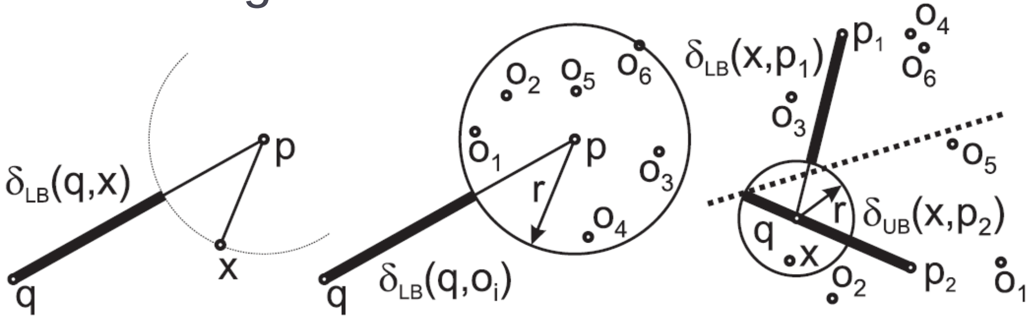
- Framework
  - Metric case similarity
    - MAM (metric access methods)
    - Useful for mapping approaches
  - General non-metric similarity
    - General NAM (nonmetric AM)
    - Black-box distance only
  - Specific non-metric similarity
    - Specific NAM
    - Additional knowledge needed



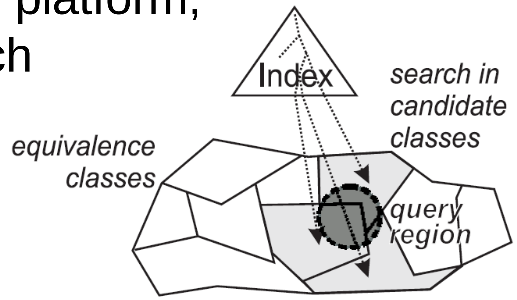
[Skopal & Bustos, 2011]

Indexing non-metric spaces – MAM

- The metric case (for completeness & mapping approaches)
  - Black-box metric distance δ needed
- Metric access methods (MAM), or metric indexes
  - Idea: pivot-based lower-bounding



  - Different implementations/designs [Zezula et al, 2005]
    - Dynamic/static database, serial/parallel/distributed platform, main/secondary memory, exact/approximate search
    - Index = set/hierarchy of metric regions, filtering
  - Examples: M-tree family, pivot tables, vp-tree, GNAT, SAT, M-index, D-file, etc.

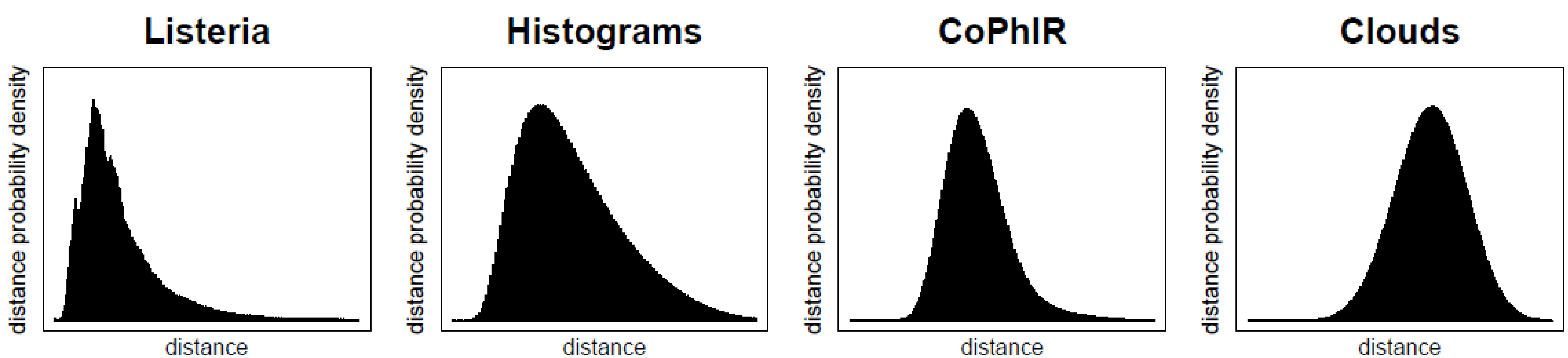


Indexing non-metric spaces – MAM & intrinsic dimensionality

- The metric postulates alone are **not a guarantee** of efficient indexing
- The structure of distance distribution indicates the **indexability** of the database
  - Intrinsic dimensionality **ρ(S,δ)** (idim) – an indexability indicator [Chávez et al., 2001]

$$\rho(\mathbb{S},\delta) = \frac{\mu^2}{2\sigma^2}$$

(μ and σ^2 are the mean and the variance of the distance distribution in **S** under δ)



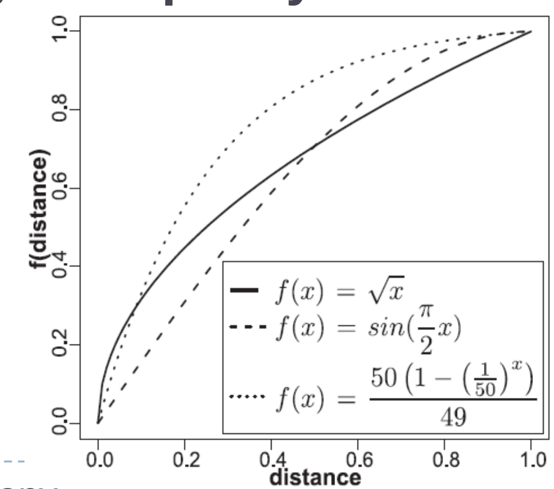
low idim (1.19) medium idim (7.5, 7.56) high idim (11.6)

Indexing non-metric spaces – mapping

- How to **index non-metric spaces**?
- Let's simplify the problem, turn them into metric ones!
- Mapping into an L_p space
  - **Pros:**
    "Easy" target space (cheap L_p distance, mostly Euclidean)
  - **Cons:**
    Approximate, static, computationally expensive mapping
- Variants of mappings into vector spaces
  - Assuming metric distance
    - FastMap, MetricMap, SparseMap, BoostMap
  - Allowing also nonmetric distance
    - Non-metric multidimensional scaling (NMDS) concept
    - Query-sensitive embedding (non-metric extension of BoostMap)

Indexing non-metric spaces – mapping

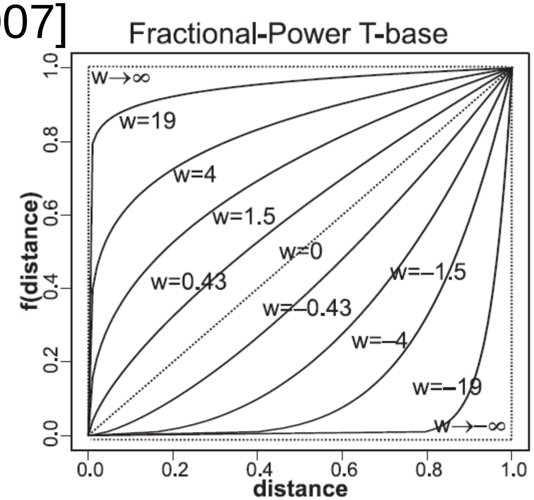
- Alternative mapping concept:
  - Do not transform whole space (the database **S** + δ), but only the distance function δ, leaving **S** unchanged
  - Suppose semimetric distance δ (metric not satisfying triangle ineq.)
- How to turn semimetric δ into a metric?
  - Consider increasing function f, such that f(0)=0, and modification f(δ)
    - i.e., f preserves the similarity ordering wrt any query
  - **Concave f increases the amount of triangle inequality in δ**
  - However, concave f also increases the intrinsic dimensionality of (**S**, f(δ)), when compared to (**S**, δ)

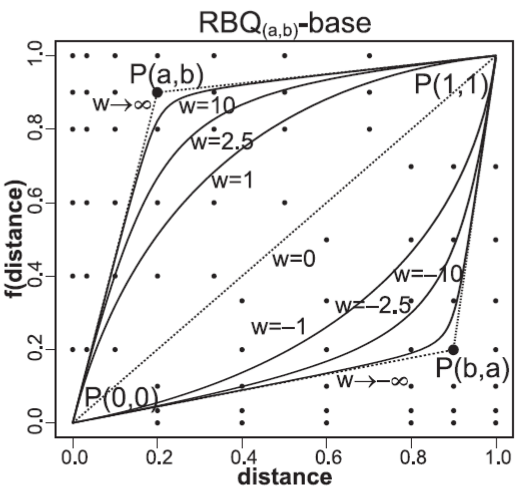


- Hence, let's find a function f that is:
  - Concave enough to turn δ into metric,
  - yet keeping idim as low as possible

Indexing non-metric spaces – mapping

- TriGen algorithm [Skopal, 2007]
  - "Metrization" of δ into f(δ)
  - Uses T-bases – set of modifying functions f, additionally parameterizable by a concavity/convexity weight w

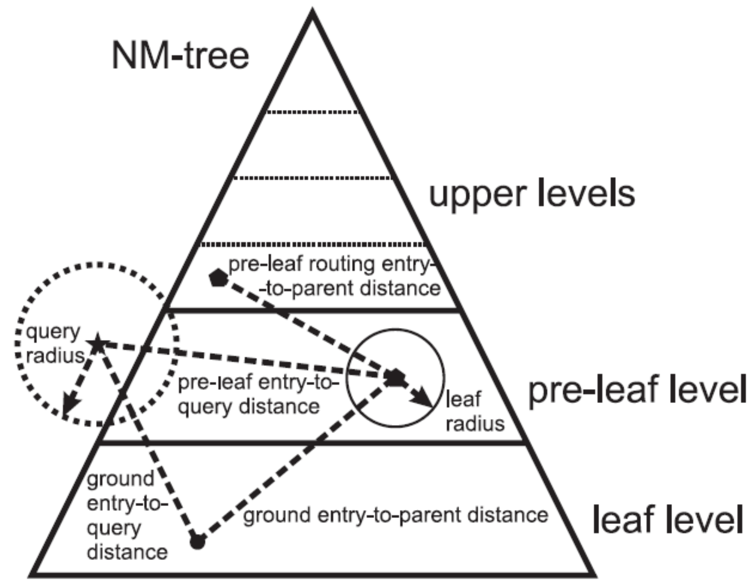




  - Uses T-error – the proportion of non-triangle triplets
    - Distance triplets sampled on S using f(δ)
  - Given a set of T-bases, δ and a sample of the database **S**, the algorithm finds the optimal **f** (T-base with w)
    - f is a candidate if T-error is below a user-defined threshold θ
    - Among the candidates the one is chosen for which idim is minimal

Indexing non-metric spaces – general NAM

- NM-tree – nonmetric M-tree
  - M-tree combined with TriGen algorithm
  - **Allows to set the retrieval error vs. performance trade-off at query time**



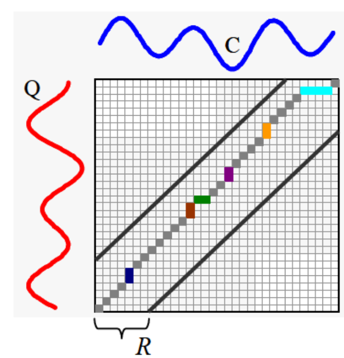
- The NM-tree idea [Skopal & Lokoč, 2008]
  - Using TriGen, find modifiers $\mathbf{f}_i$ for several T-error thresholds (including zero T-error)
  - Build M-tree using the zero T-error modified distance (i.e., full metric)
  - At query time, the T-error tolerance is a parameter
    - Each required distance value stored in M-tree is **inversely modified** from the metric one back to the original semimetric distance,
    - then it is **re-modified** using a different modifier (appropriate to the query parameter)
  - Additional requirement on T-bases – inverse symmetry, i.e., f(f(x,w),-w) = x

Indexing non-metric spaces – specific NAM

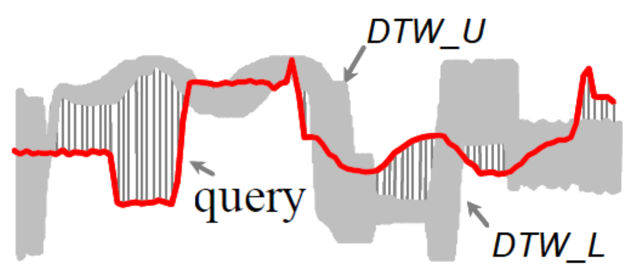
- The general techniques do not use any specific information
  - just black-box distance and a sample of the database is provided
- It is always better to use a specific solution (if developed), based on an internal knowledge, as:
  - Structure of the universe **U** (vector, string, set?)
  - The formula of δ (closed form available?)
  - Cardinality of the distance domain (discrete/continuous?)
  - Data/distance distribution in **S** (uniform/skewed?)
  - Typical query (e.g., sparse/dense vector?)
- Typically not reusable in other domains
  - Hence, hard to find a NAM specific to our setup

Indexing non-metric spaces – specific NAM

- Example – LB_Keogh for constrained DTW
  [Keogh et al, 2006]
  Lower-bounding distance, metric and cheap to compute O(n)
  - Envelope W=(DTW_U, DTW_L) created for a time series S
    $DTW\_U_i = \max(S_{i-R} : S_{i+R})$,
    $DTW\_L_i = \min(S_{i-R} : S_{i+R})$,
    R is the thickness of Sakoe-Chiba band



$$LB\_Keogh_{DTW}(Q,W) = \sqrt{\sum_{i=1}^{n} \begin{cases} (q_i - DTW\_U_i)^2 & \text{if } q_i > DTW\_U_i \\ (q_i - DTW\_L_i)^2 & \text{if } q_i < DTW\_L_i \\ 0 & \text{otherwise} \end{cases}}$$

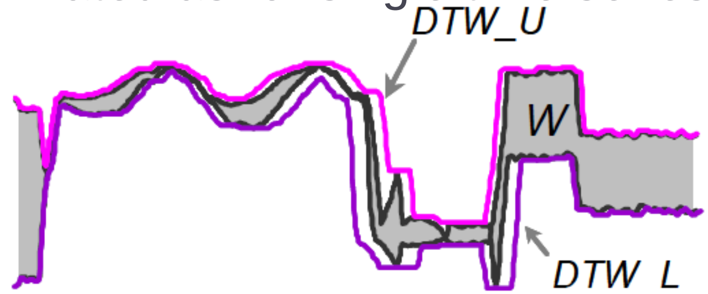


(images © Eamonn Keogh, eamonn@cs.ucr.edu)

Indexing non-metric spaces – specific NAM

- Example – LB_Keogh for constrained DTW

- Basic approach – filter & refine search
  1) Sequential search under LB_Keogh
  2) Check remaining candidates by DTW
- Extended approach – wedges
  = descriptors of multiple series
  - Wedge W = (U, L), $U_i = \max(C_{1i}, \ldots, C_{ki})$, $L_i = \min(C_{1i}, \ldots, C_{ki})$
  - W = k-dimensional rectangle, let's index it by, e.g., R-tree
  - For constrained DTW, W must be inflated as for single time series, i.e.,
    $DTW\_U_i = \max(W_{i-R} : W_{i+R})$,
    $DTW\_L_i = \min(W_{i-R} : W_{i+R})$



(image © Eamonn Keogh, eamonn@cs.ucr.edu)

Indexing non-metric spaces – specific NAM

- Example – inverted file and cosine similarity

- Used as an implementation of range query in vector model of information retrieval
  - documents $\mathbf{d}_i$, terms $\mathbf{t}_j$
  - term-by-document matrix – weights of terms in documents

$$
\begin{array}{c}
\begin{array}{ccccc} & t_1 & t_2 & & t_m \end{array} \\
\begin{array}{c} d_1 \\ d_2 \\ : \\ : \\ d_n \end{array}
\begin{pmatrix}
0.6 & 0 & \dots & 0.2 \\
0 & 0 & \dots & 0.1 \\
: & : & & : \\
: & : & & : \\
0.2 & 0.5 & \dots & 0.3
\end{pmatrix}
\end{array}
$$

- Only efficient for **cosine similarity** (or inner product) and **sparse query vector**
  - CosSim = (normed) sum of weight **multiplications**

$$\mathrm{CosSim}(\boldsymbol{d}_j, \boldsymbol{q}) = \frac{\vec{d}_j \cdot \vec{q}}{\left|\vec{d}_j\right| \cdot \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^2 \cdot \sum_{i=1}^{t} w_{iq}^2}}$$

Indexing non-metric spaces – specific NAM

- Example – inverted file and cosine similarity

- Efficient query processing
  - Visit only lists of terms having **nonzero weights** in query
  - Early termination provided when lists sorted wrt the weights

$$
\begin{array}{c}
 \\ d_1 \\ d_2 \\ : \\ : \\ d_n
\end{array}
\begin{array}{c}
\begin{array}{cccc} \text{mountain} & \text{forest} & & \text{nature} \end{array} \\
\begin{pmatrix}
0.6 & 0 & \dots & 0.2 \\
0 & 0 & \dots & 0.1 \\
: & : & & : \\
: & : & & : \\
0.2 & 0.5 & \dots & 0.3
\end{pmatrix}
\end{array}
$$

Query: <0, 0.5, 0.4>, similarity threshold = 0.05, inner product used

d_i sorted wrt the weights (desc.) →

mountain → $d_1(0.6)$, $d_n(0.2)$

forest → $d_n(0.5)$

...

nature → $d_n(0.3)$, $d_1(0.2)$, $d_2(0.1)$

Answer: $d_n(0.37)$, $d_1(0.08)$

- Cannot apply to Euclidean distance (!)
  - zero **+** nonzero weight = nonzero (all lists must be visited)

Indexing non-metric spaces

- Overview of methods for efficient non-metric search
- References to the sections of [Skopal & Bustos, 2011]

	Method	specialized/ general	approximate/ exact search	static/dynamic database	main-memory/ persistent	other characteristics	details in section
	Sequential scan	Gen.	Exact	Dynamic	Both	Requires no index	n/a
mapping	CSE	Gen.	Exact	Static	Main-mem.	Requires $O(n^2)$ space	4.5.2
mapping	TriGen	Gen.	Approx.	Static	Main-mem.	Simplifies the problem to metric case	4.5.3
mapping	Embeddings into vector spaces	Gen.	Approx.	Static	Main-mem.	Simplifies the problem to L_p space	4.5.4
mapping	Fuzzy logic	Gen.	Approx.	Static	Main-mem.	Provides transitive inequality, not implemented yet	4.5.5
general NAMs	NM-tree	Gen.	Approx.	Dynamic	Persistent	Based on M-tree, uses TriGen	4.6.1
general NAMs	QIC-M-Tree	Gen.	Exact	Dynamic	Persistent	Based on M-tree, requires user-defined metric lower bound distance	4.6.2
general NAMs	LCE	Gen.	Approx.	Static	Main-mem.	Exact only for database objects	4.6.3
general NAMs	Classification	Gen.	Approx.	Static	Main-mem.	Requires cluster analysis, limited scalability	4.6.4
general NAMs	Combinatorial approach	Gen.	Approx.	Static	Main-mem.	No implementation yet, only for NN search. Exact for large enough D.	4.6.5
specific NAMs	Inverted file	Spec.	Exact	Dynamic	Persistent	Cosine measure	4.7.2
specific NAMs	IGrid	Spec.	Exact	Static	Main-mem.	Specific L_p-like distance	4.7.3
specific NAMs	GEMINI(LB-Keogh)	Spec.	Exact	Both	Main-mem.	Uses lower bound distances	4.7.4
specific NAMs	FASTA/BLAST	Spec.	Approx.	Dynamic	Main-mem.	Approximate alignment	4.7.5

Challenges to the future

- scalability
  - mostly sequential scan nowadays, but the databases grow and get more complex, hence, indexing would be necessary
- indexability
  - how to measure indexability of nonmetric spaces?
- implementation specificity
  - specific vs. general NAMs
- efficiency vs. effectiveness
  - slower exact vs. faster approximate search
- extensibility
  - there exist other related aggregation/scoring (non-metric) concepts, to which non-metric indexing could contribute

Thank you for your attention!

… questions?

References

- **T. Skopal, B. Bustos, On Nonmetric Similarity Search Problems in Complex Domains, ACM Computing Surveys, 43(4), December 2011**
  http://siret.ms.mff.cuni.cz/skopal/pub/nmsurvey.pdf

- D. Berndt, J. Clifford. Using dynamic time warping to find patterns in time series. AAAI Workshop On Knowledge Discovery in Databases, 1994
- E. Chávez, G. Navarro, R. Baeza-Yates, J.L. Marroquín, Searching in metric spaces, ACM Computing Surveys, 33(3), 2001
- K.-S. Goh, B. Li, and E. Chang. DynDex: A dynamic and non-metric space indexer. 10th ACM International Conference on Multimedia, 2002
- D. Hoksza, J. Galgonek, Alignment-Based Extension to DDPIn Feature Extraction, International Journal of Computational Bioscience, Acta Press, 2010
- E. Keogh, L. Wei, X. Xi, S. Lee and M. Vlachos (2006) LB_Keogh Supports Exact Indexing of Shapes under Rotation Invariance with Arbitrary Representations and Distance Measures. VLDB 2006
- T. Skopal, Unified Framework for Fast Exact and Approximate Search in Dissimilarity Spaces, ACM Transactions on Database Systems, 32(4), 2007
- T. Skopal, J. Lokoč, NM-tree: Flexible Approximate Similarity Search in Metric and Non-metric Spaces, DEXA 2008, LNCS 5181, Springer
- P. Zezula, G. Amato, V. Dohnal, and M. Batko, Similarity Search: The Metric Space Approach, volume 32 of Advances in Database Systems. Springer, 2005