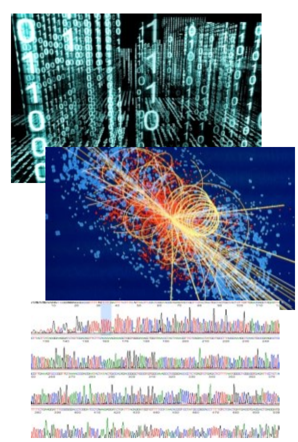
Scalable Vector Analytics
A Story of Twists and Turns

Themis Palpanas

Université Paris Cité
French University Institute

Extraction et Gestion des Connaissances (EGC) – Strasbourg (France), January 2025

In a Nutshell

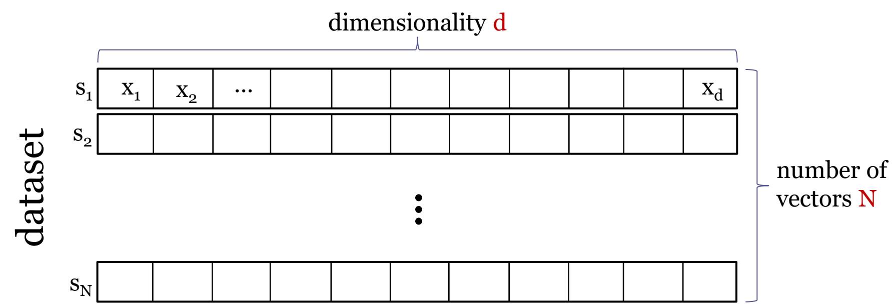


- data collected at unprecedented rates
- they enable data-driven scientific discovery
- lots of these data are high-d vectors
  - takes **days-weeks** to analyze big high-d vector collections

goal: analyze big high-d vectors in **seconds**

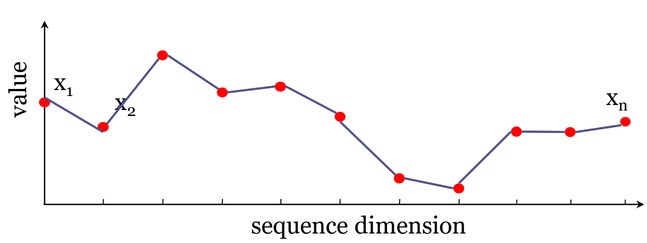
Vector Collections

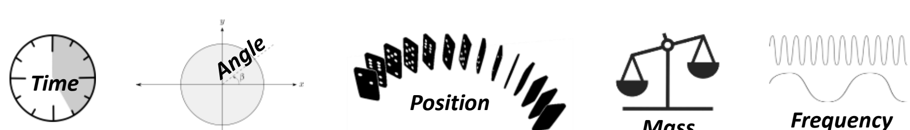
- represented as *N d*-dimensional vectors



Data Series

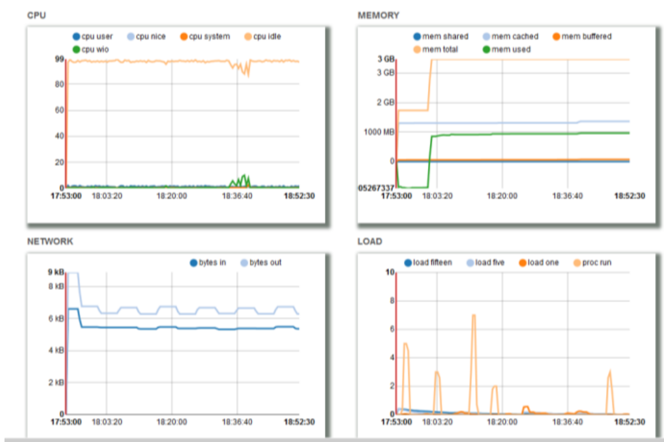
- Sequence of points ordered along some dimension





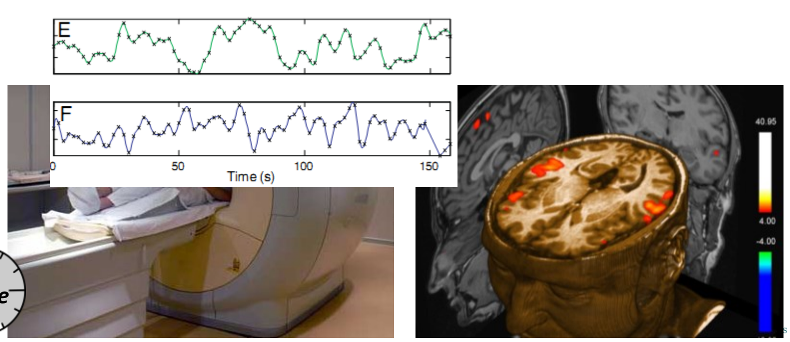
Data Centers

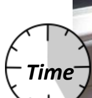
- cloud utilization/operation/health monitoring



Neuroscience

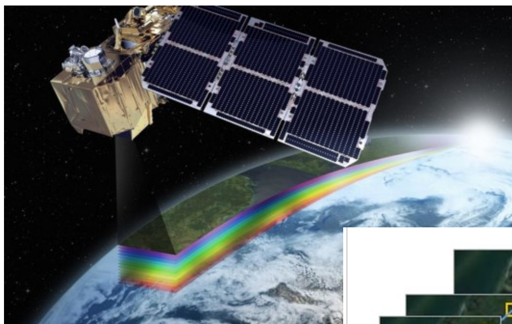
- functional Magnetic Resonance Imaging (fMRI) data
  - primary experimental tool of neuroscientists
  - reveal how different parts of brain respond to stimuli

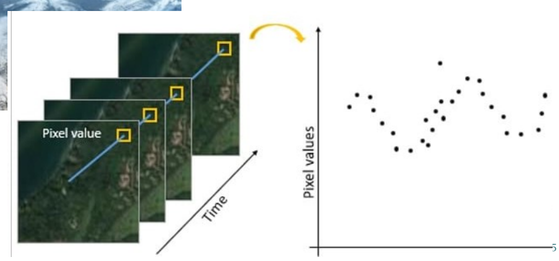




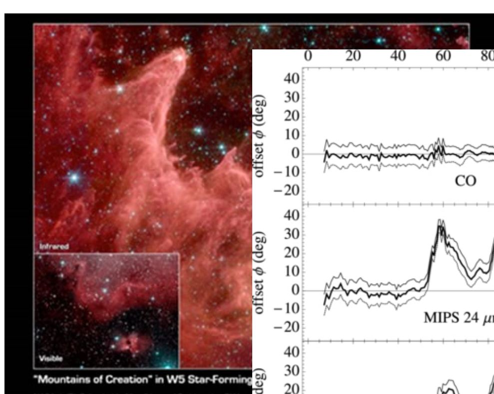
Remote Sensing

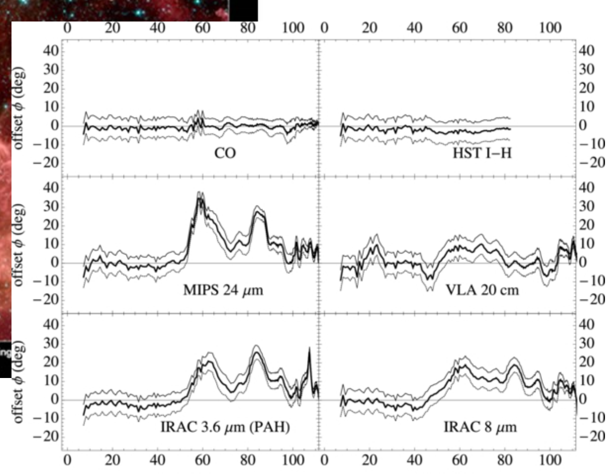
- Earth monitoring

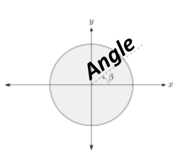




Astrophysics

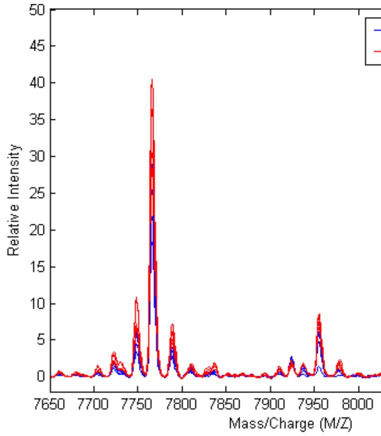


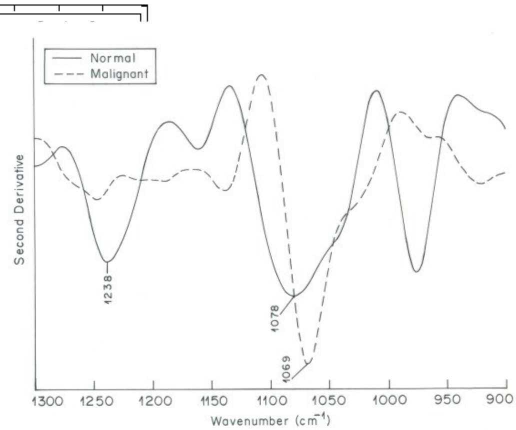




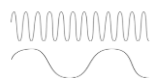
Schinnerer et al.

Medicine





Mass



Frequency

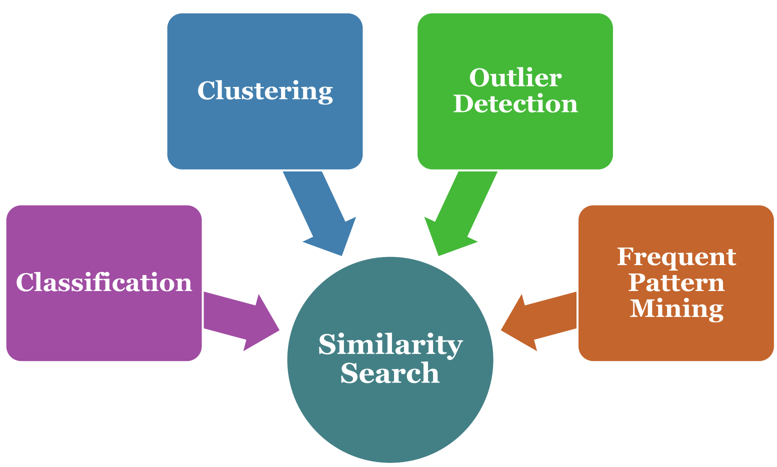
What do we want to do with them? - simple query answering

- select values in time interval
- select values in some range
- select some data series
- combinations of those

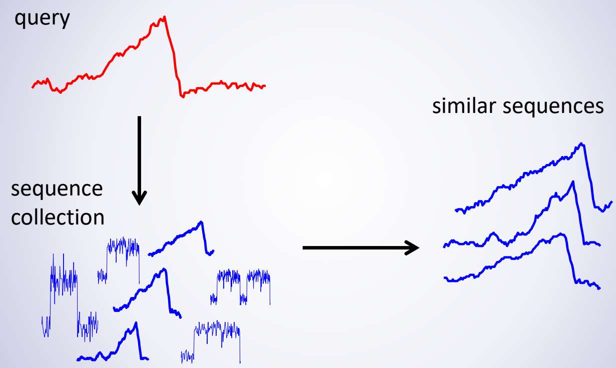
What do we want to do with them? - simple query answering

- a solved(?) problem
  - your favorite DBMS
  - …
  - InfluxData
  - kx
  - Riak TS
  - OpenTSDB
  - Gorilla/Beringei
  - TimescaleDB
  - KairosDB
  - Druid
  - …

What do we want to do with them? - complex analytics



What do we want to do with them? - complex analytics



What do we want to do with them? - complex analytics

Euclidean

$$D(X,Y) \equiv \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

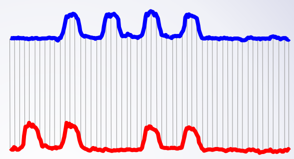
Dynamic Time Warping (DTW)

$$D_{dtw}(X,Y) = f(n,m)$$

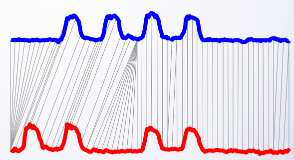
$$f(i,j) = \|x_i - y_j\| + \min \begin{cases} f(i, j-1) \\ f(i-1, j) \\ f(i-1, j-1) \end{cases}$$

What do we want to do with them? - complex analytics

Euclidean

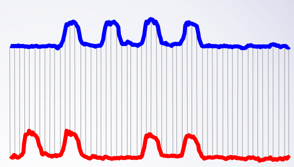


Dynamic Time Warping (DTW)



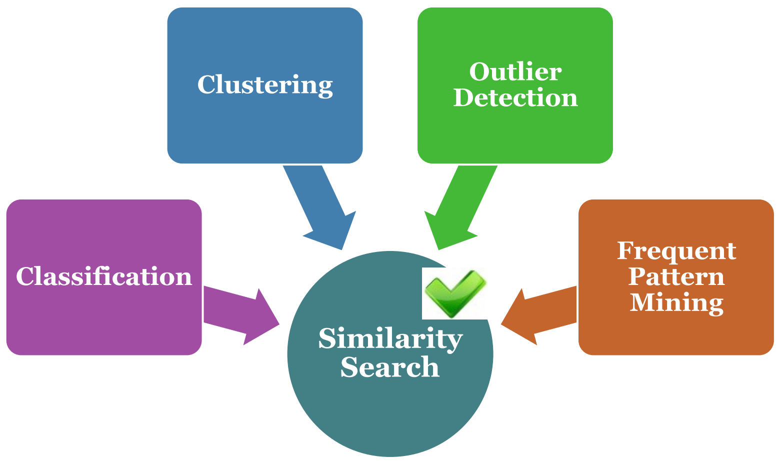
What do we want to do with them? - complex analytics

Euclidean

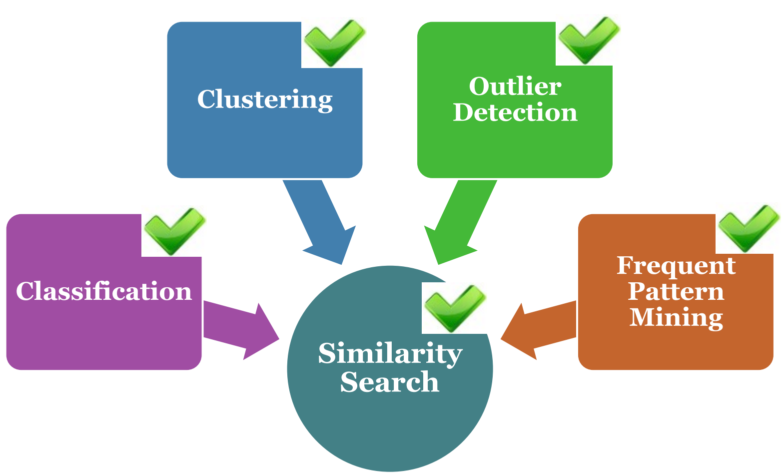


equivalence to
- Pearson's Correlation
- Cosine Similarity
- Maximum Inner Product Similarity

What do we want to do with them? - complex analytics



What do we want to do with them? - complex analytics



What do we want to do with them? - complex analytics

Clustering

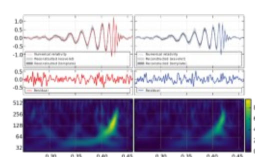
Outlier Detection

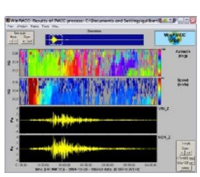
HARD, because of very high dimensionality: each data series has 100s-1000s of points!

even HARDER, because of very large size: millions to billions of data series (multi-TBs)!

Real Use-Cases

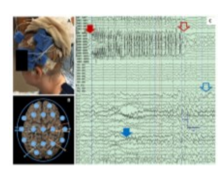
astrophysics: gravitational waves, **TB/hour**
partner: **European Gravitational Observatory (EGO)**
Pisa, Italy

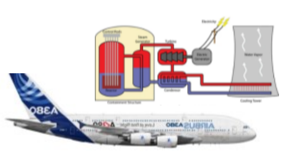




seismology: seismic sequences, **100s of TB**
partner: **Atomic Energy Commission (CEA)**
Paris, France

neuroscience: intracranial EEG sequences, **TB/patient**
partner: **Paris Brain Institute (ICM)**
Paris, France





engineering: operation monitoring, **TB-PB**
partners: **Airbus / Électricité de France (EDF)**
Toulouse / Paris, France

Nearest Neighbor (NN) Queries…

Publications

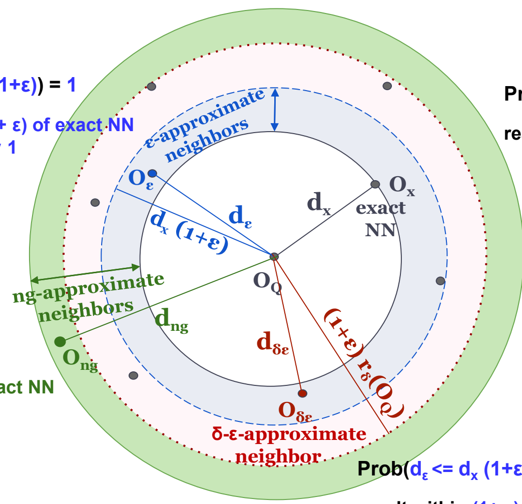
PVLDB'19

Prob(d_ε <= d_x (1+ε)) = 1

result within (1+ ε) of exact NN
with probability 1

Prob(d_x = min{d_i}) = 1

result is exact NN



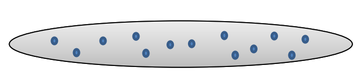
Prob(d_{ng} <>= ?) = ?

result within ? of exact NN

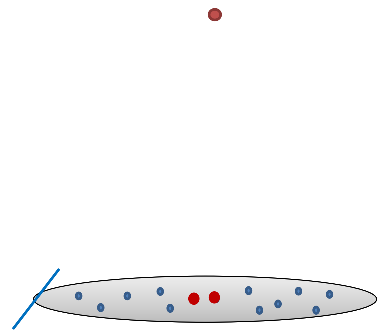
Prob(d_ε <= d_x (1+ε)) >= δ

result within (1+ ε) of exact NN
with probability at least δ

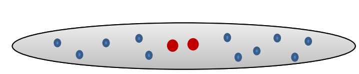
Similarity Search via
Serial Scan



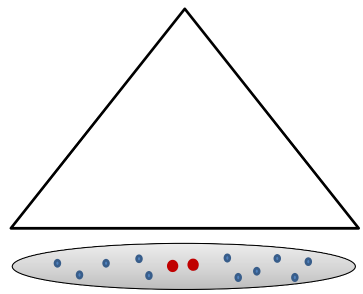
Similarity Search via
Serial Scan



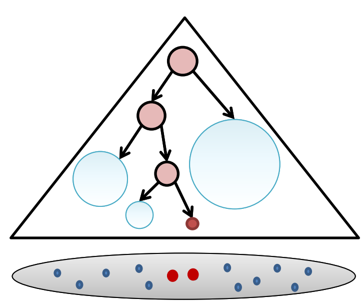
Similarity Search via
Serial Scan



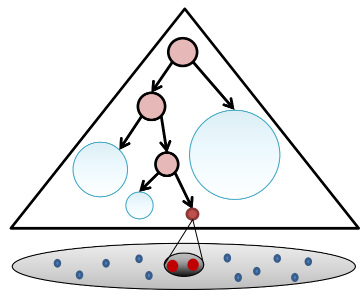
Similarity Search via
Indexing



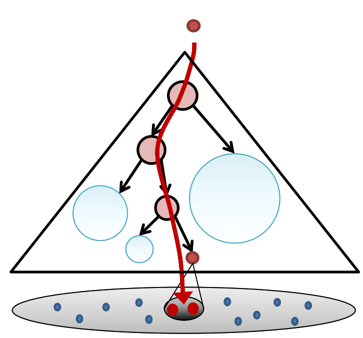
Similarity Search via
Indexing



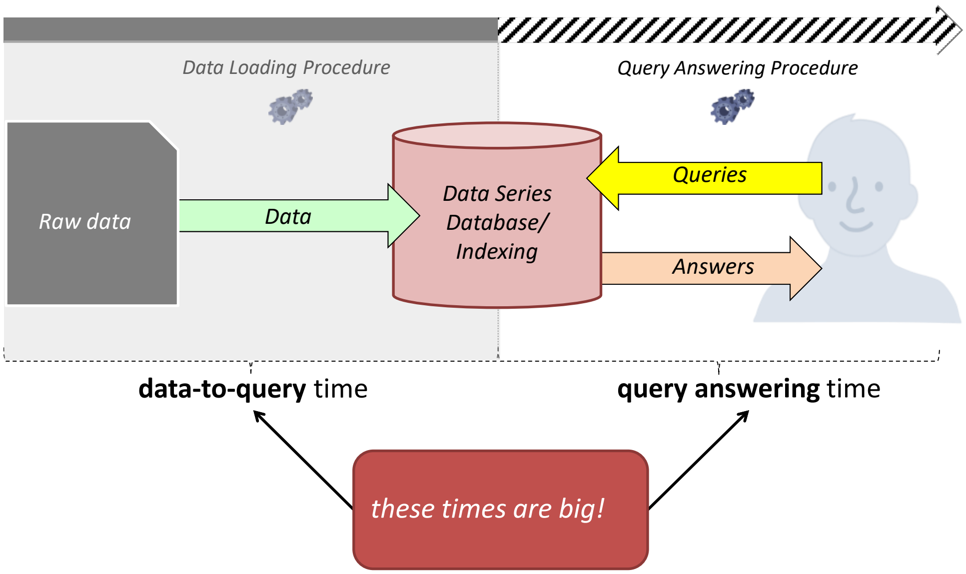
Similarity Search via
Indexing



Similarity Search via
Indexing

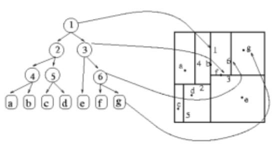


Query answering process



High-dimensional Indexes

KD-Tree

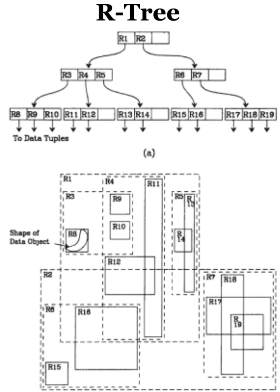


Yu Kai Him Otto, https://medium.com/

Publications

Bentley-CACM'75

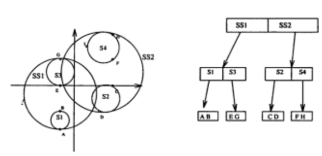
R-Tree



Publications

Guttman-SIGMOD'84

TV-Tree

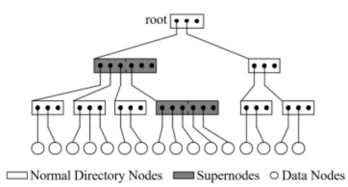


Publications

Lin et al-VLDBJ'94

High-dimensional Indexes

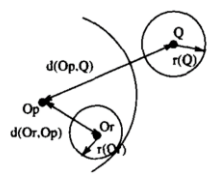
X-Tree



Publications

Berchtold et al - VLDB'96

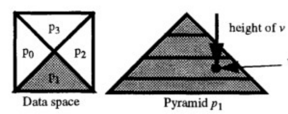
M-Tree



Publications

Ciaccia et al- VLDB'97

Pyramid technique



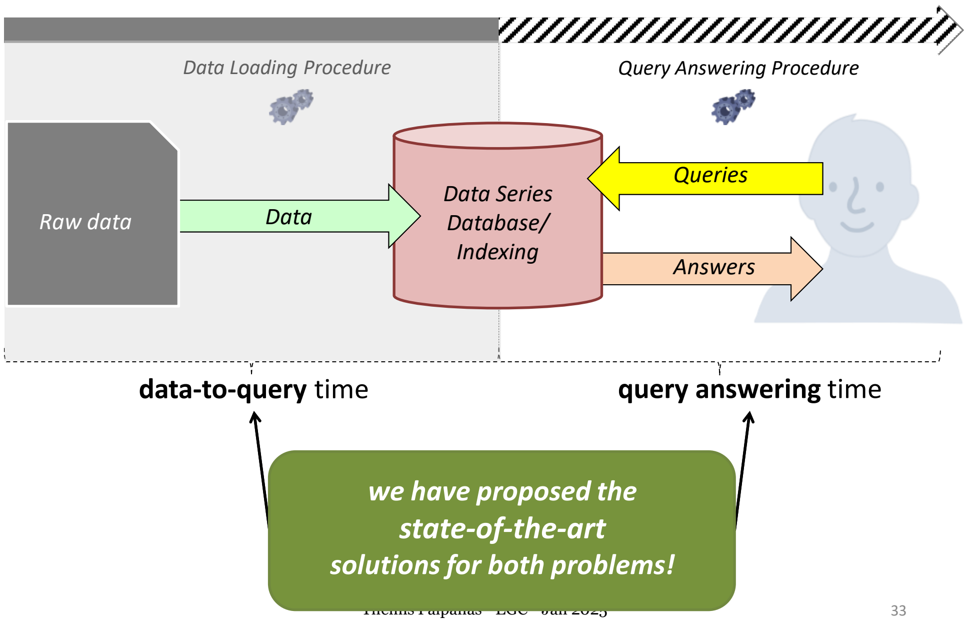
Publications

Berchtold et al - SIGMOD'98

High-dimensional Indexes in a new Era

- their world
  - focused on exact query answering
  - used relatively small dataset sizes (hundreds of thousand) and dimensionalities (few dozen)
  - tested for curse of dimensionality on uniform datasets(!)
- new world
  - looking for sublinear scalability performance on 1000x larger datasets with 100x more dimensions
  - some of these indexes (R-Trees, M-Trees) used for data series with less than impressive results
  - time series community proposed new indexes

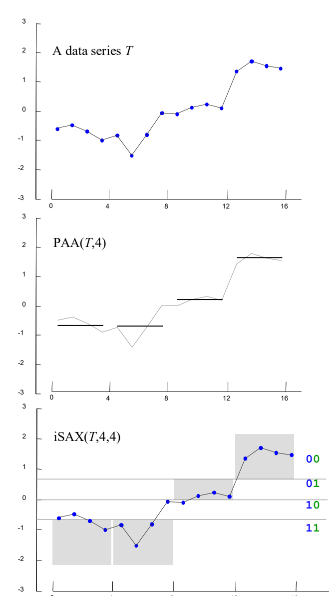
Query answering process



SAX Representation

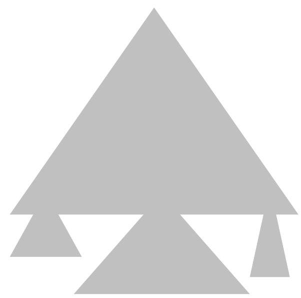
- **S**ymbolic **A**ggregate appro**X**imation (SAX)
  - **(1)** Represent data series T of length $\boldsymbol{n}$ with $\boldsymbol{w}$ segments using Piecewise Aggregate Approximation (PAA)
    - T typically normalized to $\mu = 0$, $\sigma = 1$
    - $\text{PAA}(T,w) = \overline{T} = \bar{t}_1, \ldots, \bar{t}_w$

      where $\bar{t}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} T_j$
  - **(2)** Discretize into a vector of symbols
    - Breakpoints map to small alphabet $\boldsymbol{a}$ of symbols



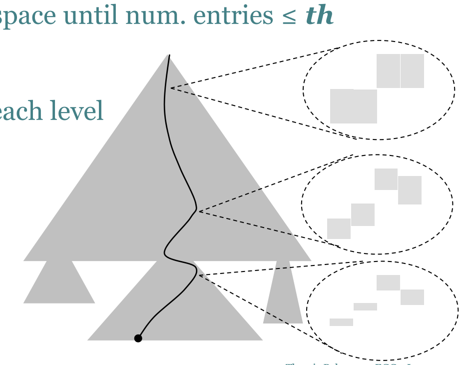
*i*SAX Index

- non-balanced tree-based index with non-overlapping regions, and controlled fan-out rate
  - base cardinality ***b*** (optional), segments ***w***, threshold ***th***
  - hierarchically subdivides SAX space until num. entries ≤ ***th***



*i*SAX Index

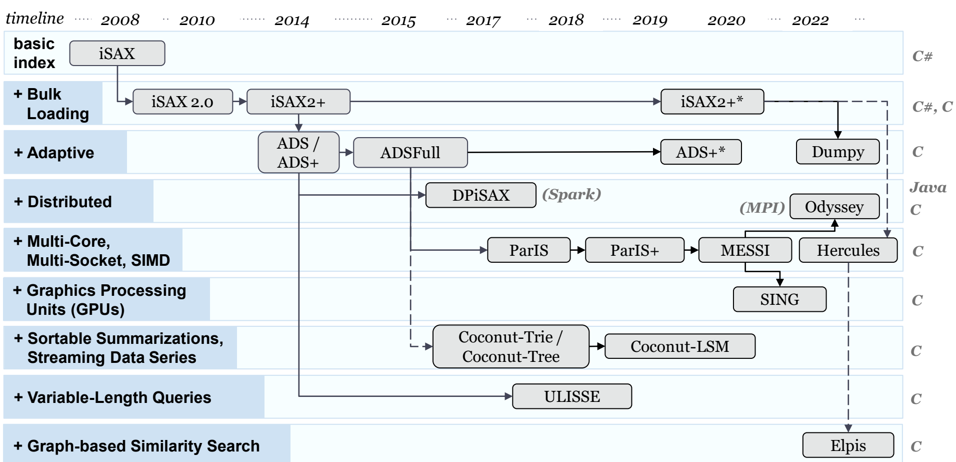
- non-balanced tree-based index with non-overlapping regions, and controlled fan-out rate
  - base cardinality ***b*** (optional), segments ***w***, threshold ***th***
  - hierarchically subdivides SAX space until num. entries ≤ ***th***
- Approximate Search
  - Match *i*SAX representation at each level
- Exact Search
  - Leverage approximate search
  - Prune search space
    - Lower bounding distance



iSAX Index Family Lineage Tree

Publications

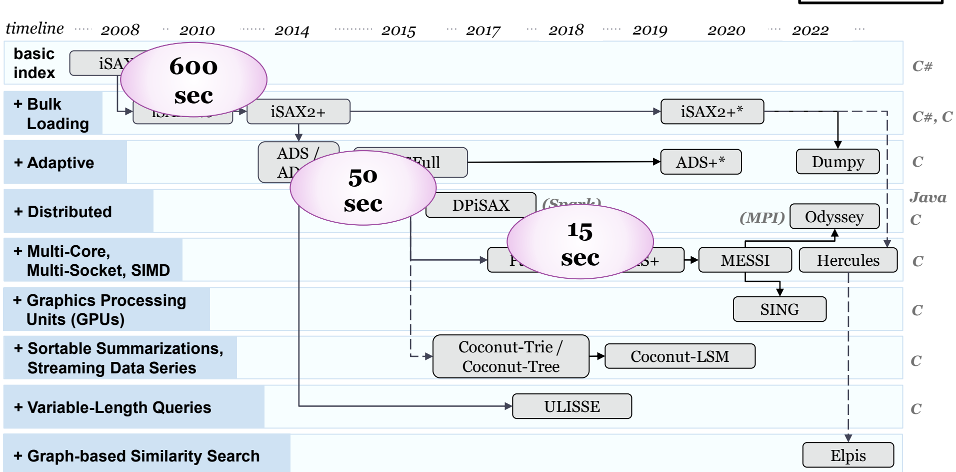
Palpanas-ISIP'19



iSAX Index Family Lineage Tree

Publications

Palpanas-ISIP'19

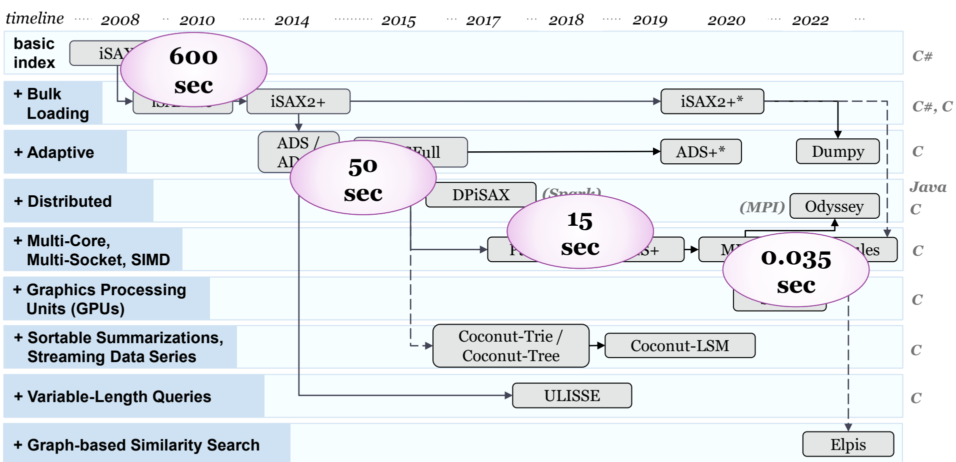


execution time for **1 similarity search query on a 100GB dataset *on disk***

iSAX Index Family Lineage Tree

Publications

Palpanas-ISIP'19



execution time for **1 similarity search query on a 100GB dataset *in memory***

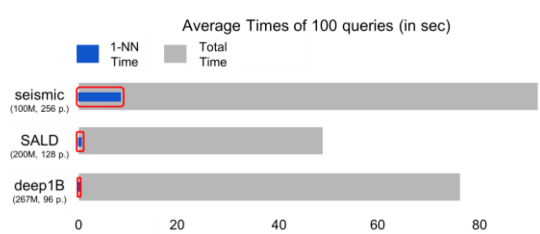
Further Advances

Publications

Gogolou - BigVis'19

Gogolou - SIGMOD'20

Gogolou - SIGMOD'20

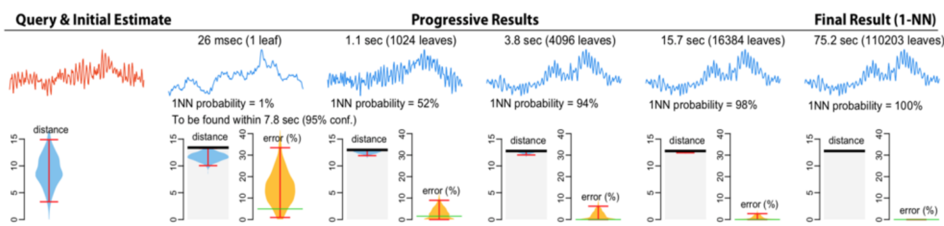


- how do we further reduce the wasted (gray) effort?
  - **progressive query answering**
    - produce intermediate answers with (probabilistic) quality guarantees

Further Advances: Progressive Query Answering

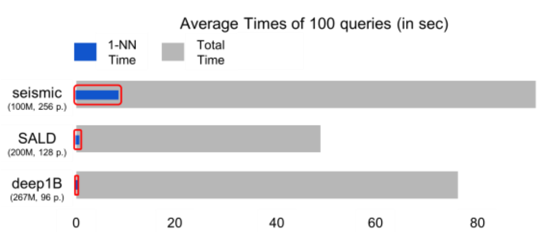
Publications
- Gogolou - BigVis'19
- Gogolou - SIGMOD'20
- Gogolou - SIGMOD'20

- interaction with users offers new opportunities
  - progressive answers
    - produce intermediate results
      - iteratively converge to final, correct solution
    - provide bounds on the errors (of the intermediate results) along the way



Further Advances

Publications
- Gogolou - BigVis'19
- Gogolou - SIGMOD'20
- Gogolou - SIGMOD'20
- Wang - KDD'21



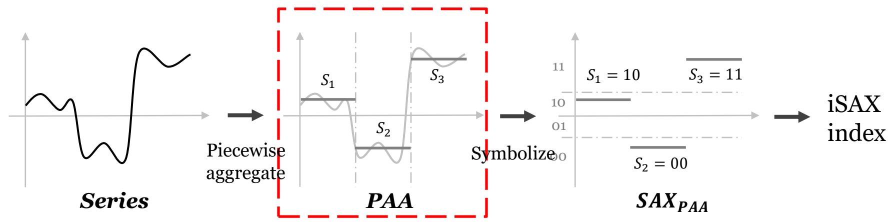
- how do we further reduce the wasted (gray) effort?
  - **progressive query answering**
    - produce intermediate answers with (probabilistic) quality guarantees
  - **learned summarizations + index structures**
    - adapt to data characteristics
    - build more efficient indexes
    - perform more effective pruning

Further Advances: Learning

Publications

Wang - KDD'21

- Series Approximation Network (SEAnet)
  - novel autoencoder architecture
  - learns deep embedding approximations
  - uses those for similarity search

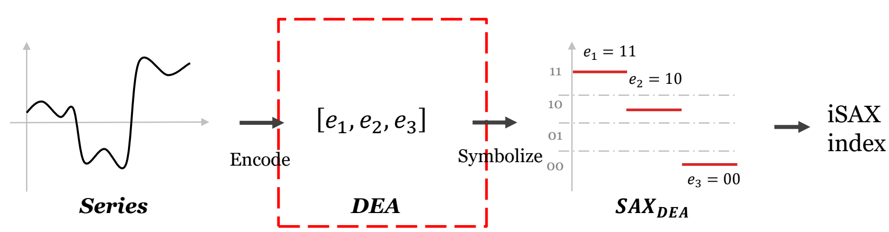


Further Advances: Learning

Publications

Wang - KDD'21

- Series Approximation Network (SEAnet)
  - novel autoencoder architecture
  - learns deep embedding approximations
  - uses those for similarity search

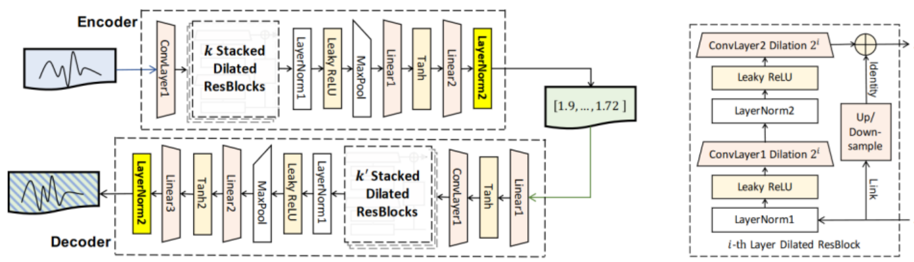


Further Advances: Learning

Publications

Wang - KDD'21

- Series Approximation Network (SEAnet)
  - is an exponentially dilated ResNet architecture + Sum of Squares regularization
  - minimizes
    - reconstruction error
    - difference between distance of two vectors in embedded space and distance in original space



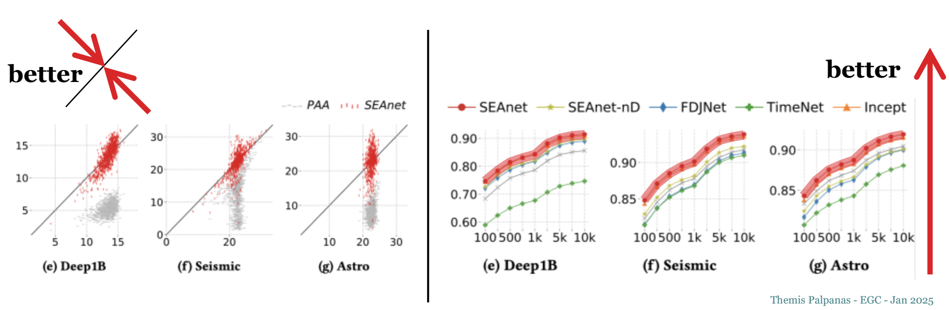
(a) SEAnet Architecture (b) Dilated ResBlock in SEAnet

Further Advances: Learning

Publications

Wang - KDD'21

- Series Approximation Network (SEAnet)
  - is an exponentially dilated ResNet architecture + Sum of Squares regularization
  - minimizes
    - reconstruction error
    - difference between distance of two vectors in embedded space and distance in original space

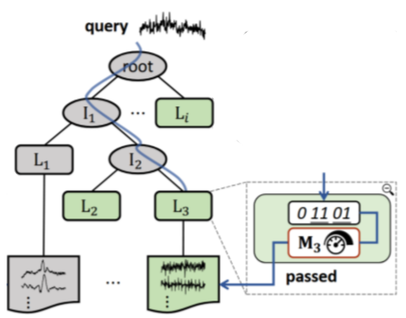


Further Advances: Learning

Publications

Wang - SIGMOD'25

- Learned Filters (LeaFi)
  - machine learning models that make pruning decisions
  - applied when pruning based on lower bounding is not possible



up to 20x more pruning
up to 32x faster query answering

Publications

WIMS'20

High-d Vectors
Beyond Data Series

- two sides of the same(?) coin
  - data series as multidimensional points
  - for a specific ordering of the dimensions
- everything we discussed applicable to high-d vectors, too!

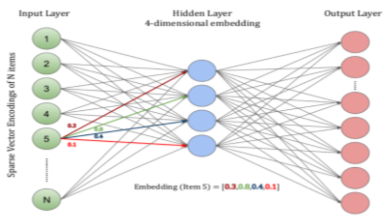
Publications

WIMS'20

High-d Vectors Beyond Data Series

- two sides of the same(?) coin
  - data series as multidimensional points
  - for a specific ordering of the dimensions
- everything we discussed applicable to high-d vectors, too!

sequences
text
images
video
graphs
molecules
...



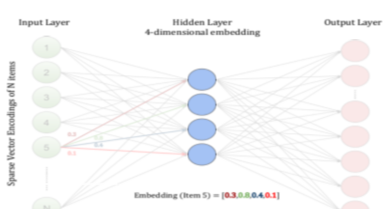
Publications

WIMS'20

High-d Vectors Beyond Data Series

- two sides of the same(?) coin
  - data series as multidimensional points
  - for a specific ordering of the dimensions
- everything we discussed applicable to high-d vectors, too!

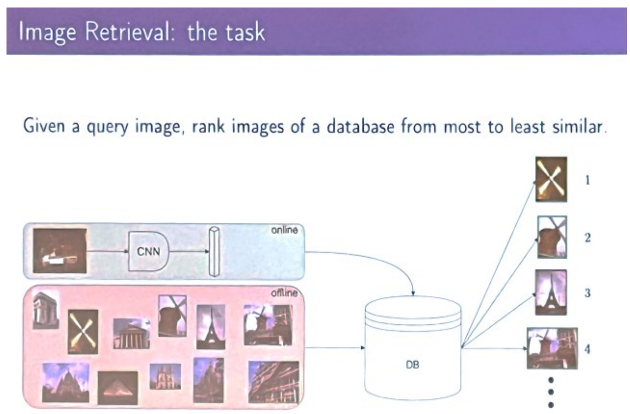
sequences
text
images
video
graphs
molecules
...



deep embeddings
high-d vectors learned using a DNN

Deep Embeddings
Similarity Search Applications

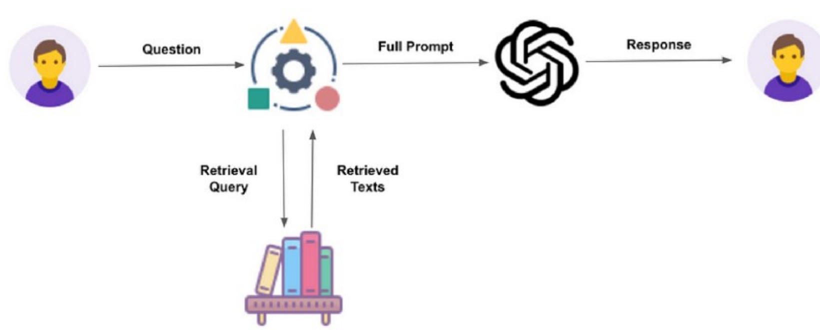
- image retrieval



Ramzi et al., Cap&RFIAP'22

Deep Embeddings
Similarity Search Applications

- image retrieval
- retrieval augmented generation (RAG)



https://pvml.com/

Deep Embeddings Similarity Search Applications

- image retrieval
- retrieval augmented generation (RAG)
- recommendations
- entity matching
- fraud detection
- drug discovery
- …

Ramzi et al., Cap&RFIAP'22

Data Series Indexes in a new Era

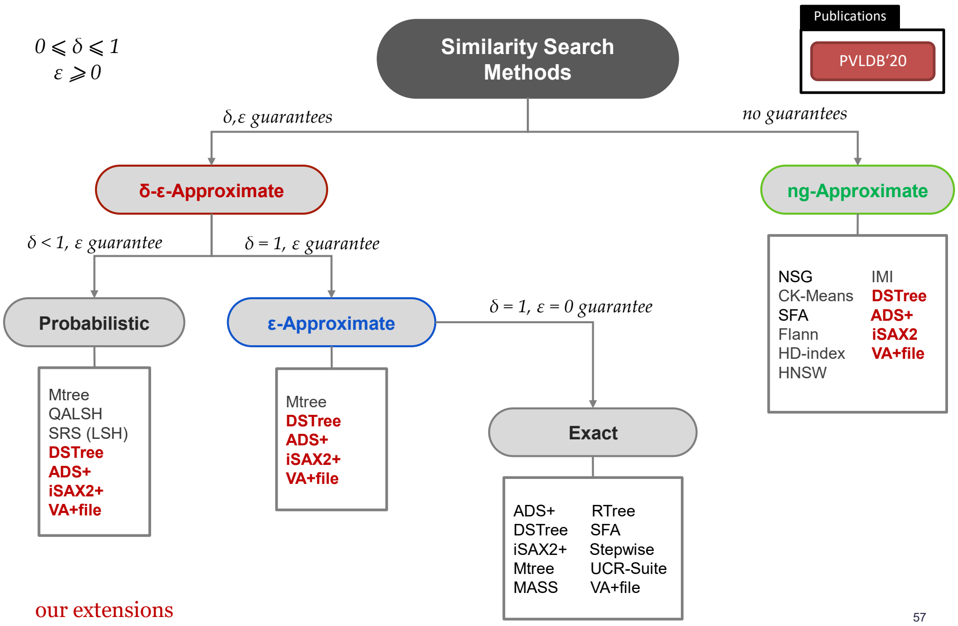
- their world
  - focused on exact query answering
  - centered discussion around data series shapes/patterns
- new world
  - looking for ultra-fast performance for applications that tolerate approximate answers
  - machine learning and related communities proposed new indexes

Publications

WIMS'20

High-d Vectors Indexes

- techniques for approximate similarity search in high-d vectors
  - [LSH (SRS)]
  - space quantization and inverted files (IMI)
  - k-NN graphs (HNSW)
- how do these high-d vector techniques compare to data series techniques?
  - have conducted extensive experimental comparison



our extensions

Data Series vs. high-d Vectors

Publications

PVLDB'20

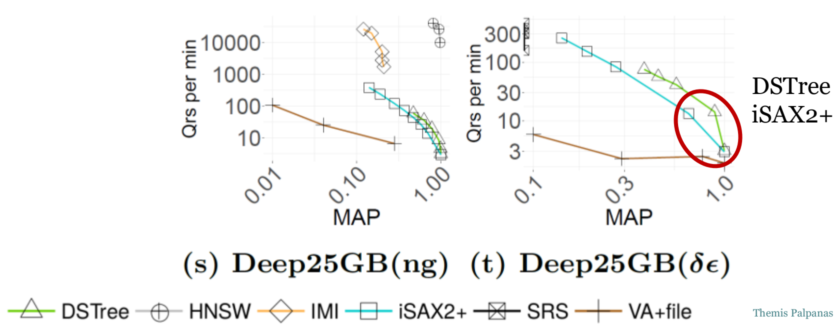
- **data series techniques** are the **overall winners**, even on **general high-d vector** data

Data Series vs. high-d Vectors

Publications

PVLDB'20

- **data series techniques** are the **overall winners**, even on **general high-d vector** data
  - perform the best for approximate queries with probabilistic guarantees (δ-ε-approximate search), in-memory and on-disk



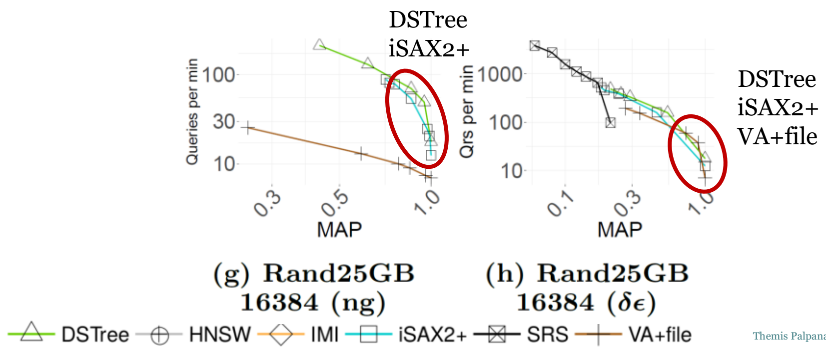
(s) Deep25GB(ng) (t) Deep25GB($\delta\epsilon$)

Data Series vs. high-d Vectors

Publications

PVLDB'20

- **data series techniques** are the **overall winners**, even on **general high-d vector** data
  - perform the best for approximate queries with probabilistic guarantees (δ-ε-approximate search), in-memory and on-disk
  - perform the best for long vectors, in-memory and on-disk

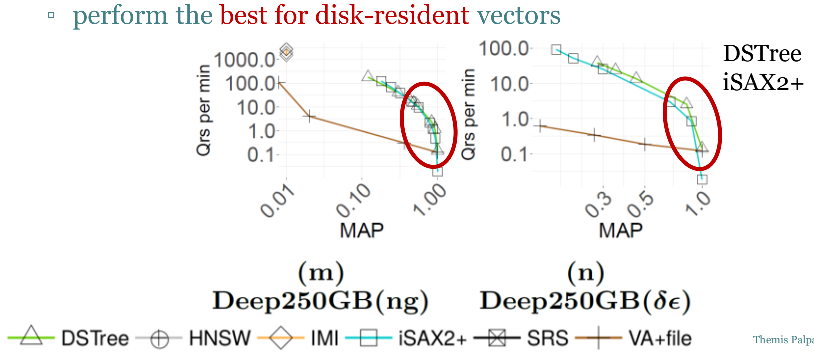


Data Series vs. high-d Vectors

Publications

PVLDB'20

- **data series techniques** are the **overall winners**, even on **general high-d vector** data
  - perform the best for approximate queries with probabilistic guarantees (δ-ε-approximate search), in-memory and on-disk
  - perform the best for long vectors, in-memory and on-disk
  - perform the best for disk-resident vectors

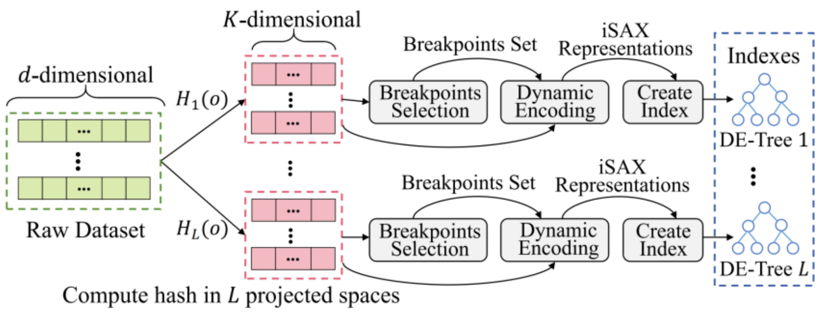


Hybrid (iSAX + LSH): DET-LSH

Publications

Wei et al. - PVLDB'24

- **DET-LSH** combines tree and LSH for efficient indexing and approximate search with probabilistic guarantees

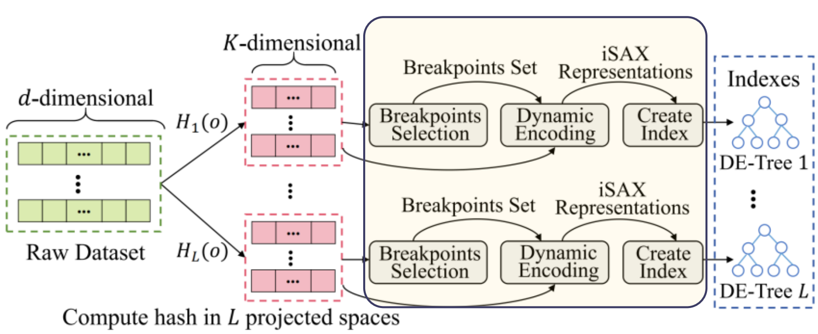


Hybrid (iSAX + LSH): DET-LSH

Publications

Wei et al. - PVLDB'24

- **DET-LSH** combines tree and LSH for efficient indexing and approximate search with probabilistic guarantees

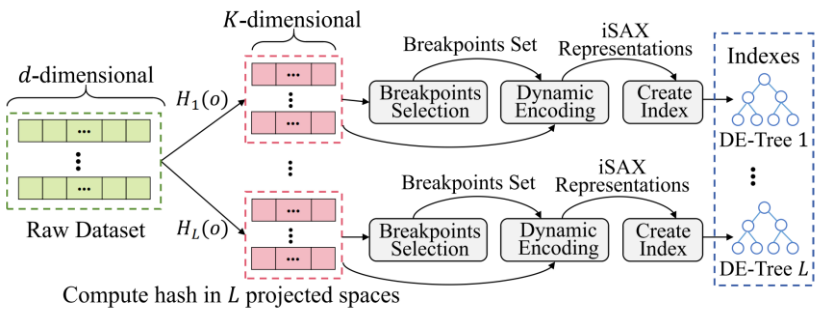


Hybrid (iSAX + LSH): DET-LSH

Publications

Wei et al. - PVLDB'24

- **DET-LSH** combines tree and LSH for efficient indexing and approximate search with probabilistic guarantees



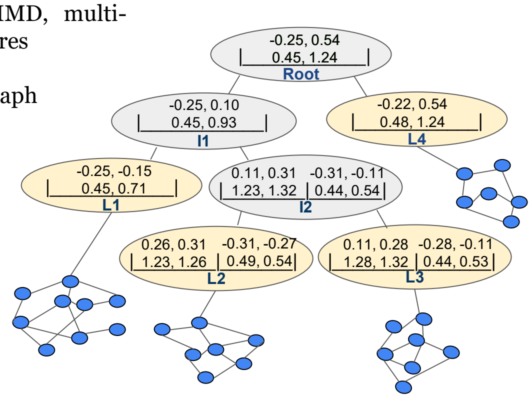
up to 6x faster indexing and 2x faster query answering (than standard LSH methods)

Hybrid (DSTree + HNSW): ELPIS
Parallel, In-Memory Indexing of Sequences

Publications

Azizi-PVLDB'23

- In-memory solution for SIMD, multi-core, multi-socket architectures
- **ELPIS** combines tree and graph structures for efficient in-memory ng-approximate vector similarity search.



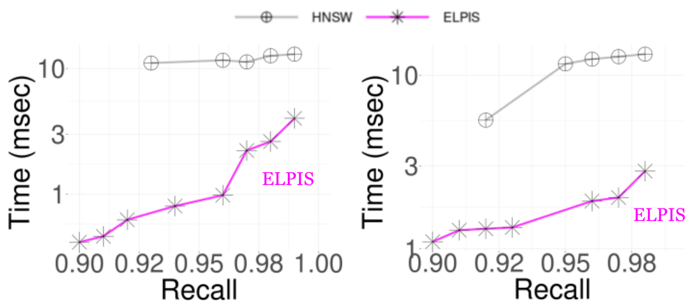
Hybrid (DSTree + HNSW): ELPIS

Publications

Azizi-PVLDB'23

Parallel, In-Memory Indexing of Sequences

- Query Performance on 1B vectors datasets (Sift, Deep)



Sift Dataset

Deep Dataset

ELPIS answers **10-NN queries** in **~3 msec** for a dataset of **1 billion vectors** with **recall 0.99**

Conclusions

- high-d vectors is a very common data type
  - across several different domains and applications
- complex high-d vector analytics are challenging
  - have very high inherent complexity
- data series management/indexing techniques provide state-of-the-art performance
  - work for data series and general high-d vectors (and embeddings)
  - lead to fast complex analytics and machine learning
- several exciting research opportunities
  - distributed solutions
  - progressive analytics
  - learned (data-adaptive) summarizations/data structures

Some more open questions…

- what are the theoretical properties of existing solutions?
  - best/expected/worst time performance
  - best/expected/worst accuracy (for approximate query answering)
- can we predict performance based on data characteristics?
  - analytical results (eg, based on different distributions)
  - learned
- how do we integrate vector similarity search in databases?
  - combine similarity search with other predicates
  - optimization
- what are the right benchmarks to evaluate high-d vector indices?
  - data and query workloads (currently: most queries are easy)
  - evaluation measures (currently: time and recall)

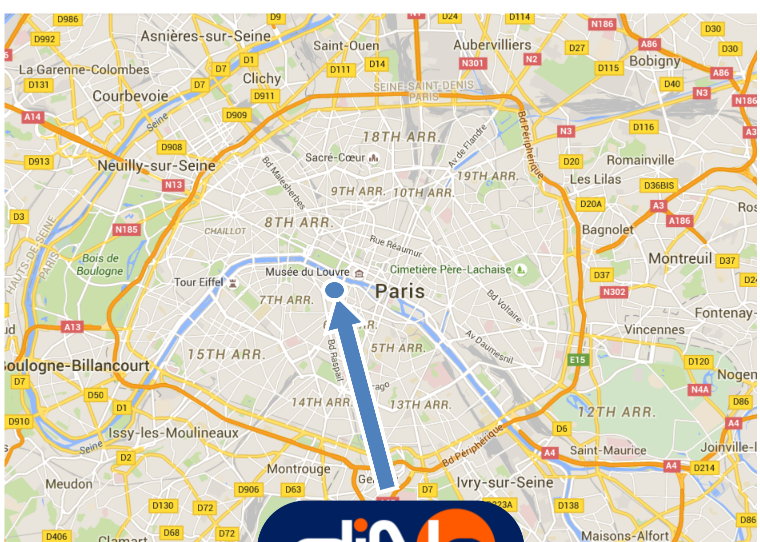
Going forward

- high-d vector similarity search relevant to many communities
  - data management
  - time series
  - information retrieval
  - text search
  - machine learning
  - deep learning
  - parallel and distributed computing

Going forward

- high-d vector similarity search relevant to many communities
  - data management
  - time series
  - information retrieval
  - text search
  - machine learning
  - deep learning
  - parallel and distributed computing
- research on this problem fragmented across communities
  - open communication channels among these communities
  - initiate discussions
- start collaborations!

Data-Intensive and Knowledge-Oriented systems



thank you!

google: **Themis Palpanas**
visit: http://**nestordb.com**

work supported by:

EU Horizon projects AI4Europe (101070000), TwinODIS (101160009),
ARMADA (101168951), DataGEMS (101188416), RECITALS (101168490)

72