New Trends in
High-D Vector Similarity Search
AI-driven, Progressive, and Distributed

Karima Echihabi

Mohammed VI Polytechnic University

Kostas Zoumpatianos

Snowflake Computing
work done while at Harvard University & University of Paris

Themis Palpanas

University of Paris & French University Institute (IUF)

International Conference on Very Large Data Bases (VLDB), August 2021

Questions This Tutorial Answers

- how important are high-dimensional (high-d) data nowadays?

- what types of analyses are performed on high-d data?
- how can we speed up such analyses?

- what are the different kinds of similarity search?
- what are the state-of-the-art high-d similarity search methods?
- how do methods designed for data series compare to those designed for general high-d vector similarity search?

- how do similarity search techniques support interactivity?
- how can AI help similarity search and vice versa?
- which similarity search techniques exploit modern hardware and distribution?

- what are the open research problems in this area?

Acknowledgements

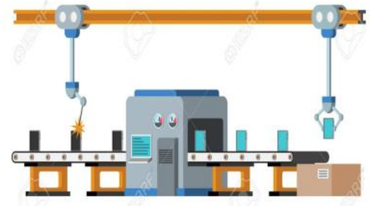
- thanks for slides to
  - Michail Vlachos
  - Eamonn Keogh
  - Panagiotis Papapetrou
  - George Kollios
  - Dimitrios Gunopulos
  - Christos Faloutsos
  - Panos Karras
  - Peng Wang
  - Liang Zhang
  - Reza Akbarinia
  - Stanislav Morozov
  - Sarath Shekkizhar
  - Marco Patella
  - Wei Wang
  - Yury Malkov
  - Matthijs Douze
  - Cong Fu
  - Arnab Bhattacharya
  - Qiang Huang
  - Artem Babenko
  - David Lowe
  - Walid Aref
  - John Paparrizos
  - Conglong Li
  - Saravanan Thirumuruganathan

Introduction, Motivation

High-d data are everywhere

Finance

Paleontology



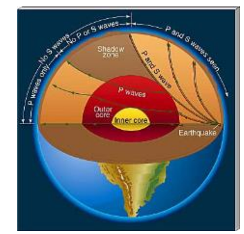
Manufacturing

Aviation

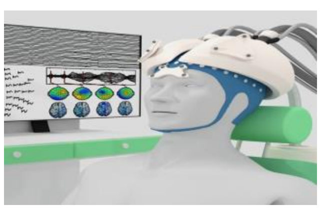
Agriculture

Astronomy

Criminology

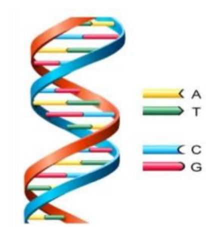


Seismology



Neuroscience

Medicine



Biology

High-d data are everywhere

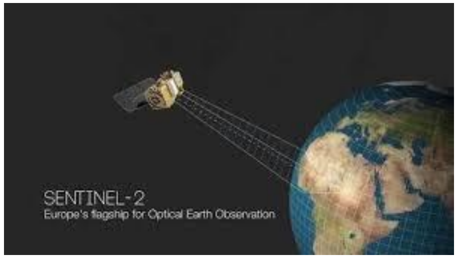
- operation health monitoring
  - classification, anomaly detection
- data integration
  - entity resolution, data discovery
- recommender systems
  - predict user interest
- information retrieval
  - similarity search
- software engineering
  - find software dependencies
- cybersecurity
  - network usage profiling, intrusion detection
- …

High-d collections are massive

≈ 500 ZB per year

≈ 130 TB

> 40 PB per day



> 5 TB per day

facebook

> 500 TB per day

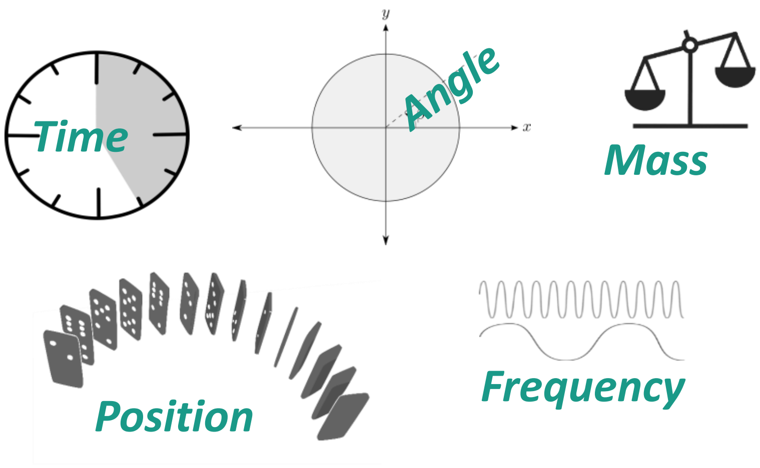
1 PB = 1 thousand TB

1 ZB = 1 billion TB

Popular High-d data

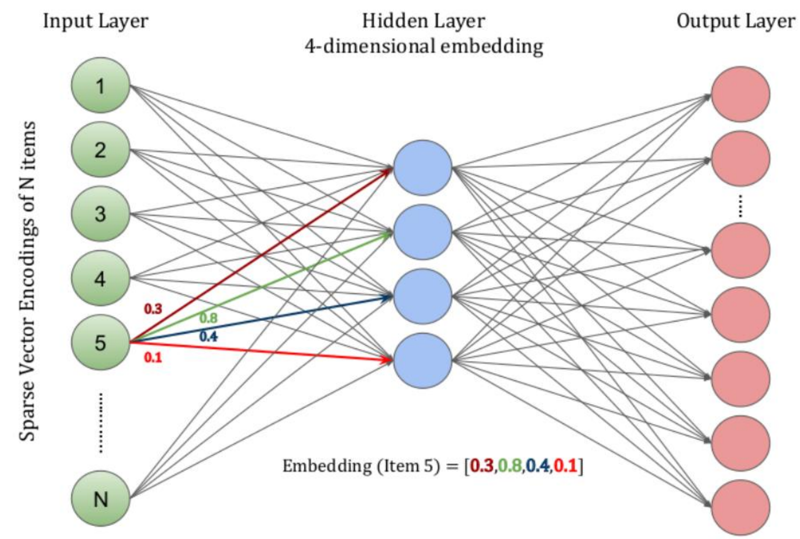
Data series

A collection of points ordered over a dimension



Deep Embeddings

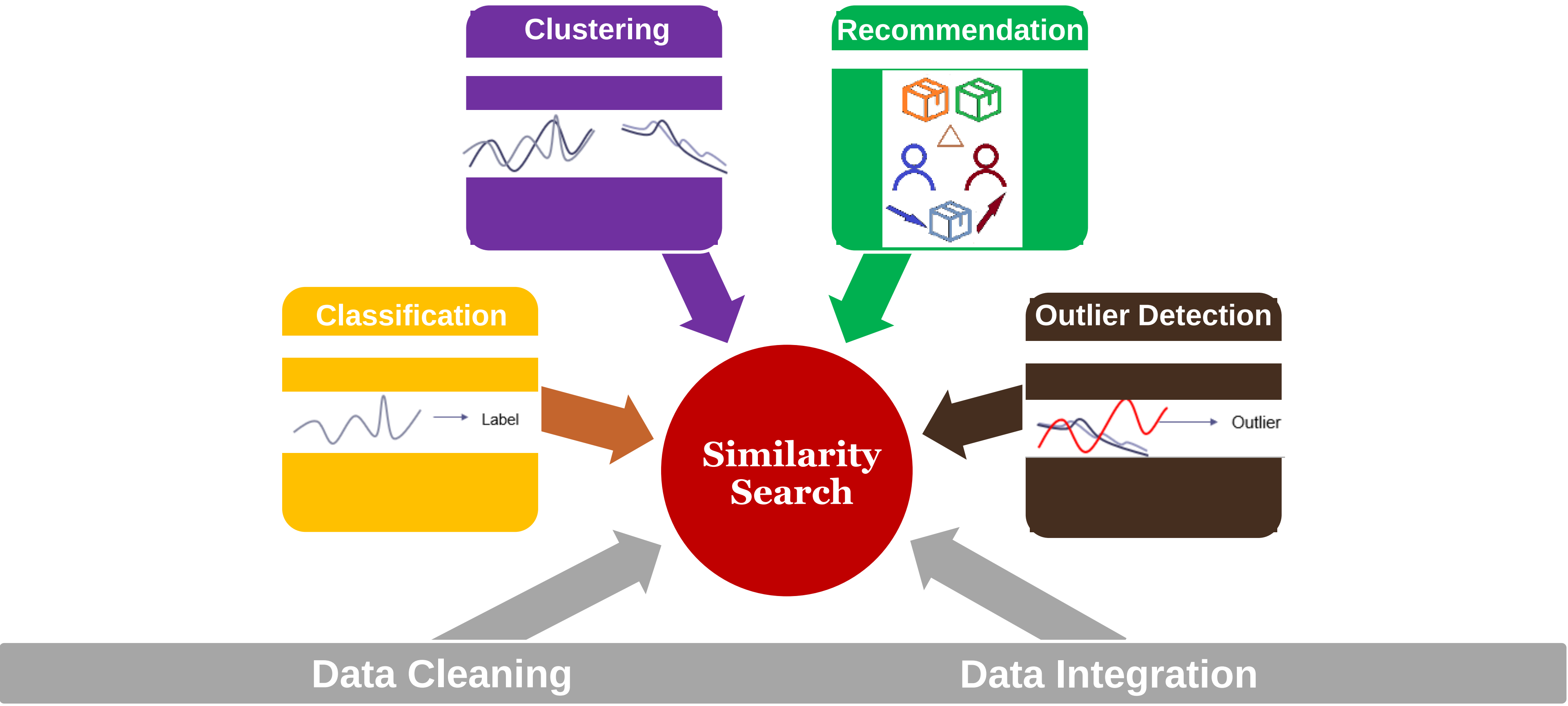
A high-d vector learned from data using a DNN



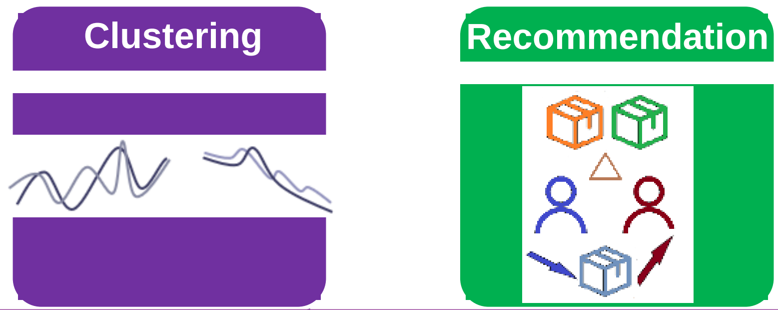
embedded
text, images, video, graphs, etc.

High-d data -> High-d vectors

Extracting value requires analytics



Extracting value requires analytics



HARD, because of very high dimensionality:
each high-d vector has 100s-1000s of dimensions!

even HARDER, because of very large size:
millions to billions of high-d vectors (multi-TBs)!

Data Cleaning

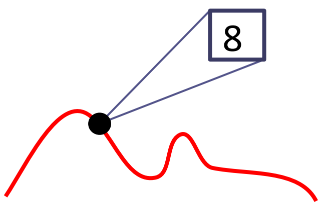
Data Integration

High-d Similarity Search

High-d Similarity Search Problem Variations

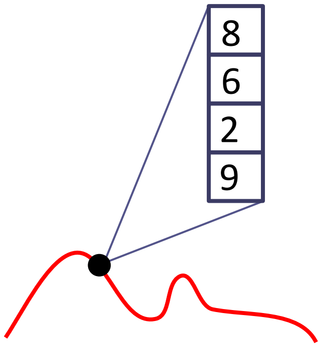
Problem Variations

Series



<u>Univariate</u>

each point represents one value (e.g., temperature)

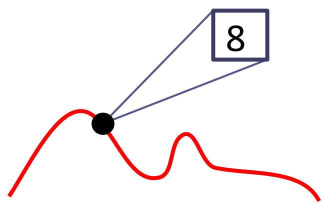


<u>Multivariate</u>

each point represents many values (e.g., temperature, humidity, pressure, wind, etc.)

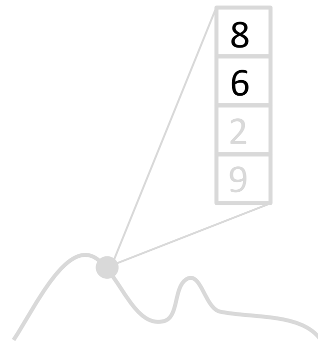
Problem Variations

Series



Univariate

each point represents one value (e.g., temperature)



Multivariate

each point represents many values (e.g., temperature, humidity, pressure, wind, etc.)

Publications

Ding-PVLDB'08

Paparrizos-SIGMOD'20

Problem Variations

Data Series Distance Measures

- similarity search is based on measuring distance between sequences
- dozens of distance measures have been proposed
  - lock-step
    - Minkowski, Manhattan, Euclidean, Maximum, DISSIM, …
  - sliding
    - Normalized Cross-Correlation, SBD, …
  - elastic
    - DTW, LCSS, MSM, EDR, ERP, Swale, …
  - kernel-based
    - KDTW, GAK, SINK, …
  - embedding
    - GRAIL, RWS, SPIRAL, SEAnet, …

Problem Variations

Data Series Distance Measures

Publications

Ding-PVLDB'08

Paparrizos-SIGMOD'20

- similarity search is based on measuring distance between sequences
- dozens of distance measures have been proposed
  - lock-step
    - Minkowski, Manhattan, Euclidean, Maximum, DISSIM, ...
  - sliding
    - Normalized Cross-Correlation, SBD, ...
  - elastic
    - DTW, LCSS, MSM, EDR, ERP, Swale, ...
  - kernel-based
    - KDTW, GAK, SINK, ...
  - embedding
    - GRAIL, RWS, SPIRAL, SEAnet, ...

Problem Variations

High-d Vectors Distance Measures

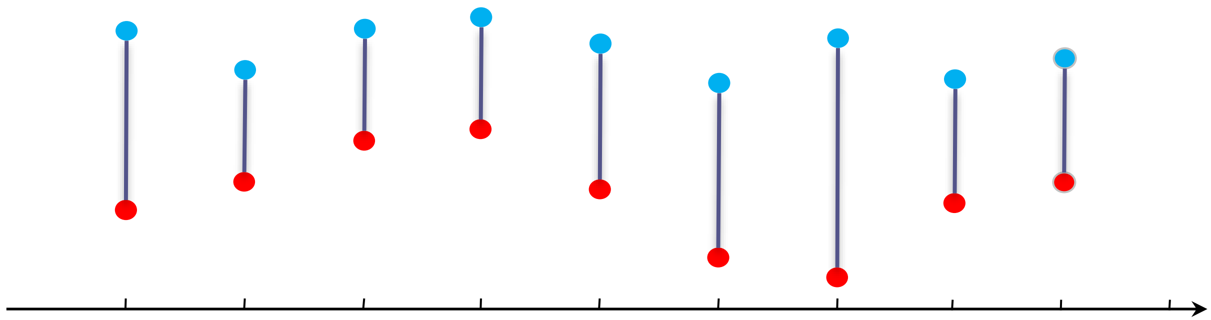
- similarity search is based on measuring distance between vectors
- A variety of distance measures have been proposed
  - L_p distances ($0<p\leq2, \infty$), (Euclidean for p = 2)
  - Cosine distance
  - Correlation
  - Hamming distance
  - …

Problem Variations

High-d Vectors Distance Measures

- similarity search is based on measuring distance between vectors
- A variety of distance measures have been proposed
  - L_p distances ($0<p\leq 2, \infty$), (Euclidean for $p = 2$)
  - Cosine distance
  - Correlation
  - Hamming distance
  - …

Euclidean Distance



- Euclidean distance
  - pair-wise point distance

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

Correlation

- measures the degree of relationship between data series
  - indicates the degree and direction of relationship
- direction of change
  - positive correlation
    - values of two data series change in same direction
  - negative correlation
    - values of two data series change in opposite directions
- linear correlation
  - amount of change in one data series bears constant ratio of change in the other data series

- useful in several applications

Pearson's Correlation (PC) Coefficient

- used to see linear dependency between values of data series of equal length, n

$$PC = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s_x}\right)\left(\frac{y_i - \bar{y}}{s_y}\right)$$

- where $\bar{x}$ is the mean: $\bar{x} = \frac{1}{n-1}\sum_{i=1}^{n} x_i$
- and s_x is the standard deviation: $s_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$

Pearson's Correlation (PC) Coefficient

- used to see linear dependency between values of data series of equal length, n

$$PC = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s_x}\right)\left(\frac{y_i - \bar{y}}{s_y}\right)$$

- takes values in [-1,1]
  - 0 – no correlation
  - -1, 1 – inverse/direct correlation
- there is a statistical test connected to PC, where null hypothesis is the no correlation case (correlation coefficient = 0)
  - test is used to ensure that the correlation similarity is not caused by a random process

PC and ED

- Euclidean distance: $$ED = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2},$$
- In case of Z-normalized data series (mean = 0, stddev = 1):

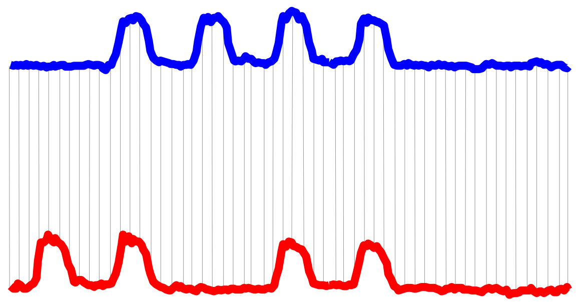
$$PC = \frac{1}{n-1}\sum_{i=1}^{n} x_i \cdot y_i$$ and $$ED^2 = 2n(n-1) - 2\sum_{i=1}^{n} x_i y_i$$

so the following formula is true: $$ED^2 = 2(n-1)(n - PC)$$

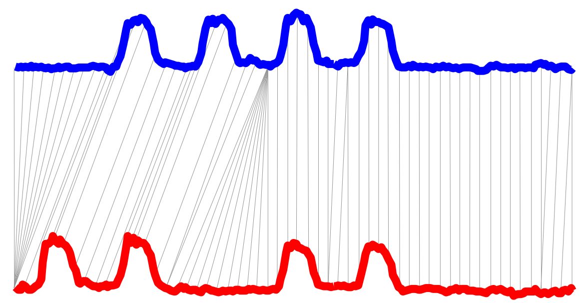
- direct connection between ED and PC for Z-normalized data series
  - if ED is calculated for normalized data series, it can be directly used to calculate the p-value for statistical test of Pearson's correlation instead of actual PC value.

Distance Measures: LCSS against Euclidean, DTW

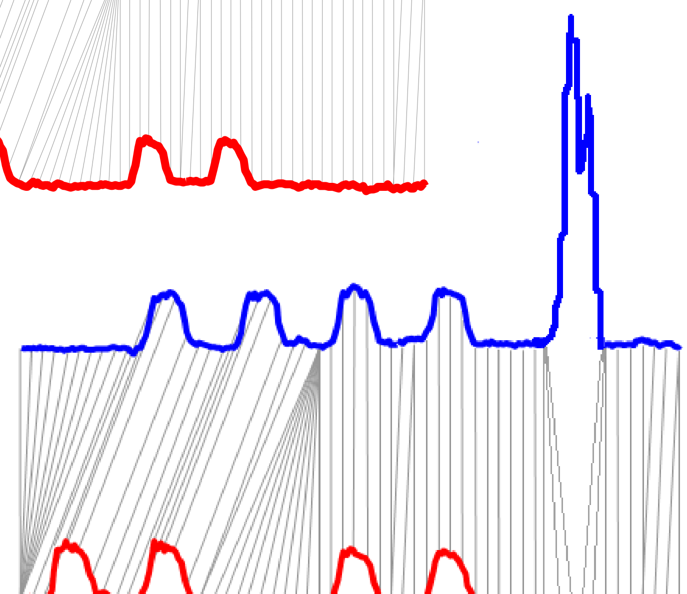
- Euclidean
  - rigid



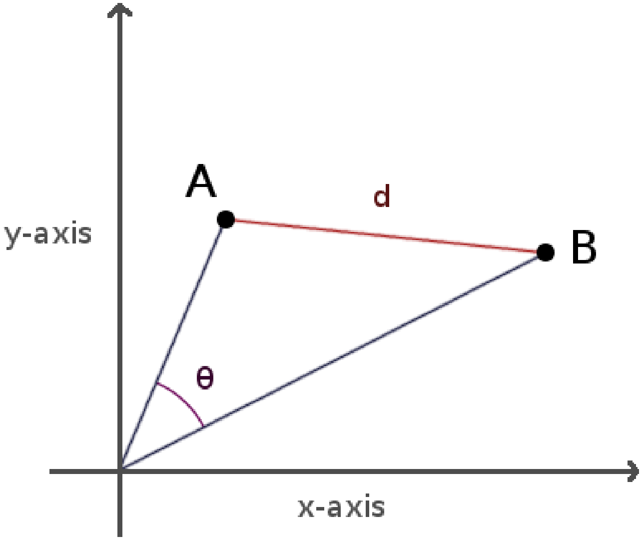
- Dynamic Time Warping (DTW)
  - allows local scaling



- Longest Common SubSequence (LCSS)
  - allows local scaling
  - ignores outliers



Distance Measures: Cosine Distance



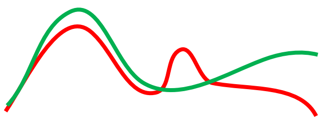
$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

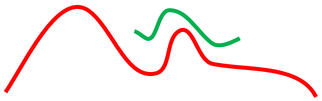
- Cosine distance = 1 - cosine similarity

- ED vs. Cosine similarity
  - If A and B are normalized to unit length in L_2, the square of ED is proportional to the cosine distance:
  - $||A||_2=||B||_2=1 \rightarrow ||A-B||_2=2-2\cos(A,B)$

Problem Variations

Queries





Whole matching

Entire query

Entire candidate

Subsequence matching

Entire query

A subsequence of a candidate

Problem Variations

Queries

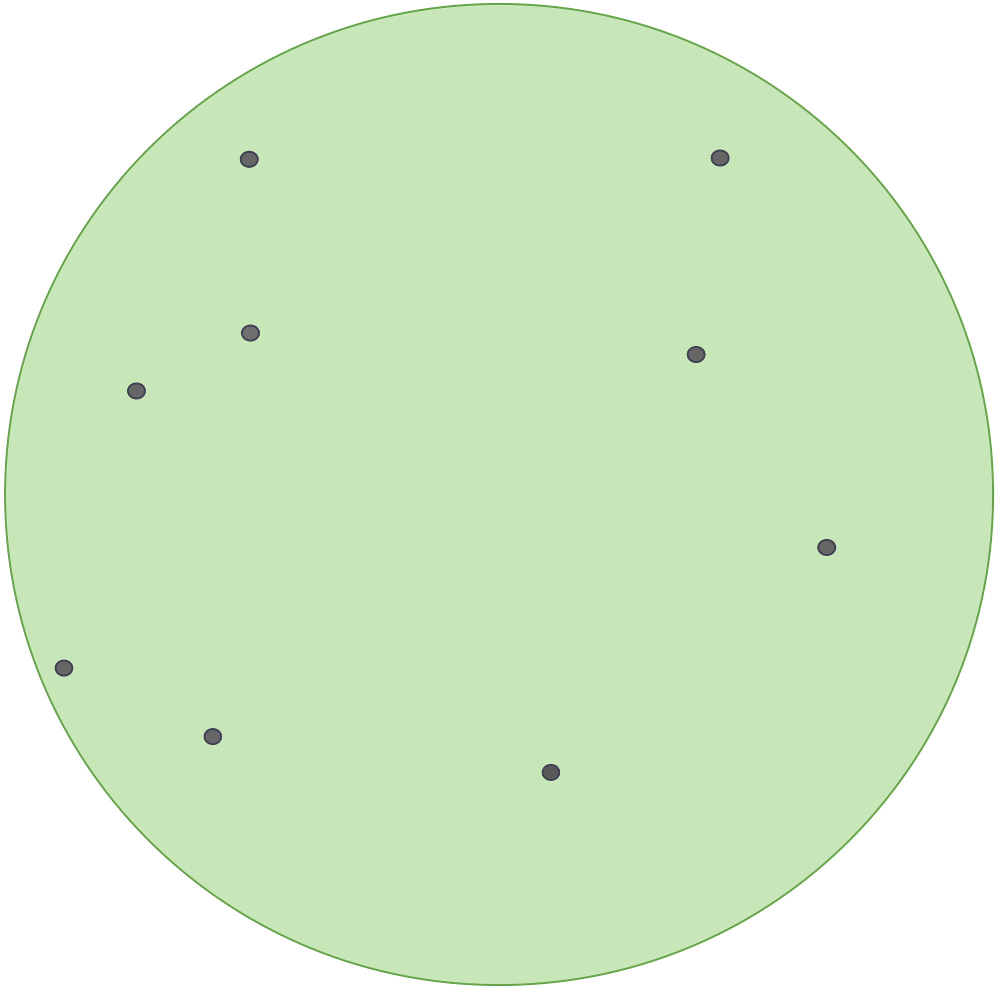
Nearest Neighbor (1NN)
k-Nearest Neighbor (kNN)
Farthest Neighbor
epsilon-Range

and more...

Nearest Neighbor (NN) Queries…

Publications

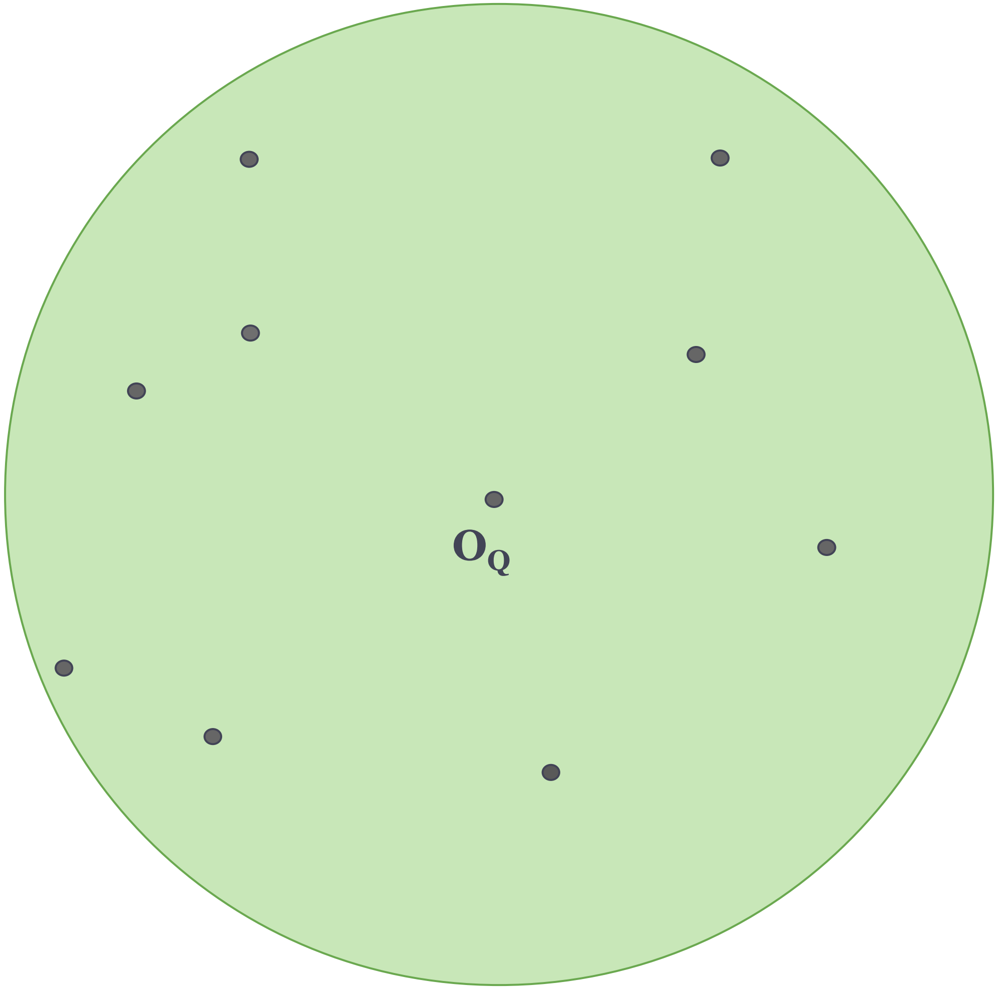
Echihabi et al. PVLDB'19



Nearest Neighbor (NN) Queries…

Publications

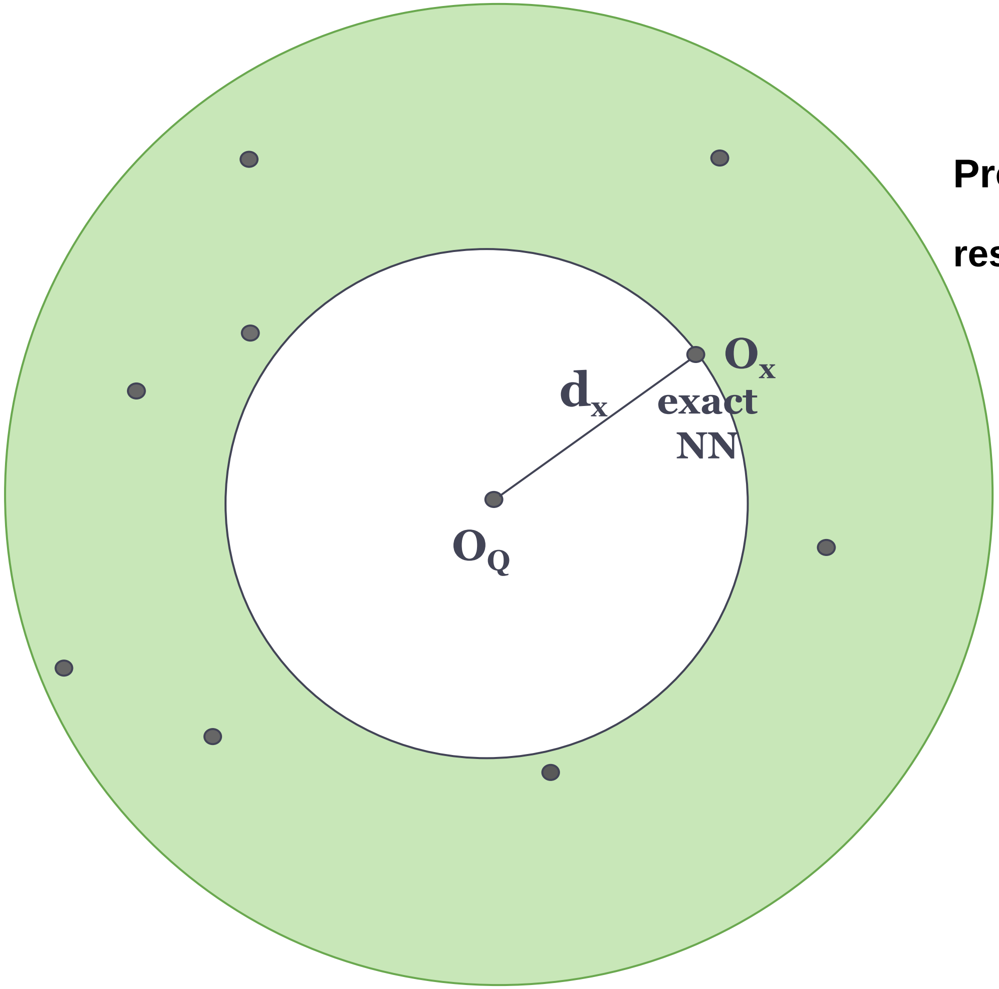
Echihabi et al. PVLDB'19



Nearest Neighbor (NN) Queries…

Publications

Echihabi et al. PVLDB'19



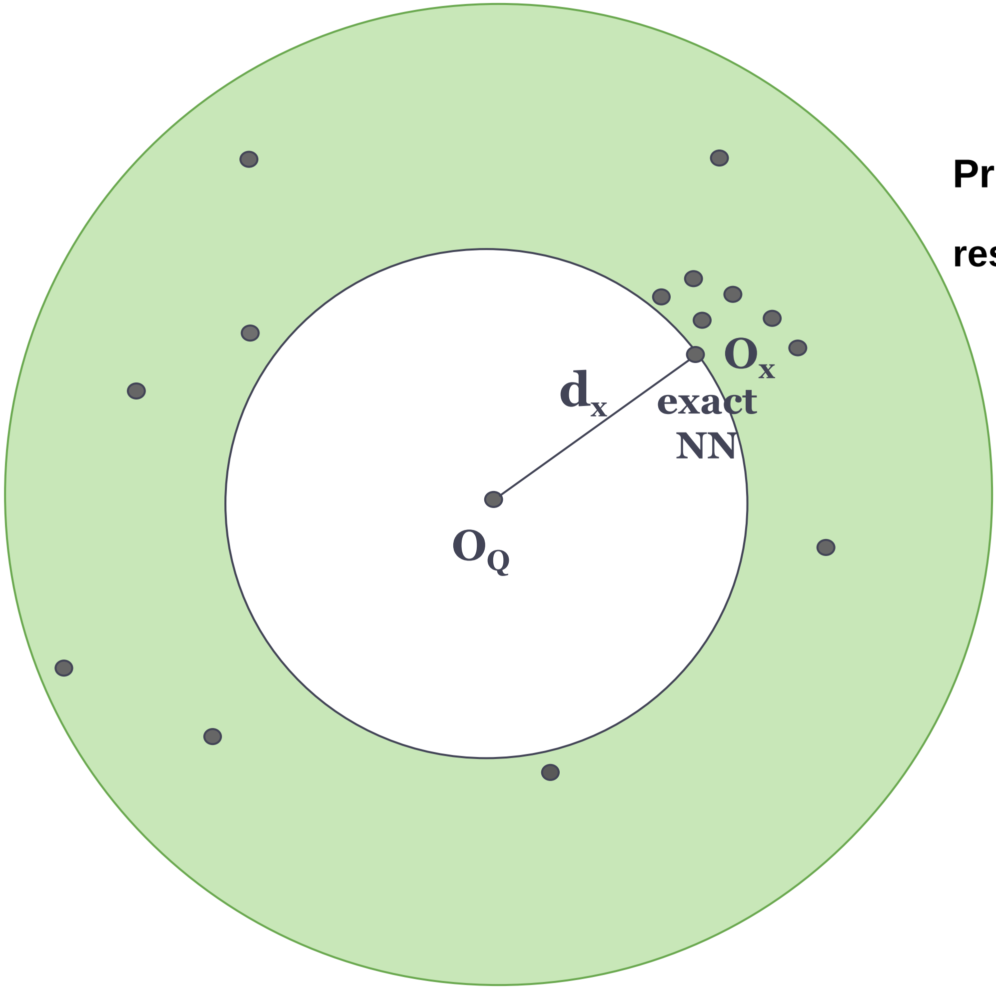
Prob($d_x = \min\{d_i\}$) = 1

result is exact NN

Nearest Neighbor (NN) Queries…

Publications

Echihabi et al. PVLDB'19



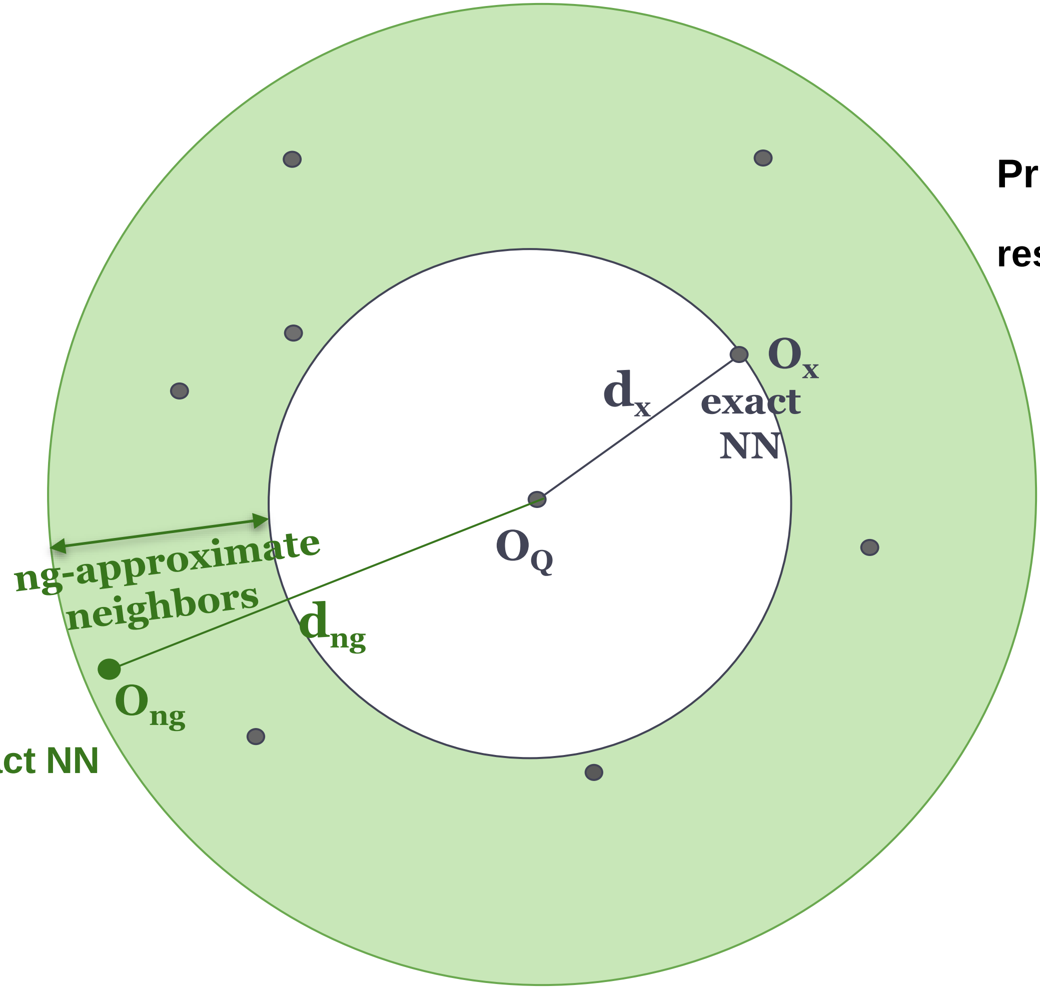
Prob(d_x = min{d_i}) = 1

result is exact NN

Nearest Neighbor (NN) Queries…

Publications

Echihabi et al. PVLDB'19



Prob($d_x = \min\{d_i\}$) = 1

result is exact NN

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

result within ? of exact NN

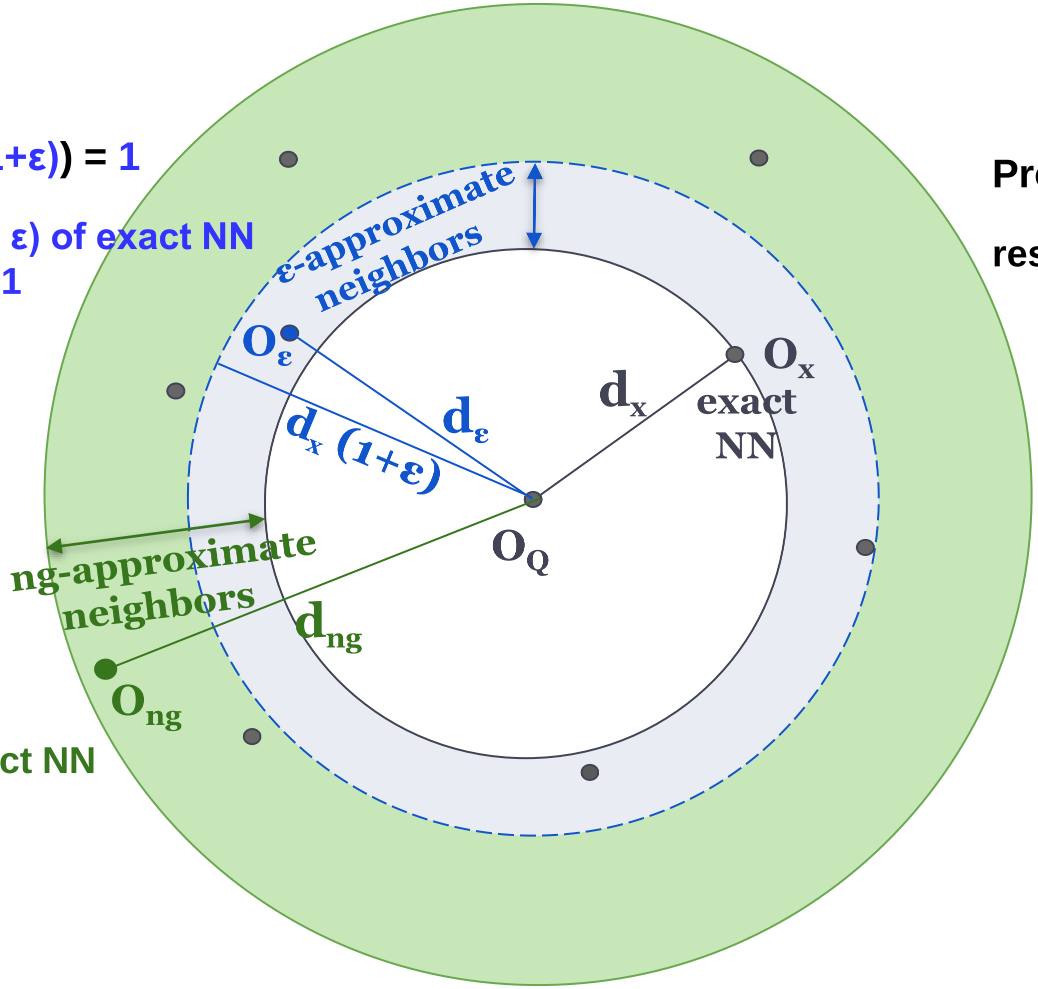
Nearest Neighbor (NN) Queries…

Publications

Echihabi et al. 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

Nearest Neighbor (NN) Queries…

Publications

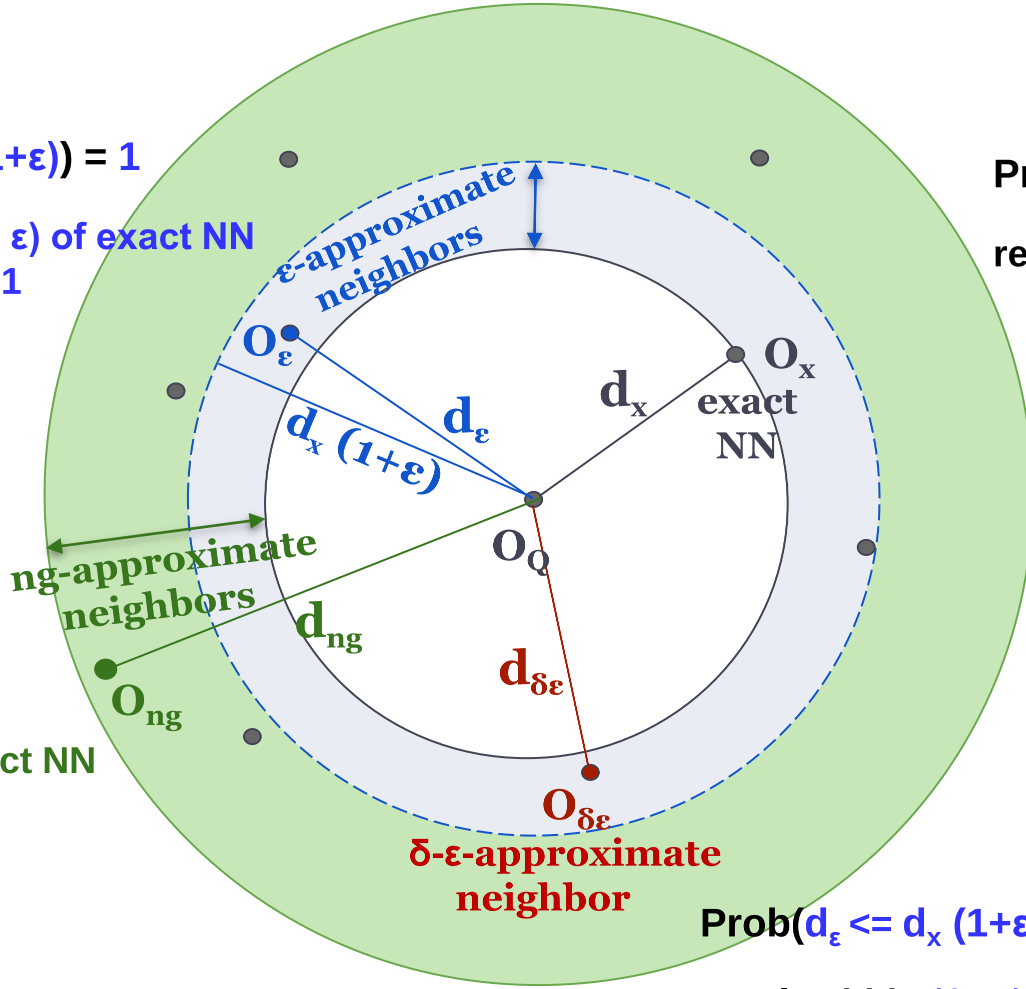
Echihabi et al. PVLDB'19

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

Nearest Neighbor (NN) Queries…

Publications

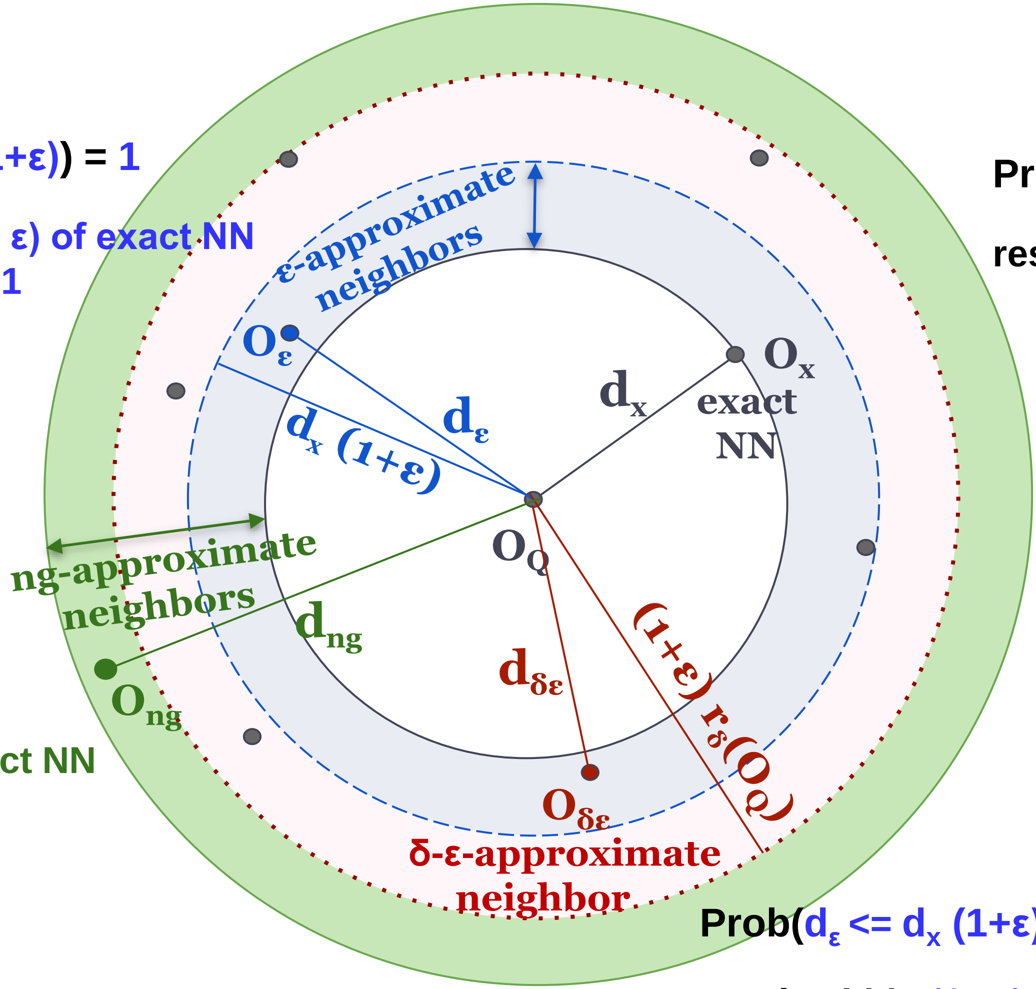
Echihabi et al. PVLDB'19

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

Maximum Inner Product Search (MIPS)

- Problem Definition:
  - Given a collection of candidate vectors S and a query Q , find a candidate vector C maximizing the inner product with the query: :
    - Given $S \subset R^d$ and $Q \in R^d$, $C = \mathrm{argmax}_{X \in S}\, Q^T X$

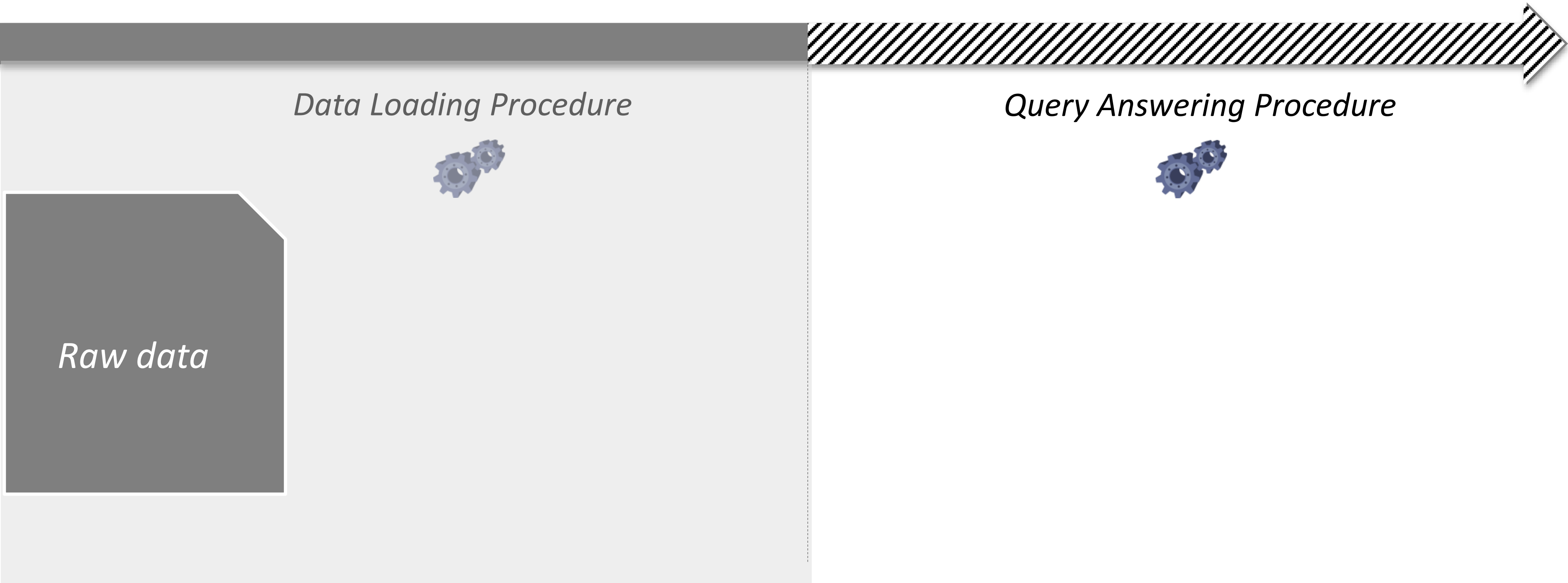
Maximum Inner Product Search (MIPS)

- **Problem Definition:**
  - Given a collection of candidate vectors S and a query Q , find a candidate vector C maximizing the inner product with the query: :
    - Given $S \subset R^d$ and $Q \in R^d$, $C = \text{argmax}_{X \in S}\, Q^T X$
  - MIPS is closely related to NN search:
    - If $\|Q\|_2 = 1$, $\|Q - X\|_2 = 1 + \|X\|_2 - 2Q^T X$
  - MIPS and NN search are equivalent when all vectors X in S have constant length c
  - Otherwise, MIPS can be converted to NN search with ED or Cosine similarity [1][2][3]

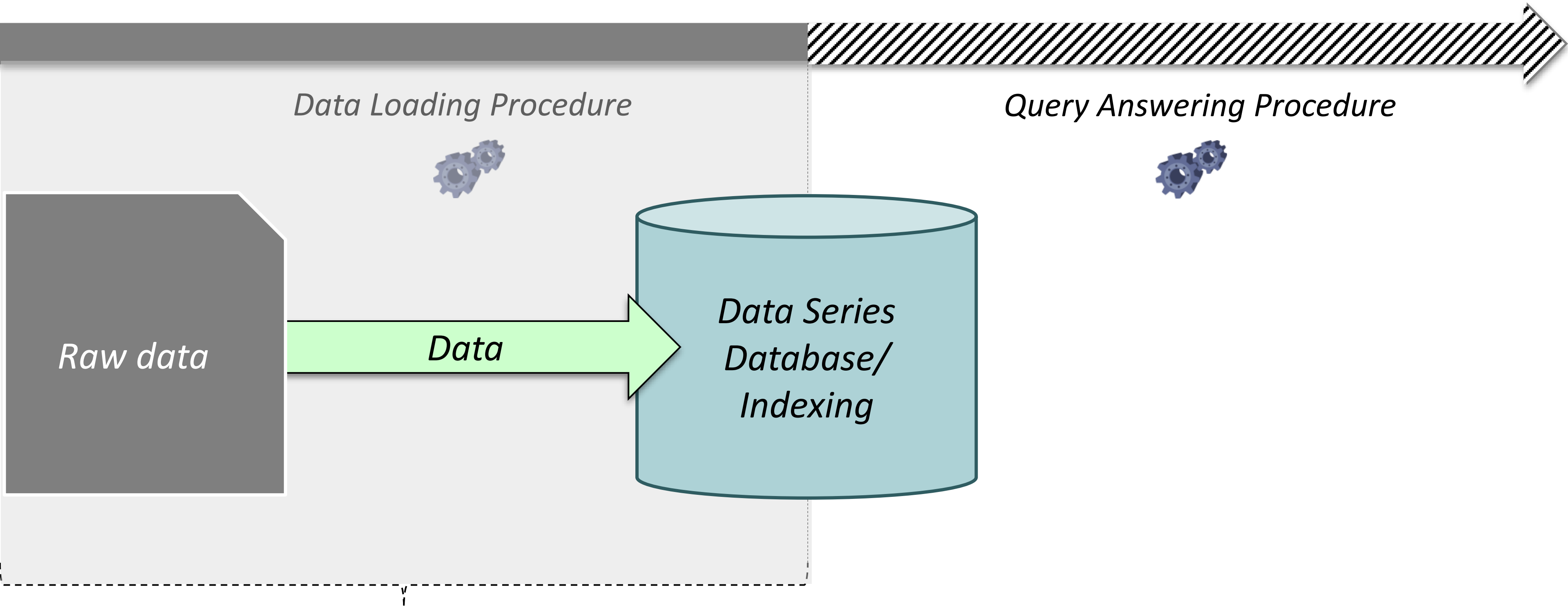
[1] Anshumali Shrivastava and Ping Li. 2014a. Asymmetric LSH (ALSH) for Sublinear Time Maximum Inner Product Search (MIPS). In NIPS. 2321–2329.
[2] Yoram Bachrach, Yehuda Finkelstein, Ran Gilad-Bachrach, Liran Katzir, Noam Koenigstein, Nir Nice, and Ulrich Paquet. 2014. Speeding Up the Xbox Recommender System Using a Euclidean Transformation for Inner-product Spaces. In RecSys. 257–264.
[3] B. Neyshabur and N. Srebro. 2014. On Symmetric and Asymmetric LSHs for Inner Product Search. ArXiv e-prints (Oct. 2014).

High-d Similarity Search Process

Similarity Search Process

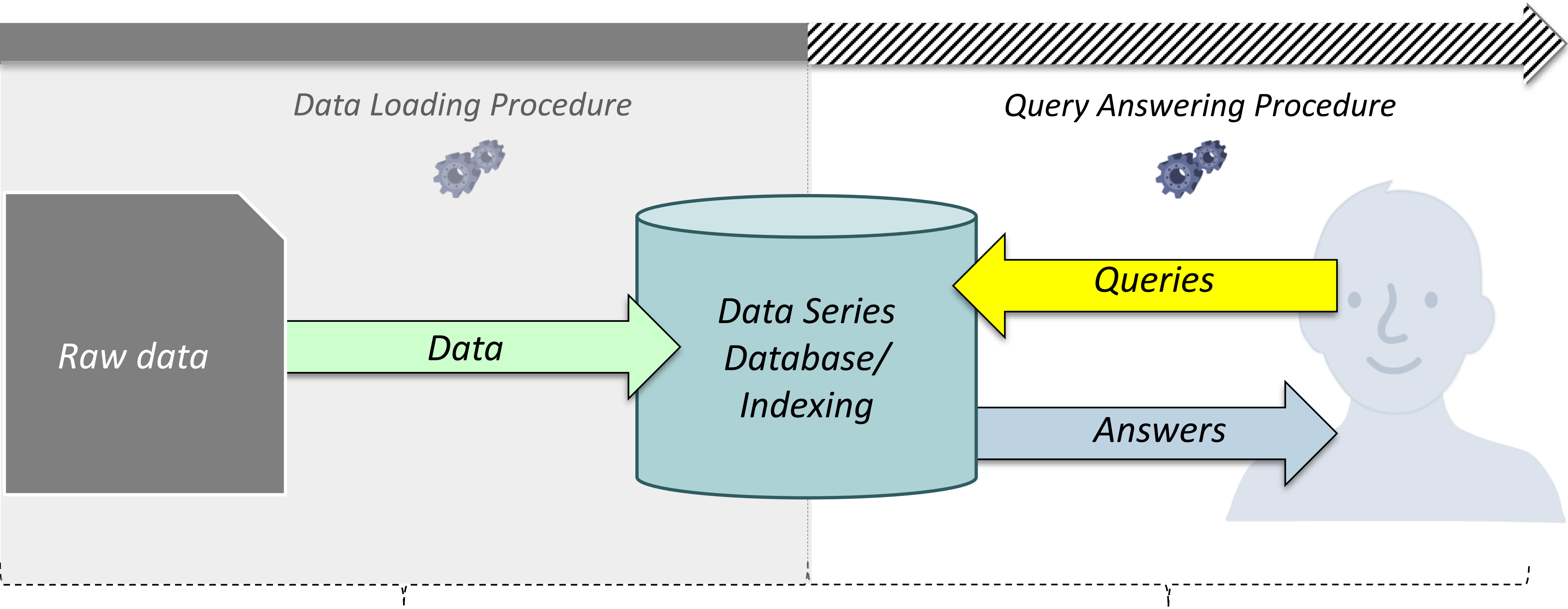


Similarity Search Process



data-to-query time

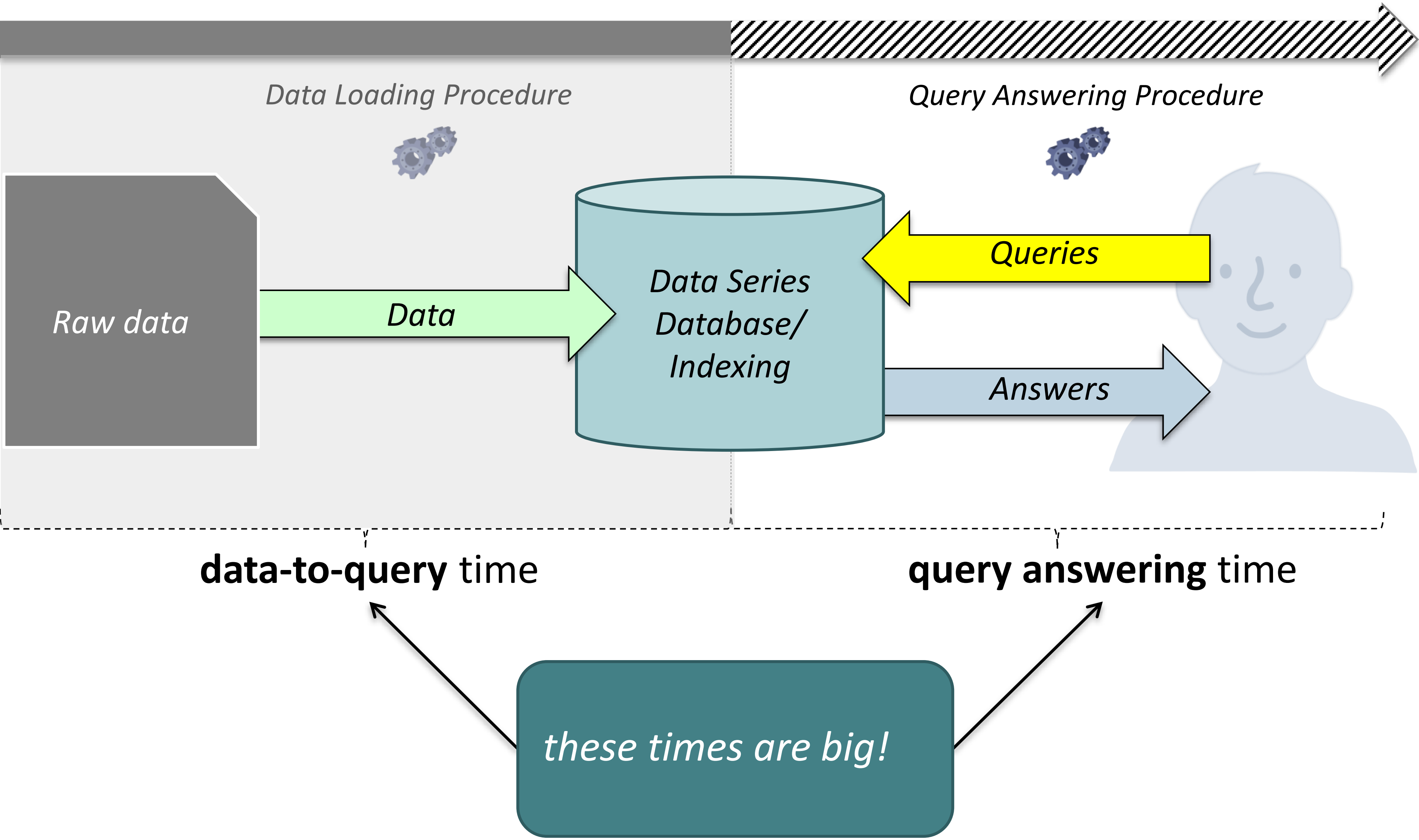
Similarity Search Process



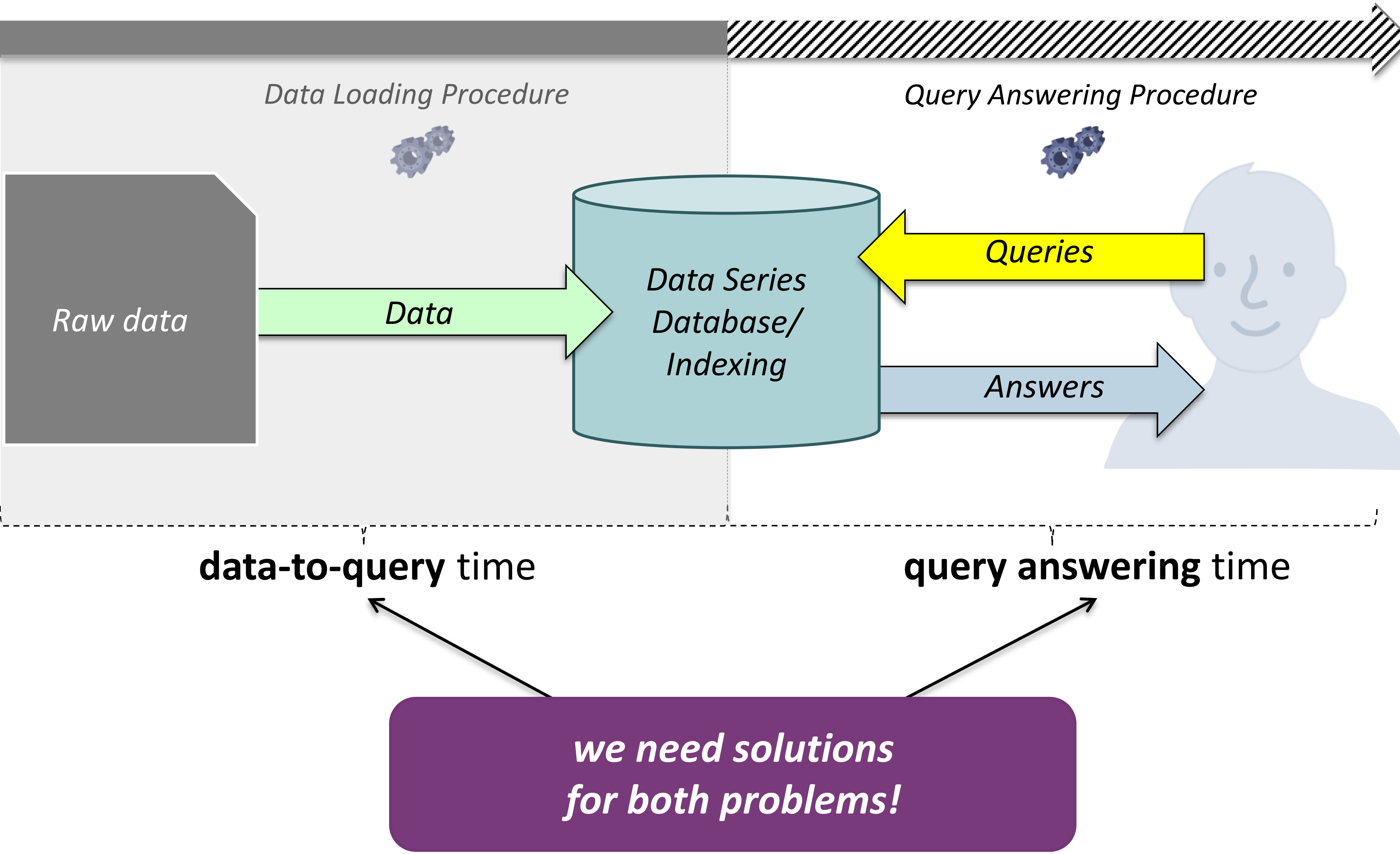
data-to-query time

query answering time

Similarity Search Process

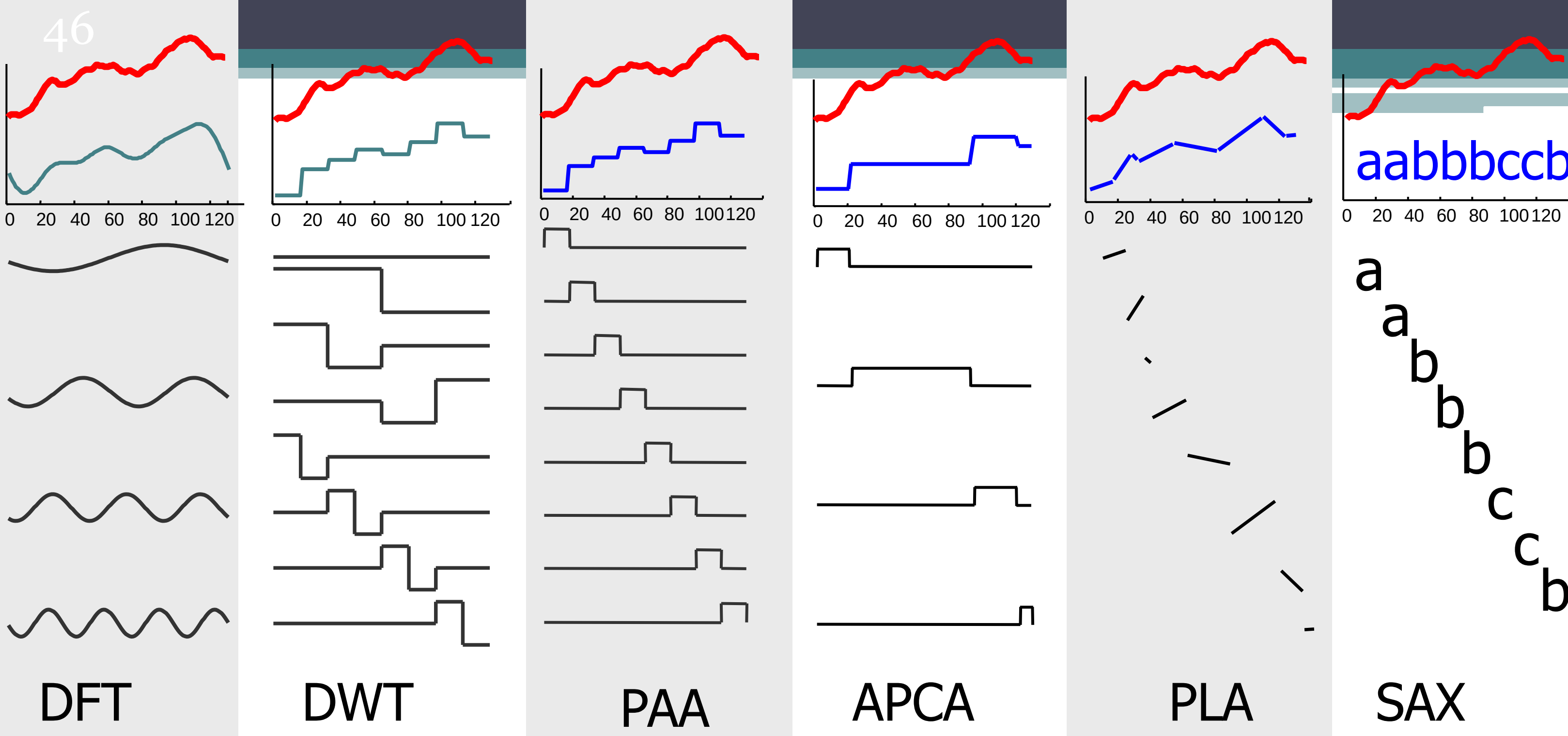


Similarity Search Process



Questions?

Data Series Similarity Search



DFT: Agrawal, Faloutsos, &. Swami. FODO 1993
Faloutsos, Ranganathan, & Manolopoulos. SIGMOD 1994

DWT: Chan & Fu. ICDE 1999

PAA: Keogh, Chakrabarti, Pazzani & Mehrotra KAIS 2000
Yi & Faloutsos VLDB 2000

APCA: Keogh, Chakrabarti, Pazzani & Mehrotra SIGMOD 2001

PLA: Morinaka, Yoshikawa, Amagasa, & Uemura, PAKDD 2001

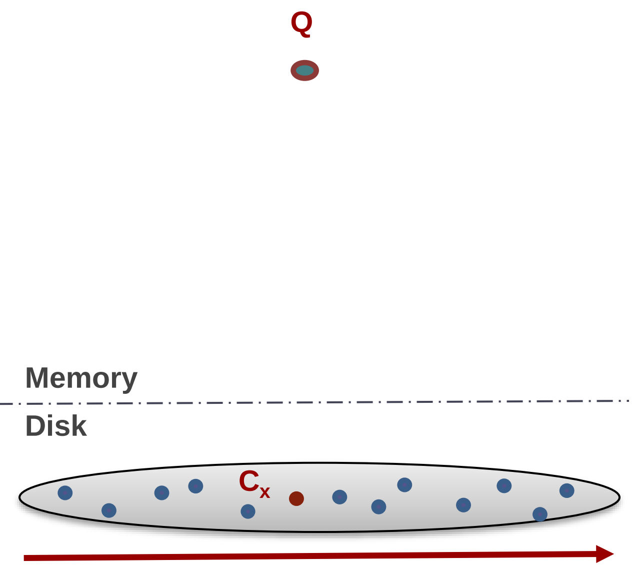
for a complete and detailed presentation, see tutorial:

Publications

Keogh - KDD‘04

Data Series Similarity Search
Classes of Methods

Similarity Matching
Serial Scan

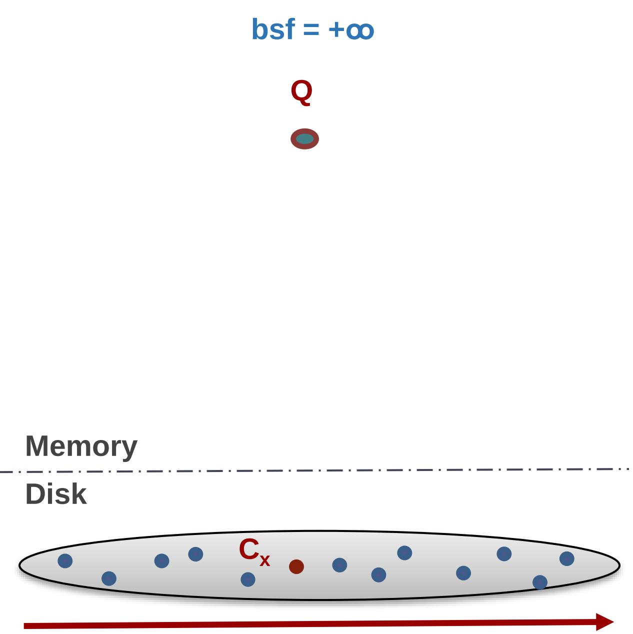


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching
Serial Scan

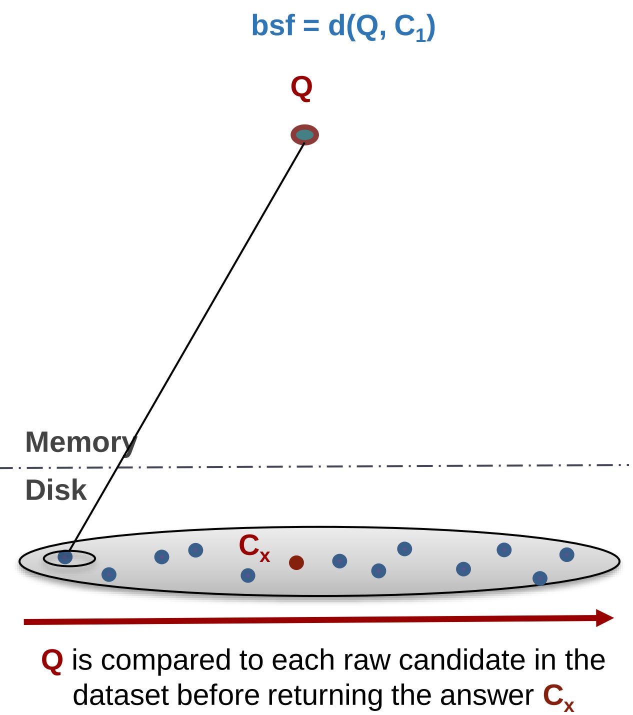


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

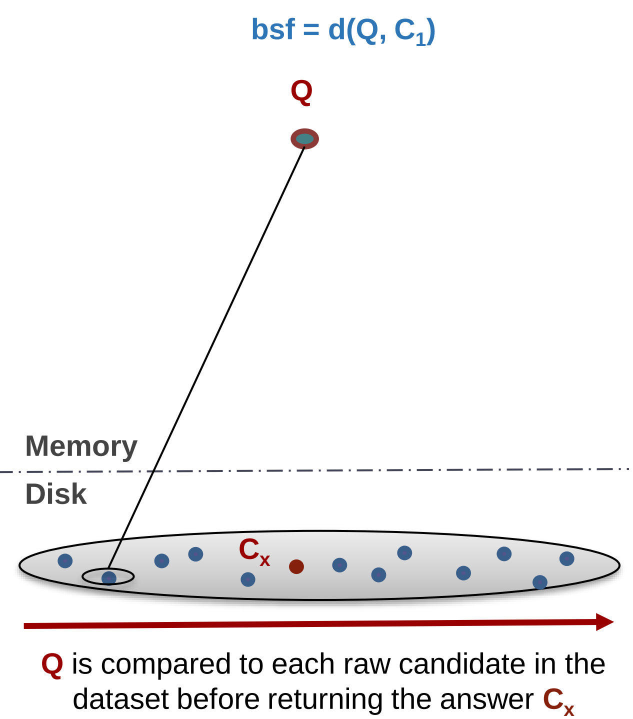
Similarity Matching
Serial Scan



(a) Serial scan

Answering a similarity search query using different access paths

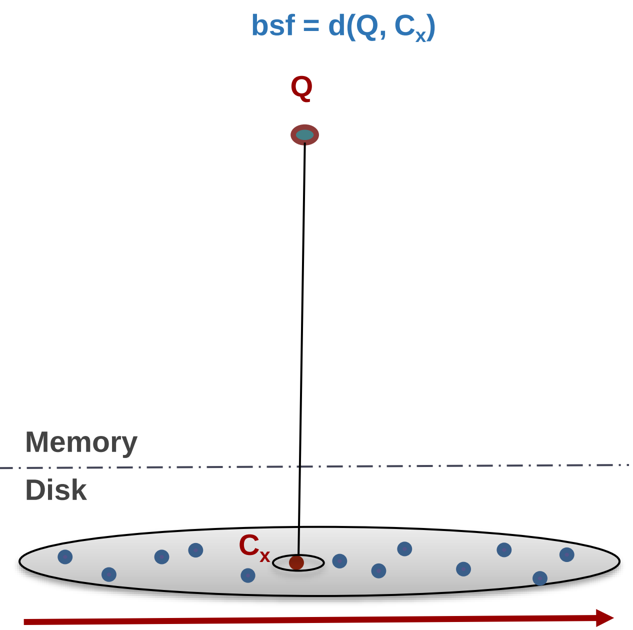
Similarity Matching
Serial Scan



(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching
Serial Scan

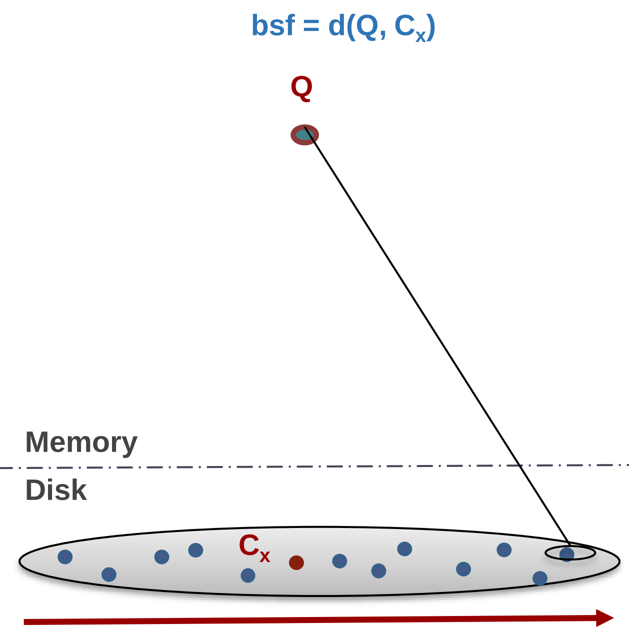


Q is compared to each raw candidate in the dataset before returning the answer **C_x**

(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching
Serial Scan

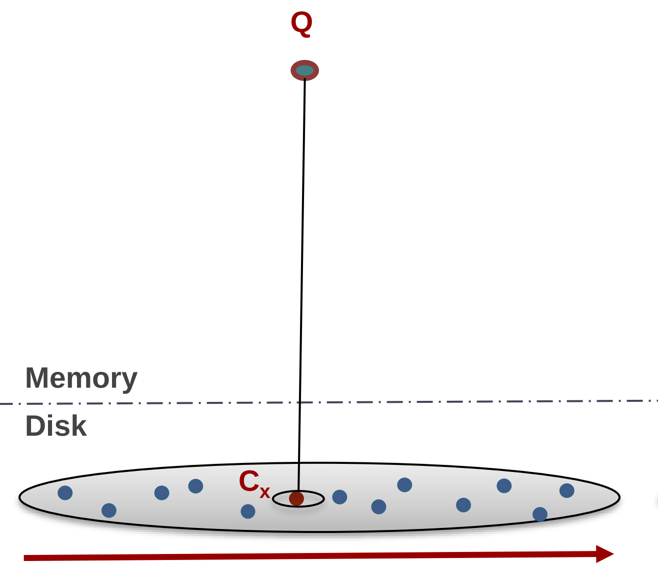


Q is compared to each raw candidate in the dataset before returning the answer **C_x**

(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching
Serial Scan

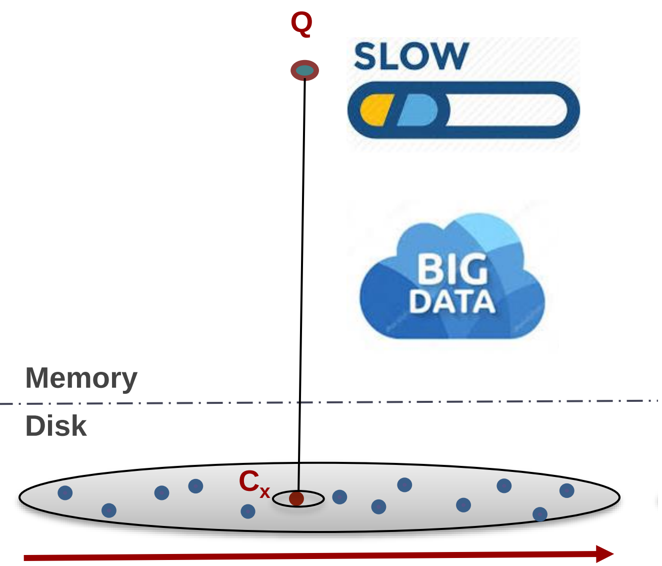


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching
Serial Scan

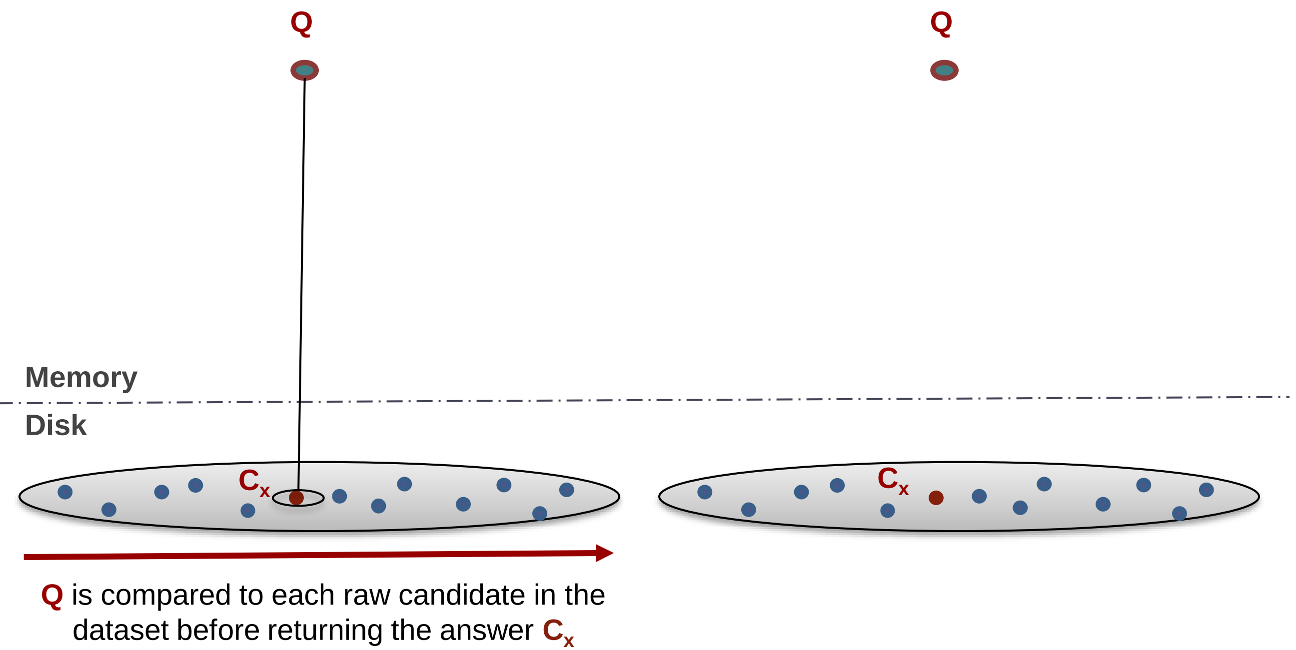


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

Indexes vs. Scans

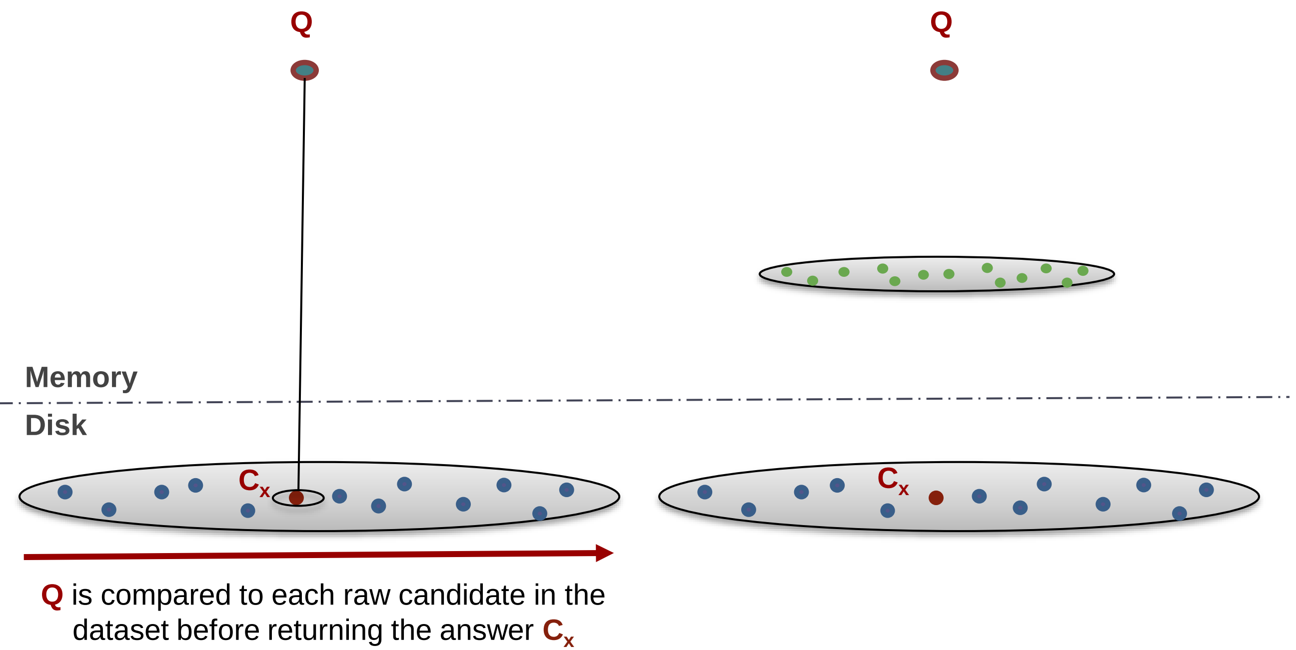


(a) Serial scan

(b) Skip-sequential scan

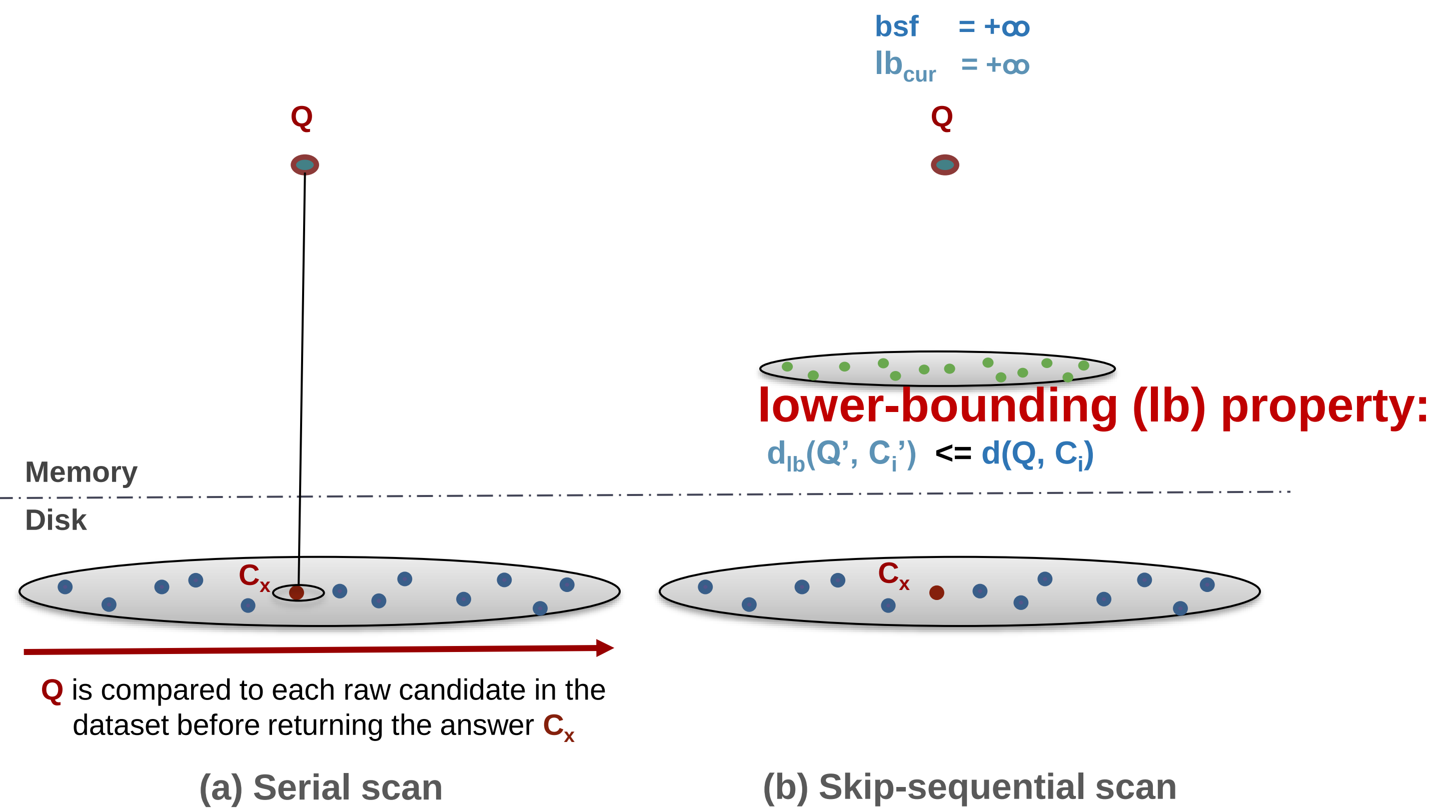
Answering a similarity search query using different access paths

Indexes vs. Scans



(a) Serial scan (b) Skip-sequential scan

Answering a similarity search query using different access paths

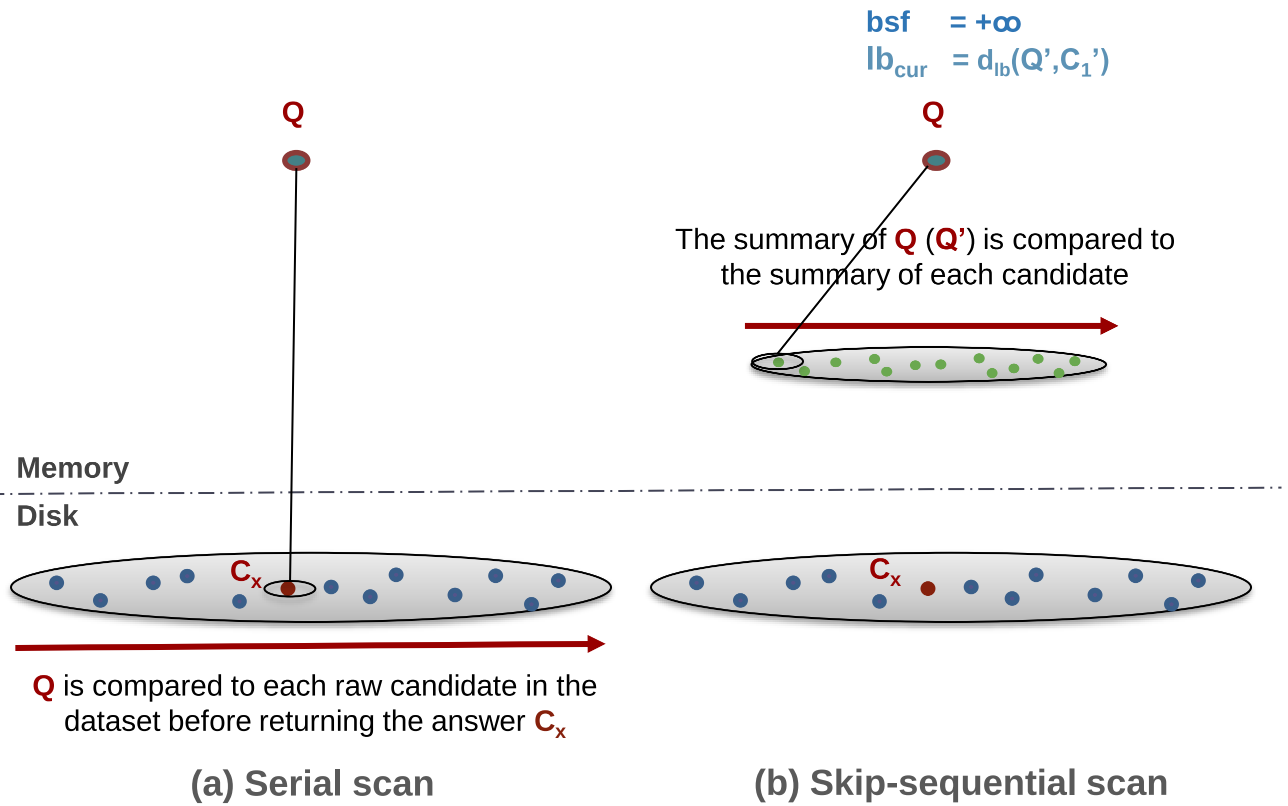


Publications

Faloutsos-SIGMOD'94

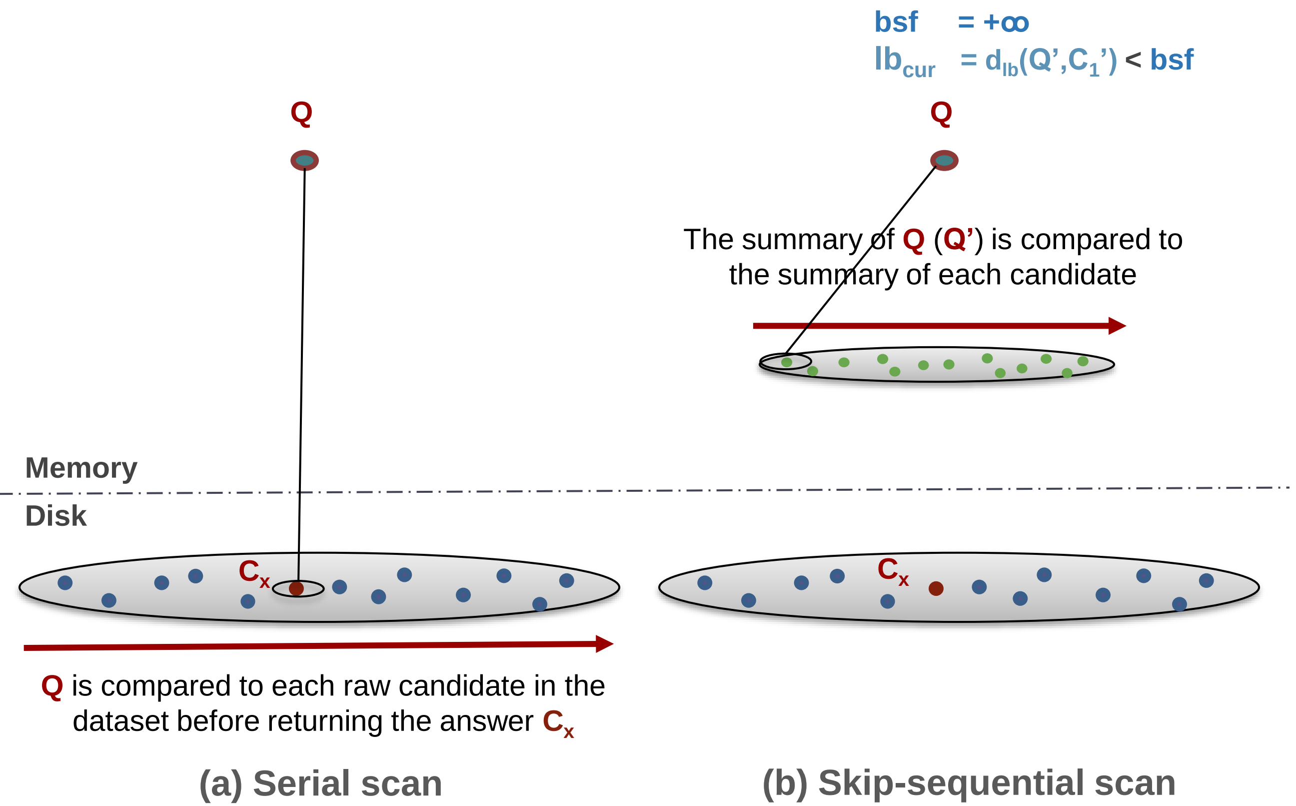
Answering a similarity search query using different access paths

Indexes vs. Scans



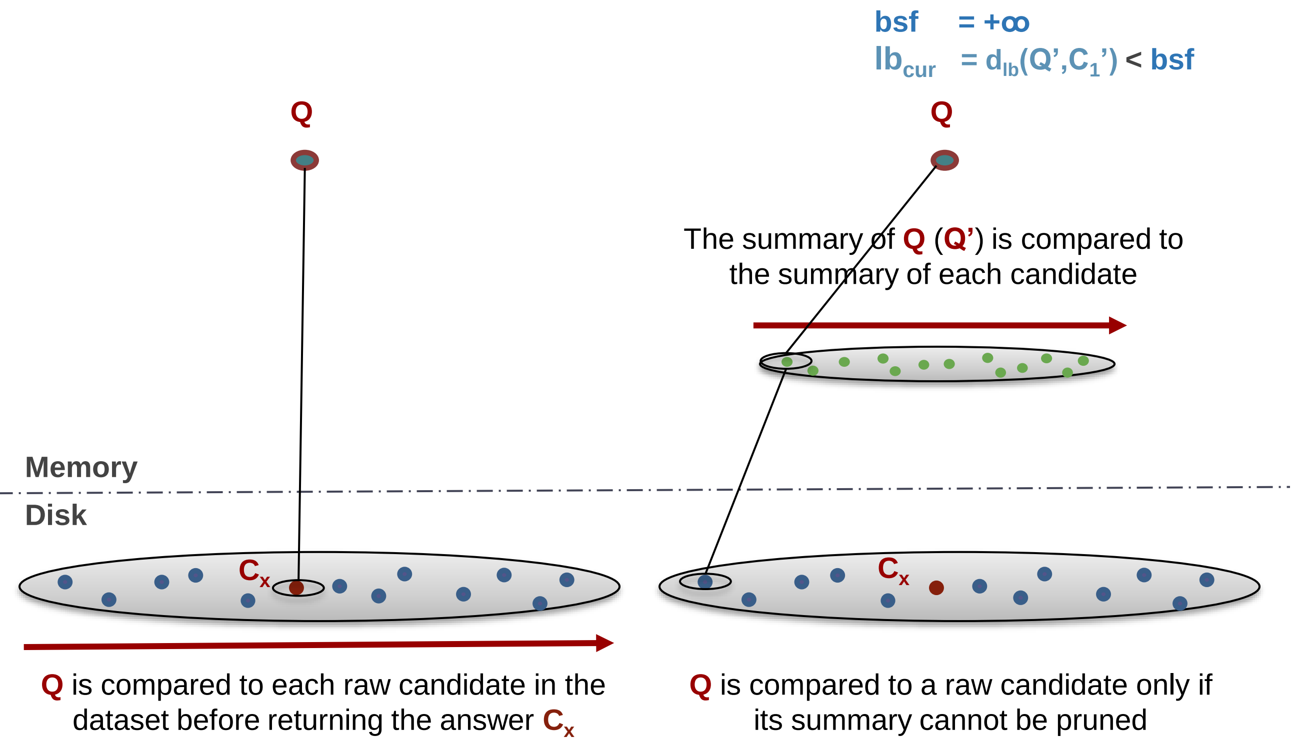
<u>**Answering a similarity search query using different access paths**</u>

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans

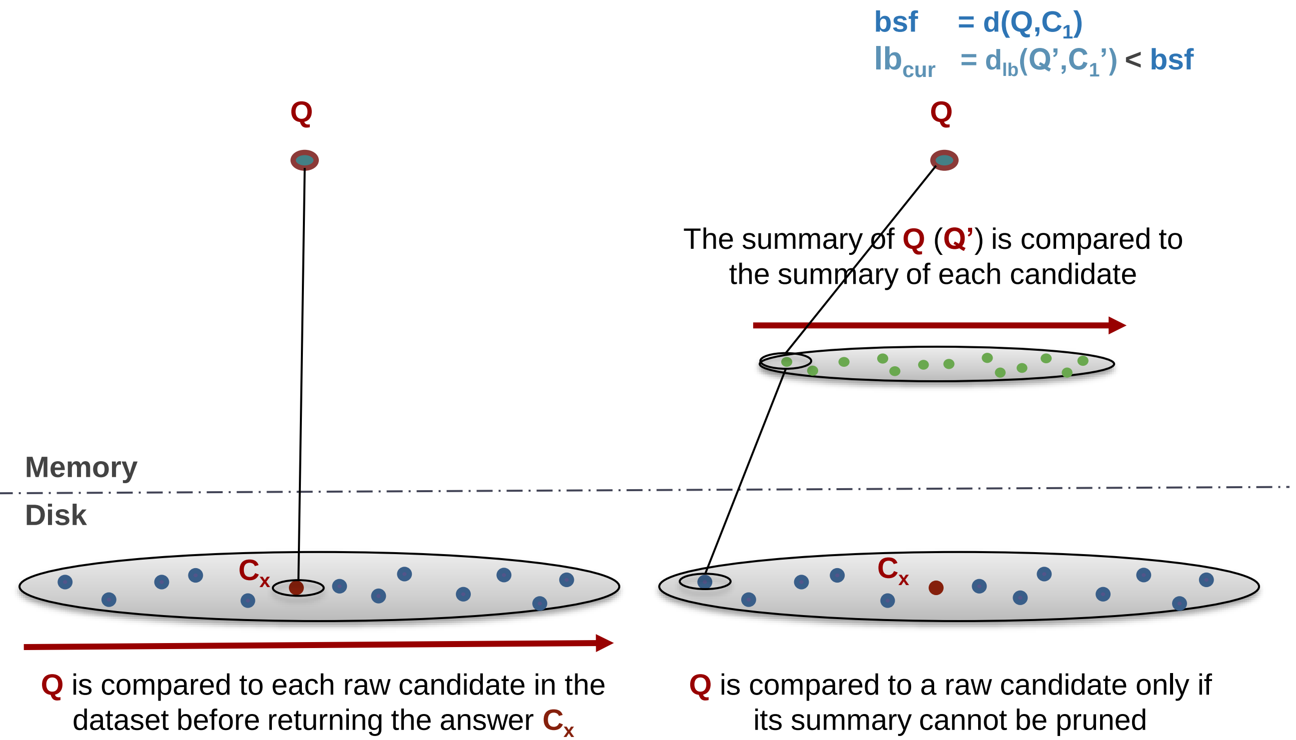


(a) Serial scan

(b) Skip-sequential scan

Answering a similarity search query using different access paths

Indexes vs. Scans

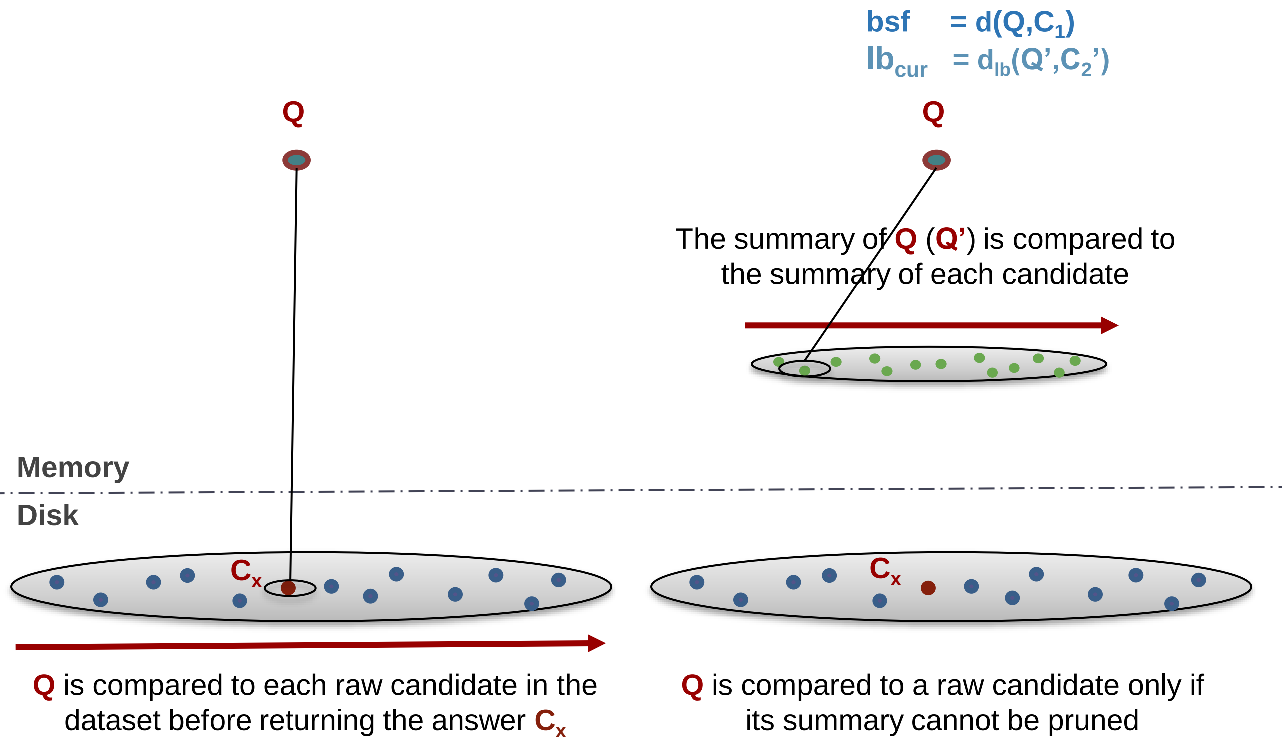


(a) Serial scan

(b) Skip-sequential scan

Answering a similarity search query using different access paths

Indexes vs. Scans

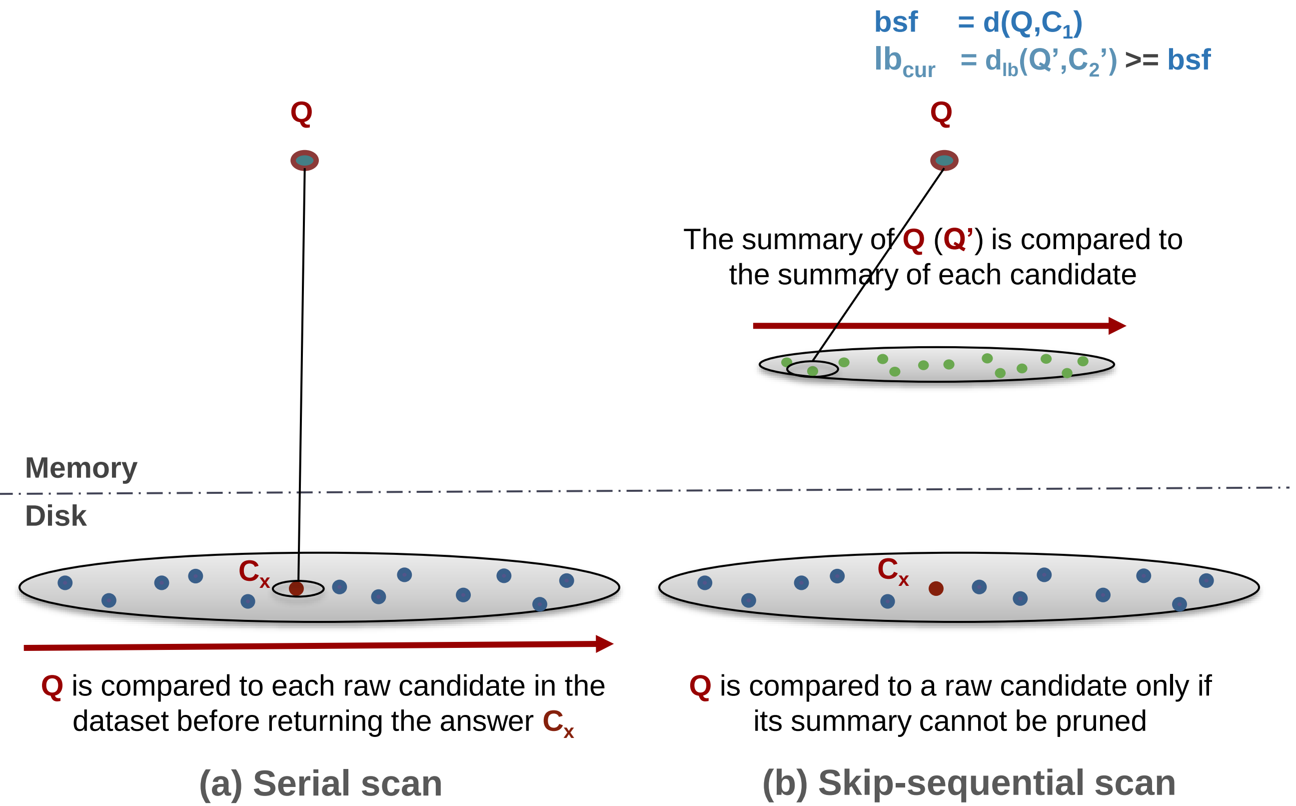


(a) Serial scan

(b) Skip-sequential scan

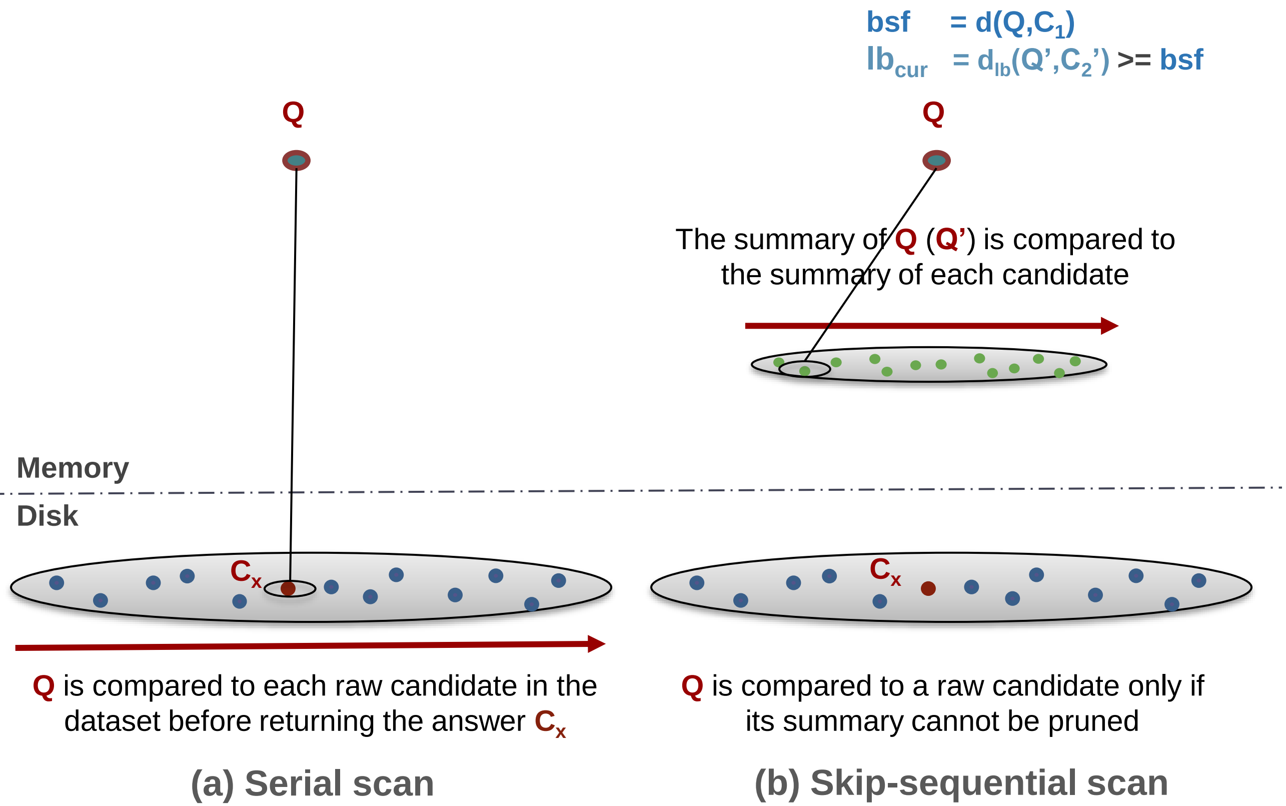
<u>Answering a similarity search query using different access paths</u>

Indexes vs. Scans



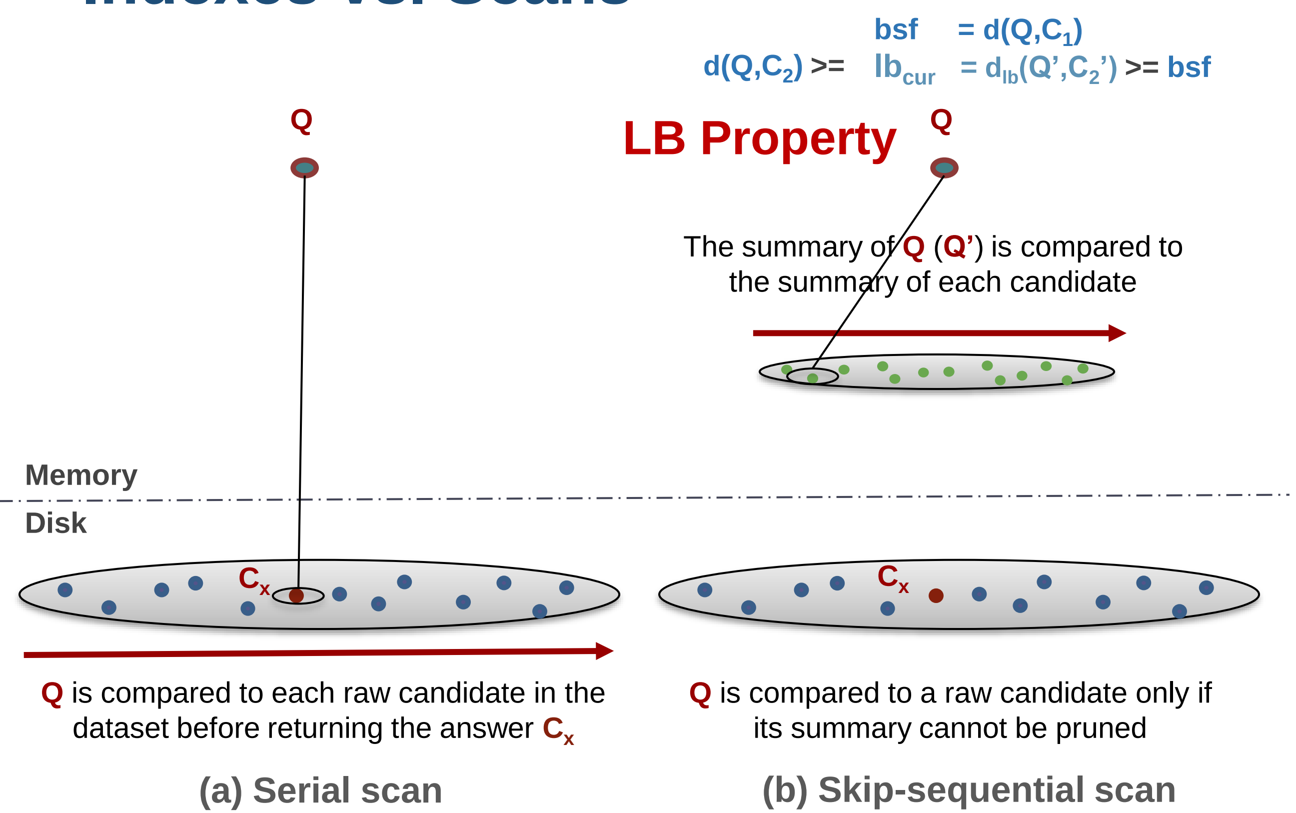
Answering a similarity search query using different access paths

Indexes vs. Scans



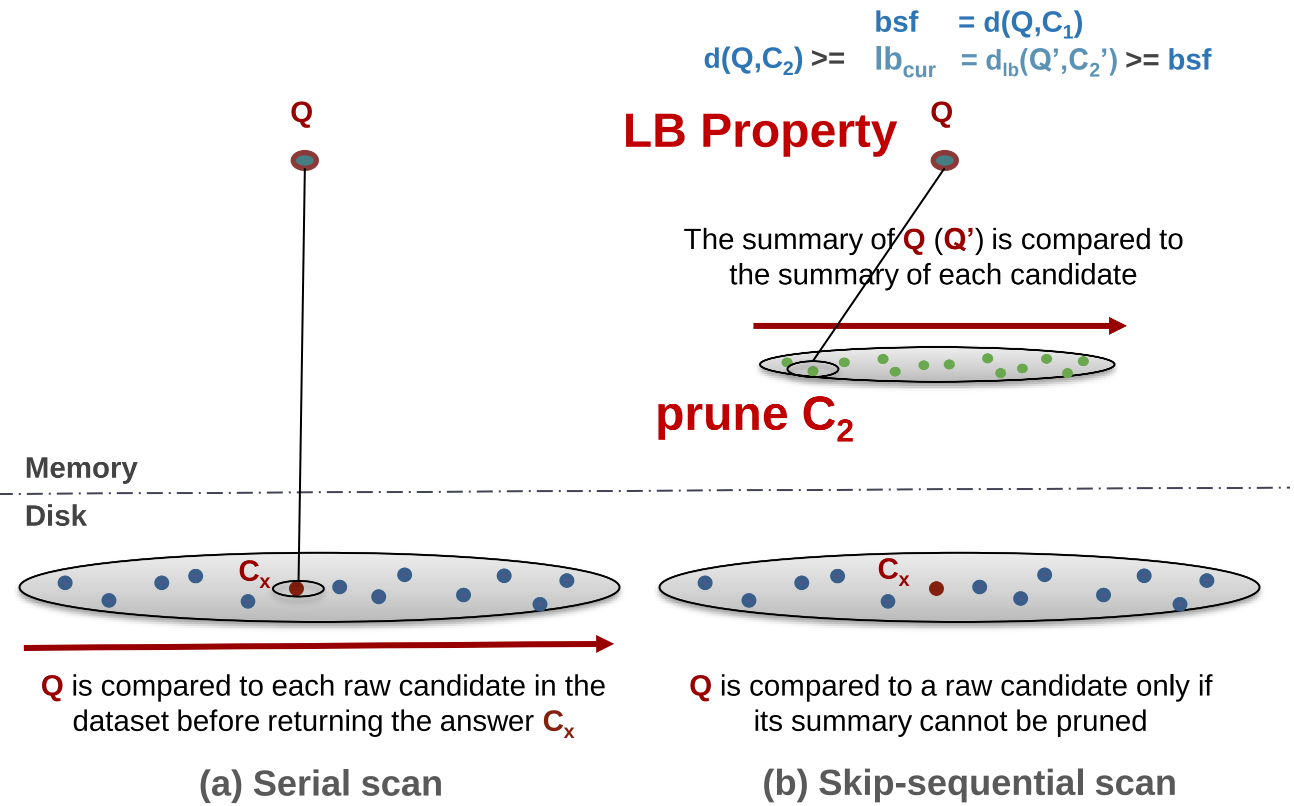
Answering a similarity search query using different access paths

Indexes vs. Scans



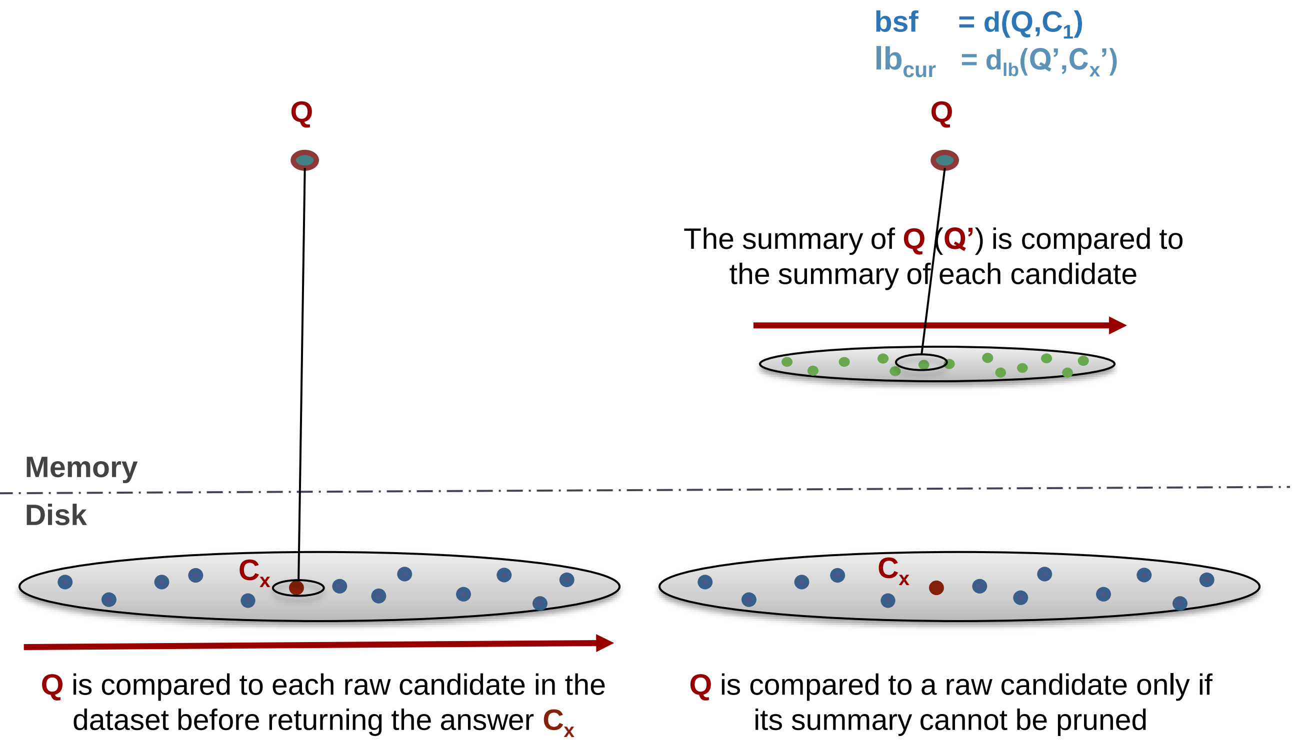
<u>**Answering a similarity search query using different access paths**</u>

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans

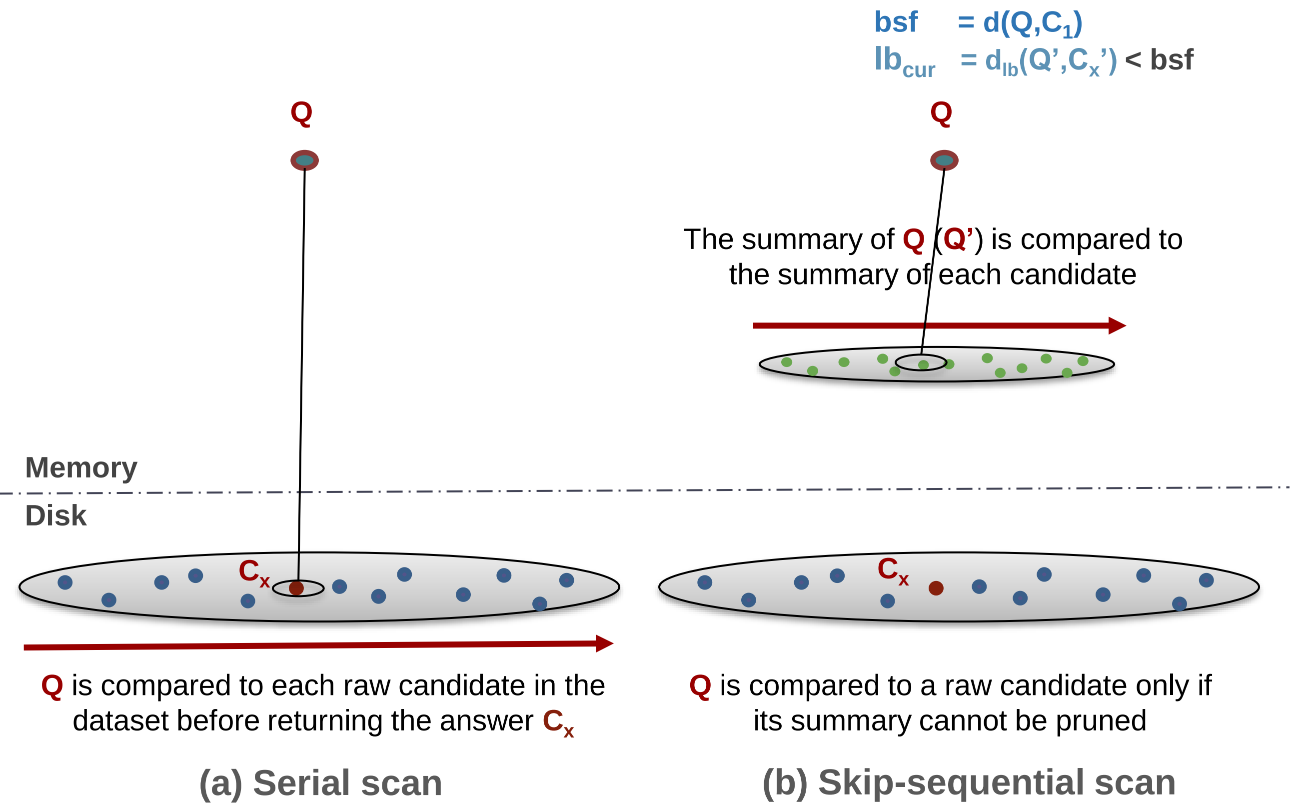


(a) Serial scan

(b) Skip-sequential scan

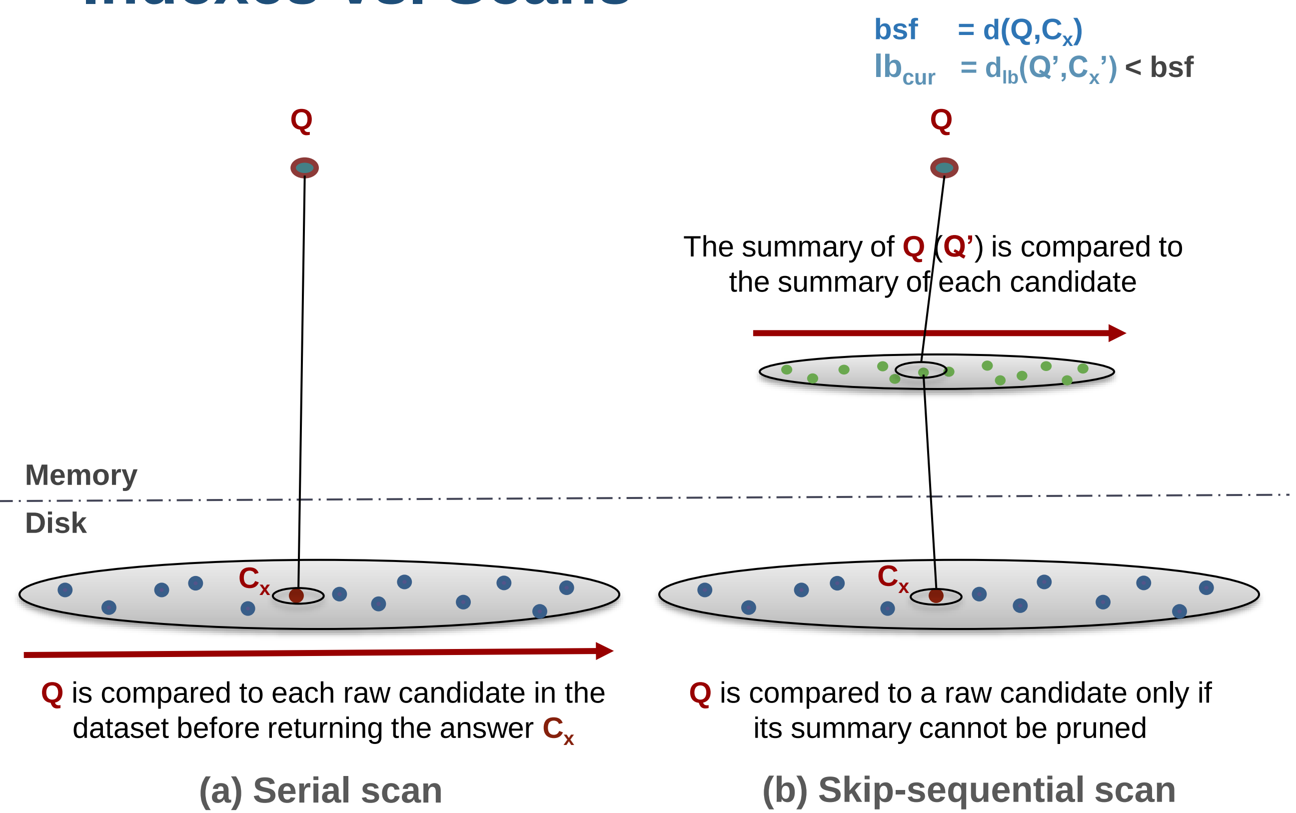
Answering a similarity search query using different access paths

Indexes vs. Scans



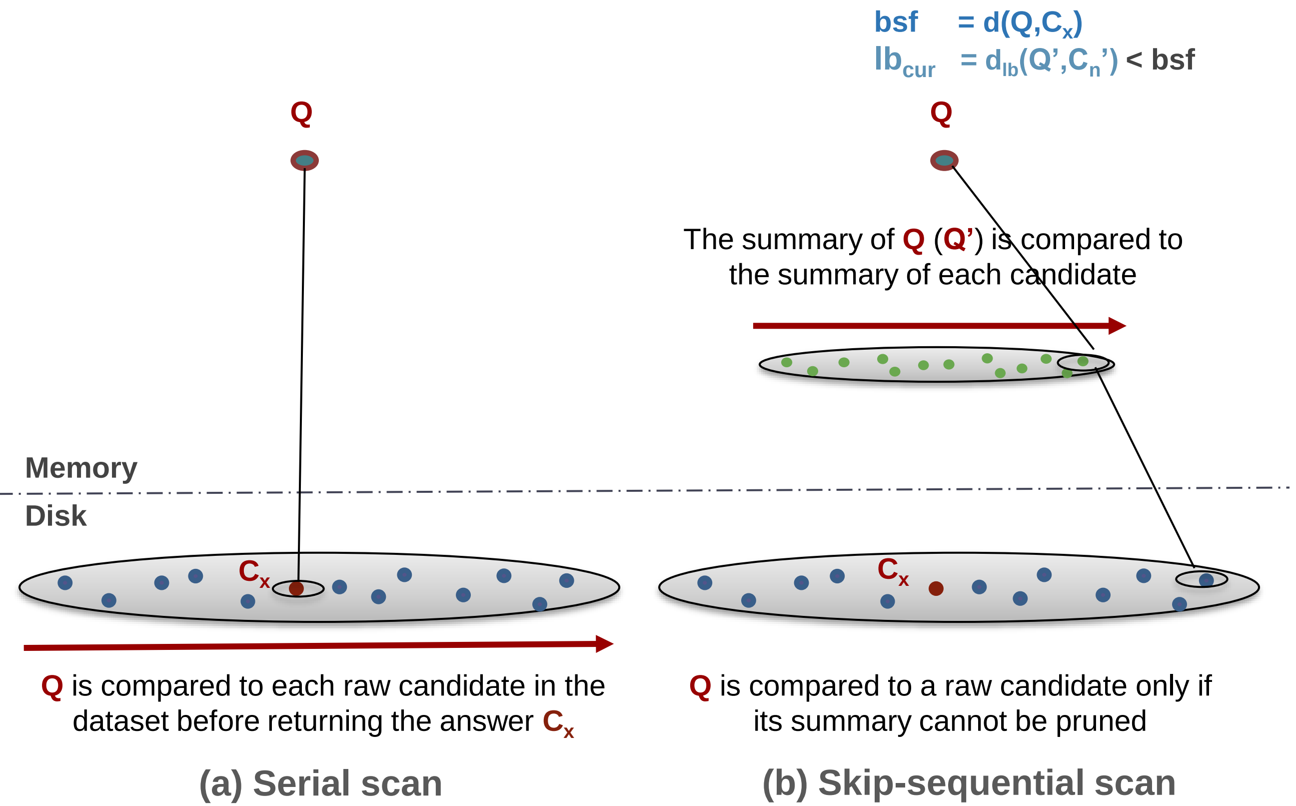
Answering a similarity search query using different access paths

Indexes vs. Scans



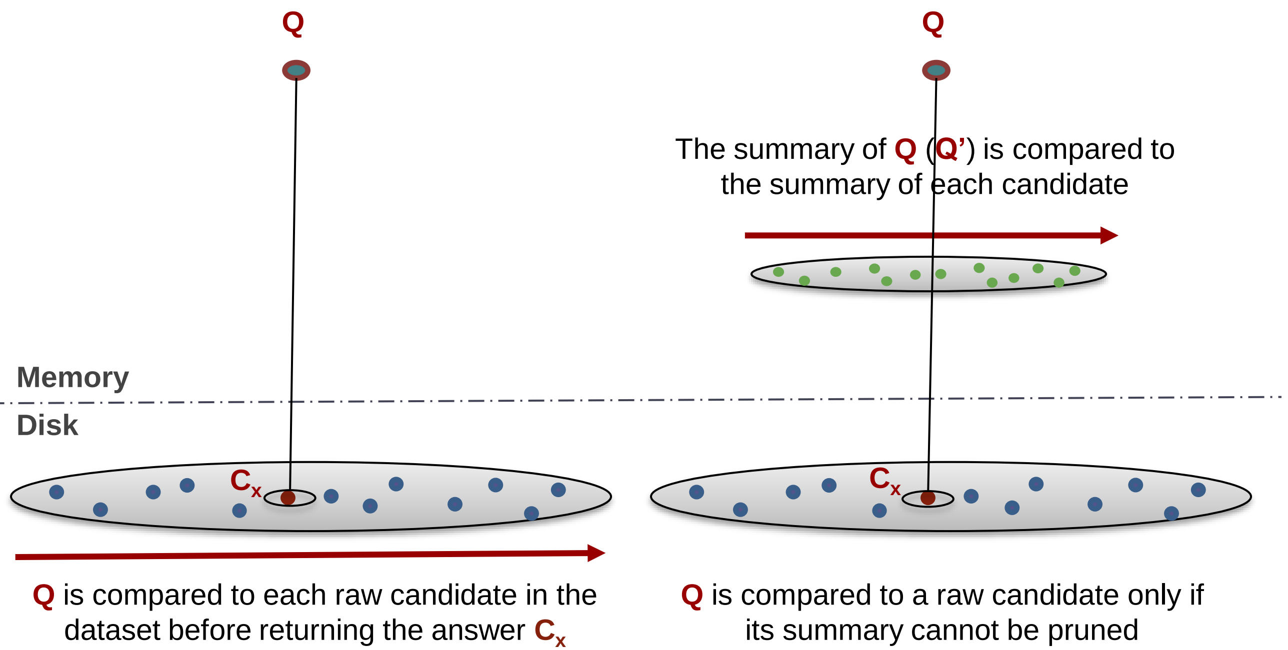
<u>**Answering a similarity search query using different access paths**</u>

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans

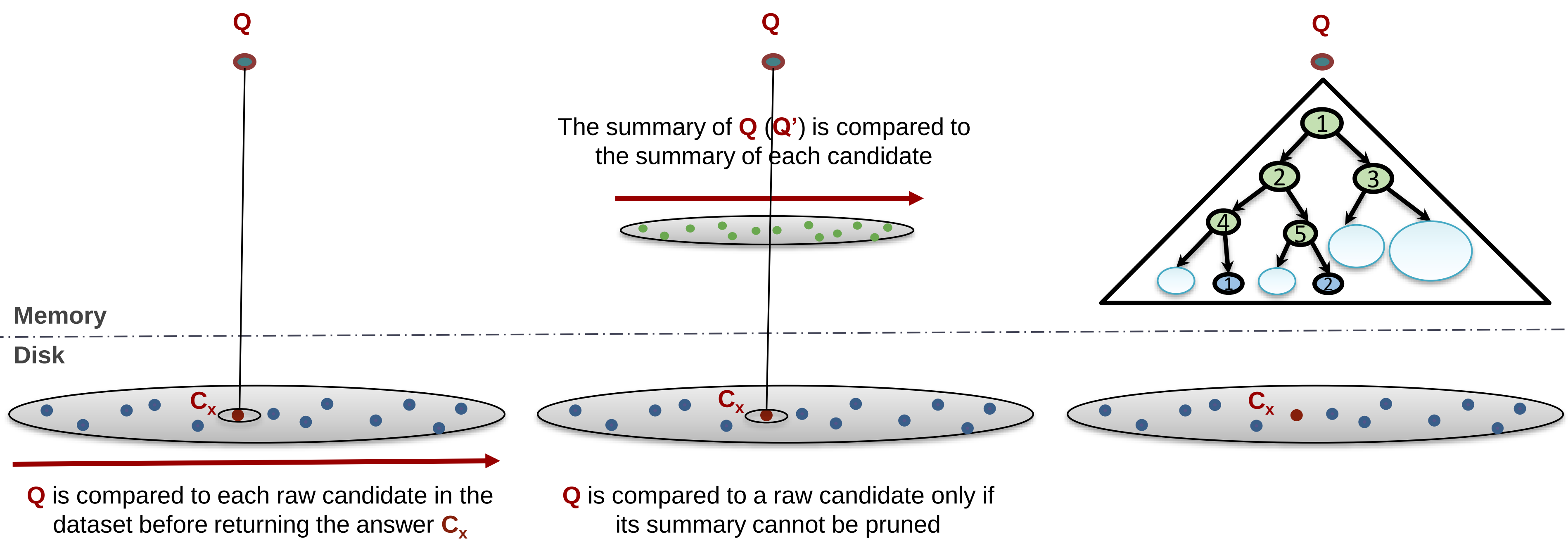


(a) Serial scan

(b) Skip-sequential scan

Answering a similarity search query using different access paths

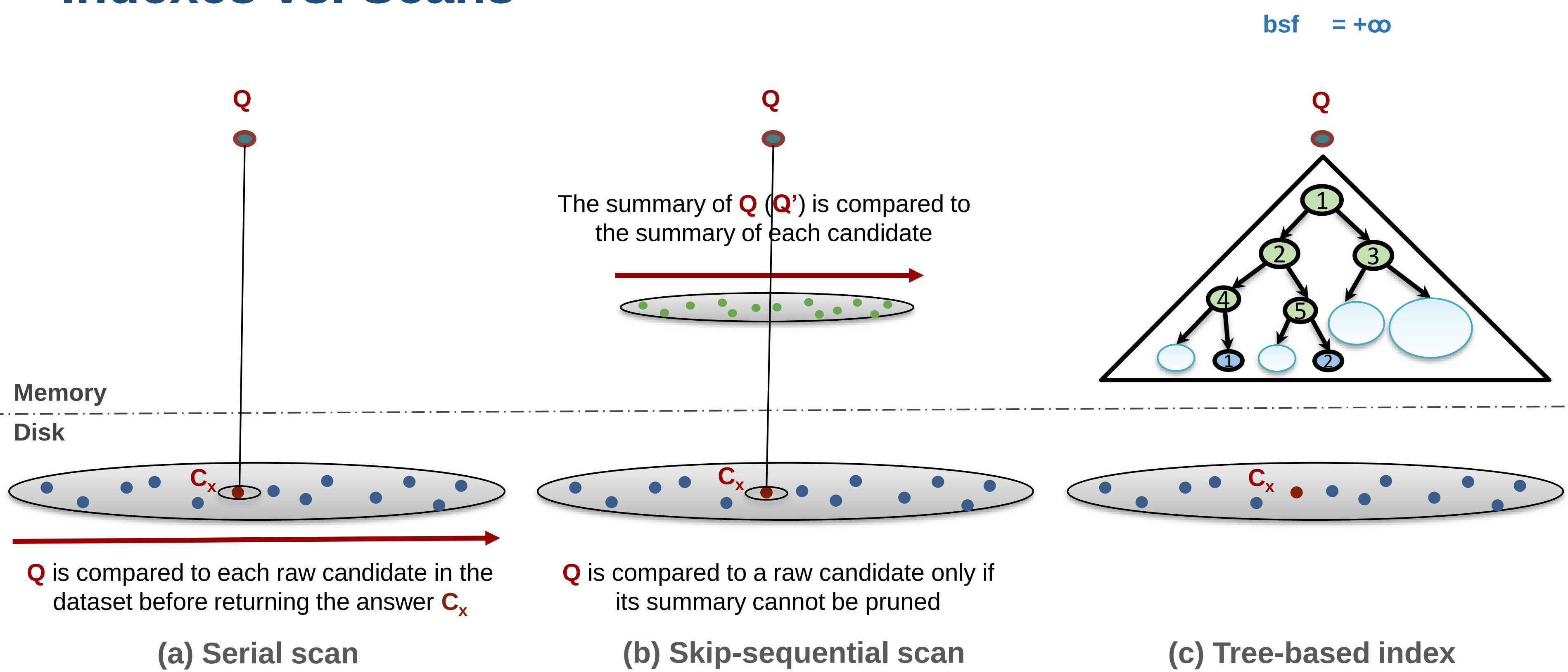
Indexes vs. Scans



(a) Serial scan (b) Skip-sequential scan (c) Tree-based index

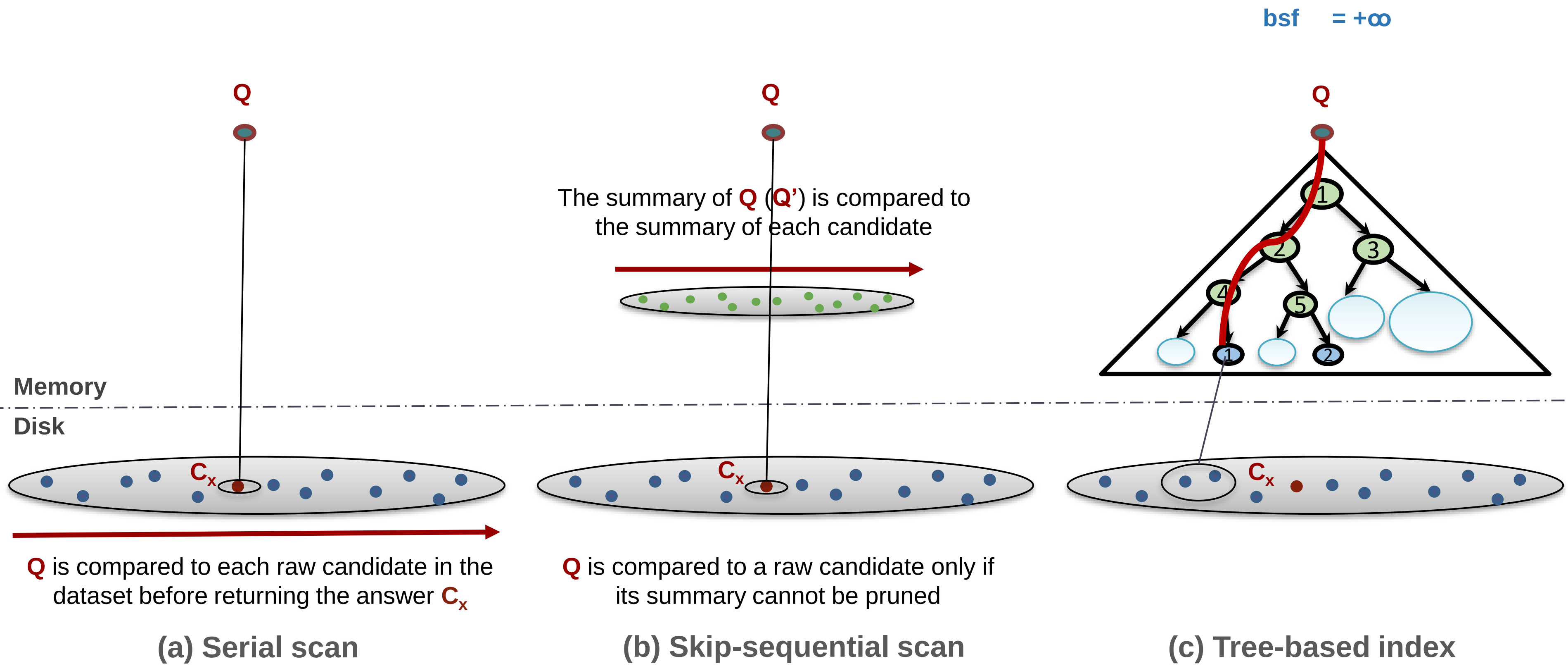
<u>Answering a similarity search query using different access paths</u>

Indexes vs. Scans



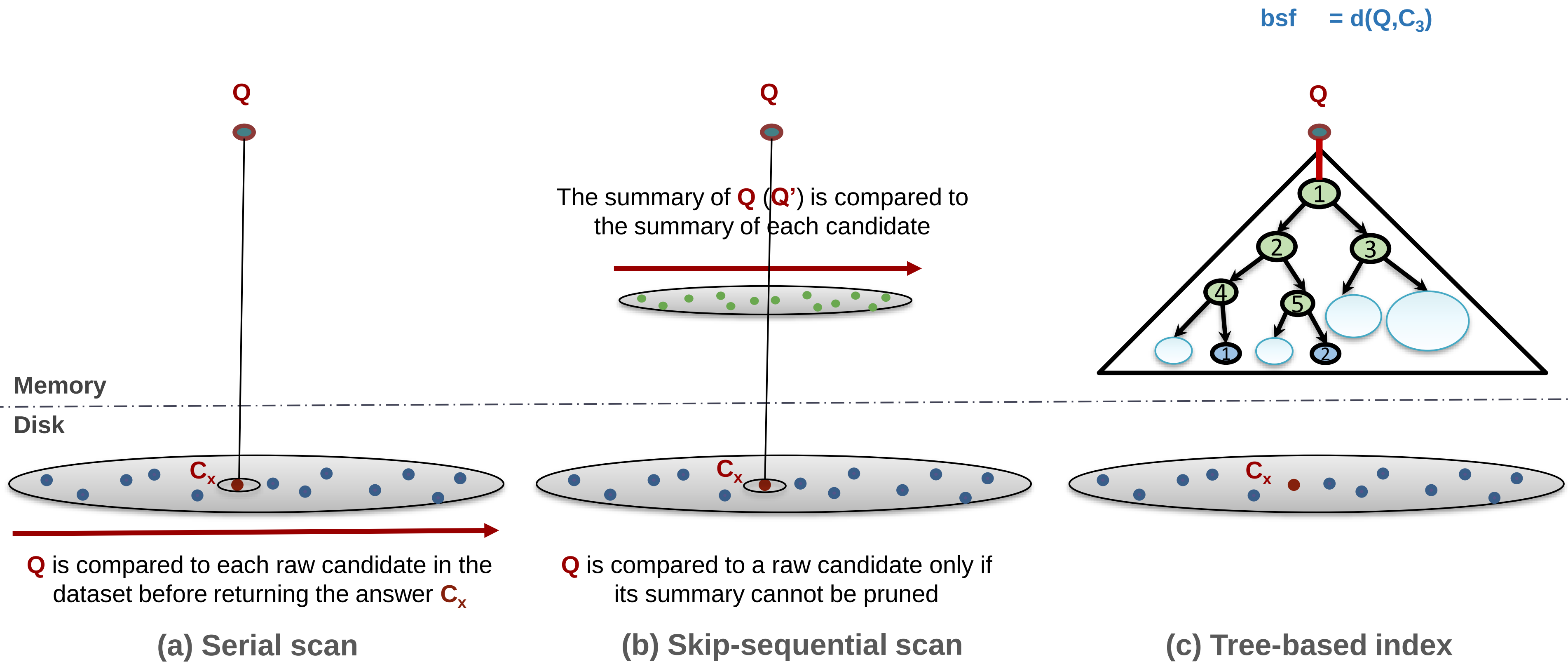
<u>**Answering a similarity search query using different access paths**</u>

Indexes vs. Scans



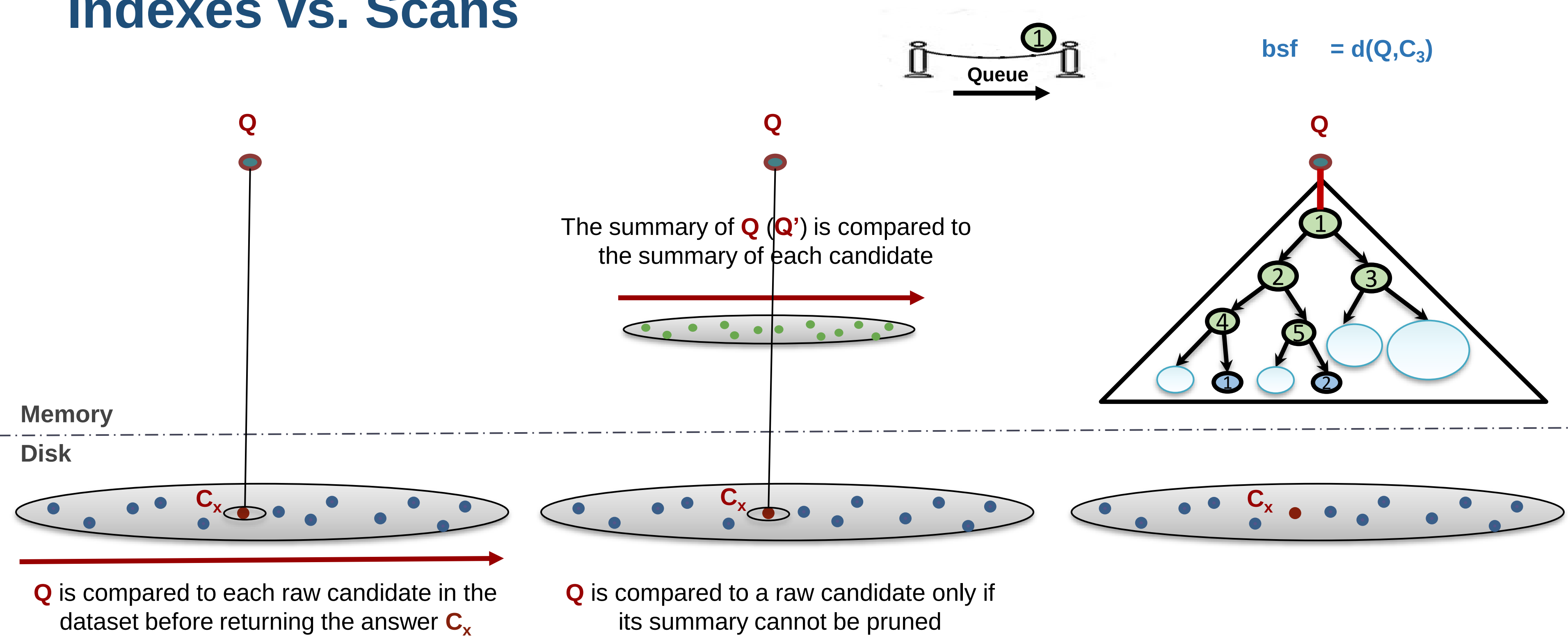
<u>**Answering a similarity search query using different access paths**</u>

Indexes vs. Scans



<u>Answering a similarity search query using different access paths</u>

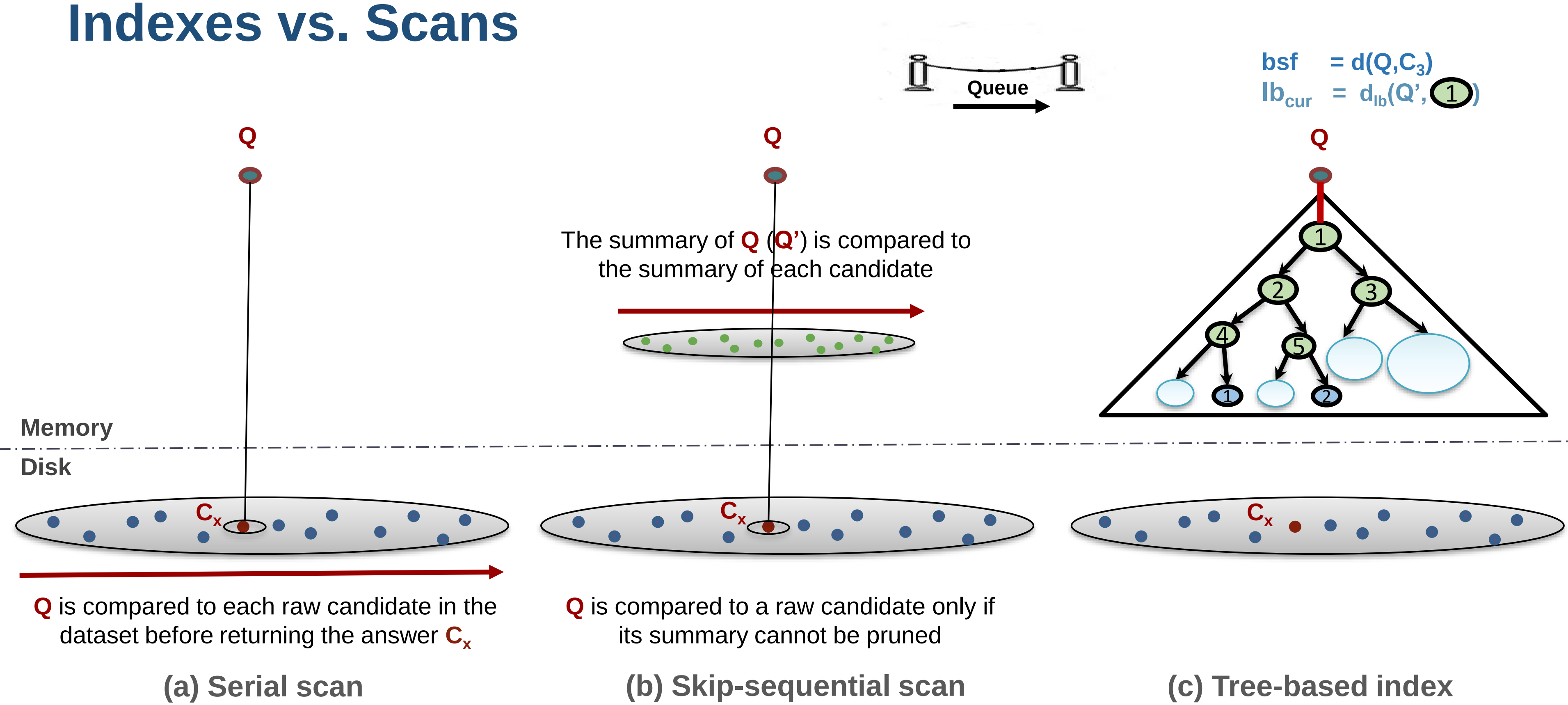
Indexes vs. Scans



(a) Serial scan (b) Skip-sequential scan (c) Tree-based index

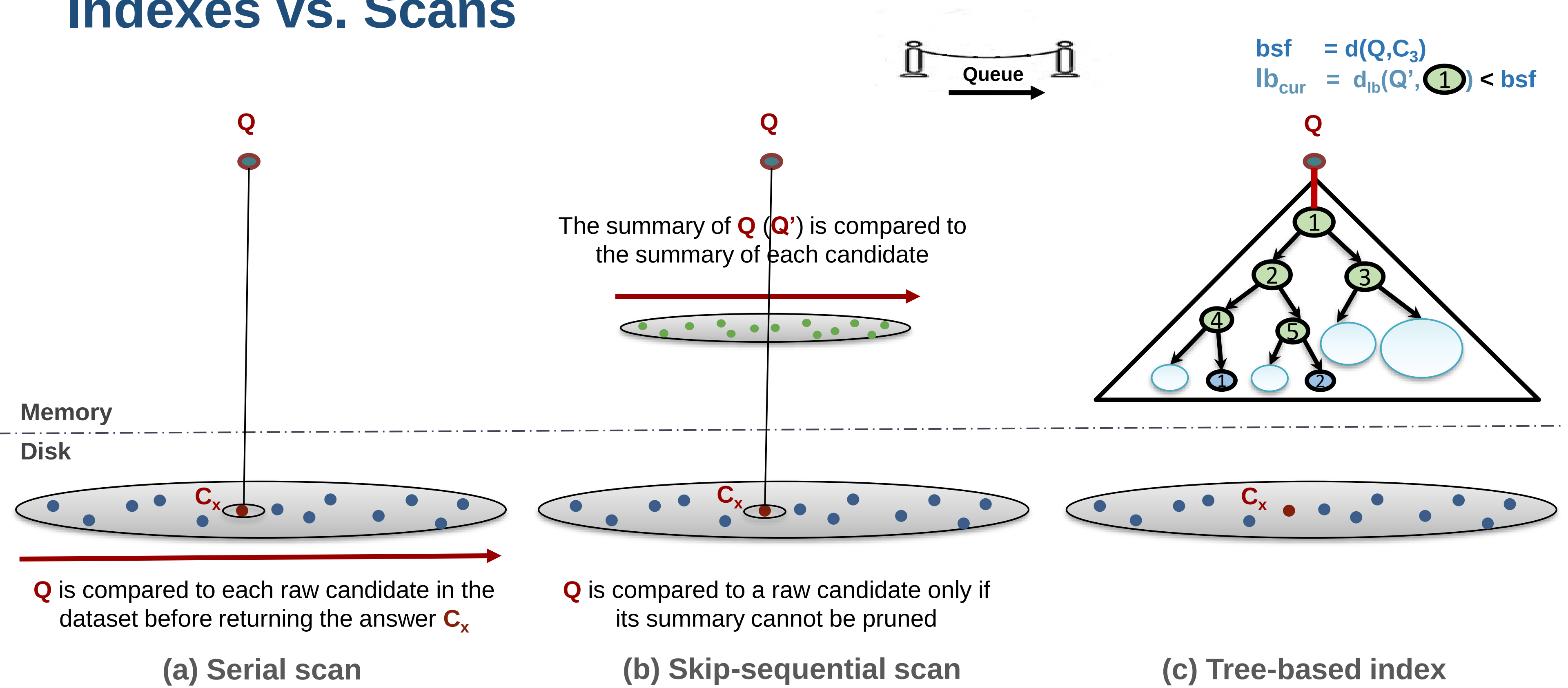
Answering a similarity search query using different access paths

Indexes vs. Scans



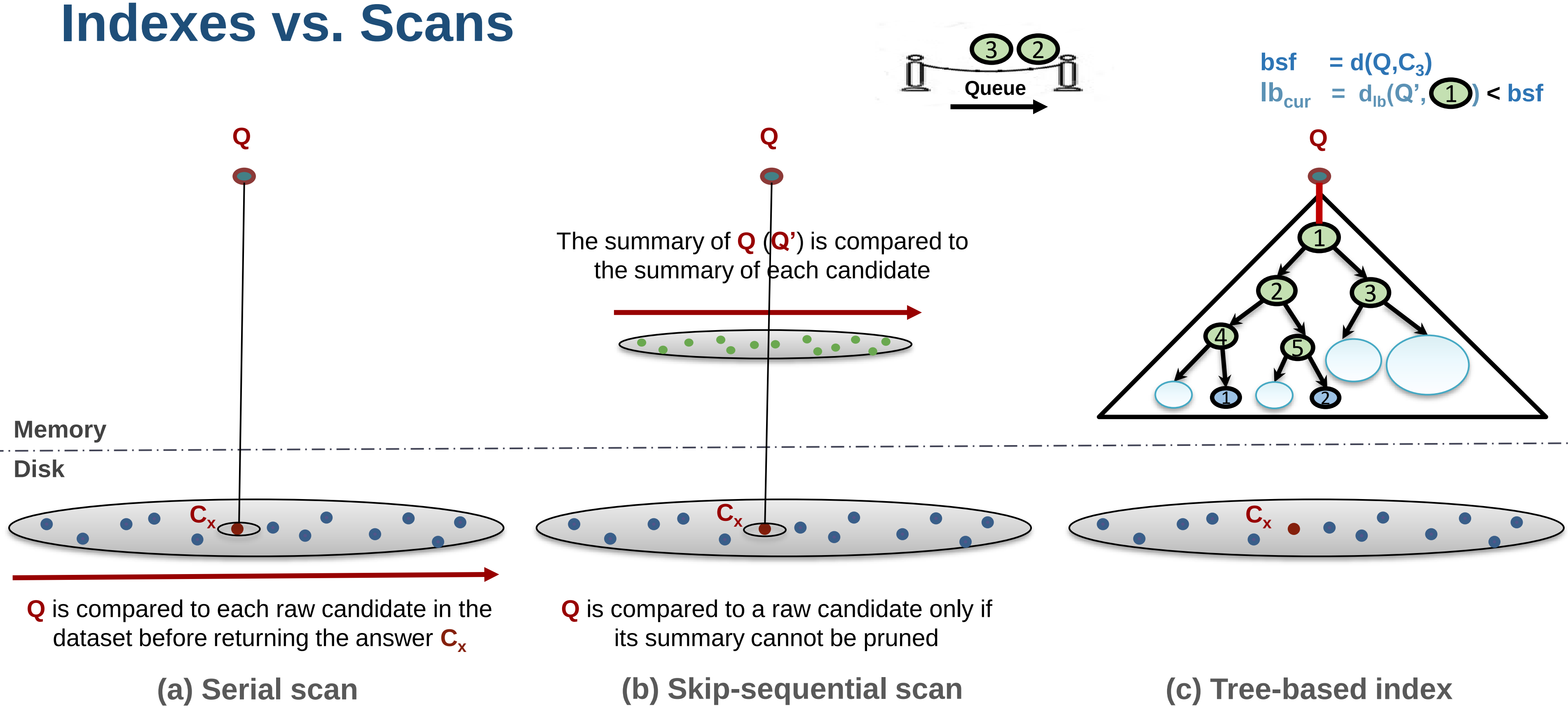
Answering a similarity search query using different access paths

Indexes vs. Scans



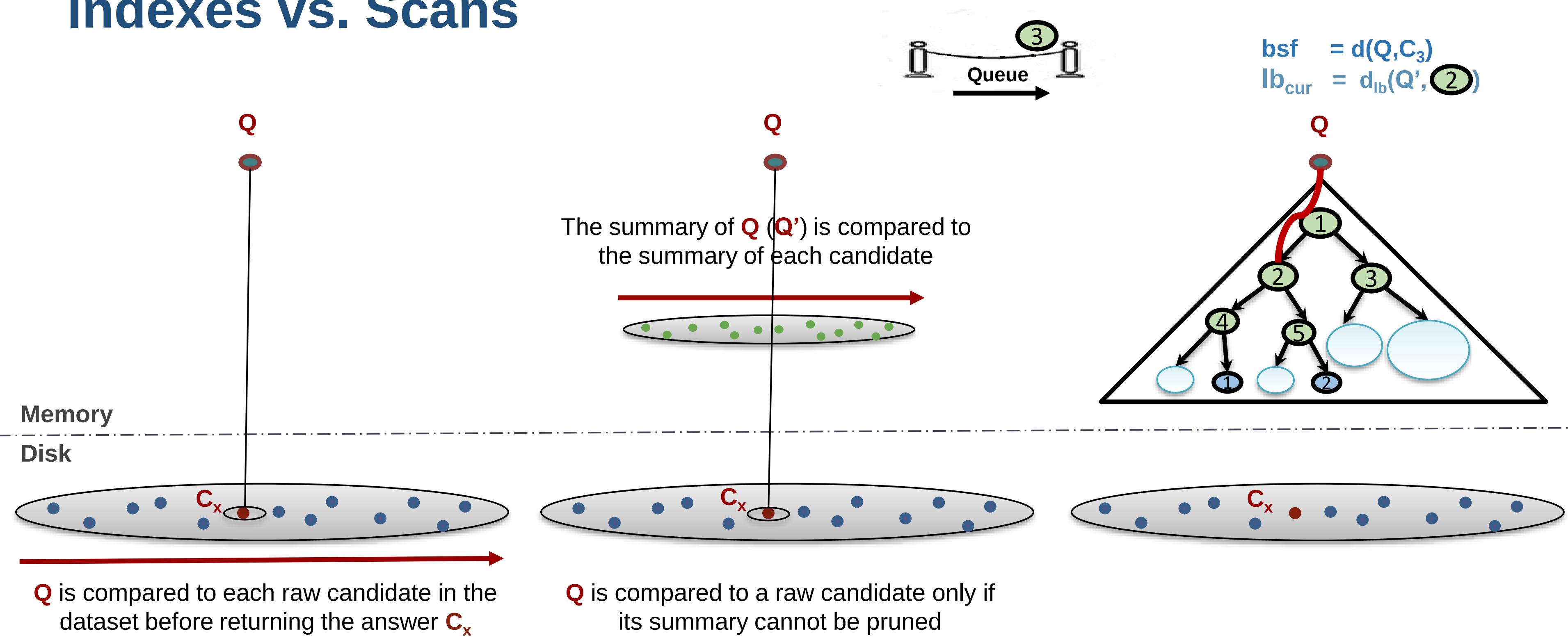
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

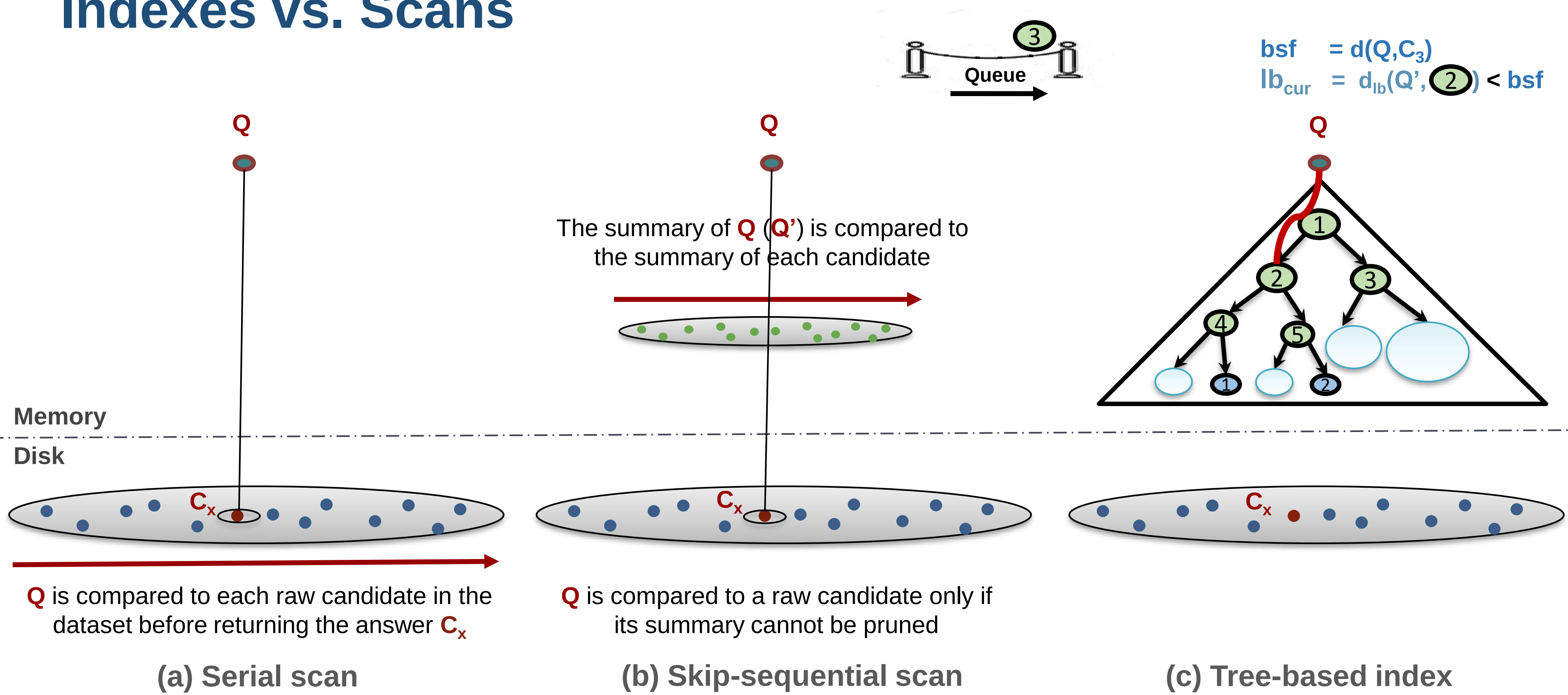
Indexes vs. Scans



(a) Serial scan (b) Skip-sequential scan (c) Tree-based index

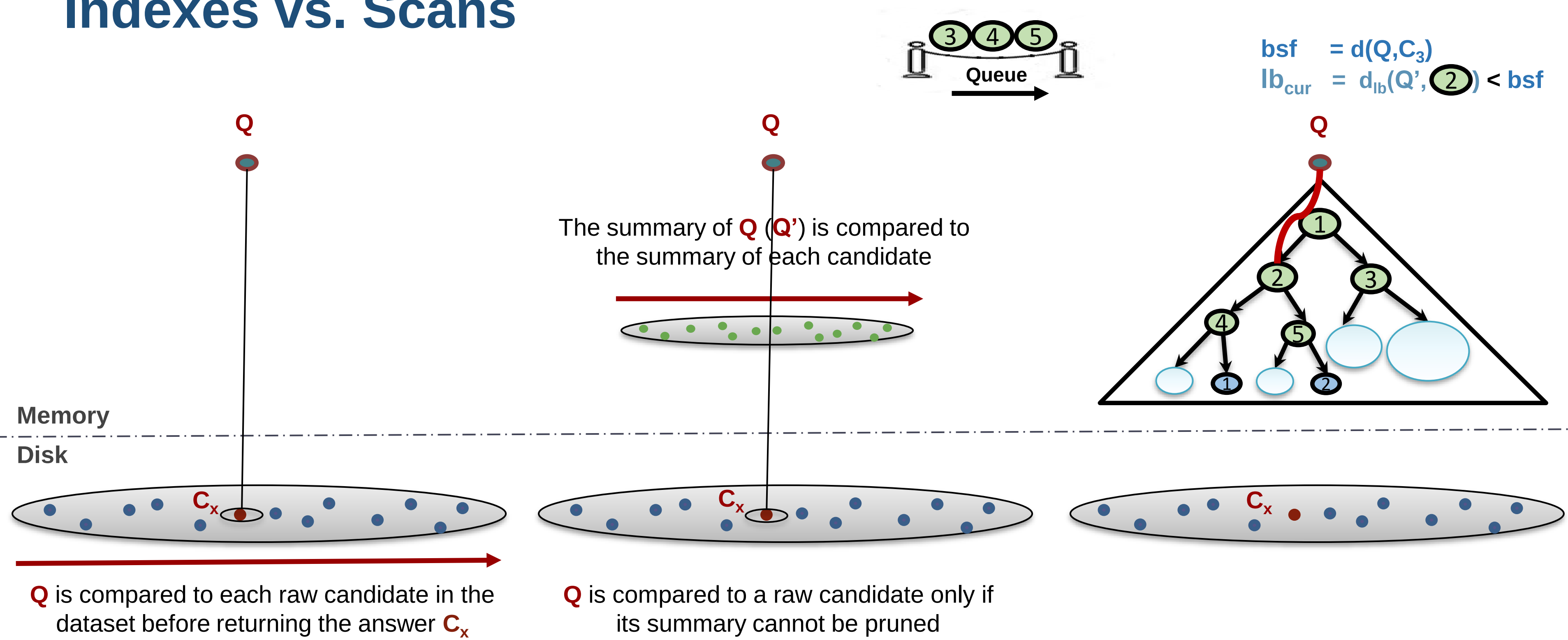
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

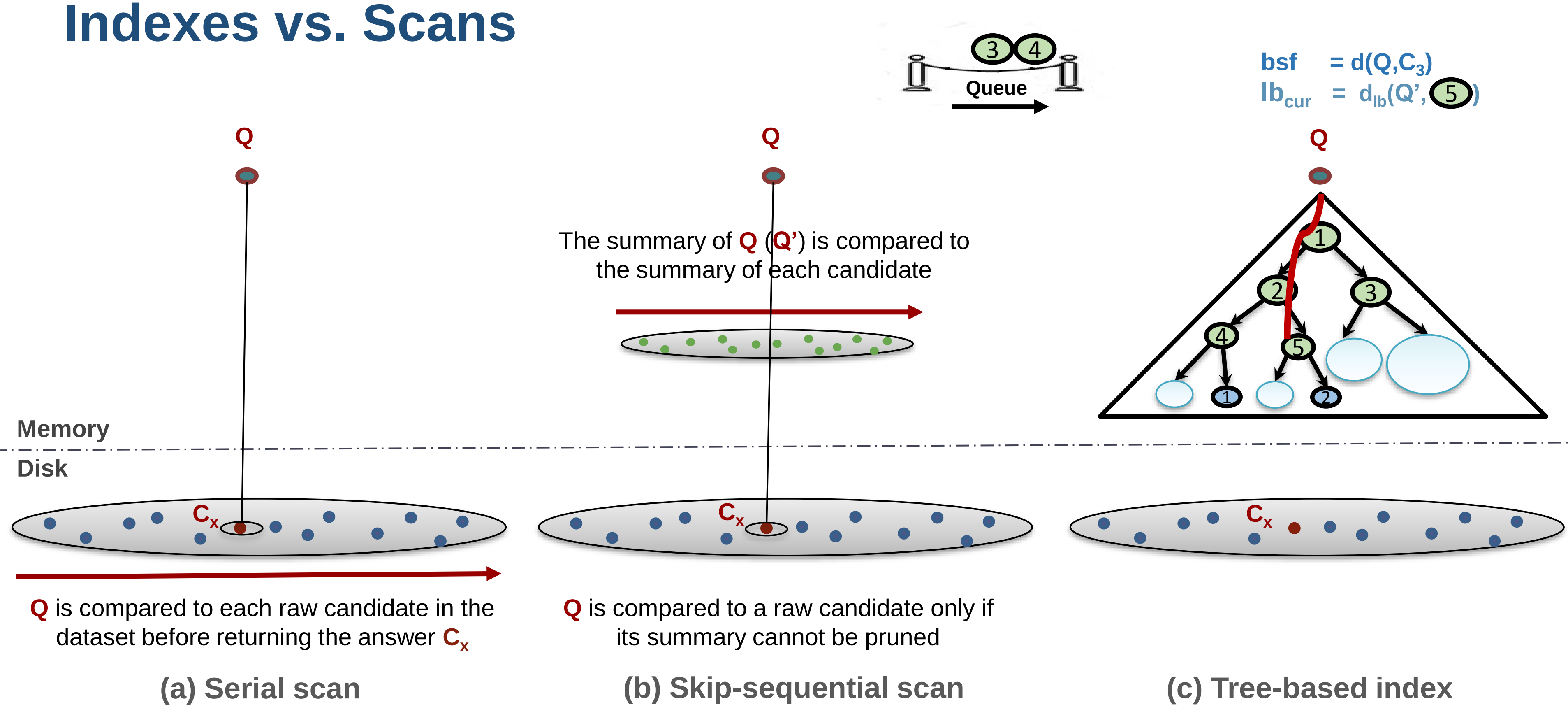
Indexes vs. Scans



(a) Serial scan (b) Skip-sequential scan (c) Tree-based index

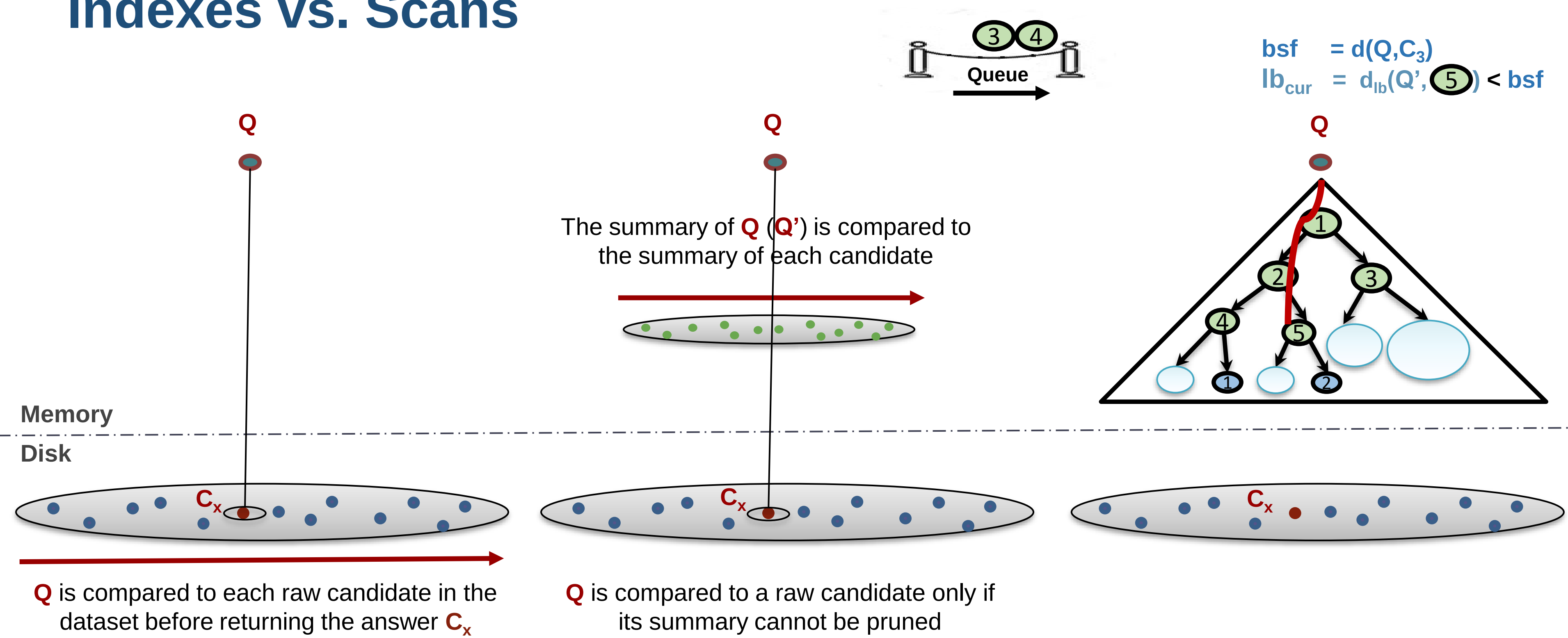
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

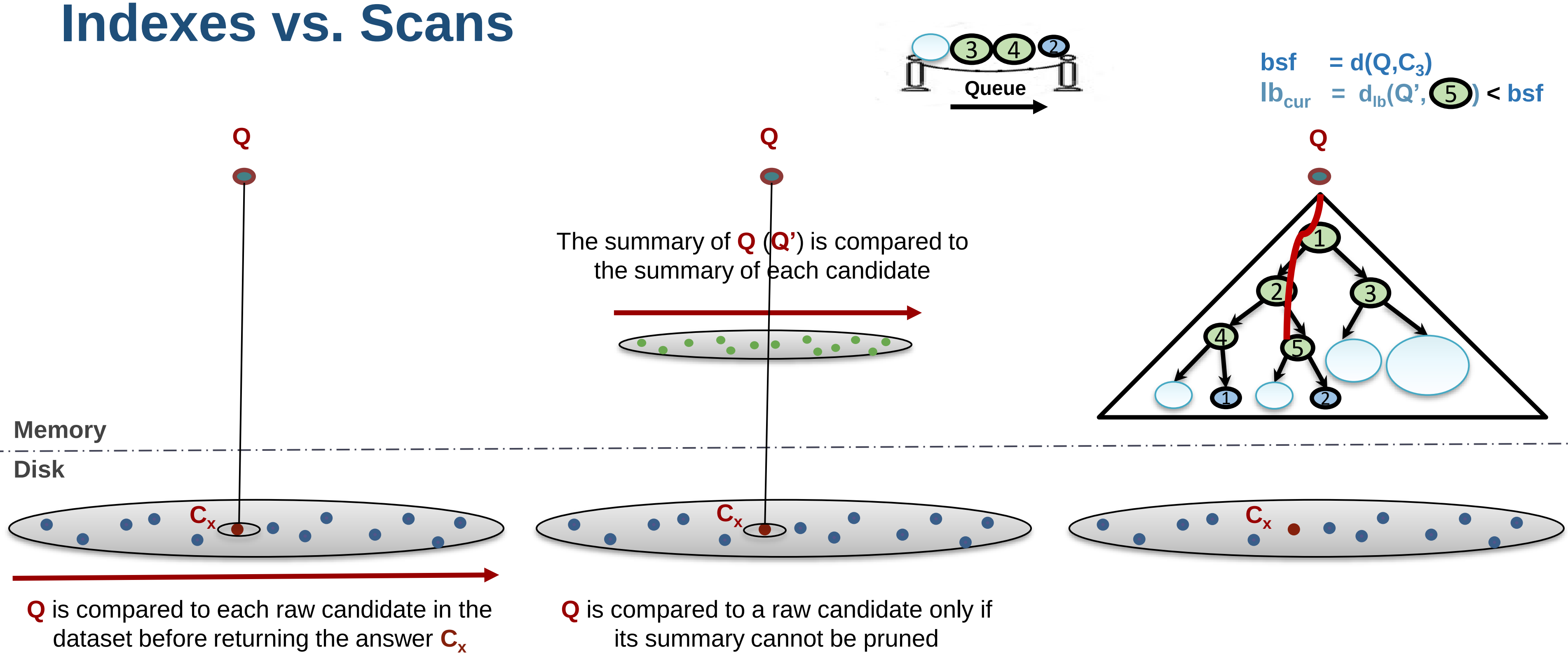
Indexes vs. Scans



(a) Serial scan (b) Skip-sequential scan (c) Tree-based index

Answering a similarity search query using different access paths

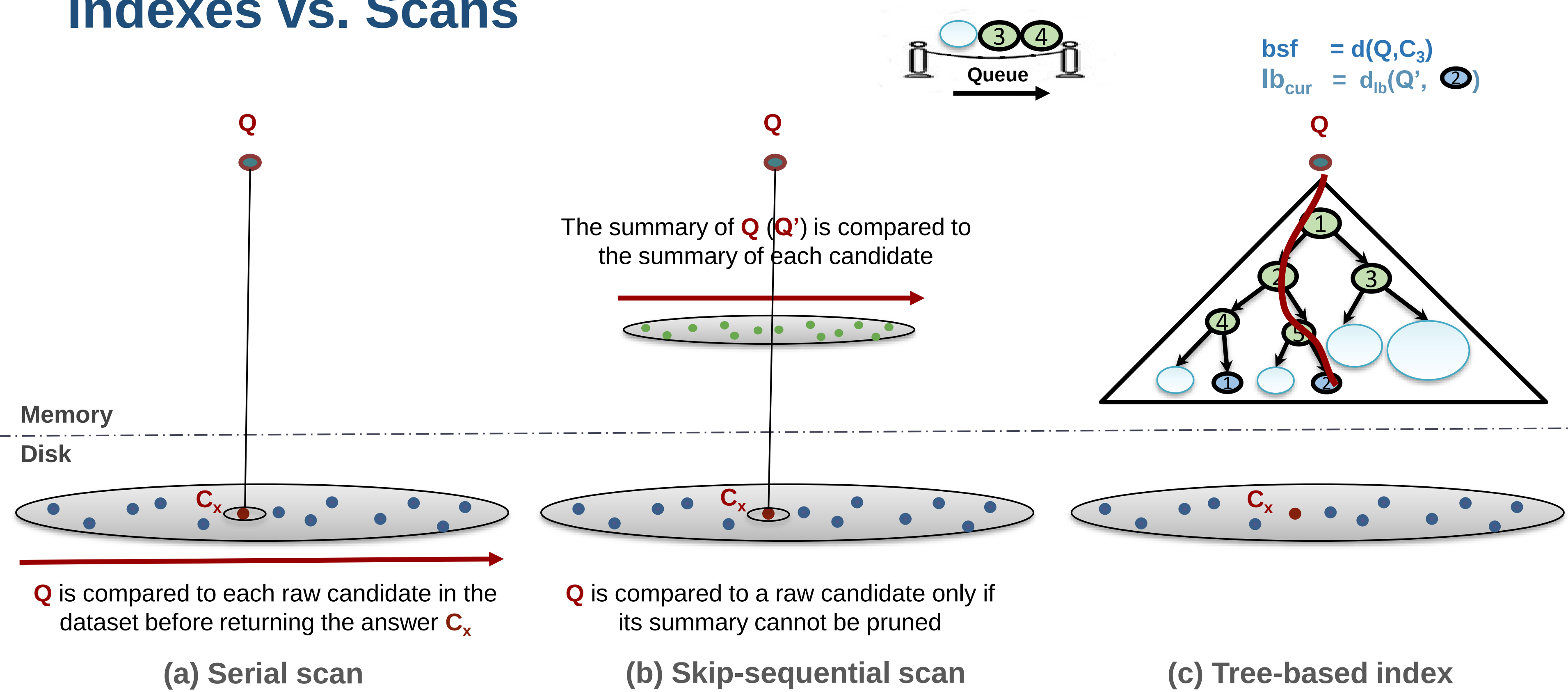
Indexes vs. Scans



(a) Serial scan (b) Skip-sequential scan (c) Tree-based index

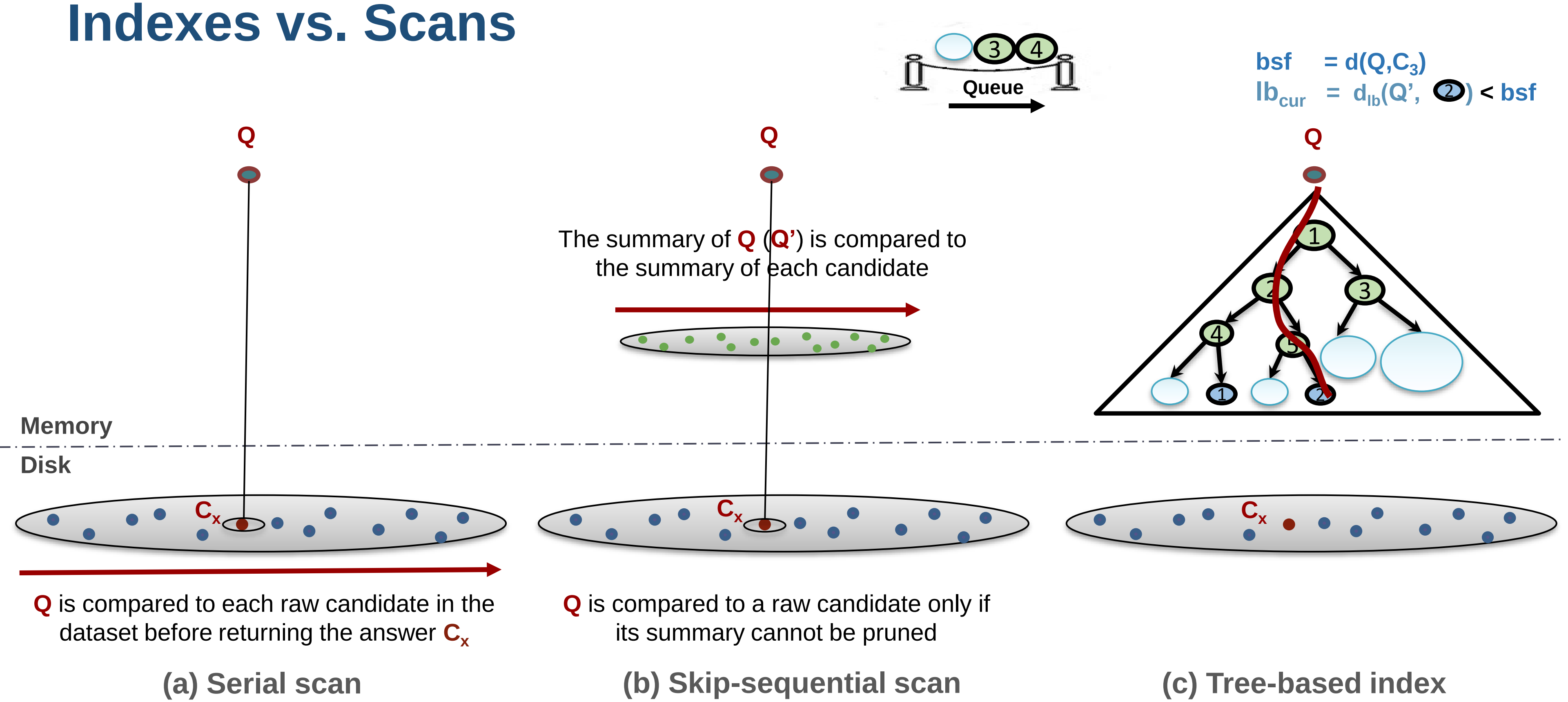
Answering a similarity search query using different access paths

Indexes vs. Scans



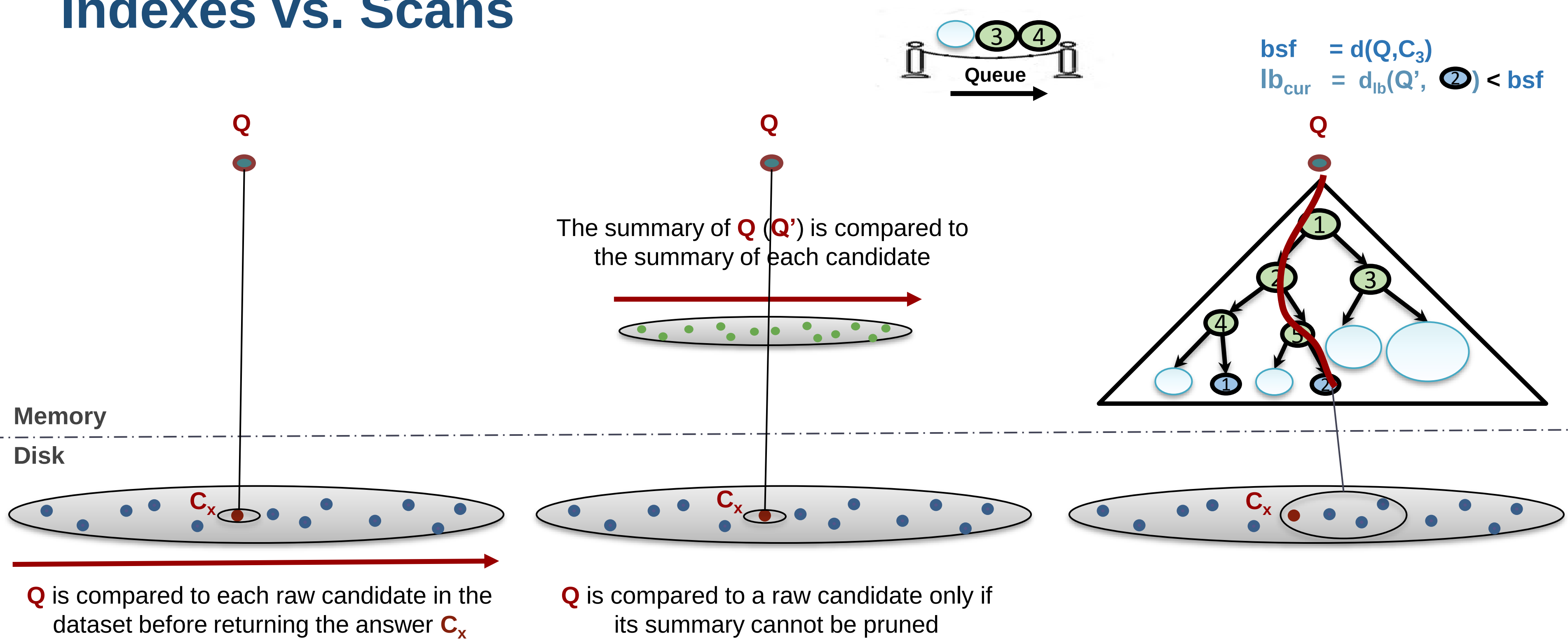
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans



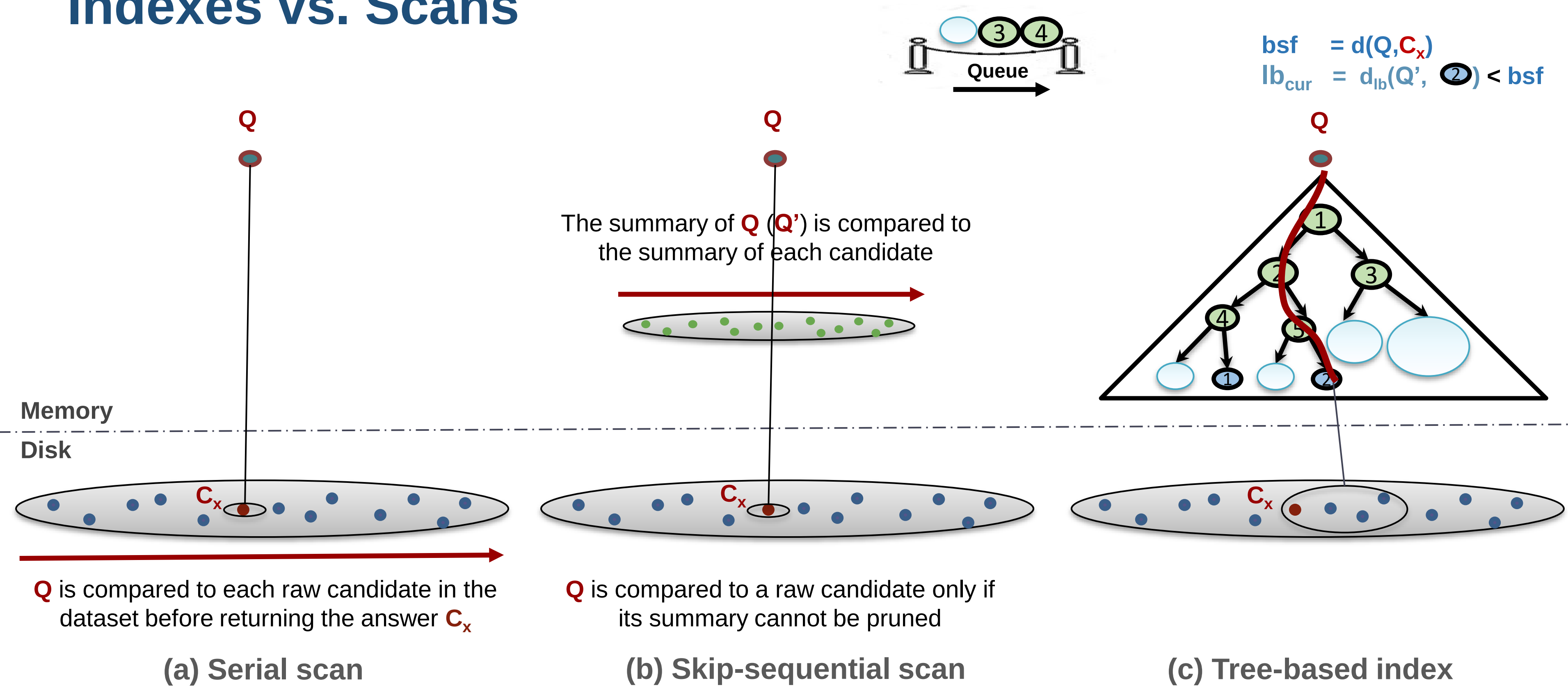
(a) Serial scan

(b) Skip-sequential scan

(c) Tree-based index

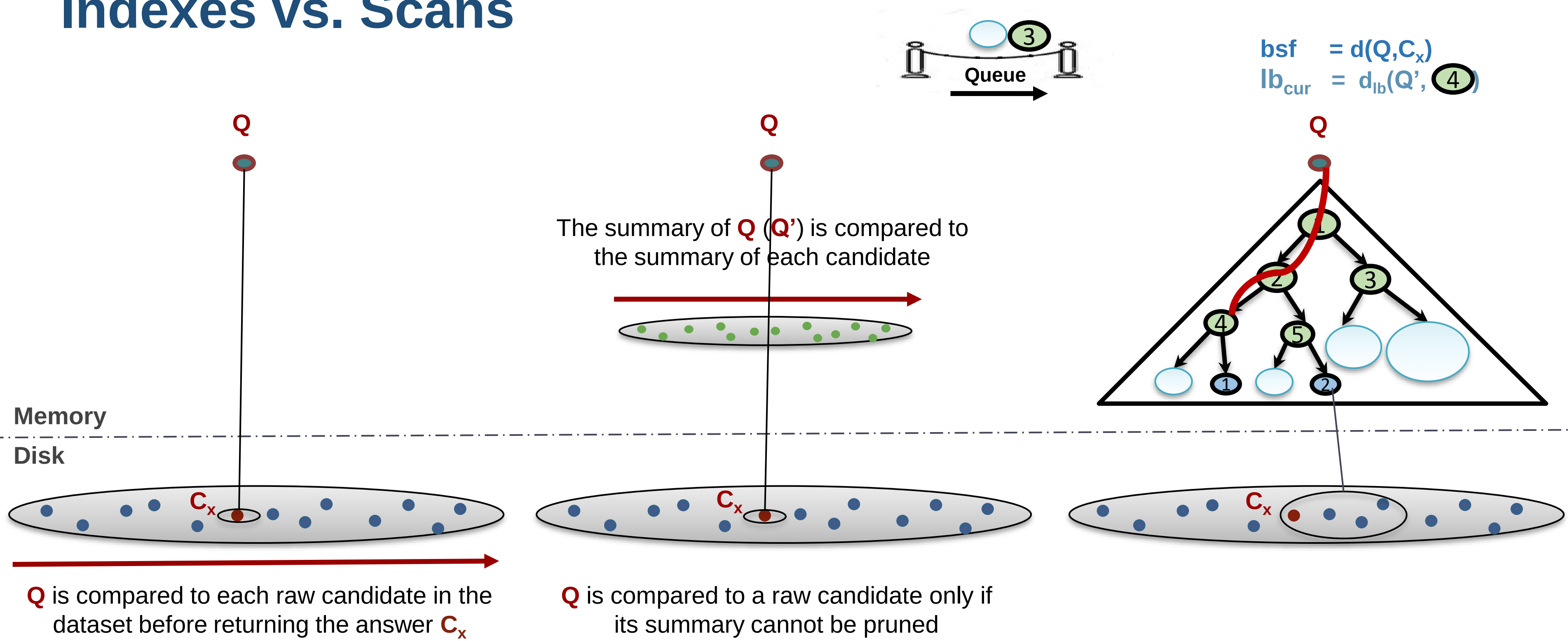
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

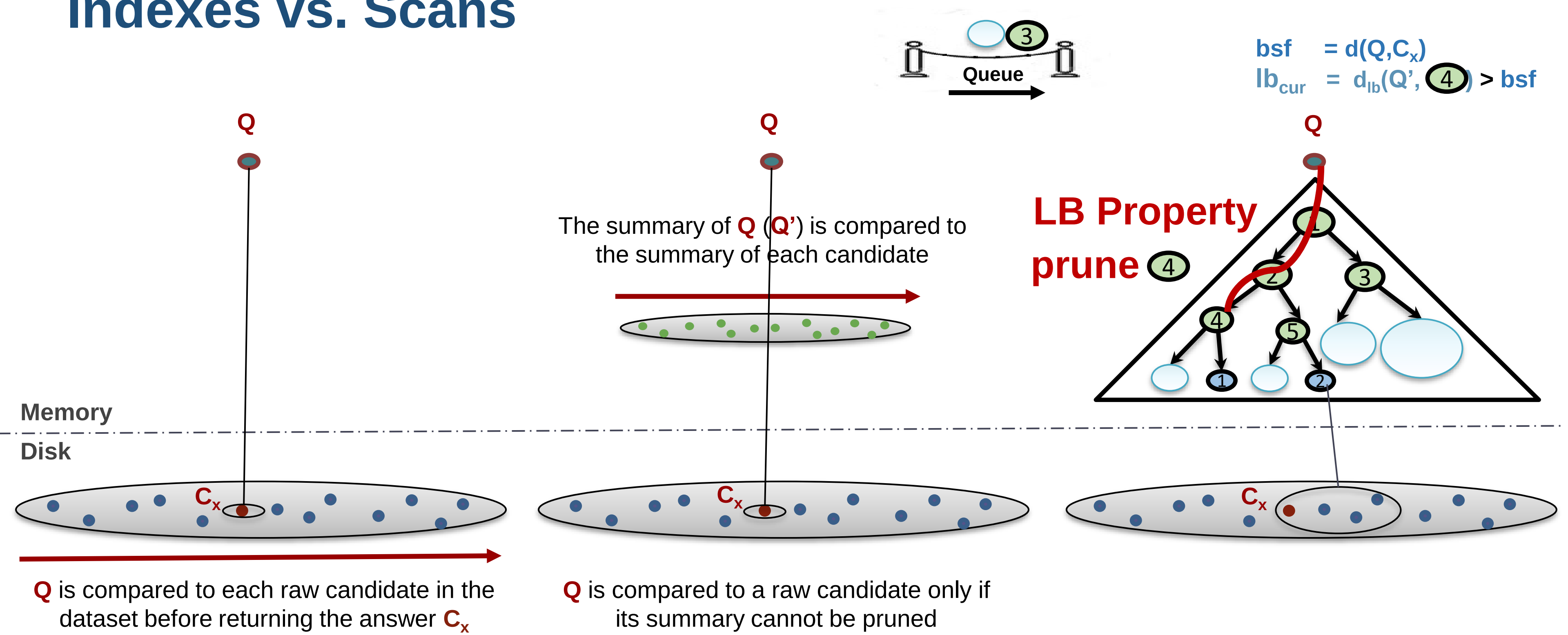
Indexes vs. Scans



(a) Serial scan (b) Skip-sequential scan (c) Tree-based index

Answering a similarity search query using different access paths

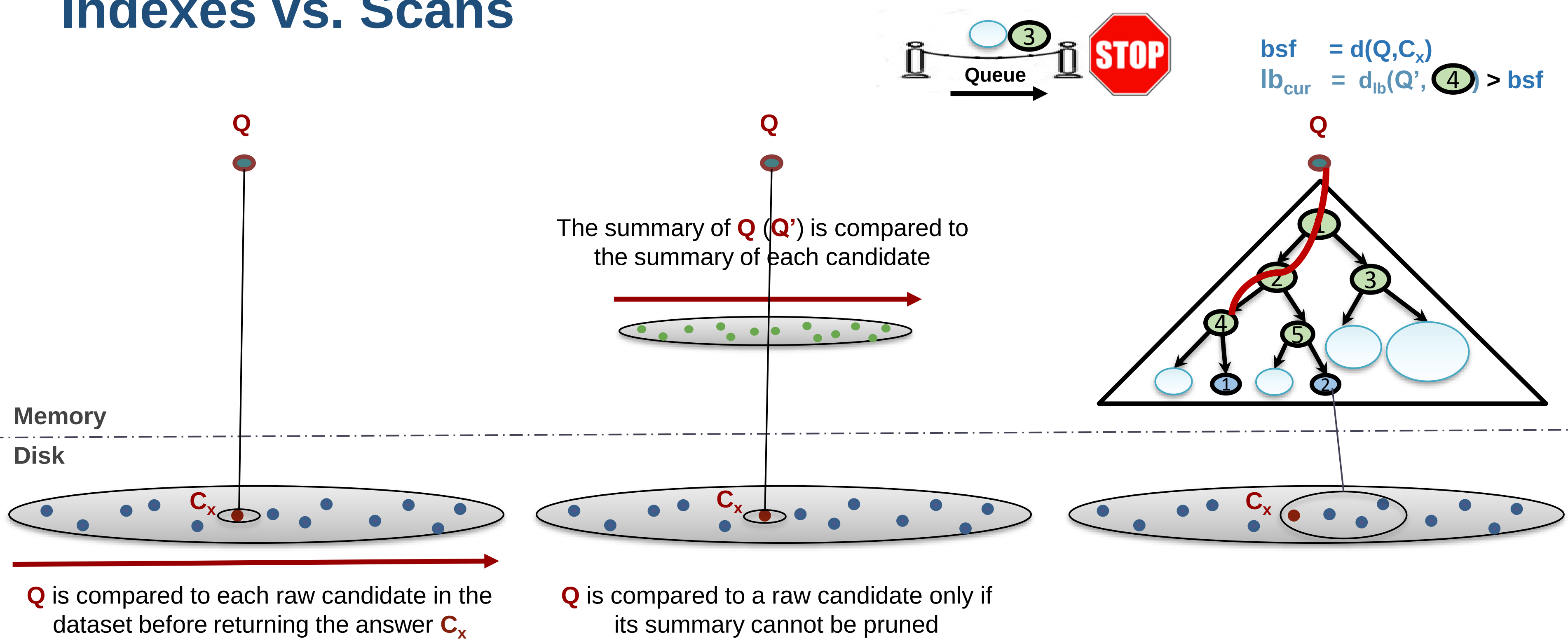
Indexes vs. Scans



(a) Serial scan (b) Skip-sequential scan (c) Tree-based index

Answering a similarity search query using different access paths

Indexes vs. Scans



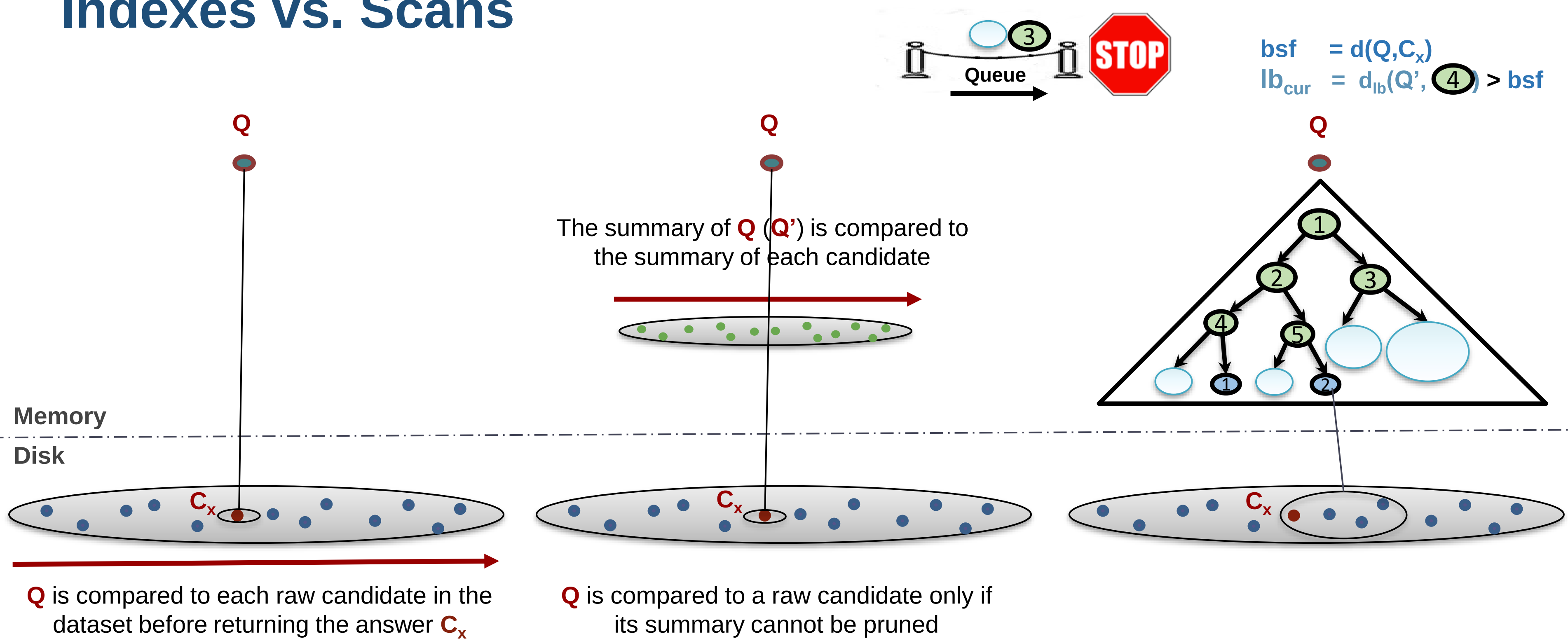
(a) Serial scan

(b) Skip-sequential scan

(c) Tree-based index

Answering a similarity search query using different access paths

Indexes vs. Scans

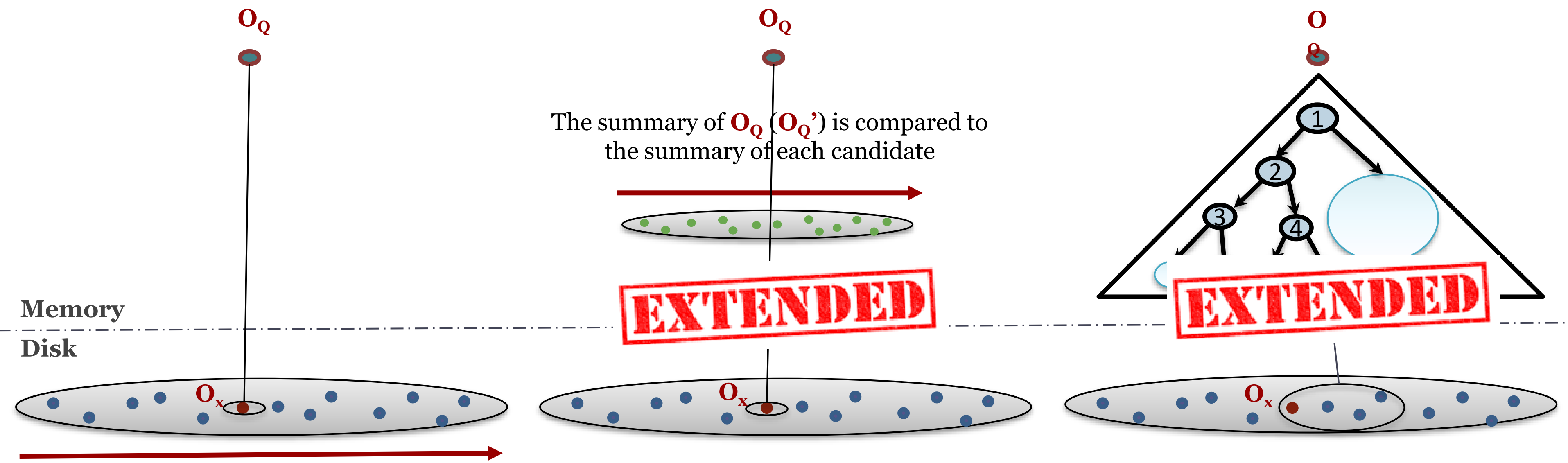


(a) Serial scan (b) Skip-sequential scan (c) Tree-based index

Answering a similarity search query using different access paths

Similarity Search
Data Series Extensions

Access Paths



O_Q is compared to each raw candidate in the dataset before returning the answer O_x

O_Q is compared to a raw candidate only if its summary cannot be pruned

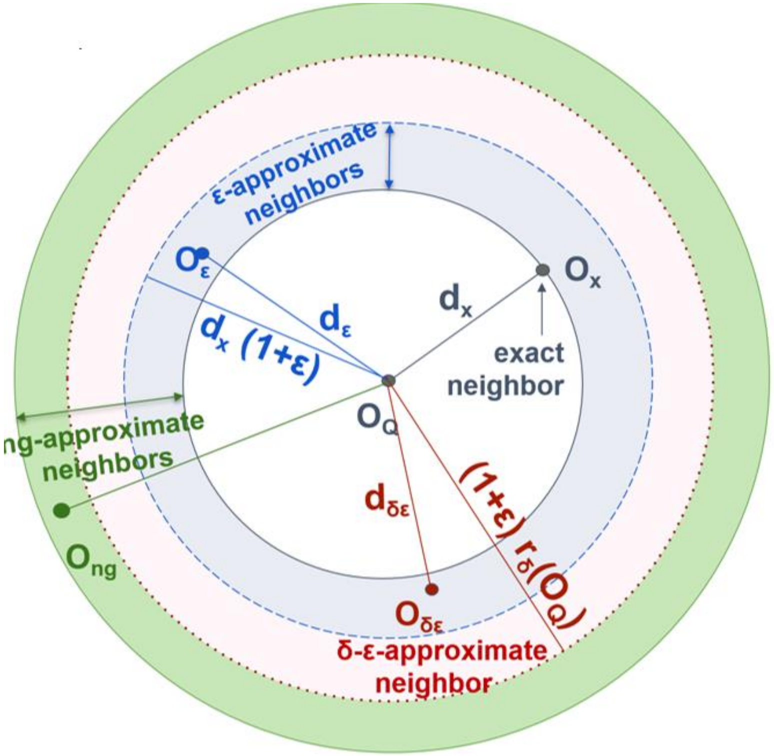
(a) Serial scan

(b) Skip-sequential scan

(c) Tree-based index

Answering a similarity search query using different access paths

Extensions: Skip-Sequential Scans

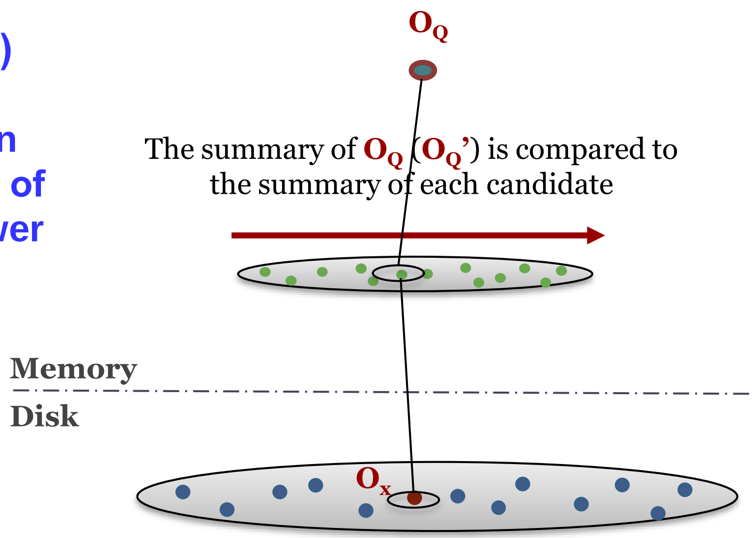


$d_\varepsilon <= d_x (1+\varepsilon)$

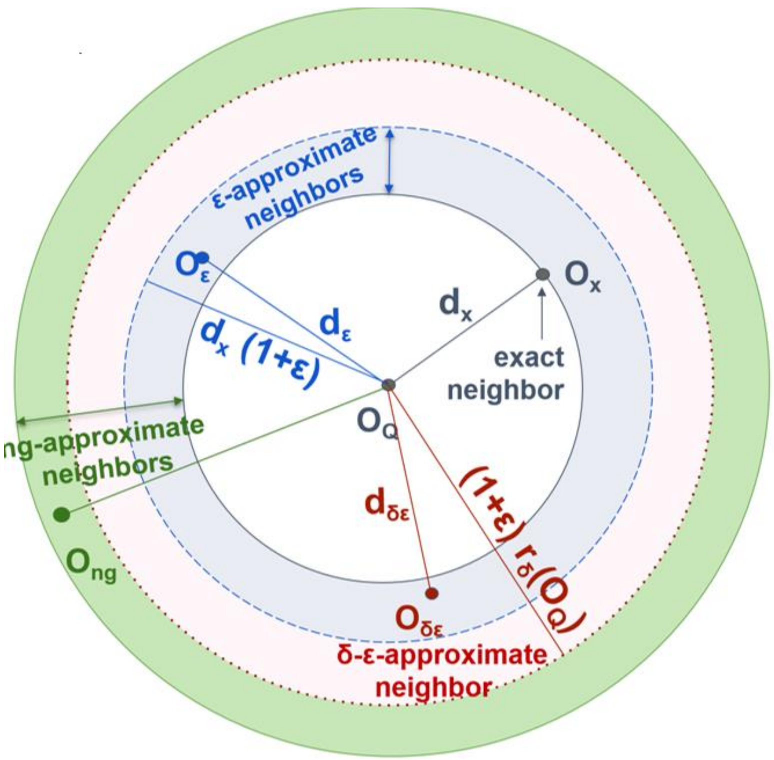
Result is within distance (1+ ε) of the exact answer

$bsf \quad = d(O_Q,O_1)$

$lb_{cur} \; = d_{lb}(O_Q', O_x') < bsf$

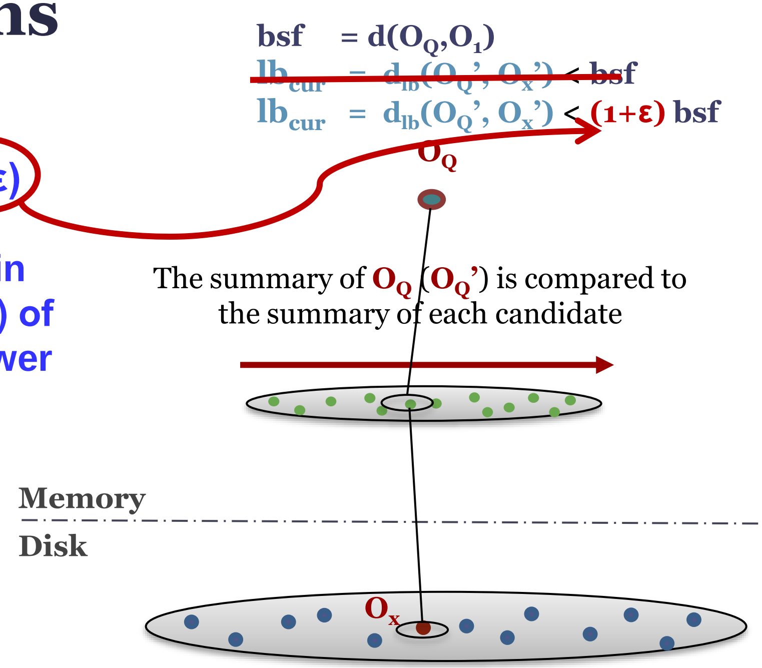


Extensions: Skip-Sequential Scans

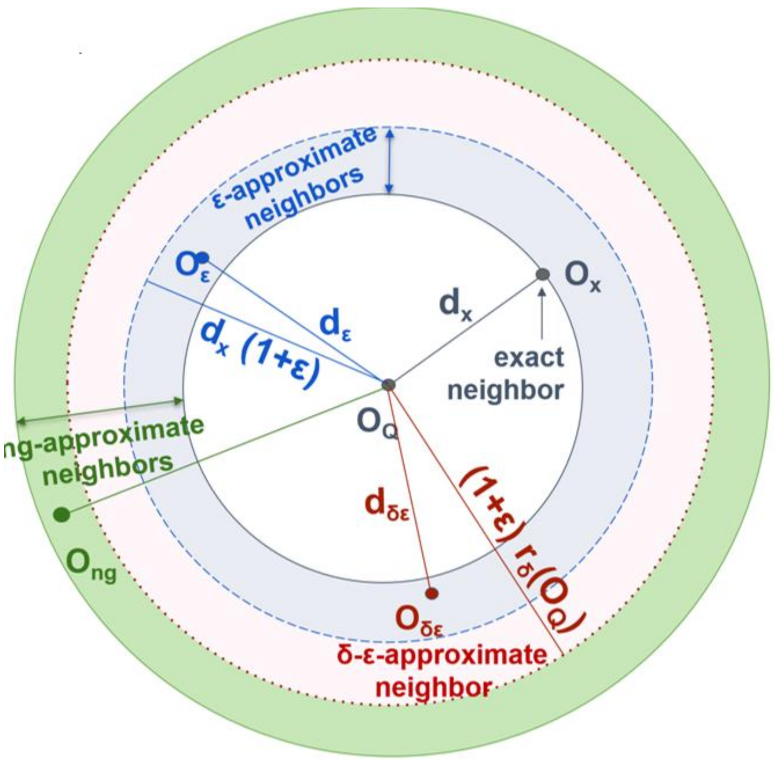


$d_\varepsilon <= d_x (1+\varepsilon)$

Result is within distance (1+ ε) of the exact answer



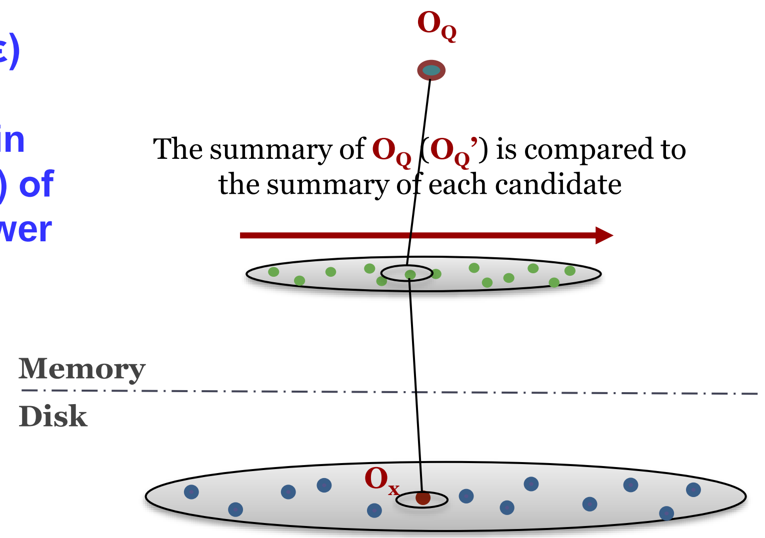
Extensions: Skip-Sequential Scans



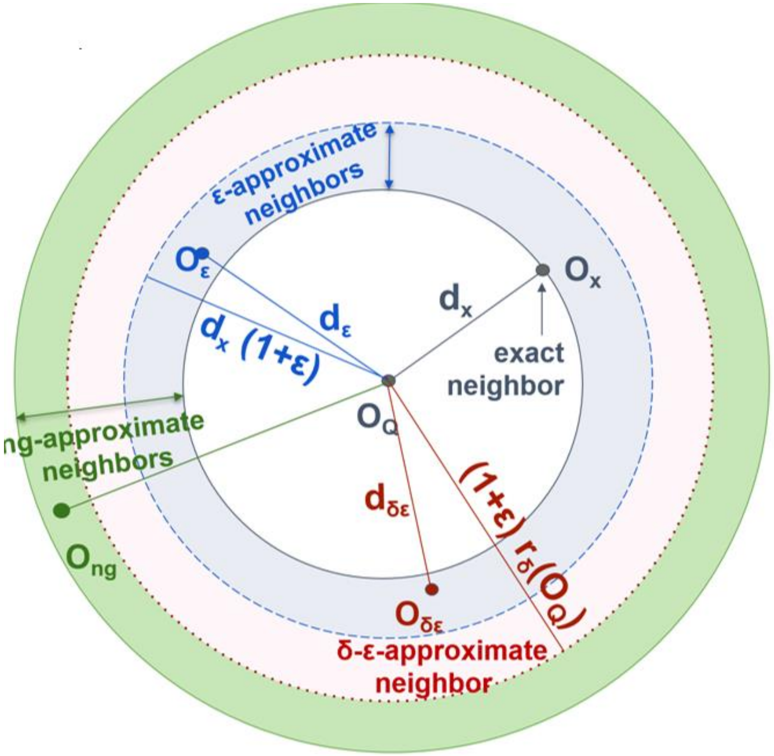
$d_ε$ <= d_x (1+ε)

Result is within distance (1+ ε) of the exact answer

bsf = $d(O_Q,O_1)$

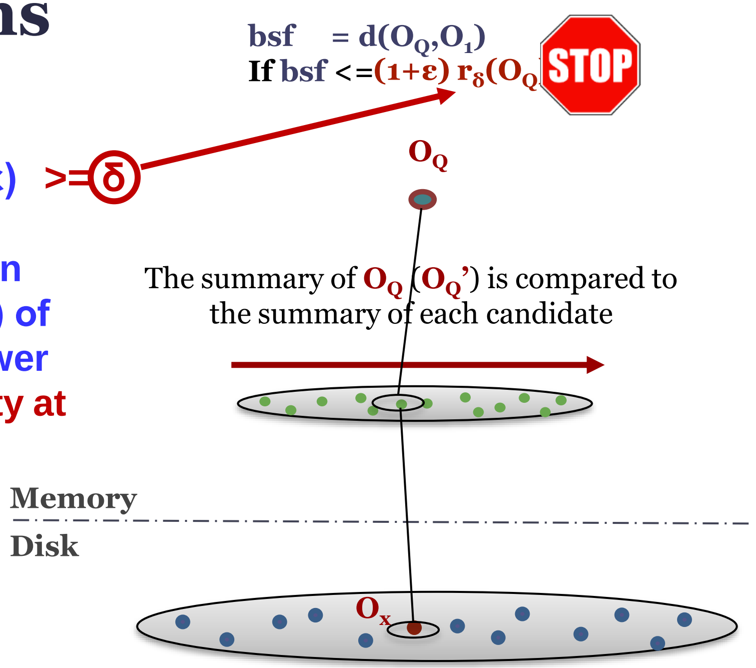


Extensions: Skip-Sequential Scans

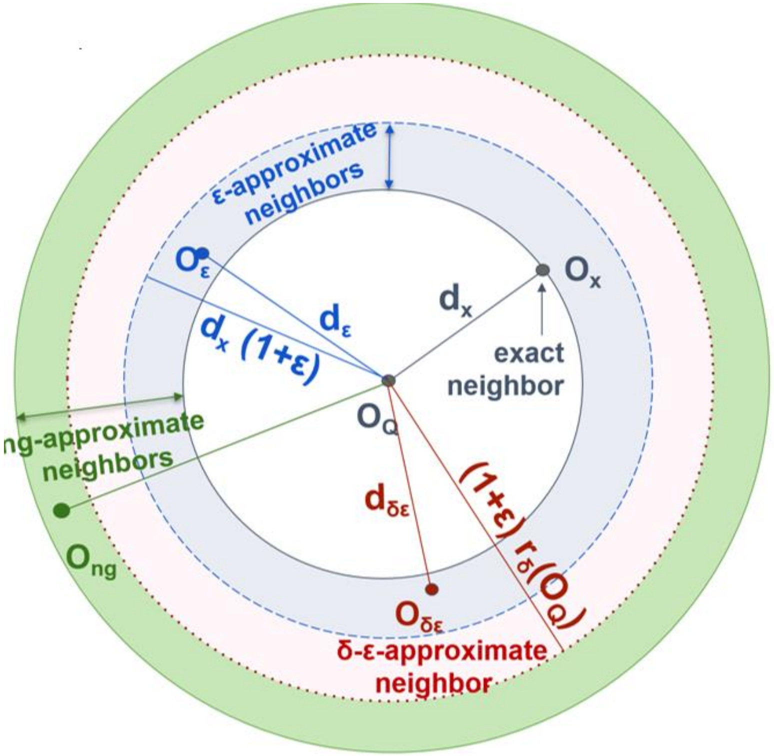


P{$d_ε$ <= d_x (1+ε) >= δ

Result is within distance (1+ ε) of the exact answer with probability at least δ

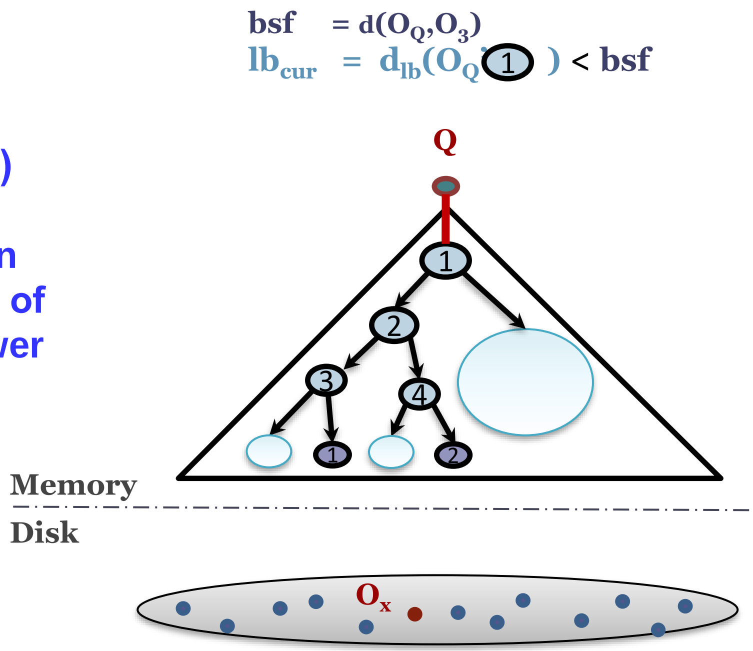


Extensions: Indexes

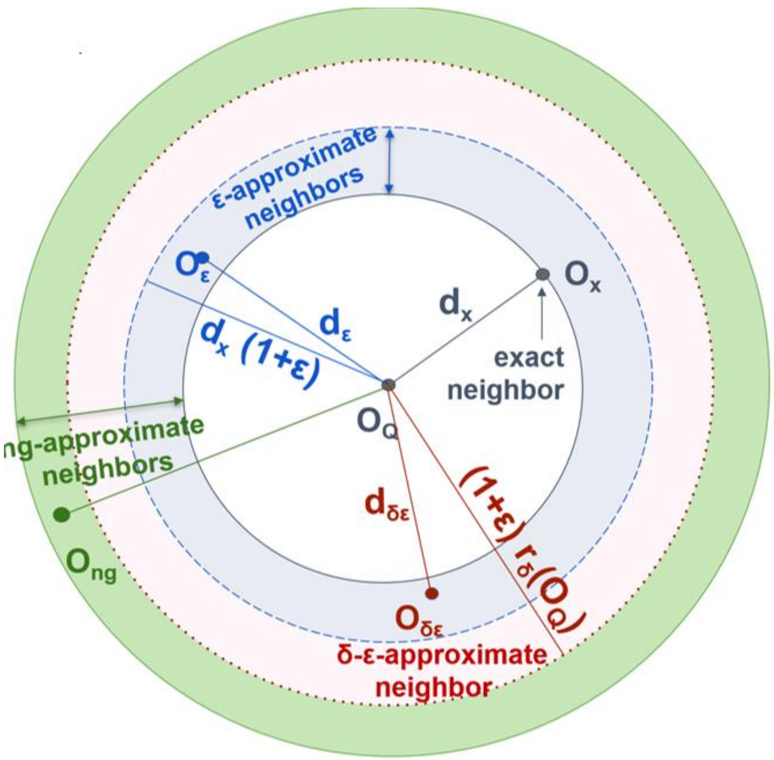


$d_\varepsilon <= d_x$ (1+ε)

Result is within distance (1+ ε) of the exact answer

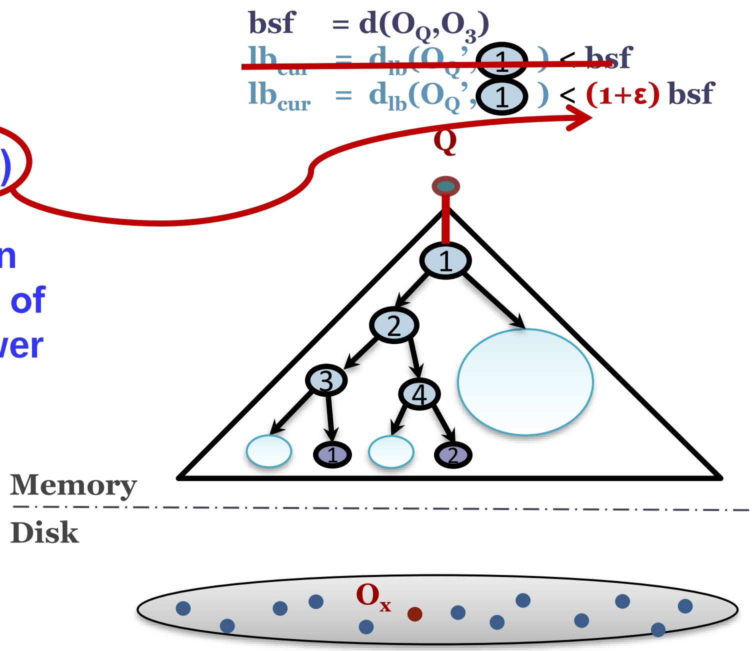


Extensions: Indexes

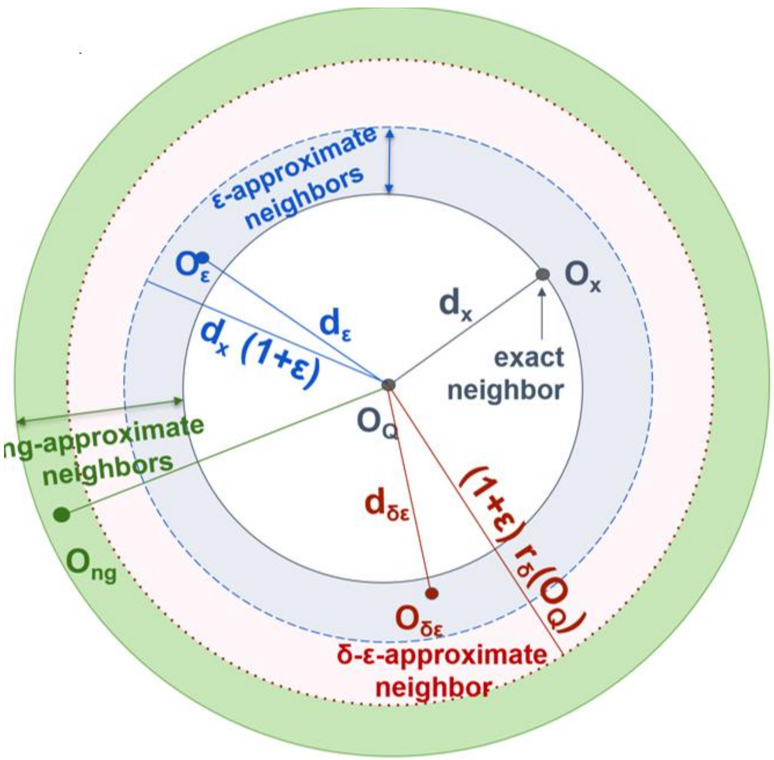


$d_\varepsilon \le d_x (1+\varepsilon)$

Result is within distance (1+ ε) of the exact answer

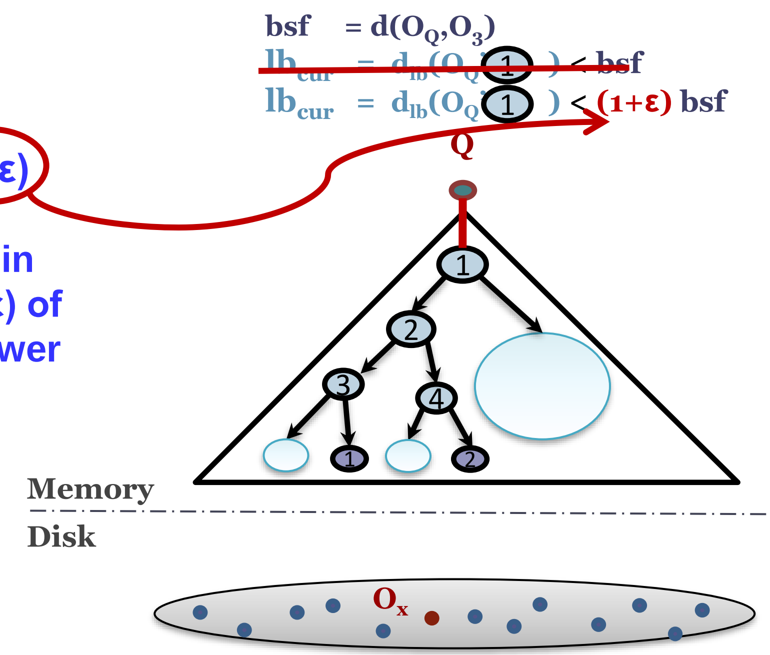


Extensions: Indexes

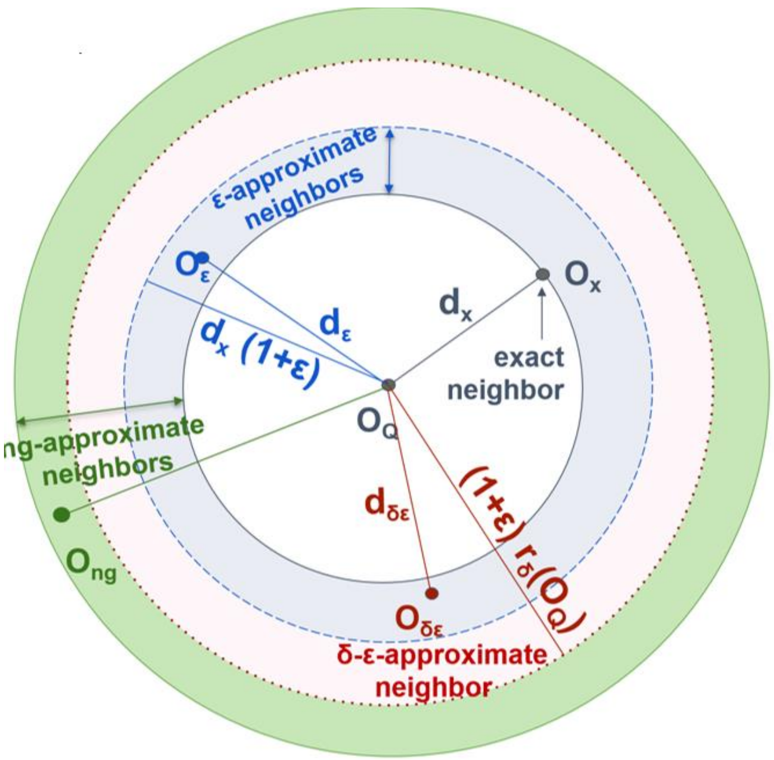


d_ε <= $d_x(1+\varepsilon)$

Result is within distance (1+ ε) of the exact answer

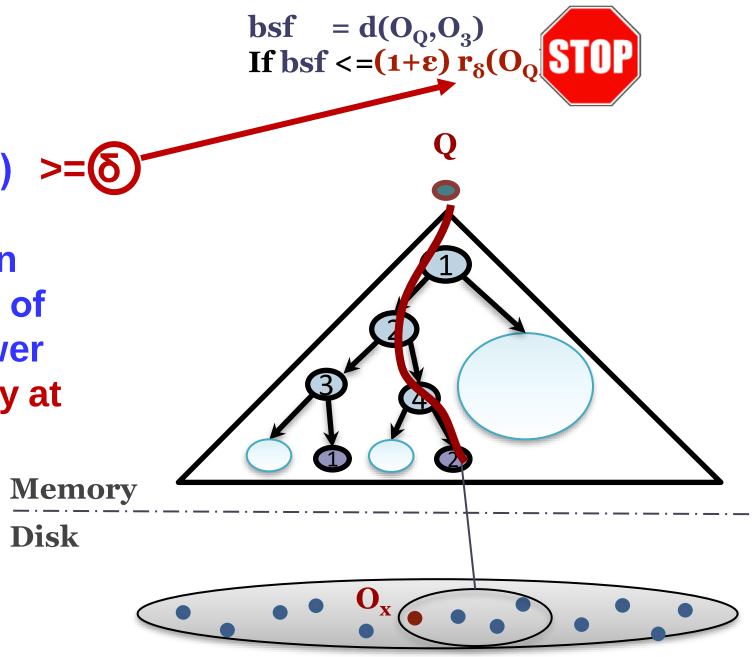


Extensions: Indexes



$P\{d_\varepsilon <= d_x\ (1+\varepsilon)\ >= \delta$

Result is within distance (1+ ε) of the exact answer with probability at least δ



Questions?

Data Series Similarity Search
State-of-the-Art Methods

Publications

Echihabi-EDBT'21

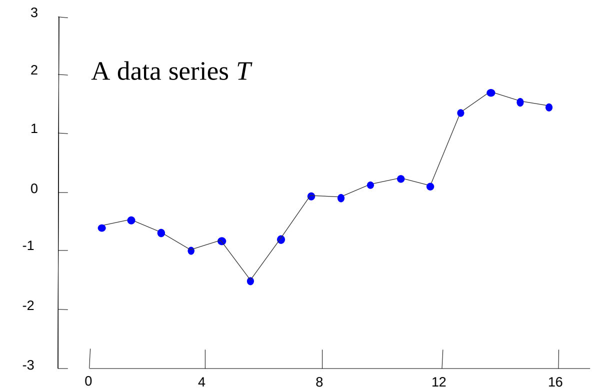
Data Series Similarity Search State-of-the-Art Methods

for a more complete and detailed presentation, see tutorial:

Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas. Big Sequence Management: Scaling Up and Out. EDBT 2021
http://helios.mi.parisdescartes.fr/~themisp/publications.html#tutorials

iSAX Summarization

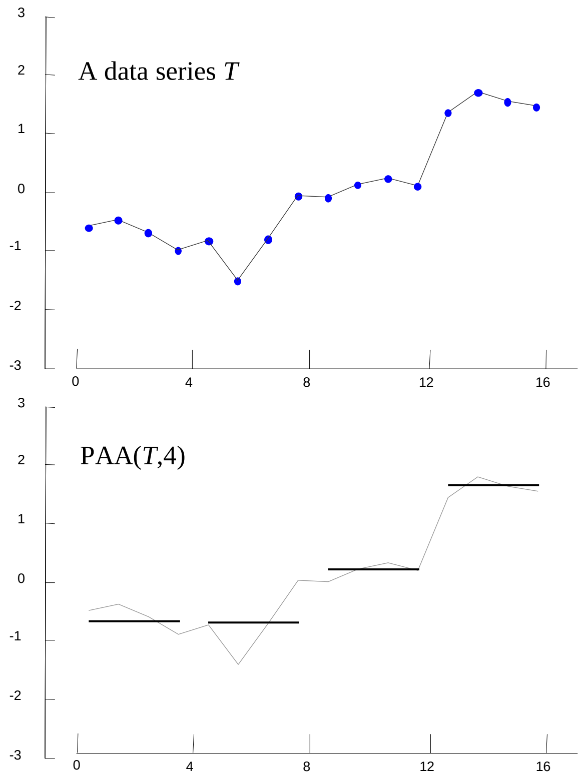
- **S**ymbolic **A**ggregate appro**X**imation (SAX)



iSAX Summarization

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

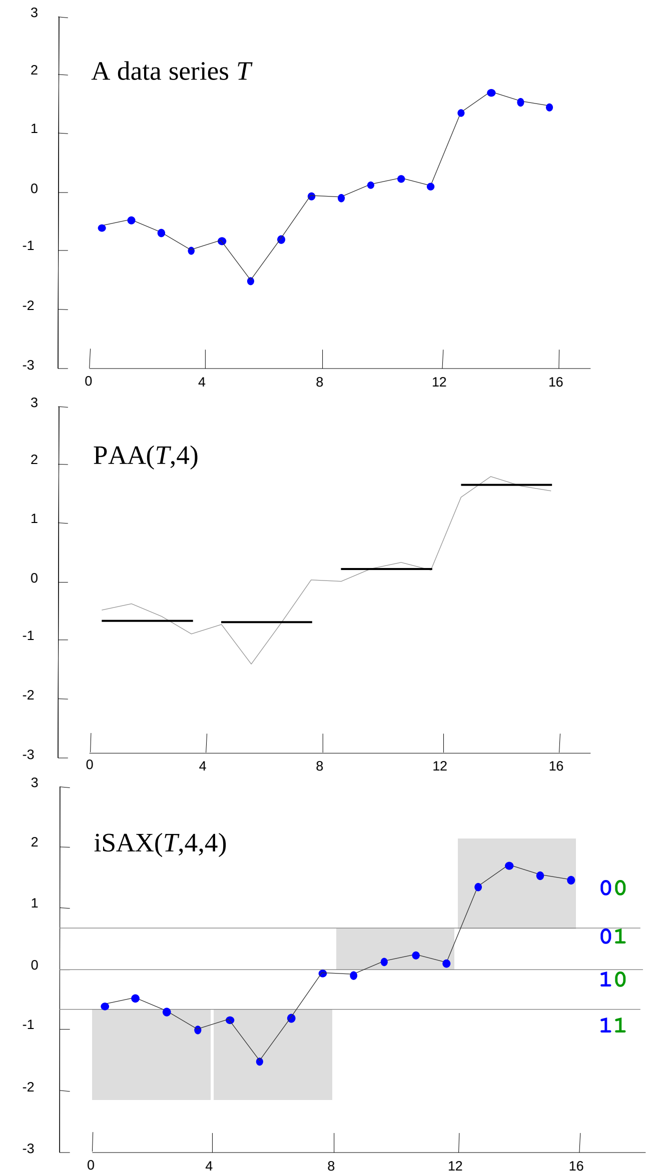
      where $\bar{t}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}i} T_j$



iSAX Summarization

- **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 $\mu = 0, \sigma = 1$
    - $\mathrm{PAA}(T,w) = \overline{T} = \bar{t}_1, \ldots, \bar{t}_w$

      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

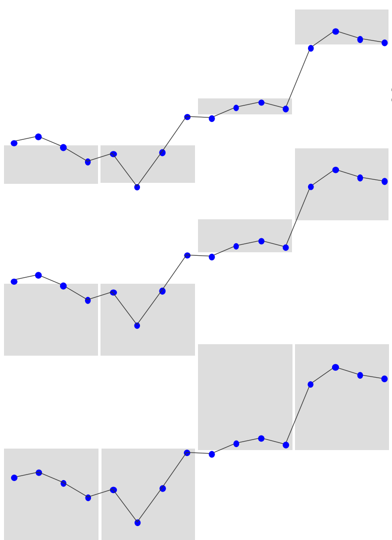


iSAX Summarization

Publications

Shieh-KDD'08

- *i*SAX representation offers a bit-aware, quantized, multi-resolution representation with variable granularity



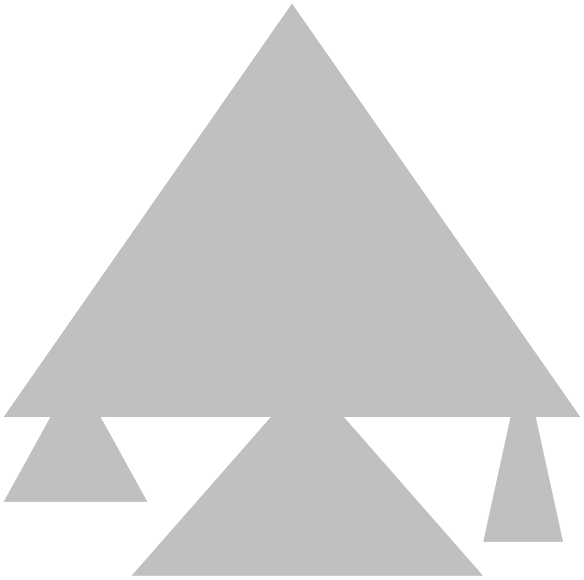
= { 6, 6, 3, 0} = {110 ,110 ,0111 ,000}

= { 3, 3, 1, 0} = {11 ,11 ,011 ,00 }

= { 1, 1, 0, 0} = {1 ,1 ,0 ,0 }

*i*SAX Index Family

- 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***

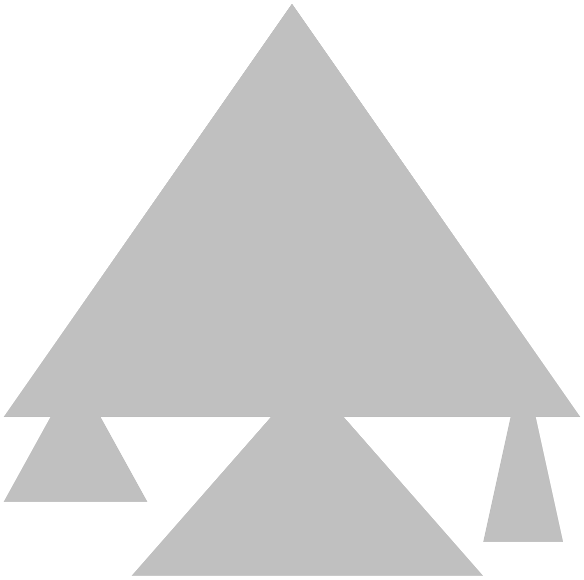


*i*SAX Index Family

- 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***

e.g., th=4, w=4, b=1

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$



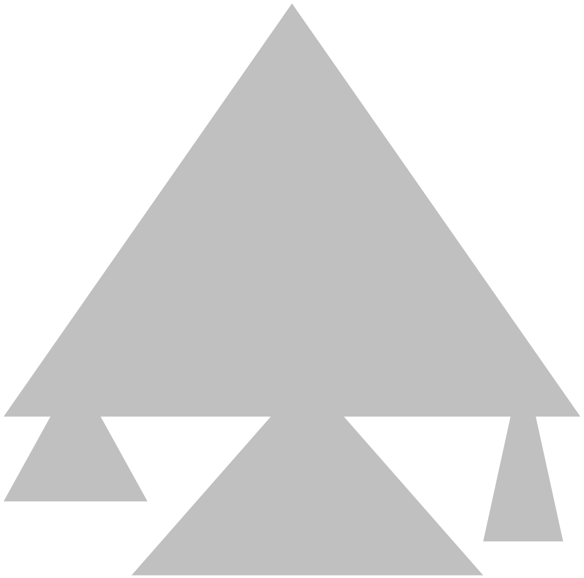
*i*SAX Index Family

- 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***

e.g., th=4, w=4, b=1

Insert:
1 1 1 0

$$\left[\begin{matrix}1 & 1 & 1 & 0\\ 1 & 1 & 1 & 0\\ 1 & 1 & 1 & 0\\ 1 & 1 & 1 & 0\end{matrix}\right.$$



*i*SAX Index Family

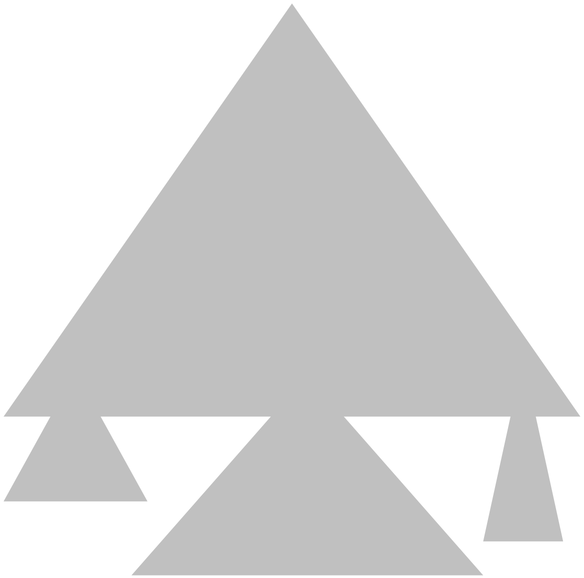
- non-balanced tree-based index with non-overlapping regions, and controlled fan-out rate
  - base cardinality ***b*** (optional), segments ***w***, threshold ***th***
  - hierarchically subdivides SAX space until num. entries ≤ ***th***

e.g., th=4, w=4, b=1

1 1 1**0** 0
1 1 1**0** 0

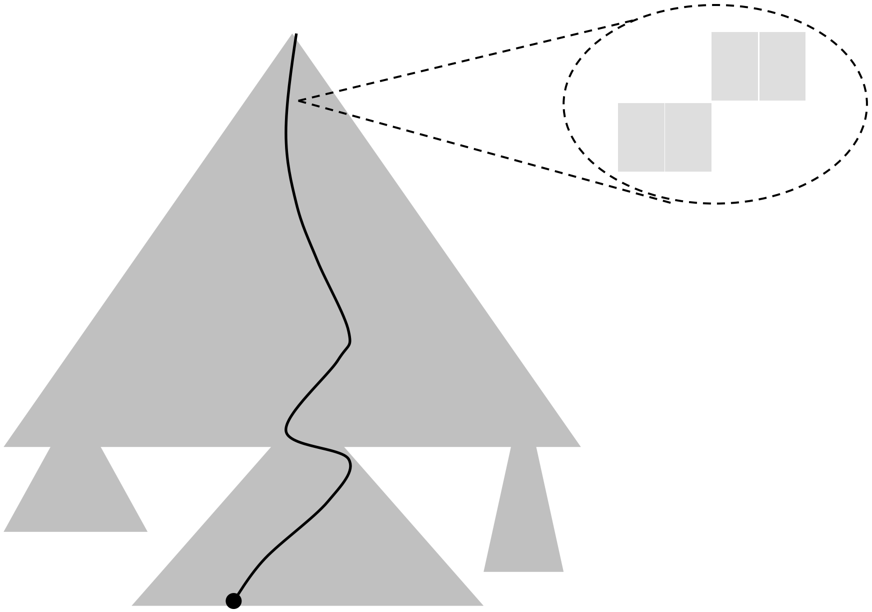
1 1 **1** 0

1 1 1**1** 0
1 1 1**1** 0
1 1 1**1** 0



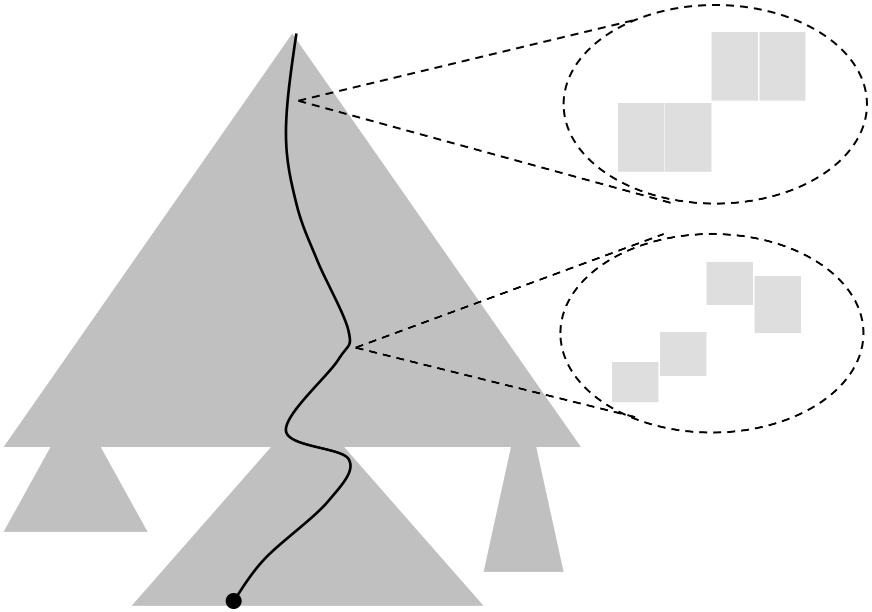
*i*SAX Index Family

- 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***



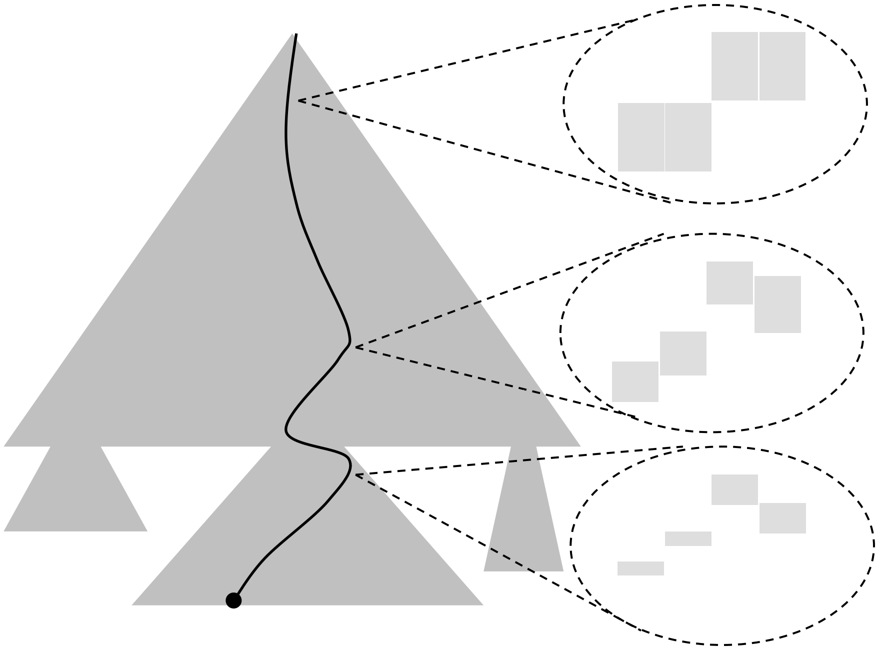
*i*SAX Index Family

- 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***



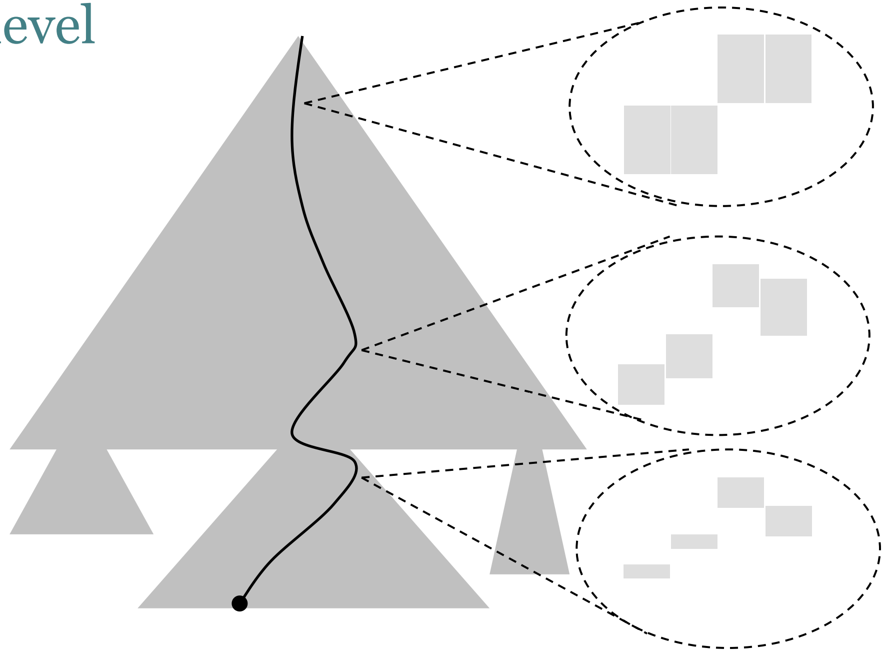
*i*SAX Index Family

- 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 Family

- 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



Publications

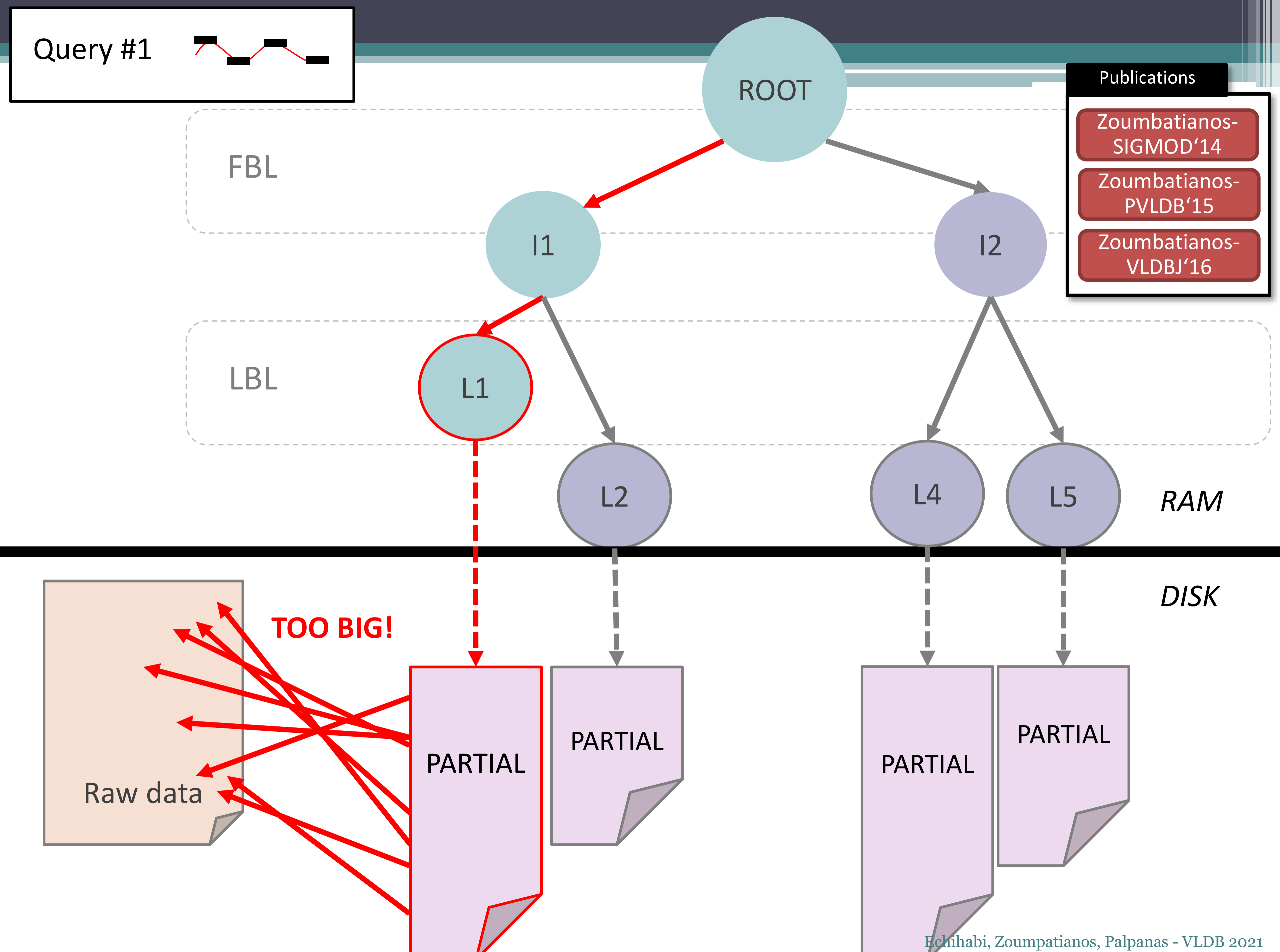
Zoumbatianos-SIGMOD‘14

Zoumbatianos-PVLDB‘15

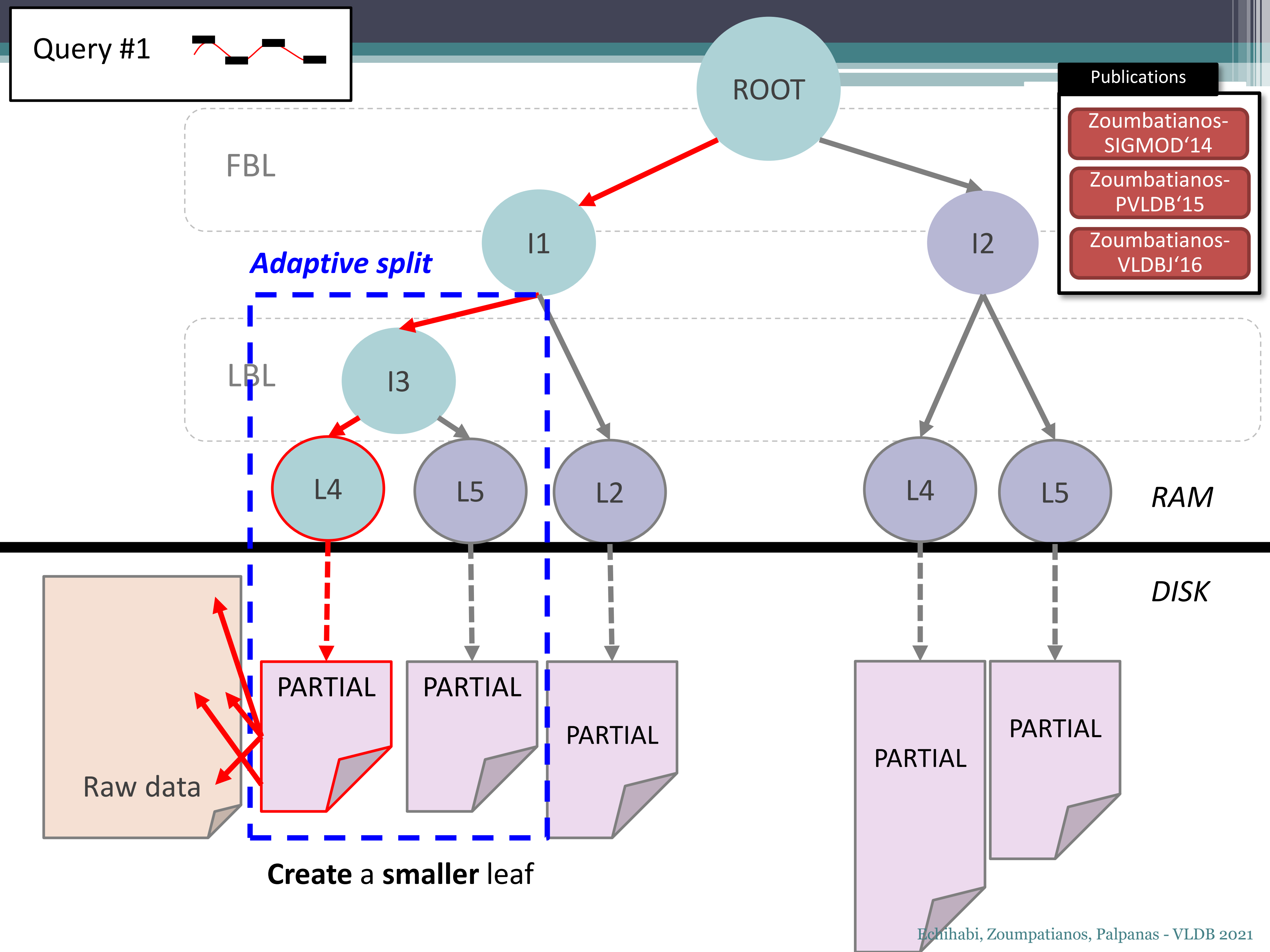
Zoumbatianos-VLDBJ‘16

ADS+

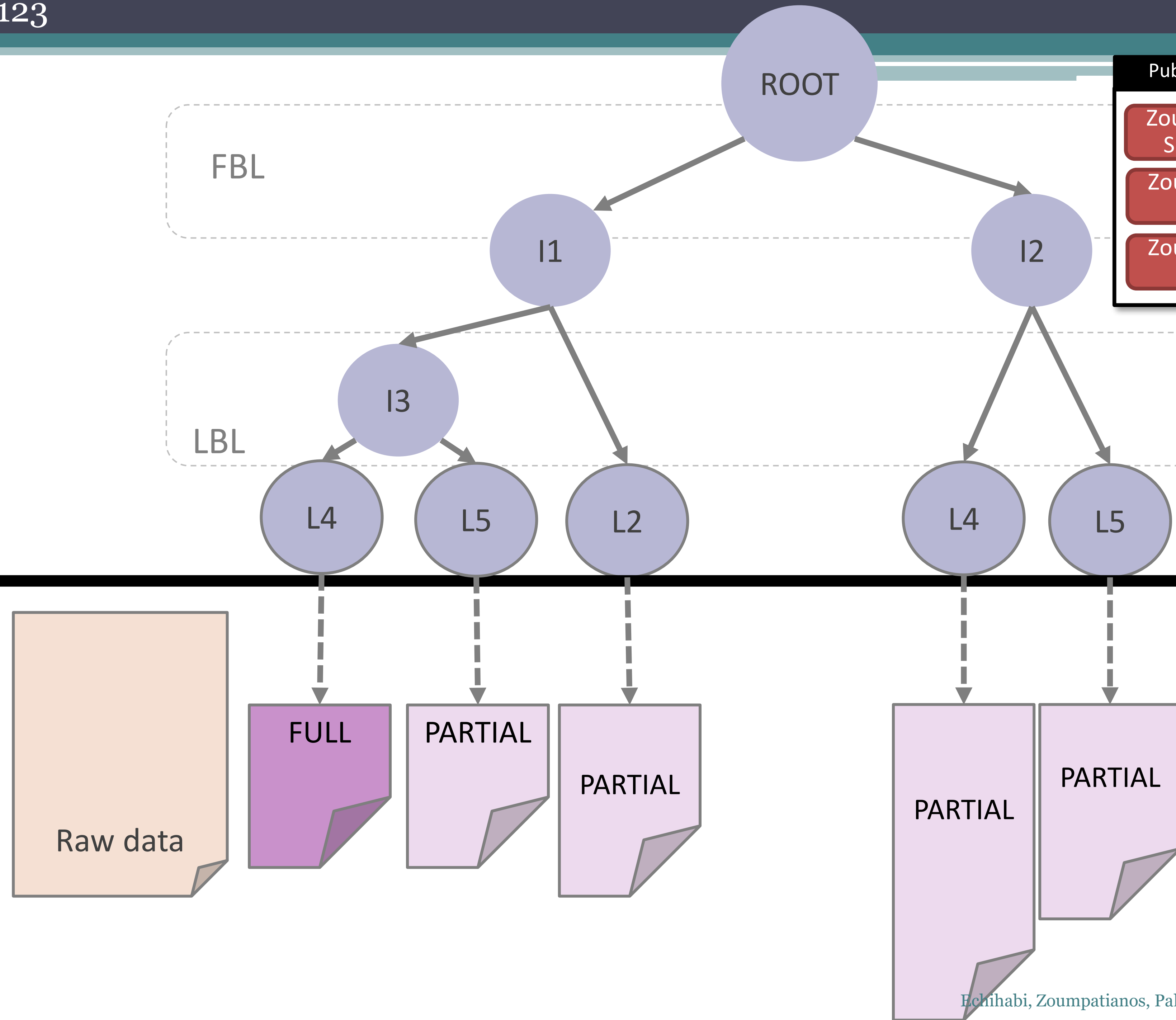
- novel paradigm for building a data series index
  - does not build entire index and then answer queries
  - starts answering queries by building the part of the index needed by those queries
- still guarantees correct answers
- intuition for proposed solution
  - builds index using only *i*SAX summaries; uses large leaf size
  - postpones leaf materialization to query time
    - only materialize (at query time) leaves needed by queries
  - parts that are queried more are refined more
    - use smaller leaf sizes (reduced leaf materialization and query answering costs)



Query #1
ROOT
FBL
I1
I2
LBL
L1
L2
L4
L5
RAM
DISK
Raw data
TOO BIG!
PARTIAL
PARTIAL
PARTIAL
PARTIAL
Publications
Zoumbatianos-SIGMOD'14
Zoumbatianos-PVLDB'15
Zoumbatianos-VLDBJ'16



Create a **smaller** leaf



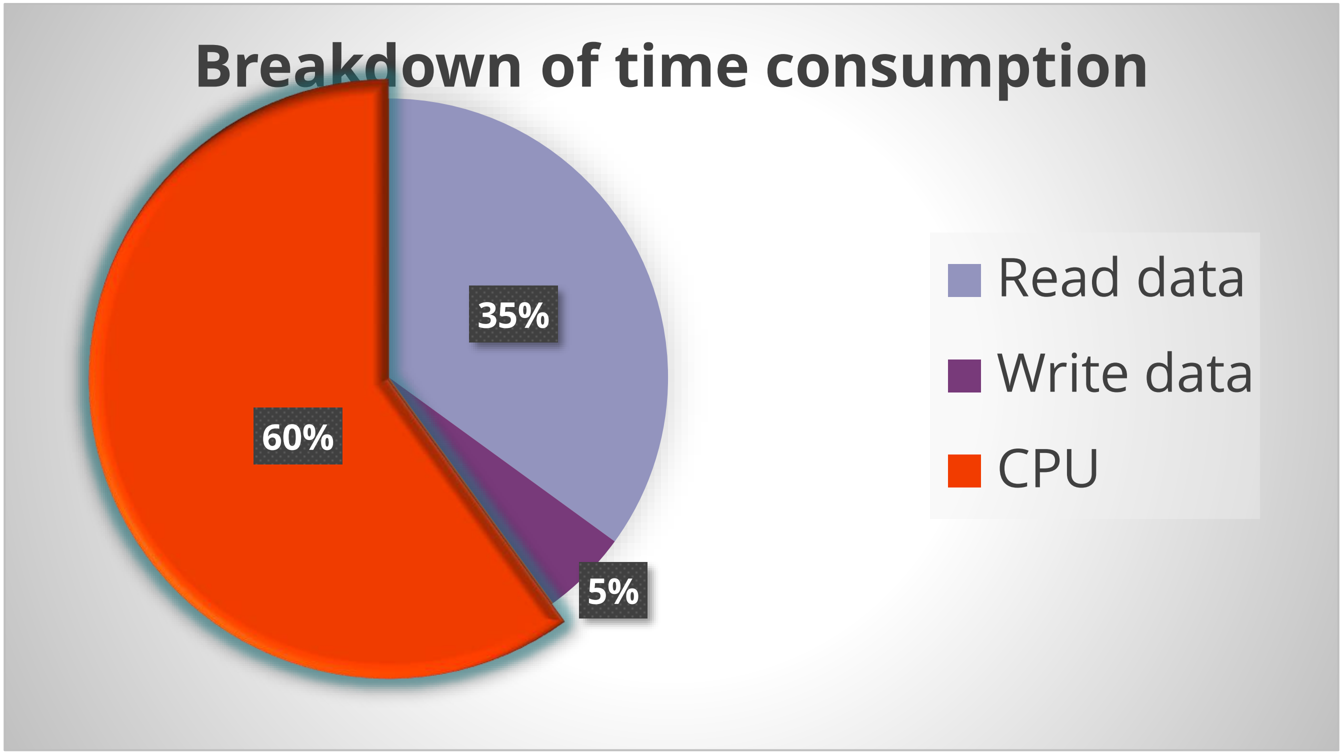
Publications

Zoumbatianos-SIGMOD'14

Zoumbatianos-PVLDB'15

Zoumbatianos-VLDBJ'16

ADS Index creation



~60% of time spent in CPU: potential for improvement!

Publications

- Yagoubi-ICDM'17
- Yagoubi-TKDE'18
- Lavchenko-KAIS'20

Parallelization/Distribution

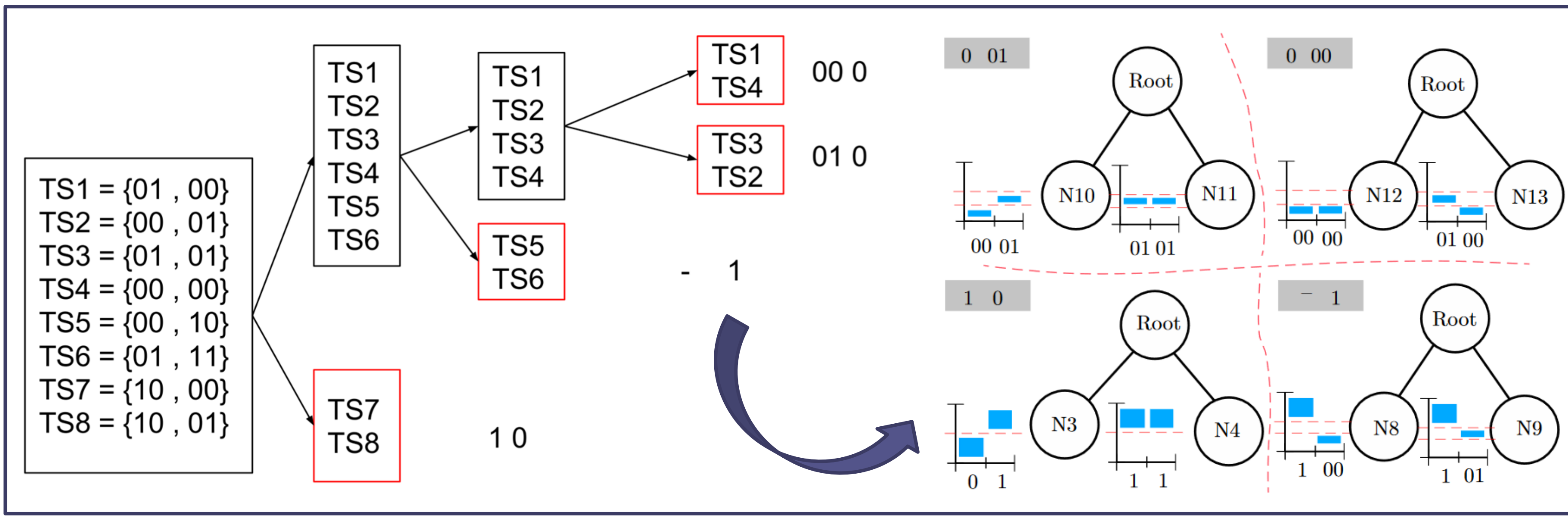
- DPiSAX: current solution for distributed processing (Spark)
  - balances work of different worker nodes

Publications

- Yagoubi-ICDM'17
- Yagoubi-TKDE'18
- Lavchenko-KAIS'20

Parallelization/Distribution

- DPiSAX: current solution for distributed processing (Spark)
  - balances work of different worker nodes



Parallelization/Distribution

Publications

- Yagoubi-ICDM'17
- Yagoubi-TKDE'18
- Lavchenko-KAIS'20

- DPiSAX: current solution for distributed processing (Spark)
  - balances work of different worker nodes
  - performs 2 orders of magnitude faster than centralized solution

Publications

- Yagoubi-ICDM'17
- Yagoubi-TKDE'18
- Lavchenko-KAIS'20
- Peng-BigData'18
- Peng-TKDE'21

Parallelization/Distribution

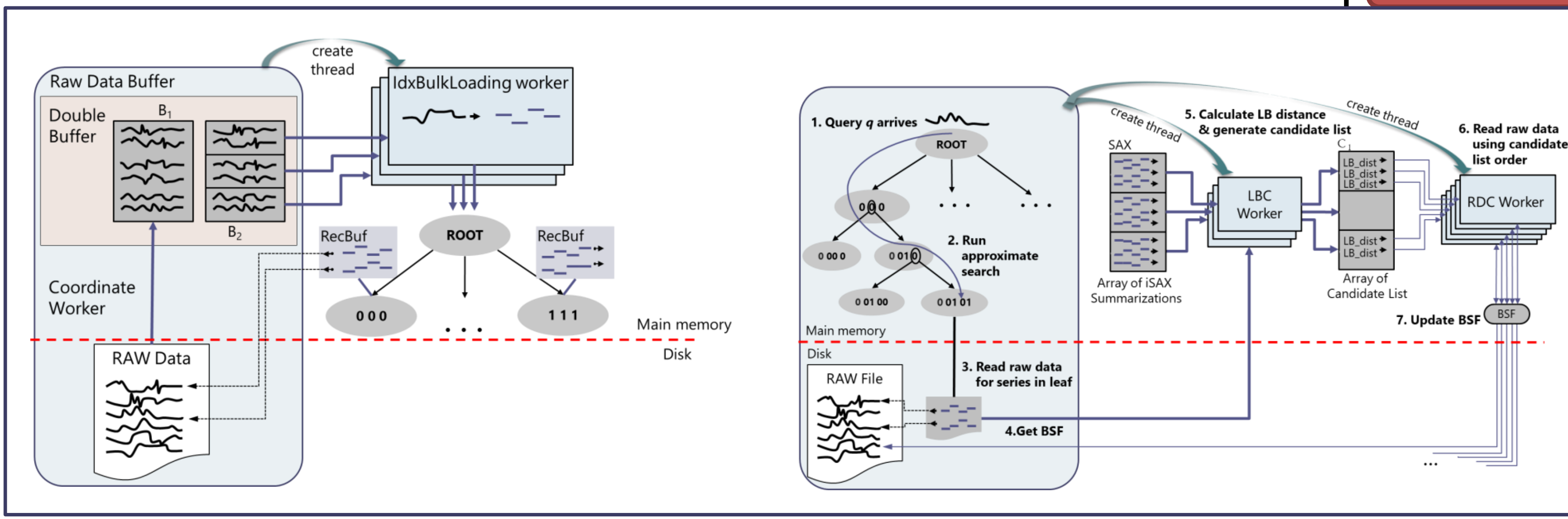
- DPiSAX: current solution for distributed processing (Spark)
  - balances work of different worker nodes
  - performs 2 orders of magnitude faster than centralized soluti
- ParIS+: current solution for modern hardware
  - completely masks out the CPU cost

Publications

- Yagoubi-ICDM'17
- Yagoubi-TKDE'18
- Lavchenko-KAIS'20
- Peng-BigData'18
- Peng-TKDE'21

Parallelization/Distribution

- DPiSAX: current solution for distributed processing (Spark)
  - balances work of different worker nodes
  - performs 2 orders of magnitude faster than centralized soluti



Publications

- Yagoubi-ICDM'17
- Yagoubi-TKDE'18
- Lavchenko-KAIS'20
- Peng-BigData'18
- Peng-TKDE'21

Parallelization/Distribution

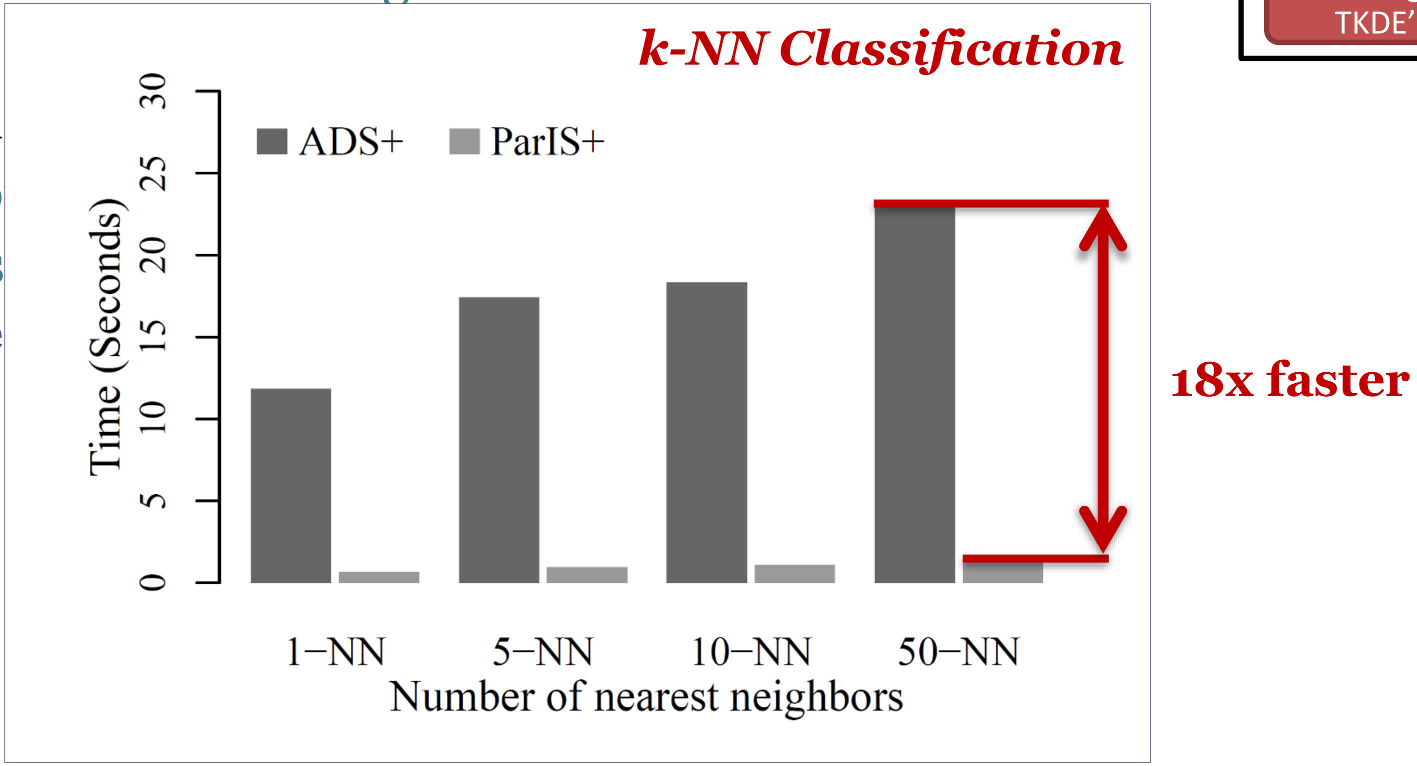
- DPiSAX: current solution for distributed processing (Spark)
  - balances work of different worker nodes
  - performs 2 orders of magnitude faster than centralized soluti
- ParIS+: current solution for modern hardware
  - masks out the CPU cost
  - answers exact queries in the order of a few secs
    - 3 orders of magnitude faster then single-core solutions

Publications

- Yagoubi-ICDM'17
- Yagoubi-TKDE'18
- Lavchenko-KAIS'20
- Peng-BigData'18
- Peng-TKDE'21

Parallelization/Distribution

- DPiSAX: current solution for distributed processing (Spark)
  - balances work of different worker nodes
  - performs 2 orders of magnitude faster than centralized soluti
- ParIS+: cu
  - masks o
  - answers
    - 3 orde

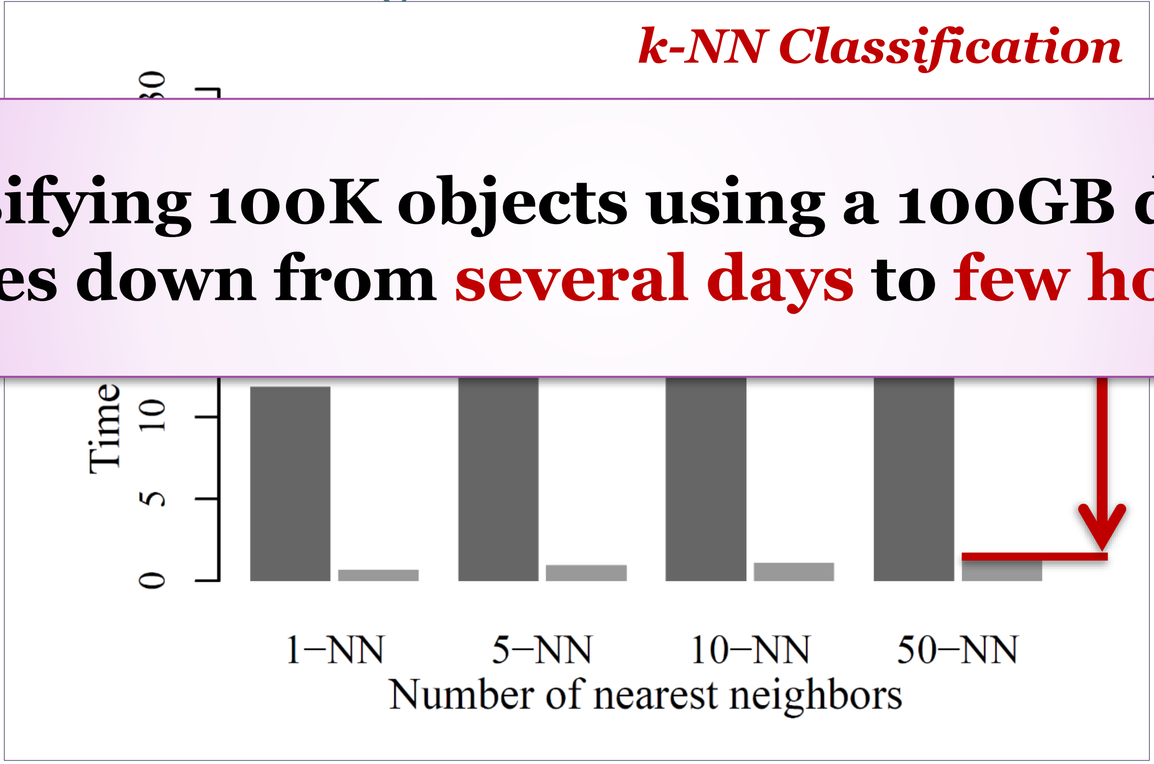


Publications

- Yagoubi-ICDM'17
- Yagoubi-TKDE'18
- Lavchenko-KAIS'20
- Peng-BigData'18
- Peng-TKDE'21

Parallelization/Distribution

- DPiSAX: current solution for distributed processing (Spark)
  - balances work of different worker nodes
  - performs 2 orders of magnitude faster than centralized soluti



- P

classifying 100K objects using a 100GB dataset goes down from several days to few hours!

Publications

- Yagoubi-ICDM'17
- Yagoubi-TKDE'18
- Lavchenko-KAIS'20
- Peng-BigData'18
- Peng-TKDE'21
- Peng-ICDE'20
- Peng-VLDBJ'21

Parallelization/Distribution

- DPiSAX: current solution for distributed processing (Spark)
  - balances work of different worker nodes
  - performs 2 orders of magnitude faster than centralized soluti
- ParIS+: current single-node parallel solution
  - masks out the CPU cost
  - answers exact queries in the order of a few secs
    - >1 order of magnitude faster then single-core solutions
- MESSI: current single-node parallel solution + in-memory data
  - answers exact queries at interactive speeds: ~50msec on 100GB

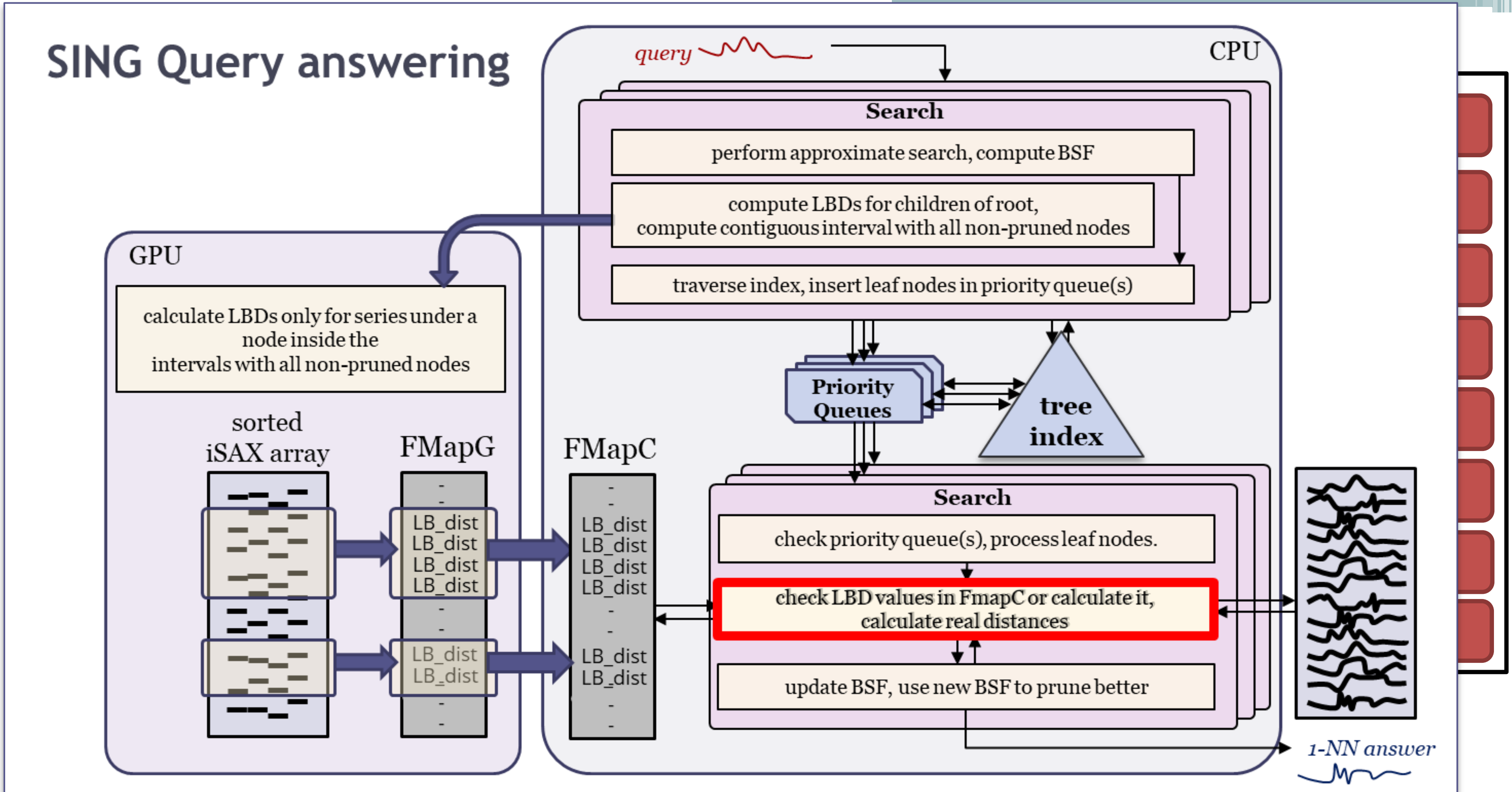
Parallelization/Distribution

Publications

- Yagoubi-ICDM'17
- Yagoubi-TKDE'18
- Lavchenko-KAIS'20
- Peng-BigData'18
- Peng-TKDE'21
- Peng-ICDE'20
- Peng-VLDBJ'21
- Peng-ICDE'21

- DPiSAX: current solution for distributed processing (Spark)
  - balances work of different worker nodes
  - performs 2 orders of magnitude faster than centralized soluti
- ParIS+: current single-node parallel solution
  - masks out the CPU cost
  - answers exact queries in the order of a few secs
    - >1 order of magnitude faster then single-core solutions
- MESSI: current single-node parallel solution + in-memory data
  - answers exact queries at interactive speeds: ~50msec on 100GB
- SING: current single-node parallel solution + GPU + in-memory data
  - answers exact queries at interactive speeds: ~32msec on 100GB

SING Query answering

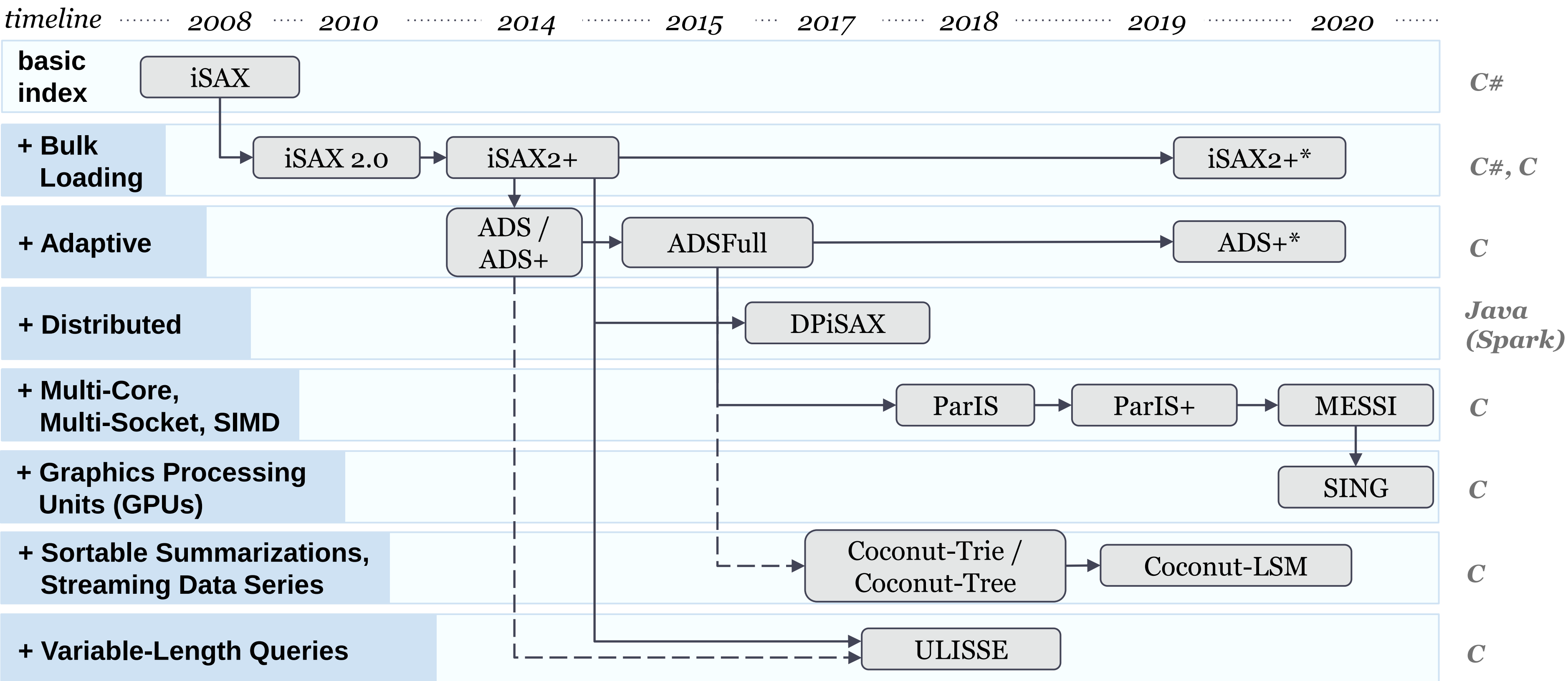


- SING: current single-node parallel solution + GPU + in-memory data
  - answers exact queries at interactive speeds: ~32msec on 100GB

iSAX Index Family

Publications

Palpanas-ISIP'19

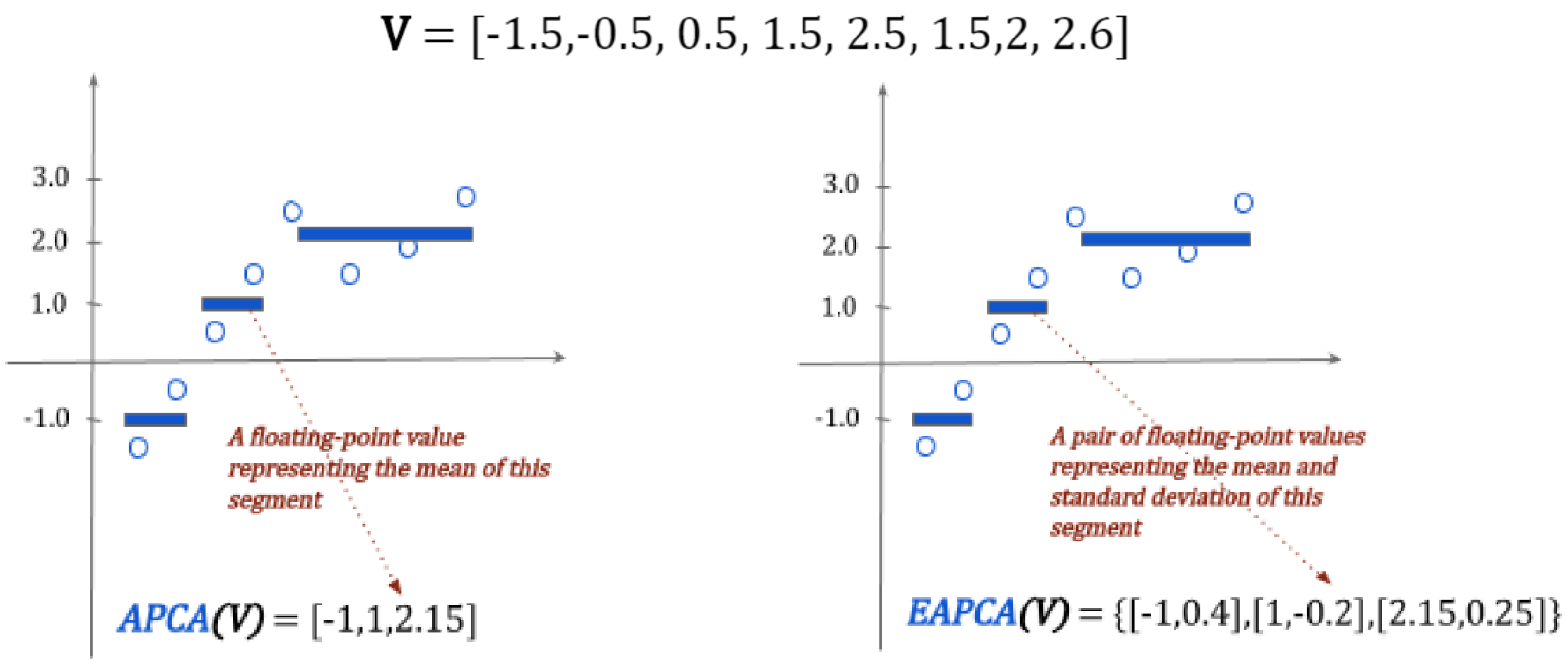


Timeline depicted on top; implementation languages marked on the right. Solid arrows denote inheritance of index design; dashed arrows denote inheritance of some of the design features; two new versions of iSAX2+/ADS+ marked with asterisk support approximate similarity search with deterministic and probabilistic quality guarantees.

DSTree
Summarization

Publications

Wang-PVLDB'13



(a) APCA

(b) EAPCA

Intertwined with indexing

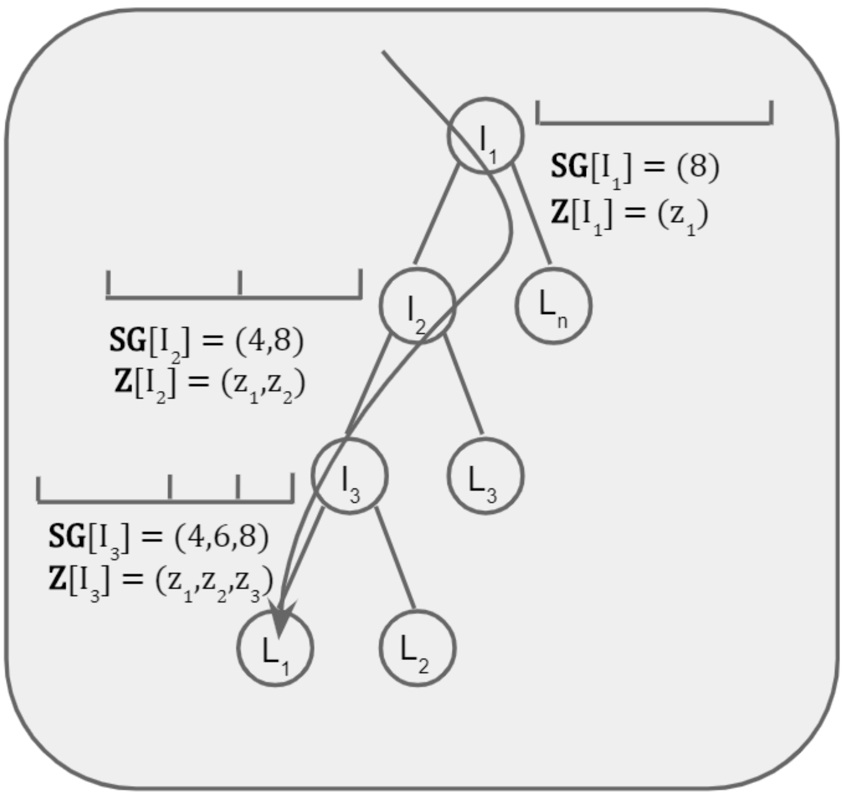
The APCA and EAPCA representations

DSTree
Indexing

Publications

Wang-PVLDB'13

$\mathbf{V} = [-1.5, -0.5, 0.5, 1.5, 2.5, 1.5, 2, 2.6]$



Each node contains
- ❑ # vectors
- ❑ segmentation **SG**
- ❑ synopsis **Z**

Each Leaf node also :
- ❑ stores its raw vectors in a separate disk file

ParSketch

Publications

Cole et al. KDD‘05

Yagoubi et al. DMKD‘18

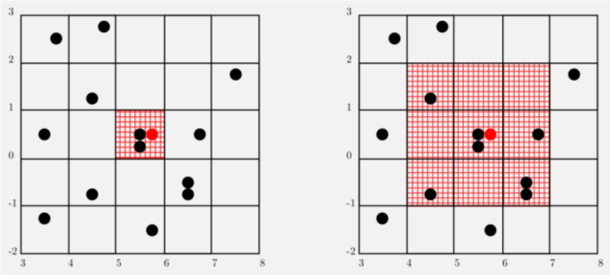
- solution for distributed processing (Spark)
  - represents data series using sketches
    - using a set of random vectors (Johnson-Lindenstrauss lemma)

$$x = (x_1, x_2, x_3, \dots x_n) \quad y = (y_1, y_2, y_3, \dots y_n) \quad z = (z_1, z_2, z_3, \dots z_n)$$

$$R1 = (r_1 1, r_1 2, r_1 3, \dots r_1 w) \quad R2 = (r_2 1, r_2 2, r_2 3, \dots r_2 w) \quad R3 = (r_3 1, r_3 2, r_3 3, \dots r_3 w) \quad R4 = (r_4 1, r_4 2, r_4 3, \dots r_4 w)$$

$$\Rightarrow (xsk1, xsk2, xsk3, xsk4) \quad (ysk1, ysk2, ysk3, ysk4) \quad (zsk1, zsk2, zsk3, zsk4)$$

  - define groups of dimensions in sketches
  - store the values of each group in a grid (in parallel)
    - each grid is kept by a node



node 1 **node 2**

  - for ng-approximate query answering (originally proposed for ε-range queries)
    - find in the grids time series that are close to the query
    - finally, check the real similarity of candidates to find the results
  - performs well for high-frequency series

Publications

- Zhang-ICDE'19
- Wu-ICDE'19
- Feng-IEEE Access'20
- Echihabi-EDBT'21

- other techniques, not covered here:
  - TARDIS
  - KV-Match (subsequence matching)
  - L-Match (subsequence matching)

- for a more complete and detailed presentation, see tutorial:
  - *Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas. Big Sequence Management: Scaling Up and Out. EDBT 2021*

Questions?

High-d Vector Similarity Search State-of-the-Art Methods

High-d Vector Similarity Search Methods

- Tree-Based Methods
- Hash-Based Methods
- Quantization-Based Methods
- Graph-Based Methods

High-d Vector Similarity Search State-of-the-Art Methods

Tree-Based Methods

Tree-Based Methods

Publications

Bentley CACM'75

- A large body of work
- Some representative methods:
  - KD-tree

Tree-Based Methods

Publications

Bentley CACM'75

Silpa-Anan CVPR'08

- A large body of work
- Some representative methods:
  - KD-tree
  - Randomized KD-tree

Tree-Based Methods

Publications

- Bentley CACM'75
- Silpa-Anan CVPR'08
- Muja et al. VISAPP'09

- A large body of work
- Some representative methods:
  - KD-tree
  - Randomized KD-tree
  - FLANN

Tree-Based Methods

Publications

- Bentley CACM'75
- Silpa-Anan CVPR'08
- Muja et al. VISAPP'09
- Ciaccia et al. VLDB'97
- Ciaccia et al. ICDE'00

- A large body of work
- Some representative methods:
  - KD-tree
  - Randomized KD-tree
  - FLANN
  - Mtree

Tree-Based Methods

- A large body of work
- Some representative methods:
  - KD-tree
  - Randomized KD-tree
  - FLANN
  - Mtree
  - HD-Index

Publications

- Bentley CACM'75
- Silpa-Anan CVPR'08
- Muja et al. VISAPP'09
- Ciaccia et al. VLDB'97
- Ciaccia et al. ICDE'00
- Arora et al. PVLDB'18

Tree-Based Methods

Publications

- Bentley CACM'75
- Silpa-Anan CVPR'08
- Muja et al. VISAPP'09
- Ciaccia et al. VLDB'97
- Ciaccia et al. ICDE'00
- Arora et al. PVLDB'18
- Echihabi et al. ICDE'21

- A large body of work
- Some representative methods:
  - KD-tree
  - Randomized KD-tree
  - FLANN
  - Mtree
  - HD-Index

- for a more complete and detailed presentation, see tutorial:
  - *Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas. High-Dimensional Similarity Search for Scalable Data Science. ICDE 2021*

High-d Vector Similarity Search State-of-the-Art Methods

Hash-Based Methods

Locality Sensitive Hashing (LSH)

Publications

Indyk et al. STOC'98

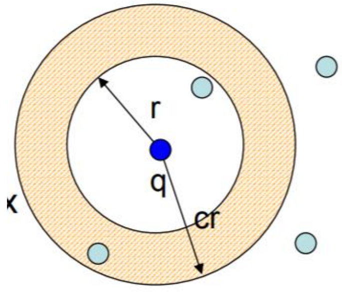
- Solution for δ-ε-approximate kNN search $\delta < 1$
- Random projections into a lower dimensional space using hashing
- Probability of collisions increases with locality

- c-Approximate r-Near Neighbor: build data structure which, for any query q:
  - If there is a point $p \in P$, $||p-q|| \le r$ Then return $p' \in P$, $||p-q|| \le c\,r$

- c-approximate nearest neighbor reduces to c-approximate near neighbor
  - Enumerate all approximate near neighbors

- Find a vector in a preprocessed set $S \subseteq \{0, 1\}\ d$ that has minimum Hamming distance to a query vector $y \in \{0, 1\}\ d$

(r_1, r_2, p_1, p_2)-sensitive [IM98]
- $\Pr[\, h(x) = h(y) \,] \ge p_1$, if $dist(x, y) \le r_1$
- $\Pr[\, h(x) = h(y) \,] \le p_2$, if $dist(x, y) \ge r_2$



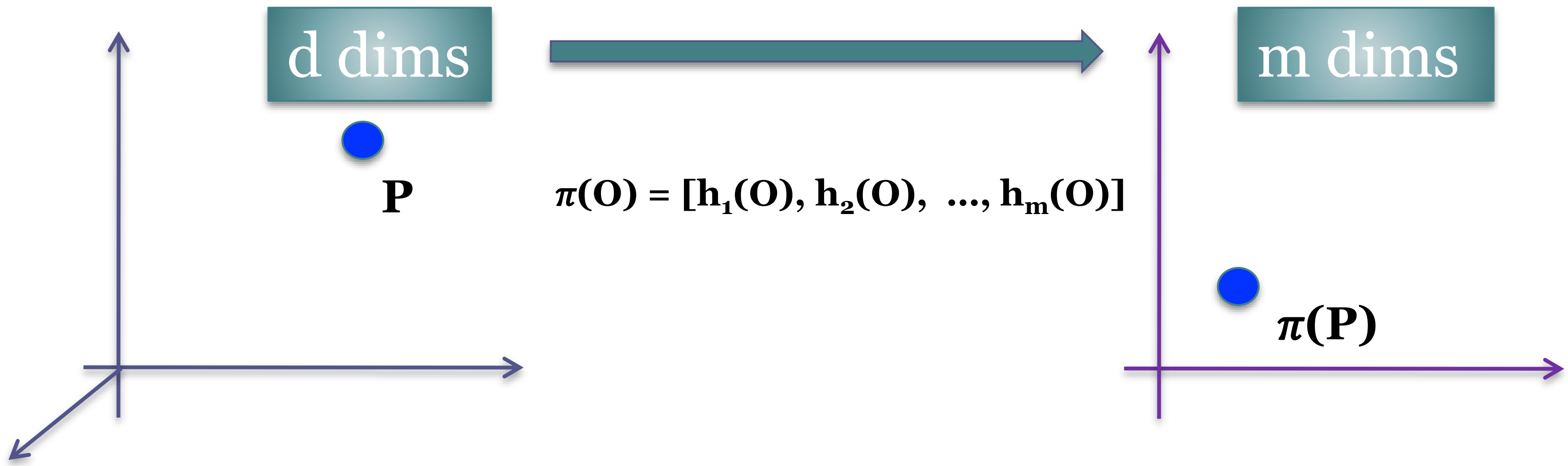
Locality Sensitive Hashing (LSH)

Publications

Andoni et al. CACM'08

- A large family
  - Different distance measures:
    - Hamming distance
    - L_p ($0 < p \leq 2$): use p-stable distribution to generate the projection vector
    - Angular distance (simHash)
    - Jaccard distance (minhash)
  - Tighter Theoretical Bounds
  - Better query efficiency/smaller index size

Probabilistic Mapping

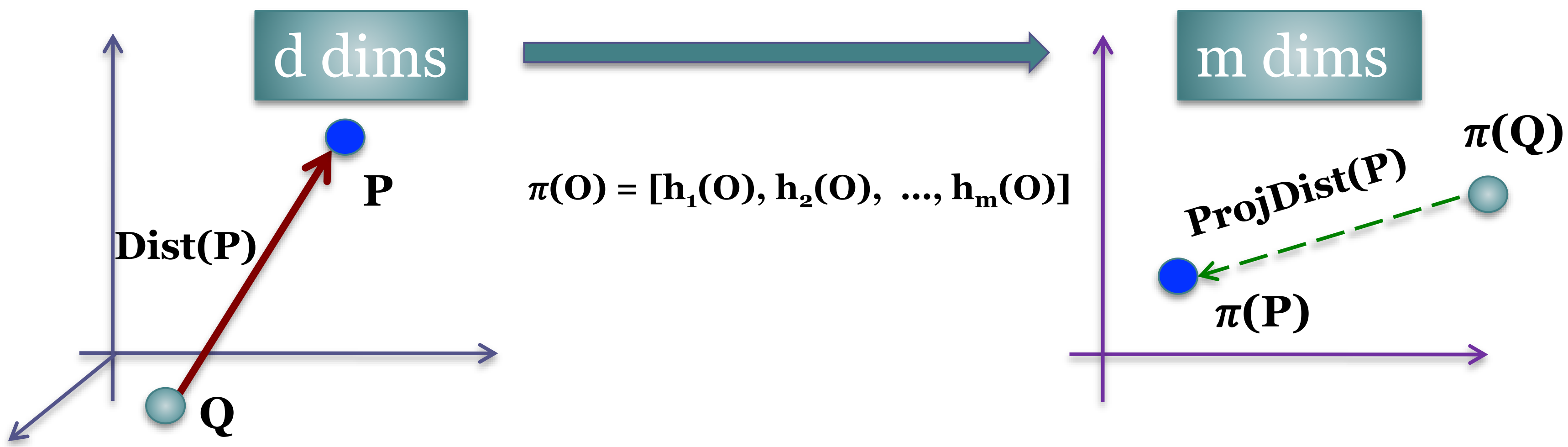


$\pi(O) = [h_1(O), h_2(O), \ldots, h_m(O)]$

- Probabilistic, linear mapping from the original space to the projected space

Slide by W. Wang

Probabilistic Mapping



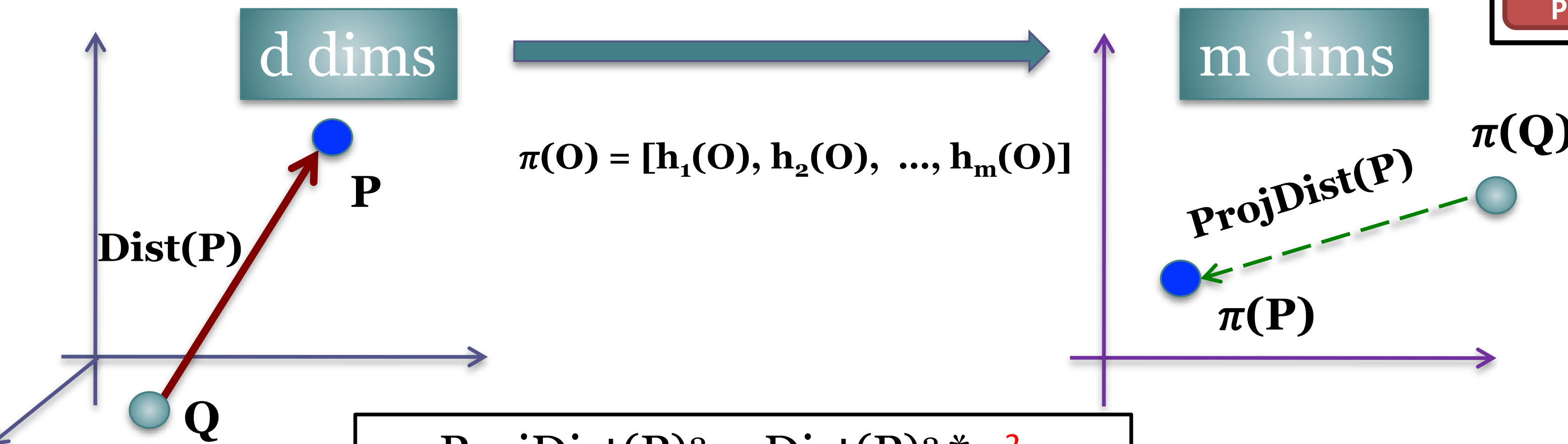
- Probabilistic, linear mapping from the original space to the projected space
- What about the distances (wrt Q or π(Q)) in these two spaces?

Slide by W. Wang

SRS

Publications

Sun et al. PVLDB' 14



$$\text{ProjDist}(P)^2 \sim \text{Dist}(P)^2 * \chi^2_m$$

- Given that ProjDist(P) ≤ r, what can we infer about Dist(P)?
  - If Dist(P) ≤ R, then Pr[ProjDist(P) ≤ r] ≥ $\Psi_m((r/R)^2)$
  - If Dist(P) > cR, then Pr[ProjDist(P) ≤ r] ≤ $\Psi_m((r/cR)^2)$ = t
  - (some probability) at most O(tn) points with ProjDist ≤ R
  - (constant probability) one of the O(tn) points has Dist ≤ R

- This solves the so-called (R, c)-NN queries ➔ returns a c^2 ANN
- Using another algorithm & proof ➔ returns a c-ANN

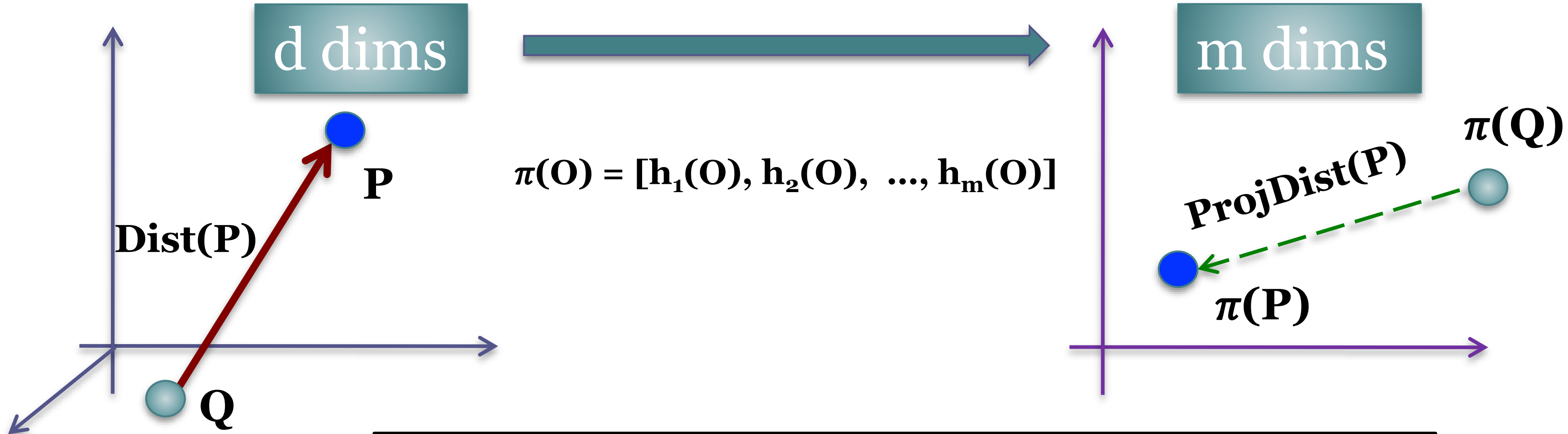
Slide by W. Wang

C2LSH/QALSH

Publications

Huang et al. PVLDB' 15

Gan et al. SIGMOD'12



Collision wrt w: if $|h_i(P) - h_i(Q)| \le w$

- Given that P's #collision $\ge \alpha m$, what can we infer about Dist(P)?
  - If Dist(P) ≤ R, then Pr[#collision $\ge \alpha m$] $\ge \gamma_1$
  - If Dist(P) > cR, then Pr[#collision $\ge \alpha m$] $\le \gamma_2$
  - (some probability) at most O(γ_2*n) points with #collision $\ge \alpha m$
  - (constant probability) one of the O(γ_2*n) points has #collision $\ge \alpha m$

Slide by W. Wang

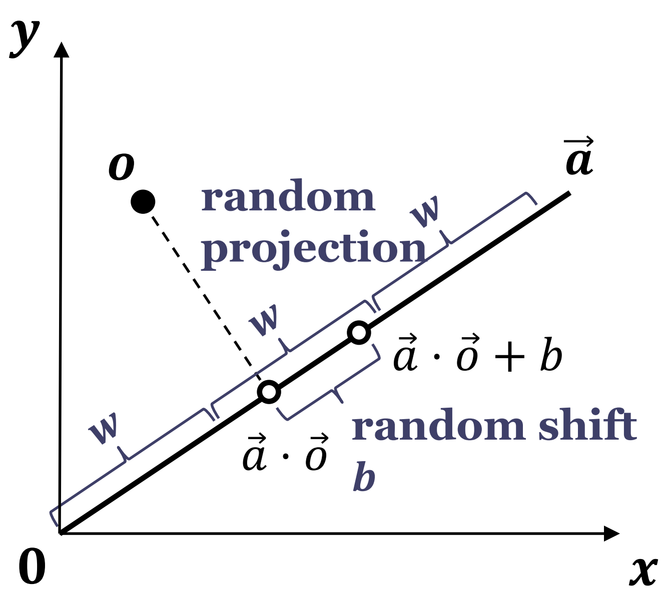
Query-oblivious LSH functions

Publications

Huang et al. PVLDB'15

- **The query-oblivious LSH functions for Euclidean distance:**

$$h_{\vec{a},b}(o) = \left\lfloor \frac{\vec{a}\cdot\vec{o}+b}{w} \right\rfloor$$



Query-Oblivious Bucket Partition:

- Buckets are ***statically*** determined before any query arrives;
- Use the **origin (i.e., "o") as anchor**;
- If $\boldsymbol{h_{\vec{a},b}(o) = h_{\vec{a},b}(q)}$, we say ***o* and *q* collide** under $h_{\vec{a},b}(\cdot)$.

Slide by Q. Huang

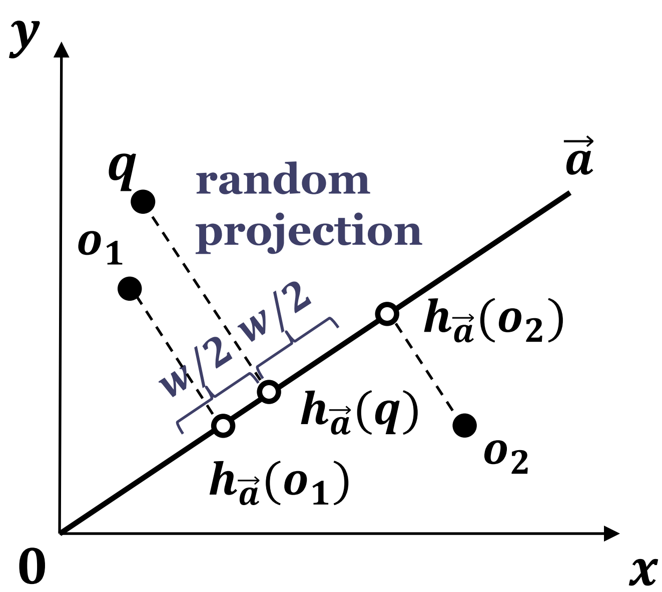
QALSH

Publications

Huang et al. PVLDB' 15

- **Query-aware LSH function = random projection + query-aware bucket partition**

$$h_{\vec{a}}(o) = \vec{a} \cdot \vec{o}$$



Query-Aware Bucket Partition:

- Buckets are **dynamically** determined when q arrives;
- Use "$h_{\vec{a}}(q)$" **as anchor** ;
- If an object o falls into the **anchor bucket**, i.e., $|h_{\vec{a}}(o) - h_{\vec{a}}(q)| \le \frac{w}{2}$, we say **$o$ and q collide** under $h_{\vec{a}}(\cdot)$.

Slide by Q. Huang

VHP

Publications

Lu et al. PVLDB’ 20

- Solution for δ-ε-approximate kNN search
  - Indexing:
    - Store LSH projections with independent B+ trees.
  - Querying
    - Impose a virtual hypersphere in the original high-d space
    - Keep enlarging the virtual hypersphere to accommodate more candidate until the success probability is met

Slide by W. Wang

Some Comparisons

Publications

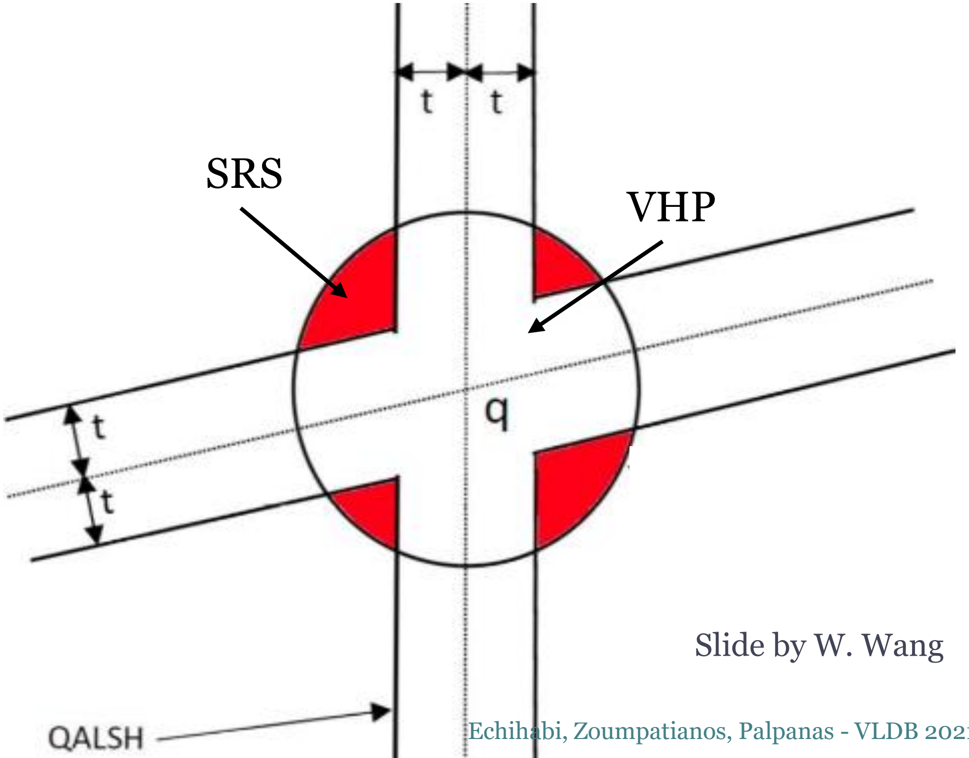
Huang et al. PVLDB' 15

Candidate Conditions

Method	Collision Count	(Observed) Distance	Max Candidates
SRS	= m	≤ r	T
QALSH	≥ αm	n/a	βn
VHP	≥ i (i = 1, 2, …, m)	≤ l_i	βn

Candidate Regions

VHP = SRS ∩ QALSH



Slide by W. Wang

High-d Vector Similarity Search
State-of-the-Art Methods

Quantization-Based Methods

Quantization

- A lossy compression process that maps a set of infinite numbers to a finite set of codewords that together constitute the codebook:
  - Scalar Quantization
    - Operates on the individual dimensions of the original vector independently
  - Vector Quantization
    - Considers the original vector as a whole
  - Product Quantization
    - Splits the original vector of dimension d into m smaller subvectors, on which a lower-complexity vector quantization is performed. The codebook consists of the cartesian product of the codebooks of the m subquantizers.
  - Scalar and vector quantization are special cases of product quantization, where m is equal to d and 1, respectively

VA-file

Publications

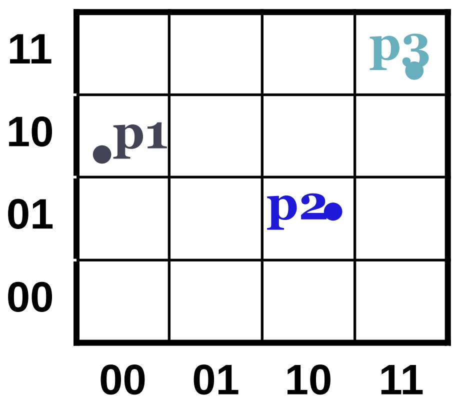
Blott et. al VLDB'98

- A solution for exact kNN search

- The basic idea of the VA-file is to speed-up the sequential scan by exploiting a "Vector Approximation"
- Each dimension of the data space is partitioned into 2^{b_i} intervals using b_i bits (scalar quantization)
  - E.g.: the 1st coordinate uses 2 bits, which leads to the intervals 00,01,10, and 11
- Thus, each coordinate of a point (vector) requires now b_i bits instead of 32

- The VA-file stores, for each point of the dataset, its approximation, which is a vector of $\sum_{i=1,D} b_i$ bits

Data space



Feature values

p1	0.1 0.6
p2	0.7 0.4
p3	0.9 0.3

VA-file

p1	00 10
p2	10 01
p3	11 11

Slides by M. Patella.

VA-file

Publications

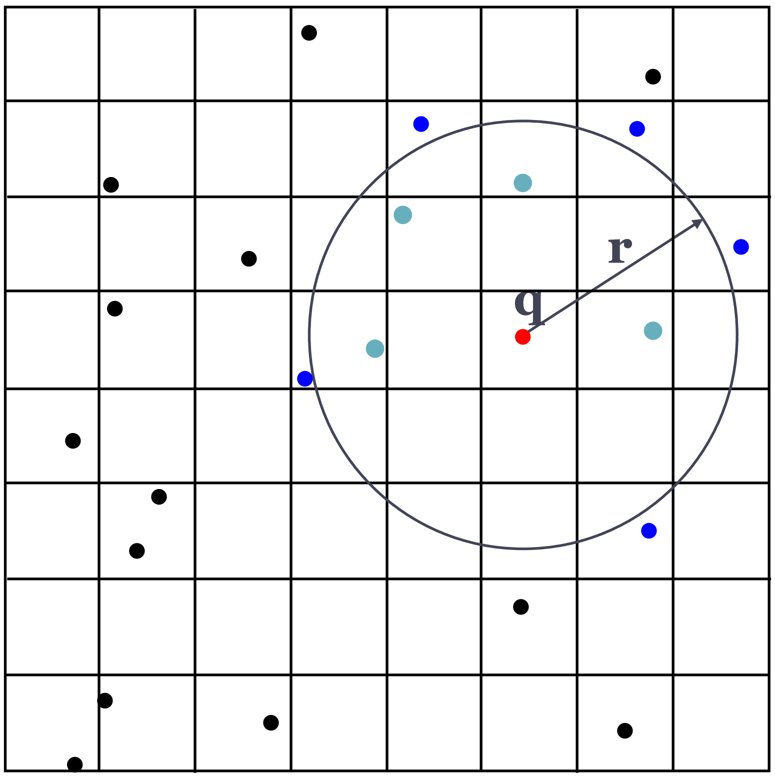
Blott et. al
VLDB'98

- Query processing with the VA-file is based on a filter & refine approach

- For simplicity, consider a range query

  Filter: the VA file is accessed and only the points in the regions that intersect the query region are kept

  Refine: the feature vectors are retrieved and an exact check is made



actual results
false drops
excluded points

Slides by M. Patella.

VA+file

Publications

Ferhatosmanoglu et al.
CIKM'00

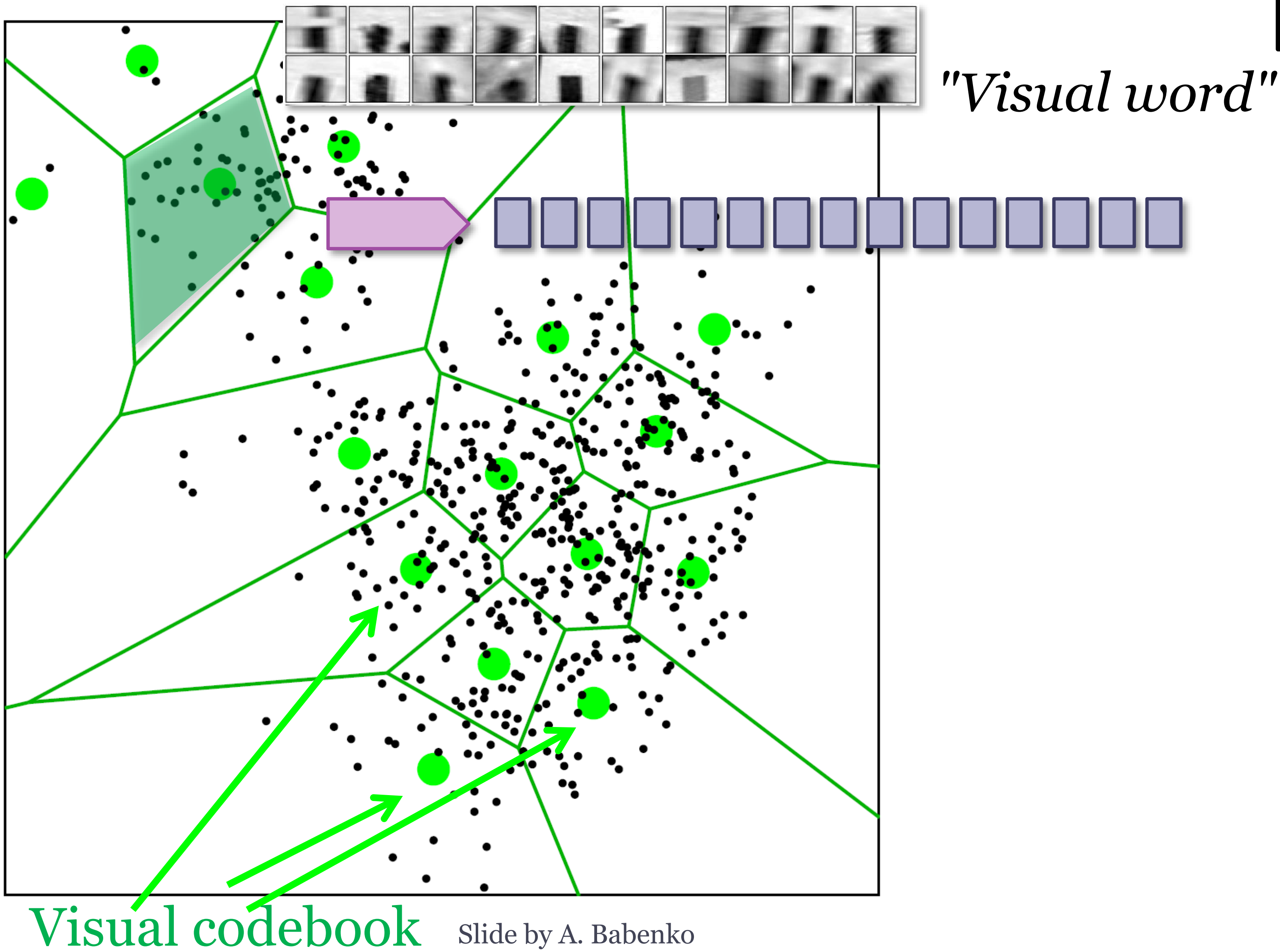
- Solution for exact kNN search

- An improvement of the VA-file method:
  - Does not assume that neighboring dimensions are uncorrelated
  - Decorrelates the data using KLT
  - Allocates bits per dimension in a non-uniform fashion
  - Partitions each dimension using k-means instead of equi-depth

The Inverted Index

Publications

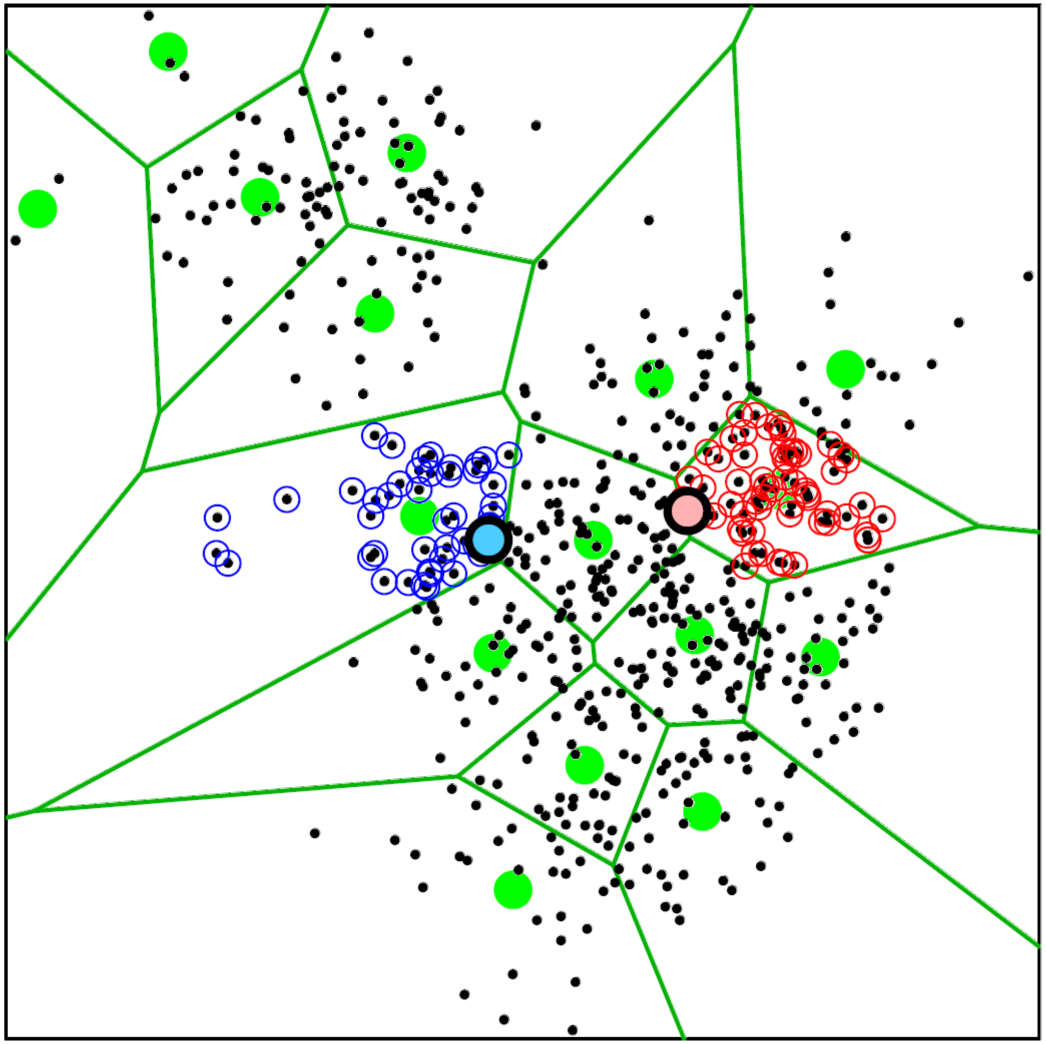
Sivic et al. ICCV' 03



Slide by A. Babenko

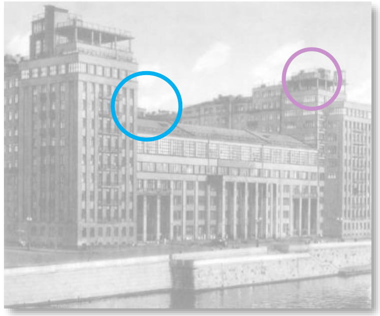
Querying the Inverted Index

Publications
Sivic et al. ICCV' 03



Slide by A. Babenko

Query:



- Have to consider several words for best accuracy
- Want to use as big codebook as possible

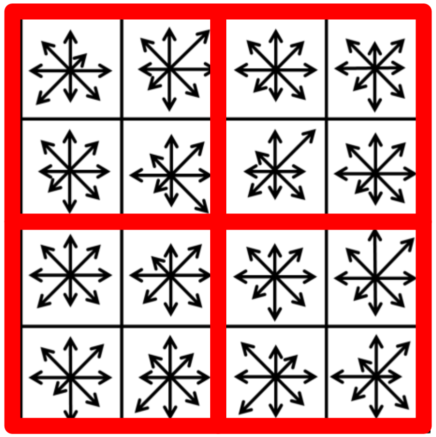
conflict

- Want to spend as little time as possible for matching to codebooks

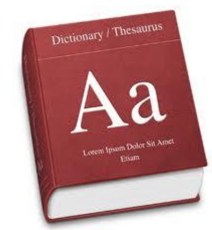
Product Quantization

Publications

Jegou et al. TPAMI' 11



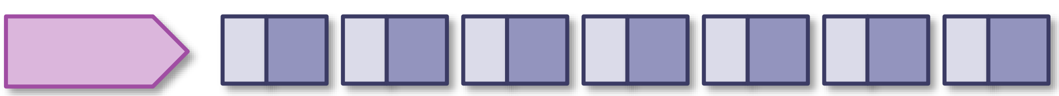
1. Split vector into correlated subvectors
2. use separate small codebook for each chunk



Quantization vs. Product quantization:

For a budget of 4 bytes per descriptor:

1. Can use a single codebook with 1 billion codewords
2. Can use 4 different codebooks with 256 codewords each



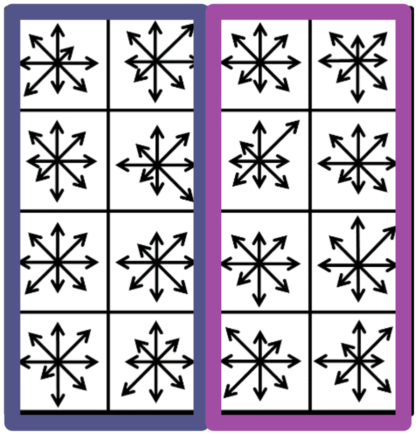
IVFADC+ variants (state-of-the-art for billion scale datasets) =
inverted index for indexing + product quantization for reranking

Slide by A. Babenko

The Inverted Multi-Index

Publications

Babenko et al. TPAMI' 12

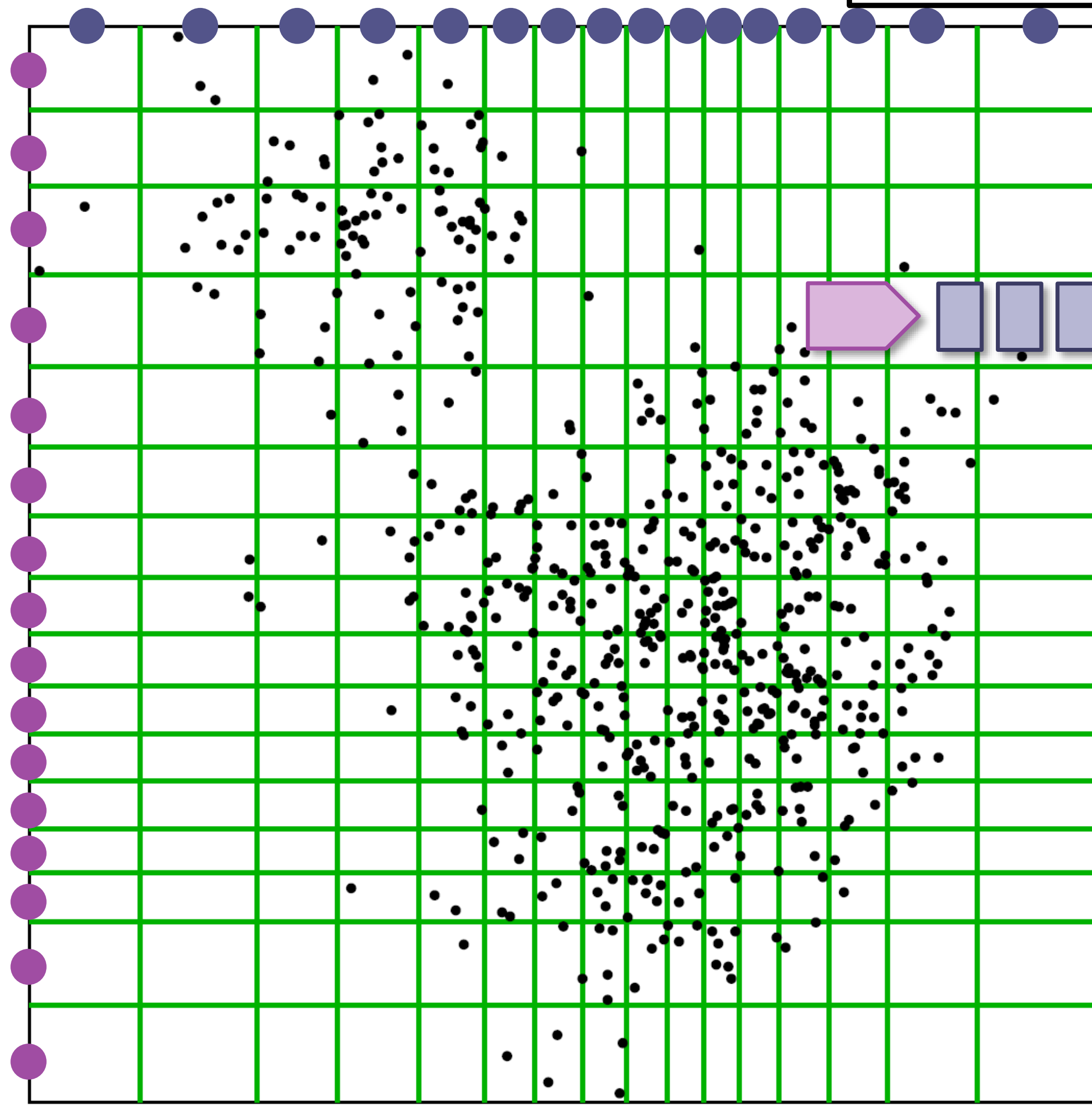


Idea: use product quantization for indexing

Main advantage:
For the same K, much finer subdivision achieved

Main problem:
Very non-uniform entry size distribution

Slide by A. Babenko



Querying the Inverted Multi-Index

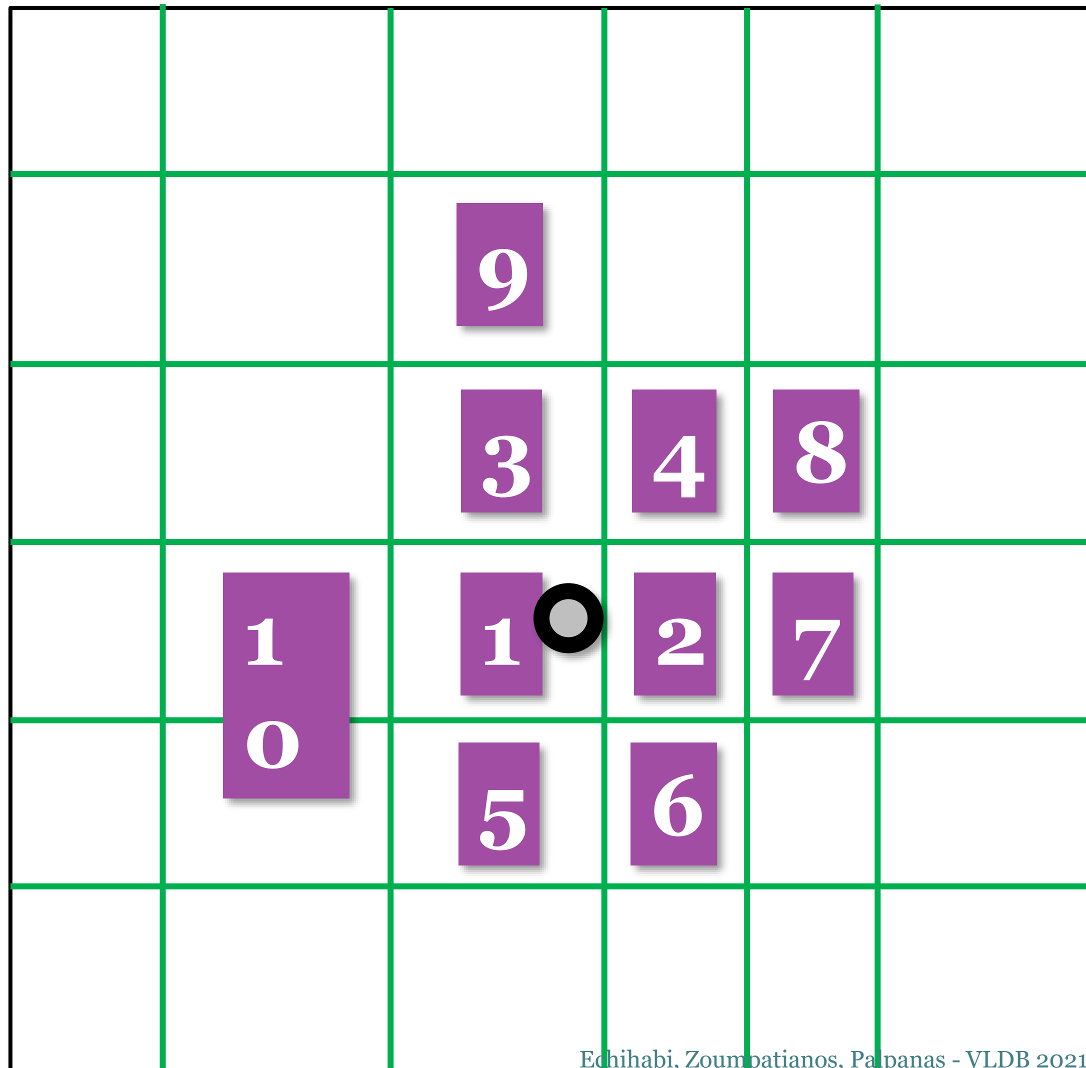
Publications

Babenko et al. TPAMI' 12

Answer to the query:

Input: query

Output: stream of entries



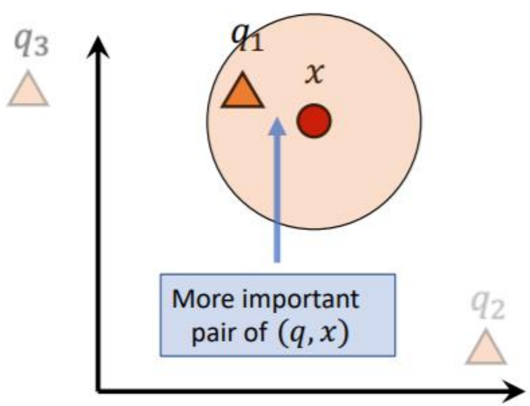
Slide by A. Babenko

Publications

Guo-ICML'20

Google ScaNN

- Quantization-based similarity search using MIPS
  - A novel score-aware loss function:
    - The approximation error on the pairs which have a high inner product is far more important than that of pairs whose inner product is low.

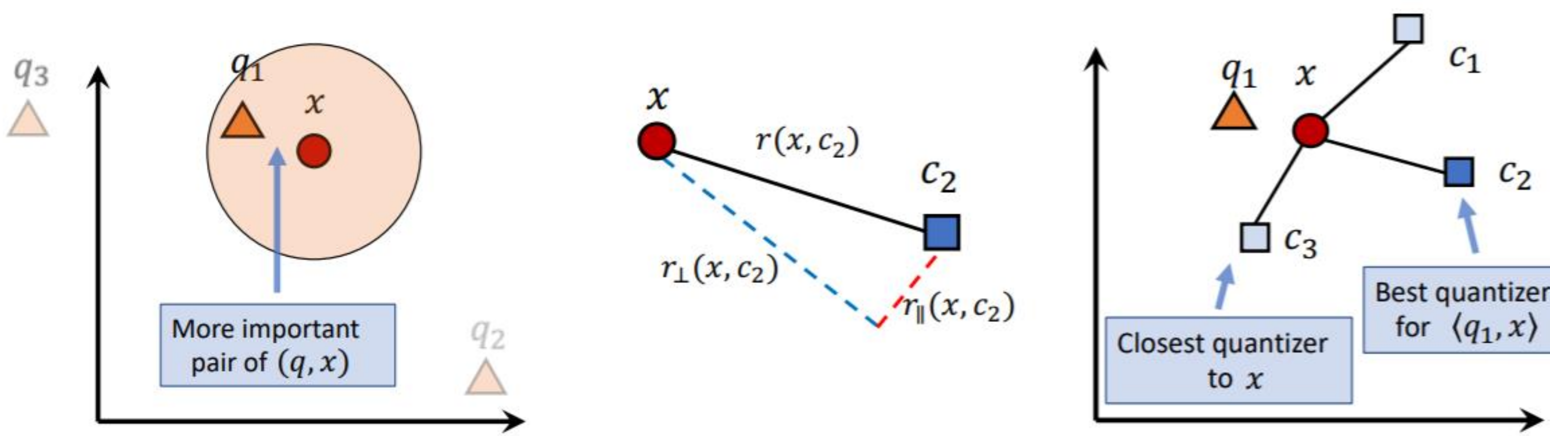


Publications

Guo-ICML'20

Google ScaNN

- Quantization-based similarity search using MIPS
  - A novel score-aware loss function:
    - The approximation error on the pairs which have a high inner product is far more important than that of pairs whose inner product is low.



High-d Vector Similarity Search State-of-the-Art Methods

Graph-Based Methods

Conceptual Graphs

- Voronoi/Delaunay Diagrams
- kNN Graphs
- Navigable Small World Graphs
- Relative Neighborhood graphs

The Delaunay Diagram

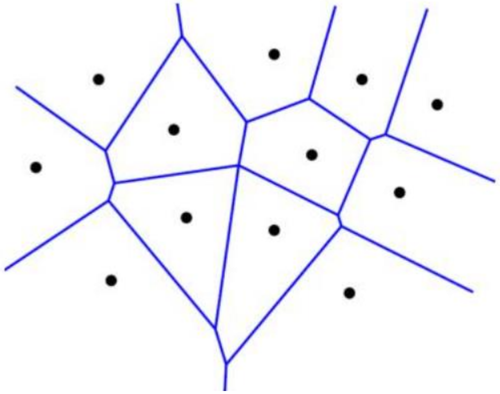
Publications

Delaunay
CSMN' 39

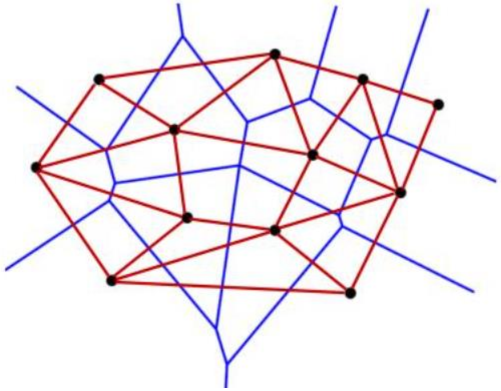
Delaunay Diagram – Dual of Voronoi Diagram

- The VD is constructed by decomposing the space using a finite number of points, called sites into regions, such that each site is associated to a region consisting of all points closer to it than to any other site.

- The DT is the dual of the VD, constructed by connecting sites with an edge if their regions share a side.



Voronoi Diagram



Delaunay Diagram

kNN Graphs

Publications

Anastasiu et al. CIKM' 15

Dong et al. WWW' 11

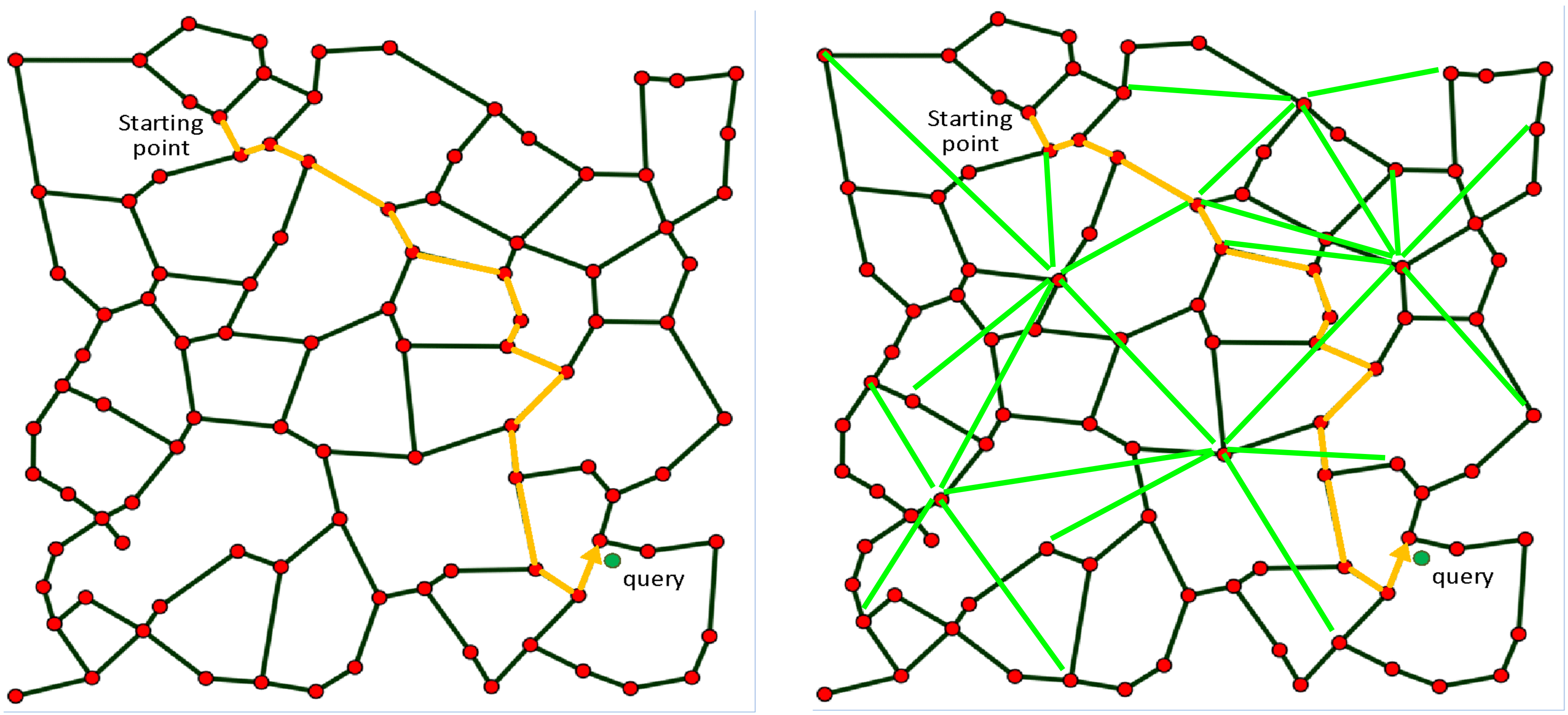
- Exact kNN graphs on n d-dimensional points:
  - Each point in the space is considered a node
  - A directed edge is added between nodes node A and B (A -=> B) if B is a kNN of A
  - $O(dn^2)$
  - Example: L2knng
- Approximate kNN Graphs:
  - LSH
  - Heuristics
    - Example: NN-Descent: "*a neighbor of a neighbor is also likely to be a neighbor*"

NSW Graphs

Publications

Kleinberg STOC' 00

- Augment approximate kNN graphs with long range links:
  - Milgram experiment
  - Shorten the greedy algorithm path to log(N)

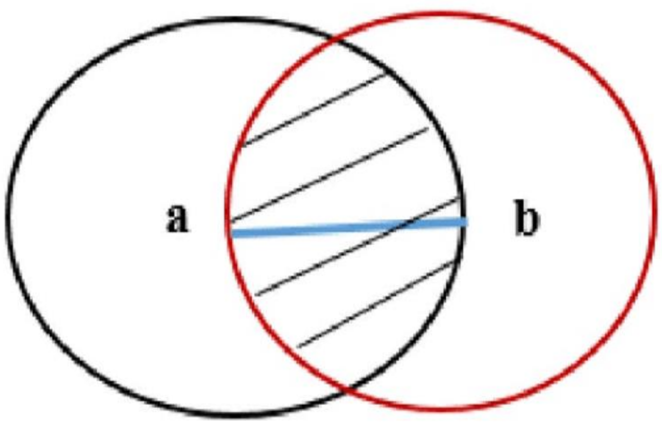


Publications

Toussaint
Pat. Recognit.'80

Relative Neighbourhood graph (RNG)

- A superset of the minimal spanning tree (MST) and a subset of the Delaunay Diagram.
- Two algorithms for obtaining the RNG of n points on the plane:
  - An algorithm for 1-d space in o(n2) time
  - Another algorithm for d-dimensional spaces running in o(n3).
- An edge is constructed between two vertices if there is no vertex in the intersection of the two balls

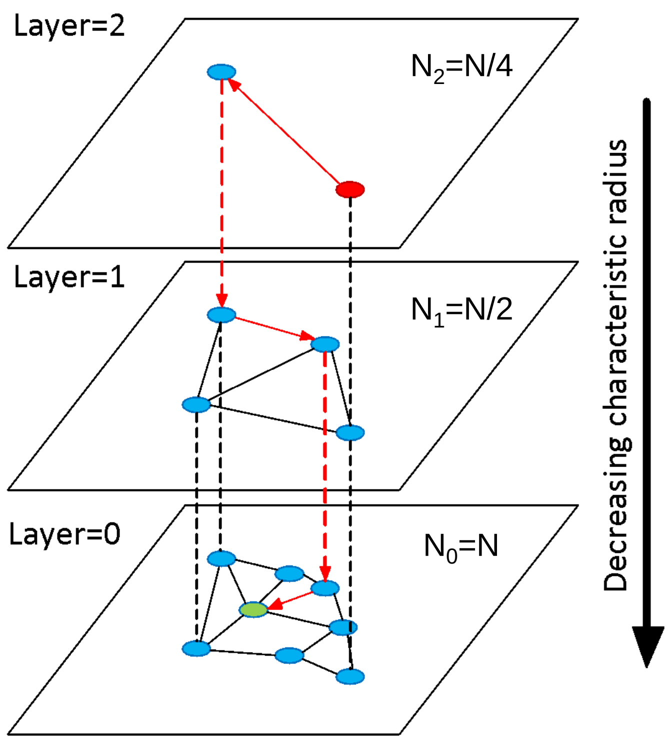


HNSW

Publications

Malkov et al. TPAMI' 20 Arxiv'16

- In HNSW we split the graph into layers (fewer elements at higher levels)
- Search starts for the top layer. Greedy routing at each level and descend to the next layer.
- Maximum degree is capped while paths ~ log(N) → log(N) complexity scaling.
- Incremental construction



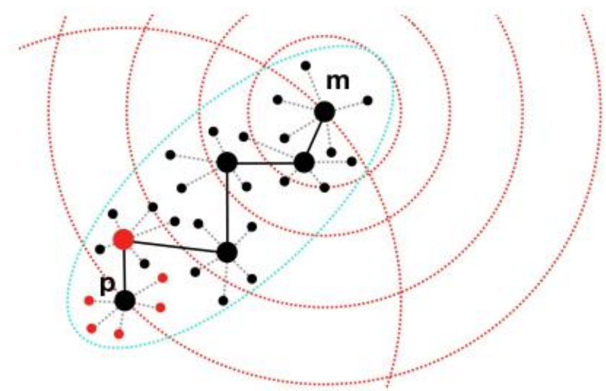
Slides by Malkov

Navigating Speading-out Graph (NSG)

Publications

Fu et al. PVLDB' 19

- RNGs do not guarantee monotonic search
  - There exists at least one monotonic path. Following this path, the query can be approached with the distance decreasing monotonically
- Propose a Monotonic RNG (MRNG)
- Build an approximate *k*NN graph.
- Find the <u>*Navigating Node*</u>. (All search will start with this fixed node – center of the graph).
- For each node p, find a relatively small candidate neighbour set. (*sparse*)
- Select the edges for p according to the definition of MRNG. (*low complexity*)
- leverage Depth-First-Search tree (*connectivity*)



Slides by Fu

Other tutorials

- for a more complete and detailed presentation, see tutorials:
  - Jianbin Qin, Wei Wang, Chuan Xiao, Ying Zhang: Similarity Query Processing for High-Dimensional Data. PVLDB. 13(12): 3437-3440 (2020).
  - *Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas. High-Dimensional Similarity Search for Scalable Data Science. ICDE 2021*

High-d Vector Similarity Search State-of-the-Art Methods

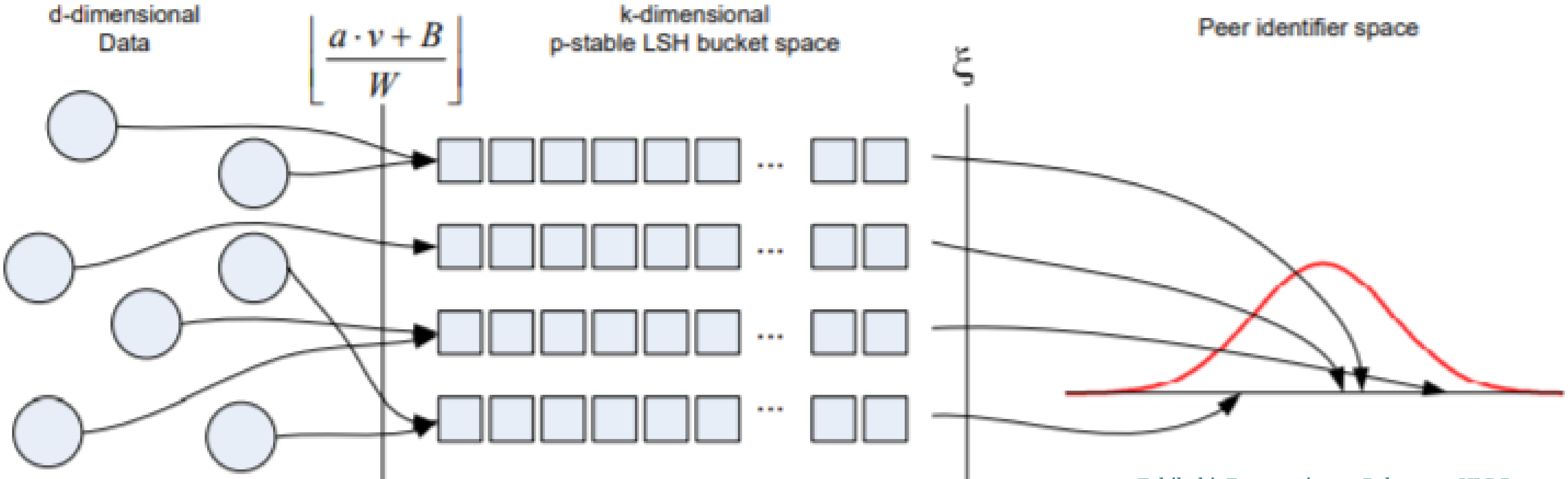
Modern Hardware & Distribution

Distributed LSH

Publications

Haghani-EDBT'09

- A two-level mapping strategy
  - Condition 1: assign buckets likely to hold similar data to the same peer.
  - Condition 2: have a predictable output distribution which fosters fair load balancing.
- Theoretical guarantees on locality preserving properties of the mapping
- Significant improvement over state-of-the-art



Layered LSH

Publications

Bahmani-CIKM'12

- One of the early works
- Entropy-based LSH in Euclidean space
- Apache Hadoop for disk-based version
- Twitter storm for in-memory version
- Theoretical guarantees
  - Only for the single hash tables setting

PLSH

Publications

Sundaram-PVLDB'13

- In-memory, multi-core, distributed LSH
  - Designed for text data (angular distance)
- Main idea
  - Use a caching strategy to improve online index construction
  - Insert-optimized delta tables to hold indexes of new data
    - Merge periodically with main index structures
  - Eliminate duplicate data using a bitmap-based strategy
  - Model to predict performance
- Experiments on a billion-tweet dataset on 100 nodes
  - 1-2.5 ms per query
  - Streaming 100 millions of tweets per day

RDH

Publications

Durmaz-PLR'19

- Distributed similarity search for images
- Main ideal:
  - Randomly splits and distributes the dataset over compute nodes
  - Each node builds an LSH index over its data subset
  - Same hash functions used in all nodes
  - No communication between nodes
  - Network used to send hash functions and
- 8x faster (with 10 nodes) than state-of-the-art while maintaining similar accuracy

FAISS

Publications

Johnson-ITBD'21

- Facebook's library for similarity search
  - CPU and GPU implementations
- FAISS GPU:
  - Quantization-based inverted index
  - kNN graph
- Experiments
  - Up to 8.5x faster than other GPU-based techniques
  - 5x-10x faster than corresponding CPU implementation on a single GPU
  - Near linear speedup with multiple GPUs over a single GPU
  - 95 million images in 35 minutes, and of a graph connecting 1 billion vectors in less than 12 hours on 4 Maxwell Titan X GPUs

Questions?

Experimental Comparisons: Similarity Search Methods

How do similarity search methods compare?

- several methods proposed in last 3 decades by different communities
  - never carefully compared to one another

- we now present results of extensive experimental comparison

Experimental Comparisons: A Taxonomy

Methods

Publications

Echihabi-PVLDB'18

Echihabi-PVLDB'19

Similarity Search Methods

Methods

Publications

Echihabi-PVLDB'18

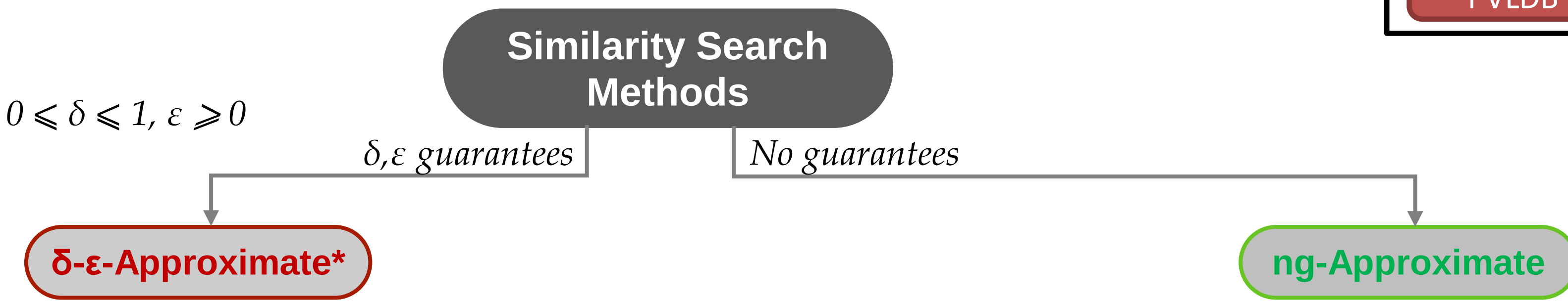
Echihabi-PVLDB'19

Methods

Publications

Echihabi-PVLDB'18

Echihabi-PVLDB'19



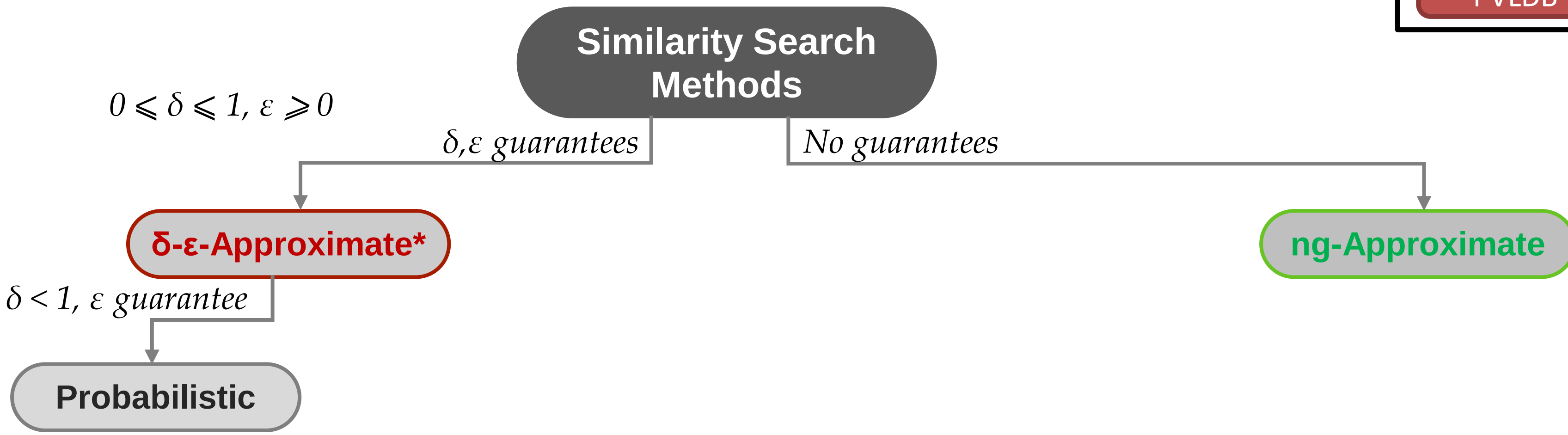
* result is within distance (1+ ε) of the exact answer with probability δ

Publications

Echihabi-PVLDB'18

Echihabi-PVLDB'19

Methods



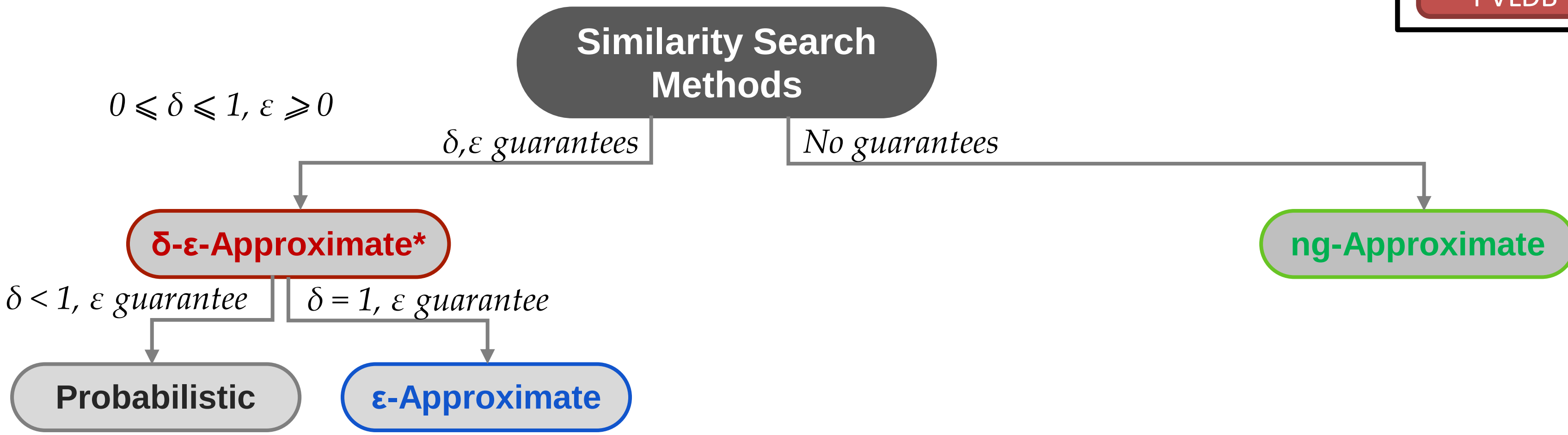
* result is within distance (1+ ε) of the exact answer with probability δ

Methods

Publications

Echihabi-PVLDB‘18

Echihabi-PVLDB‘19



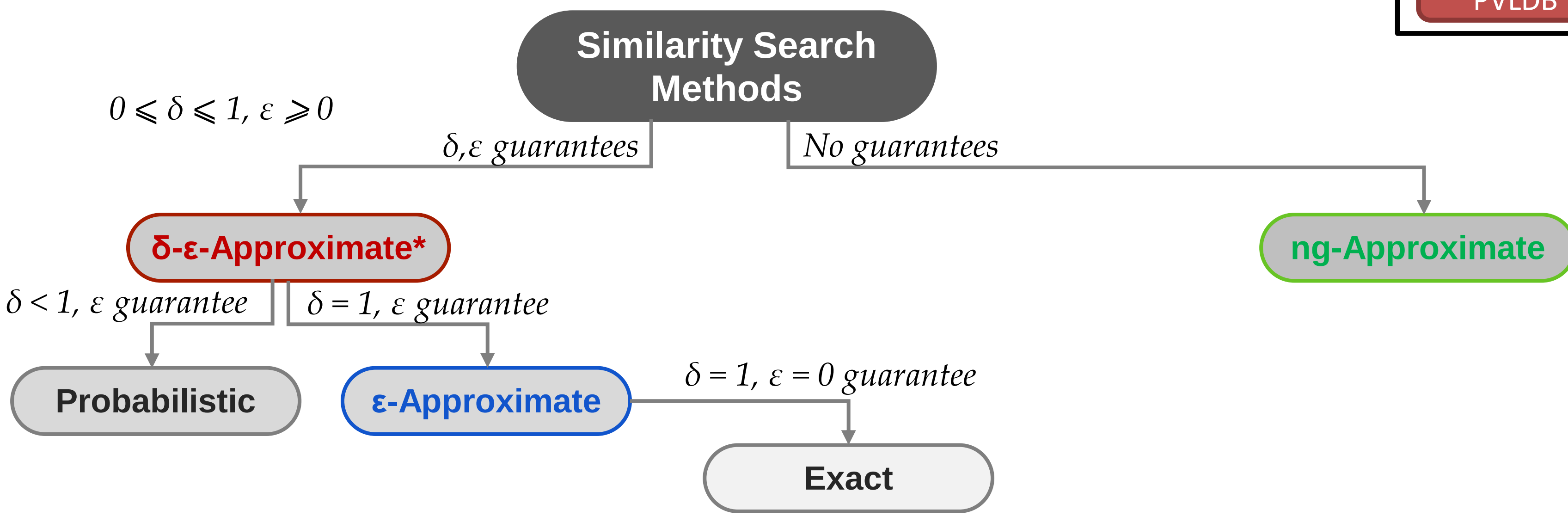
*** result is within distance (1+ ε) of the exact answer with probability δ**

Methods

Publications

Echihabi-PVLDB'18

Echihabi-PVLDB'19

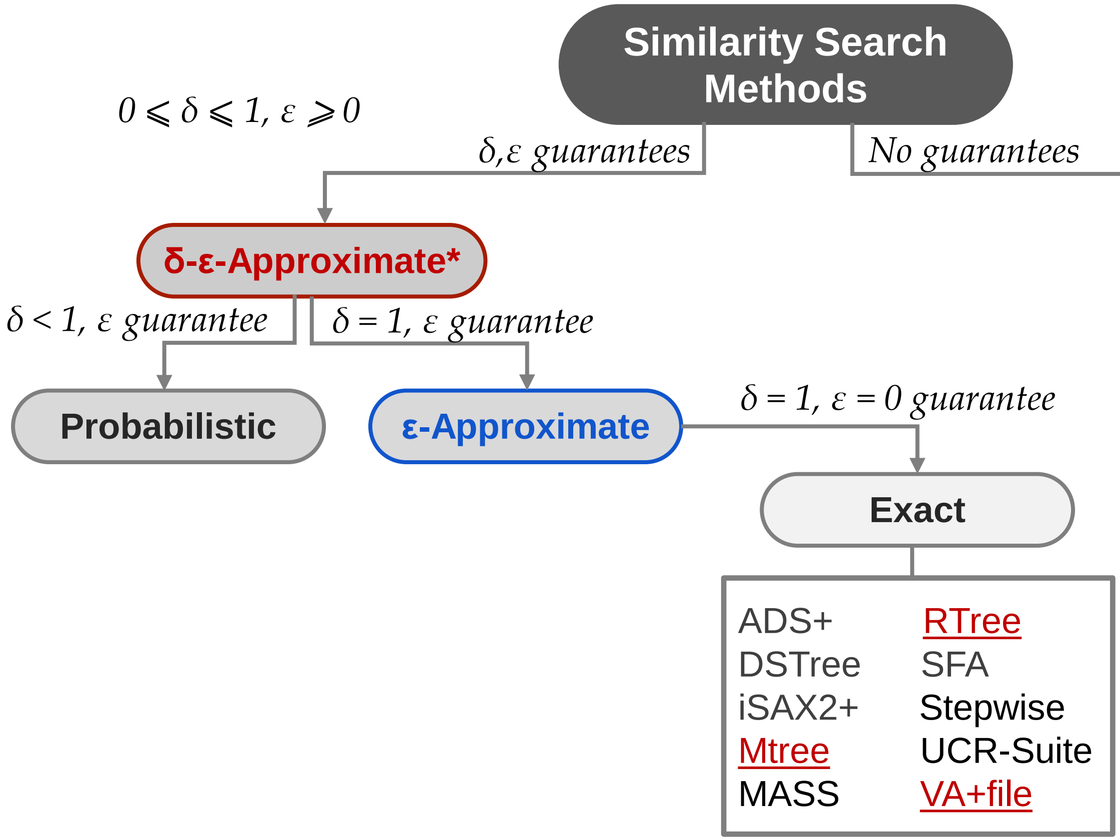


* result is within distance (1+ ε) of the exact answer with probability δ

Methods

Techniques for data Series
Techniques for High-D vectors

Publications
Echihabi-PVLDB‘18
Echihabi-PVLDB‘19

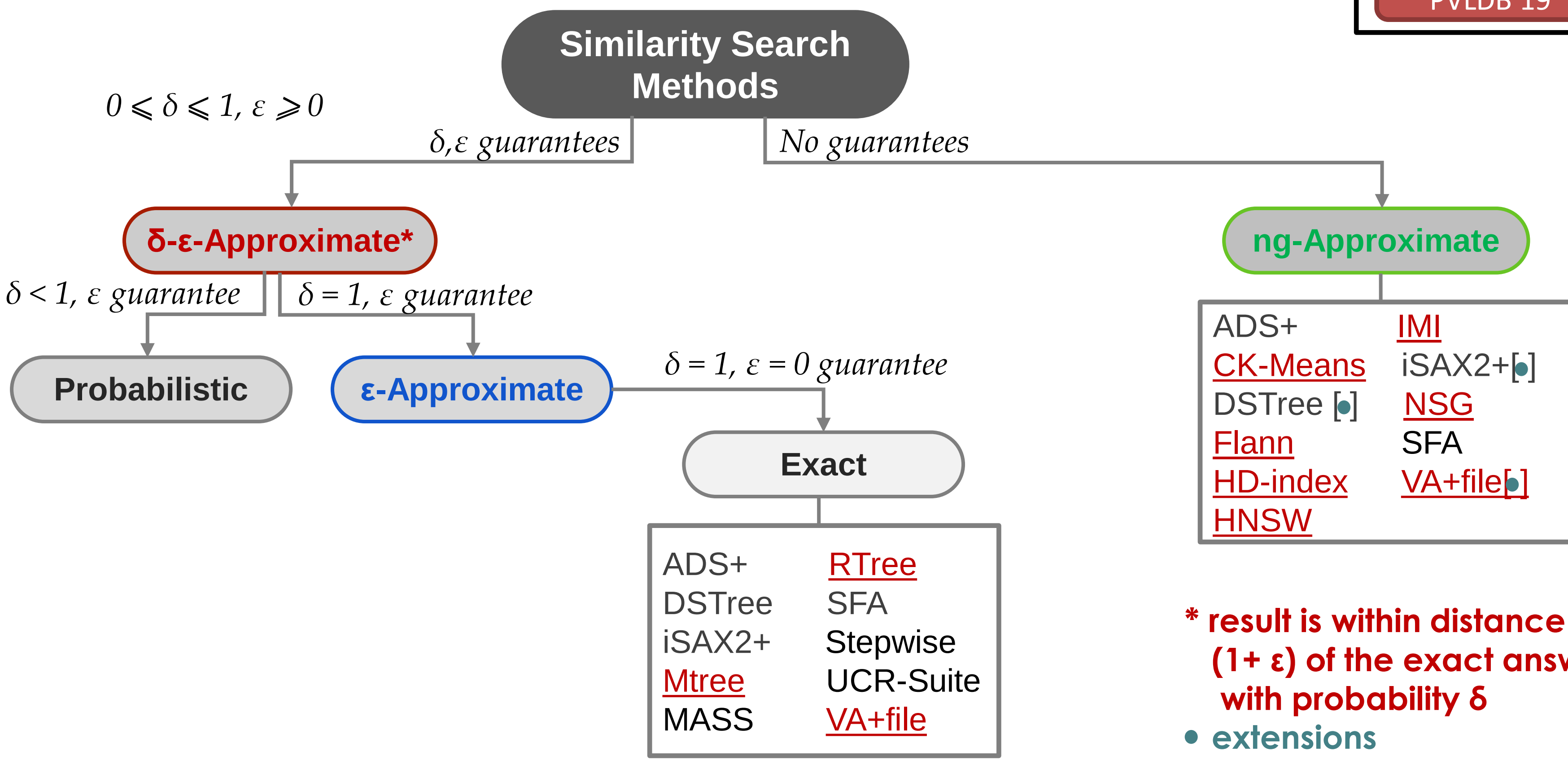


*** result is within distance (1+ ε) of the exact answer with probability δ**

Methods

Techniques for data Series
Techniques for High-D vectors

Publications
Echihabi-PVLDB'18
Echihabi-PVLDB'19



*** result is within distance (1+ ε) of the exact answer with probability δ**

• extensions

Methods

Publications
- Echihabi-PVLDB'18
- Echihabi-PVLDB'19

Techniques for data Series
Techniques for High-D vectors

Similarity Search Methods

$0 \leqslant \delta \leqslant 1, \varepsilon \geqslant 0$

δ,ε guarantees → **δ-ε-Approximate***
- *δ < 1, ε guarantee* → **Probabilistic**
- *δ = 1, ε guarantee* → **ε-Approximate**
  - ADS+[•]
  - DSTree[•]
  - iSAX2+ [•]
  - Mtree
  - VA+file[•]
  - *δ = 1, ε = 0 guarantee* → **Exact**
    - ADS+
    - DSTree
    - iSAX2+
    - Mtree
    - MASS
    - RTree
    - SFA
    - Stepwise
    - UCR-Suite
    - VA+file

No guarantees → **ng-Approximate**
- ADS+
- CK-Means
- DSTree [•]
- Flann
- HD-index
- HNSW
- IMI
- iSAX2+[•]
- NSG
- SFA
- VA+file[•]

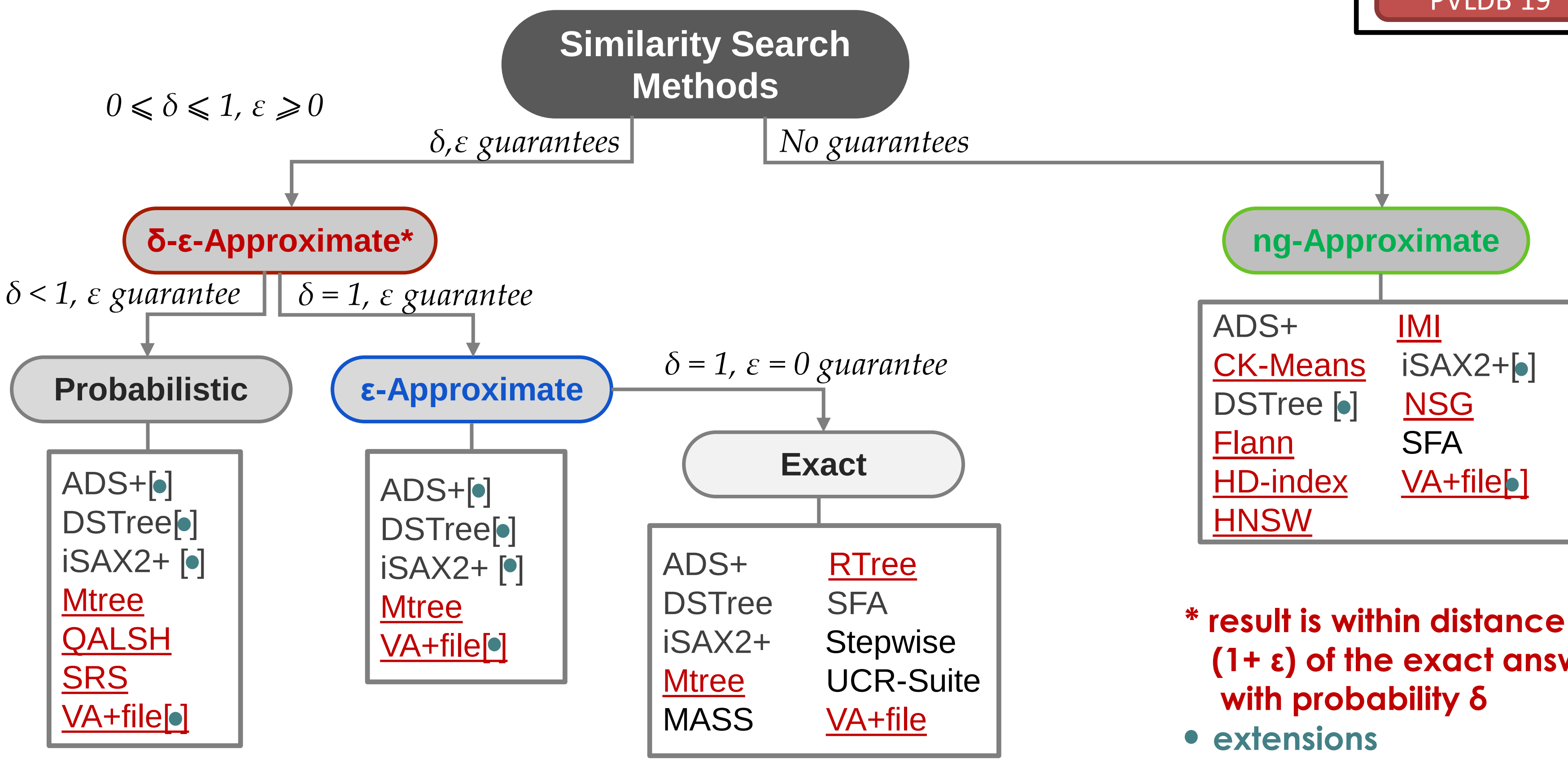
*** result is within distance (1+ ε) of the exact answer with probability δ**

• extensions

Methods

Techniques for data Series
Techniques for High-D vectors

Publications
- Echihabi-PVLDB'18
- Echihabi-PVLDB'19



*** result is within distance (1+ ε) of the exact answer with probability δ**
- **extensions**

Experimental Comparisons: Exact Query Answering

Experimental Framework

Publications

Echihabi-PVLDB'18

- Hardware
  - HDD and SSD
- Datasets
  - Synthetic (25GB to 1TB) and 4 real (100 GB)
- Exact Query Workloads
  - 100 – 10,000 queries
- Performance measures
  - Time, #disk accesses, footprint, pruning, Tightness of Lower Bound (TLB), etc.
- C/C++ methods (4 methods reimplemented from scratch)

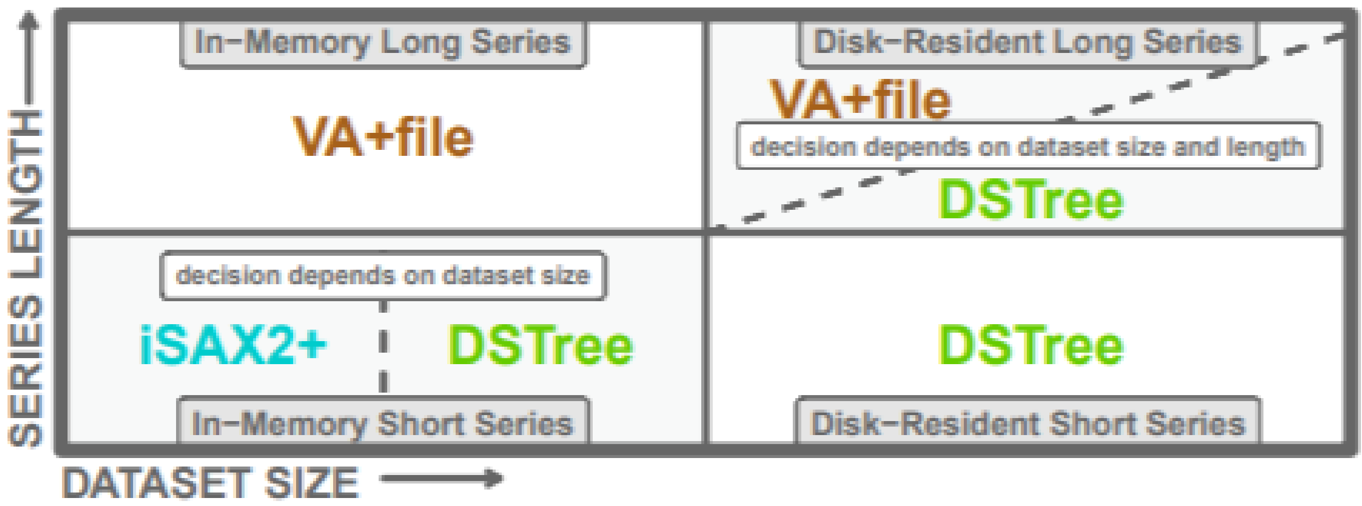
- Procedure:
  - Step 1: Parametrization
  - Step 2: Evaluation of individual methods
  - Step 3: Comparison of best methods

Recommendations

Publications

Echihabi-PVLDB'18

Scenario: Indexing and answering 10K exact queries on HDD



Unexpected Results

Publications

Echihabi-PVLDB'18

- Some methods do not scale as expected (or not at all!)
- Brought back to the spotlight two older methods VA+file and DSTree
  - New reimplementations outperform by far the original ones
- Optimal parameters for some methods are different from the ones reported in the original papers
- Tightness of Lower Bound (TLB) does not always predict performance

Insights

Publications

Echihabi-PVLDB'18

- Results are sensitive to:
  - Parameter tuning
  - Hardware setup
  - Implementation
  - Workload selection
- Results identify methods that would benefit from modern hardware

Experimental Comparisons: Approximate Query Answering

Experimental Framework

Publications

Echihabi-PVLDB'19

- Datasets
  - In-memory and disk-based datasets
  - Synthetic data modeling financial time series
  - Four real datasets from deep learning, computer vision, seismology, and neuroscience (25GB-250GB)
- Query Workloads
  - 100 – 10,000 kNN queries k in [1,100]
  - ng-approximate and δ-ε-approximate queries (exact queries used as yardstick)
- C/C++ methods (3 methods reimplemented from scratch)
- Performance measures
  - Efficiency: time, throughput, #disk accesses, % of data accessed
  - Accuracy: average recall, mean average precision, mean relative error
- Procedure:
  - Step 1: Parametrization
  - Step 2: Evaluation of indexing/query answering scalability in-memory
  - Step 3: Evaluation of indexing/query answering scalability on-disk
  - Step 4: Additional experiments with best-performing methods on disk

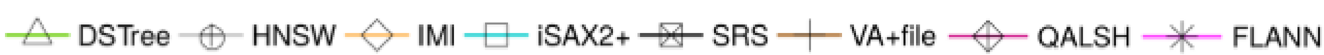
Approximate Methods Covered in Study

		Matching Accuracy				Representation		Implementation		
		exact	ng-appr.	ϵ-appr.	δ-ϵ-appr.	Raw	Reduced	Original	New	Disk-resident Data
Graphs	HNSW		[99]			✓		C++		
	NSG		[58]			✓		C++		
Inv. Indexes	IMI		[16, 60]				OPQ	C++		✓
LSH	QALSH				[69]		Signatures	C++		
	SRS				[136]		Signatures	C++		
Scans	VA+file	[55]	•	•	•		DFT	MATLAB	C	✓
Trees	Flann		[107]			✓		C++		
	DSTree	[146]	[146]	•	•		EAPCA	Java	C	✓
	HD-index		[11]				Hilbert keys	C++		✓
	iSAX2+	[30]	[30]	•	•		iSAX	C#	C	✓

- **Our extensions**

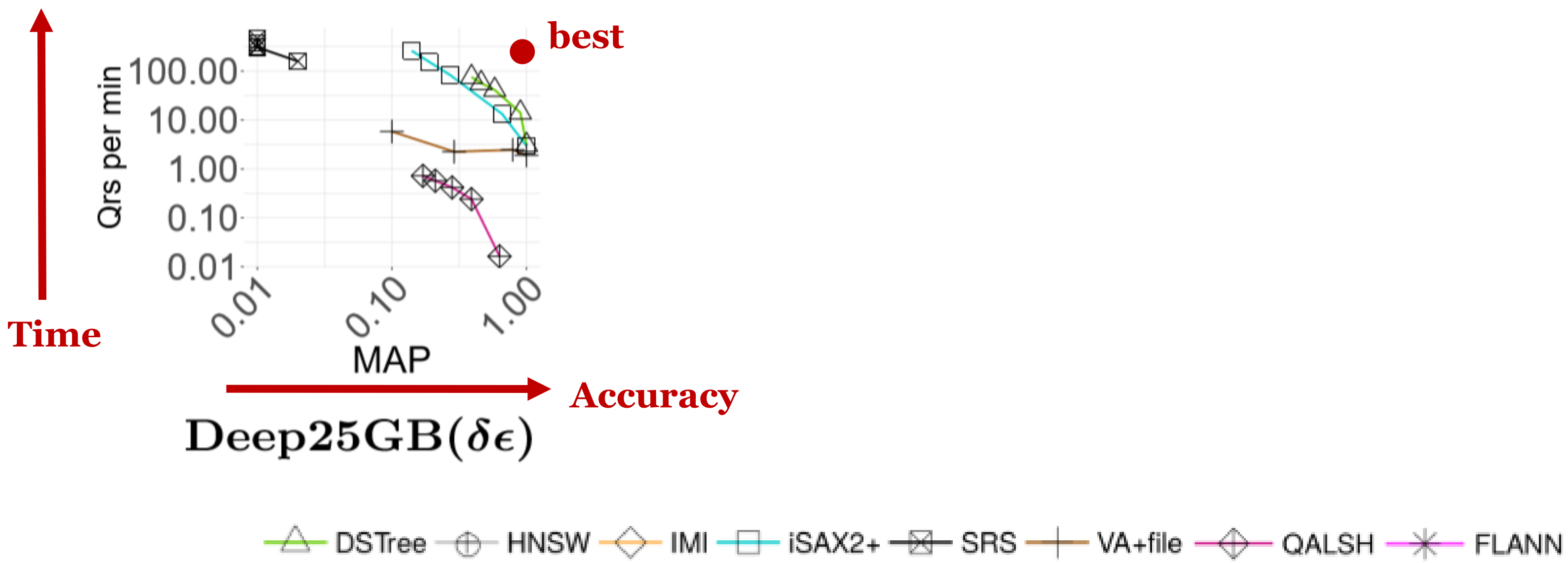
Unexpected Results

- New data series extensions are the overall winners even for general high-d vectors
  - perform the best for approximate queries with probabilistic guarantees (δ-ε-approximate search)



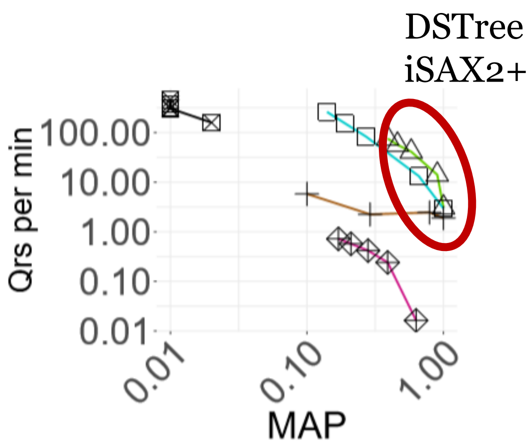
Unexpected Results

- New data series extensions are the overall winners even for general high-d vectors
  - perform the best for approximate queries with probabilistic guarantees (δ-ε-approximate search)

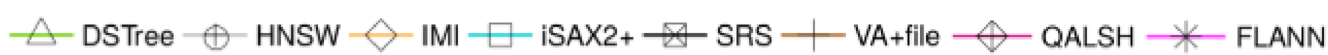


Unexpected Results

- New data series extensions are the overall winners even for general high-d vectors
  - perform the best for approximate queries with probabilistic guarantees (δ-ε-approximate search), in-memory

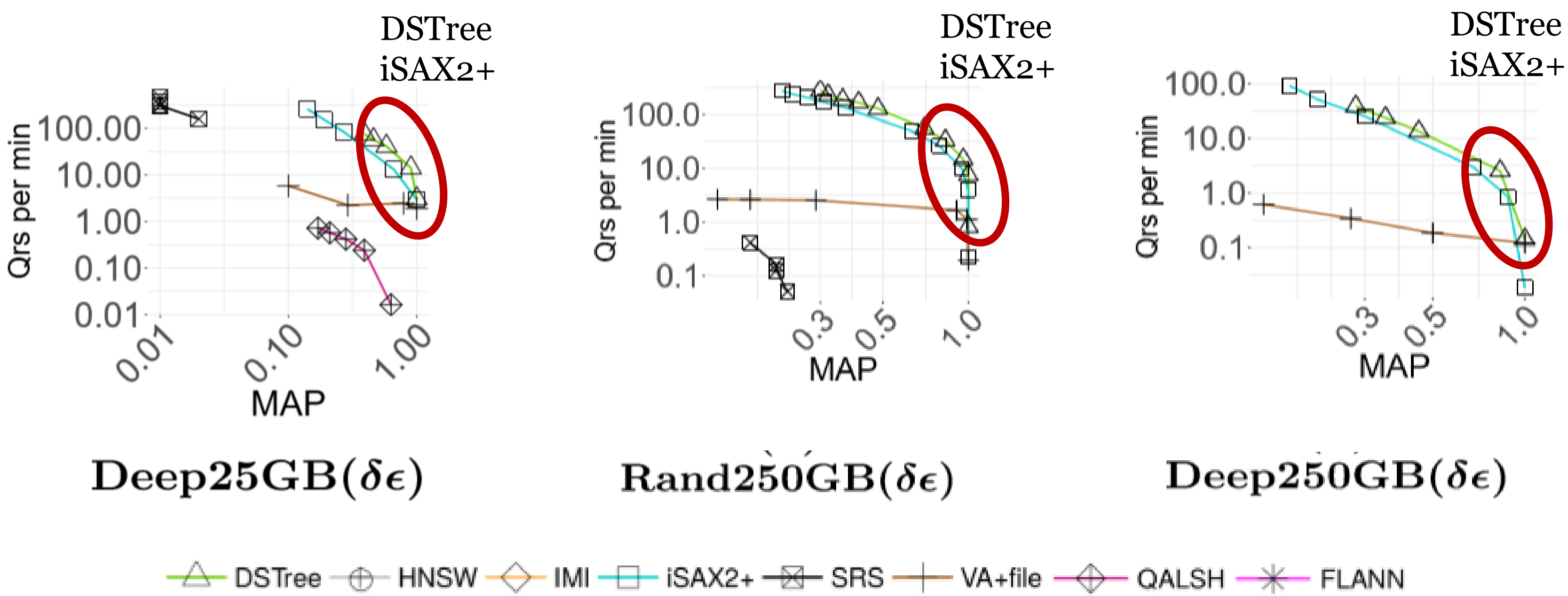


Deep25GB($\delta\epsilon$)



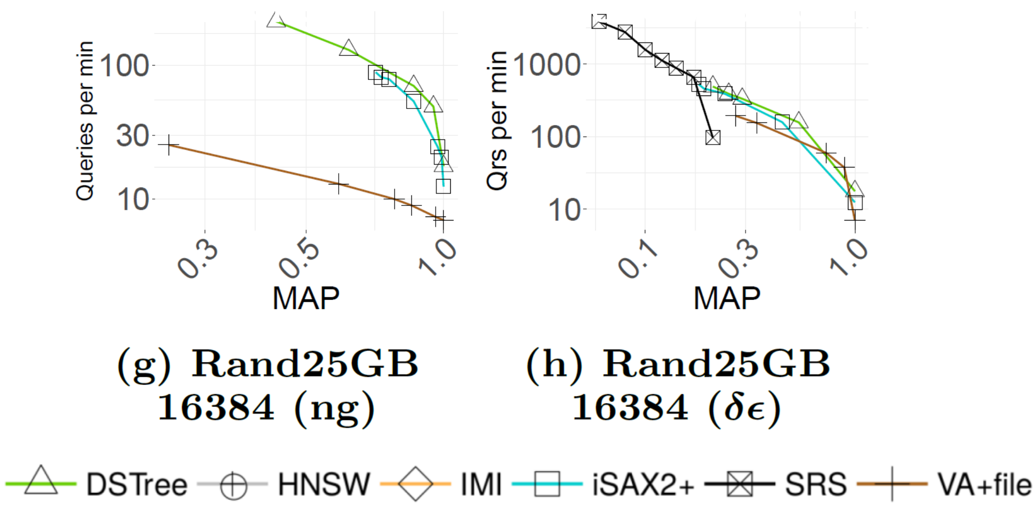
Unexpected Results

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



Unexpected Results

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

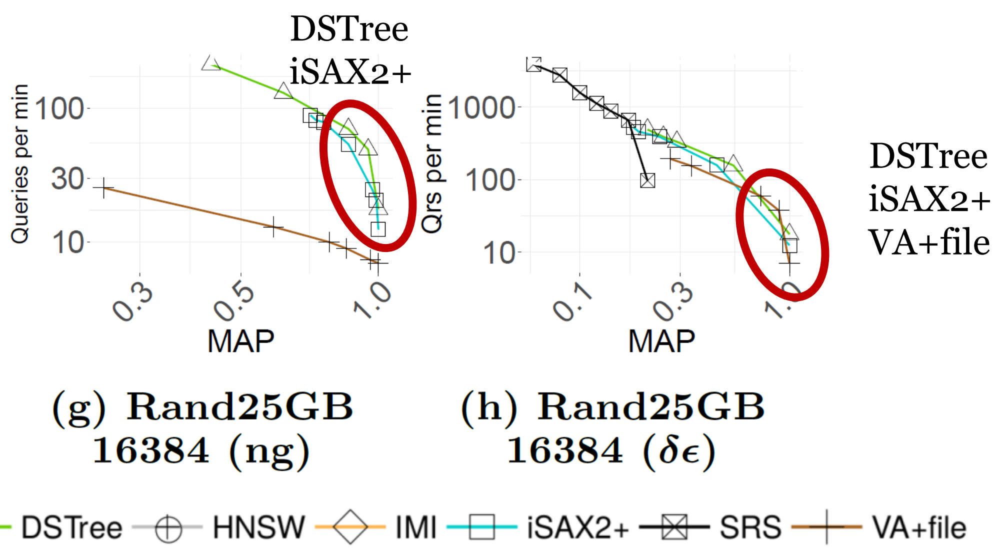


(g) Rand25GB 16384 (ng)

(h) Rand25GB 16384 ($\delta\epsilon$)

Unexpected Results

- New data series extensions are the overall winners even for general high-d vectors
  - 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

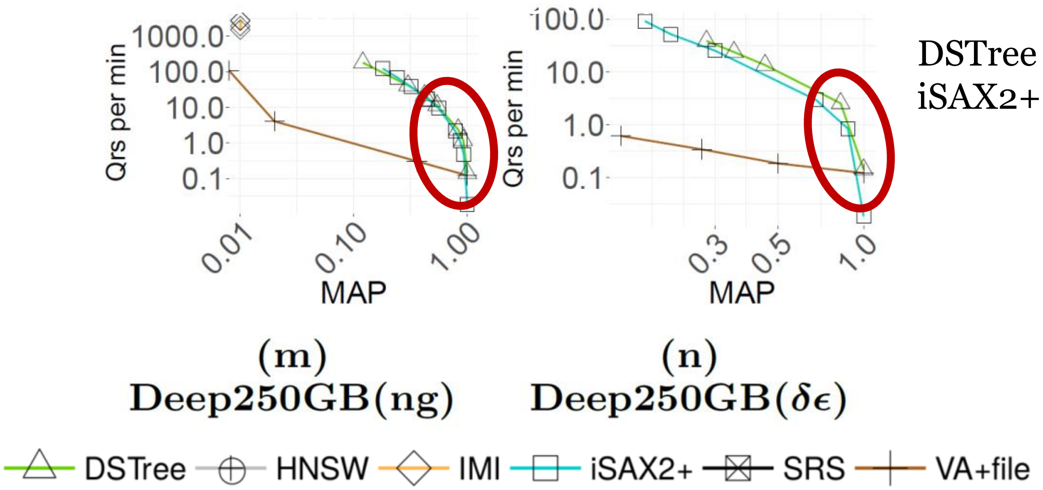


(g) Rand25GB 16384 (ng)

(h) Rand25GB 16384 (δϵ)

Unexpected Results

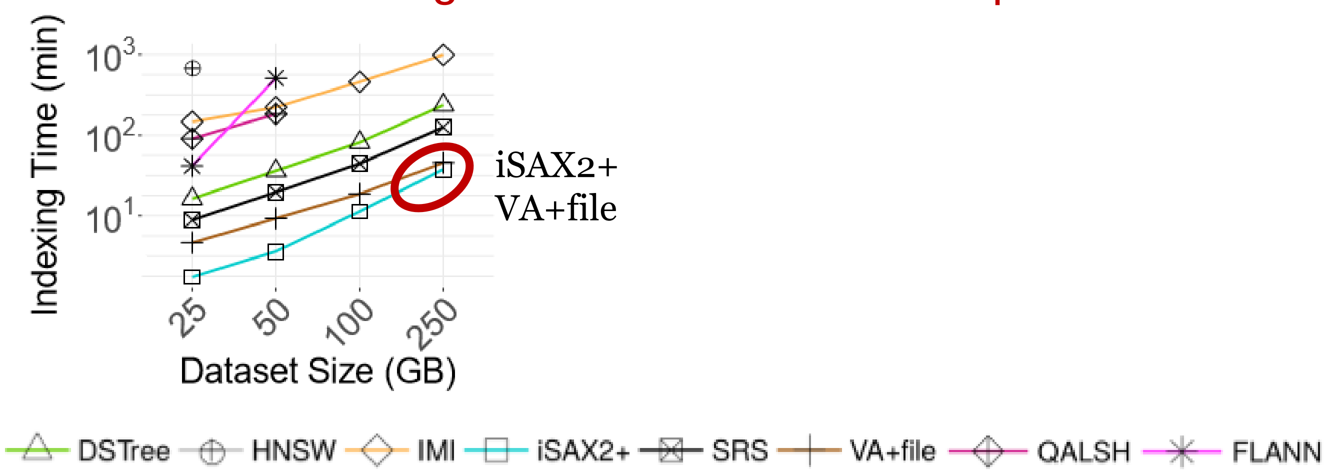
- New data series extensions are the overall winners even for general high-d vectors
  - perform the best for approximate queries with probabilistic guarantees (δ-ε-approximate search), in-memory and on-disk
  - perform the best for long vectors, in-memory and on-disk
  - perform the best for disk-resident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



(m) Deep250GB(ng) (n) Deep250GB($\delta\epsilon$)

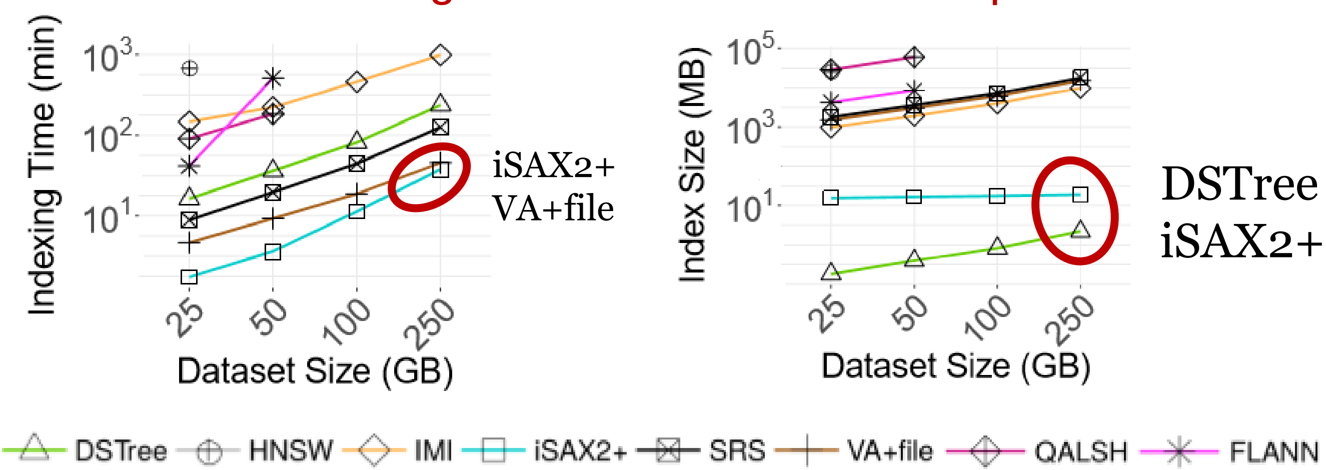
Unexpected Results

- New data series extensions are the overall winners even for general high-d vectors
  - 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
  - are fastest at indexing and have the lowest footprint



Unexpected Results

- New data series extensions are the overall winners even for general high-d vectors
  - 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
  - are fastest at indexing and have the lowest footprint



Insights

Exciting research direction for approximate similarity search in high-d spaces:

Insights

Exciting research direction for approximate similarity search in high-d spaces:

Currently two main groups of solutions exist:

approximate search solutions
without guarantees
relatively efficient

Insights

Exciting research direction for approximate similarity search in high-d spaces:

Currently two main groups of solutions exist:

approximate search solutions
without guarantees
relatively efficient

approximate search solutions
with guarantees
relatively slow

Insights

Exciting research direction for approximate similarity search in high-d spaces:

Currently two main groups of solutions exist:

approximate search solutions	approximate search solutions
without guarantees	with guarantees
relatively efficient	relatively slow

We show that it is possible to have efficient approximate algorithms with guarantees

Insights

Approximate state-of-the-art techniques for high-d vectors are not practical:

Insights

Approximate state-of-the-art techniques for high-d vectors are not practical:

- LSH-based techniques
  - slow, high-footprint, low accuracy (recall/MAP)

Insights

Approximate state-of-the-art techniques for high-d vectors are not practical:

LSH-based techniques

slow, high-footprint, low accuracy (recall/MAP)

kNNG-based techniques

slow indexing, difficult to tune, in-memory, no guarantees

Insights

Approximate state-of-the-art techniques for high-d vectors are not practical:

- LSH-based techniques
  - slow, high-footprint, low accuracy (recall/MAP)
- kNNG-based techniques
  - slow indexing, difficult to tune, in-memory, no guarantees
- Quantization-based techniques
  - slow indexing, difficult to tune, no guarantees

Insights

Approximate state-of-the-art techniques for high-d vectors are not practical:

- LSH-based techniques
  - slow, high-footprint, low accuracy (recall/MAP)
- kNNG-based techniques
  - slow indexing, difficult to tune, in-memory, no guarantees
- Quantization-based techniques
  - slow indexing, difficult to tune, no guarantees

All suffer a serious limitation:
accuracy determined during index-building & query answering

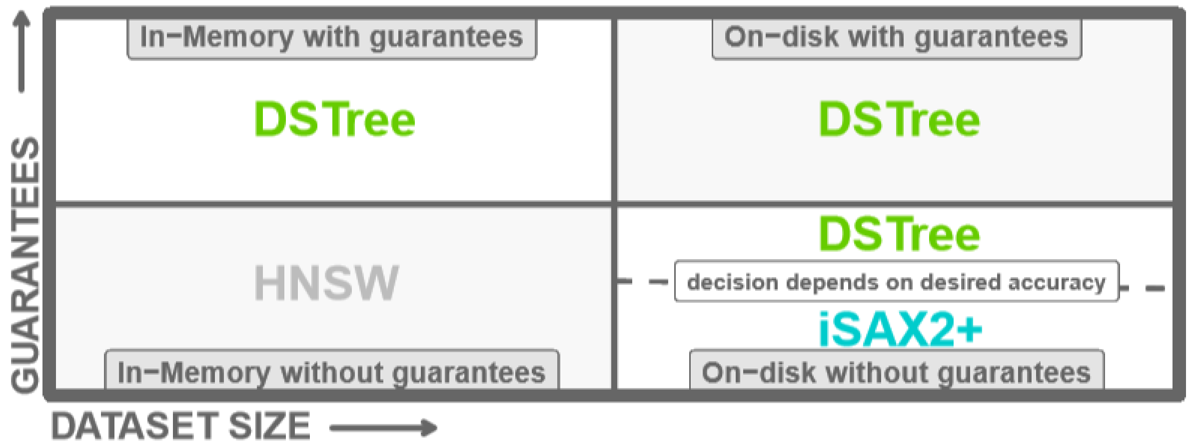
Recommendations for approx. techniques

Data series approaches are the overall winners!

The only exception is HNSW for in-memory ng-approximate queries using an existing index

Recommendations

Scenario: Answering a query workload using an existing index



Experimental evaluation of graph-based methods

Publications

Wang-PVLDB'2021

- A variety of evaluation criteria
  - Indexing:
    - Construction efficiency, index size, graph quality
  - Search
    - Efficiency, accuracy, candidate set size, query path length, memory overhead,
- 13 graph-based methods
- 8 real datasets and 12 synthetic datasets
  - Largest contains 2M vectors

Experimental evaluation of graph-based methods

Publications

Wang-PVLDB'2021

- Recommendations

Scenario	Algorithm
S1: A large amount of data updated frequently	NSG, NSSG
S2: Rapid construction of KNNG	KGraph, EFANNA, DPG
S3: Data is stored in external memory	DPG, HCNNG
S4: Search on hard datasets	HNSW, NSG, HCNNG
S5: Search on simple datasets	DPG, NSG, HCNNG, NSSG
S6: GPU acceleration	NGT
S7: Limited memory resources	NSG, NSSG

Questions?

Progressive Similarity Search

Interactive Analytics

- analytics over high-d data is computationally expensive
  - very high inherent complexity

- may not always be possible to remove delays
  - but could try to hide them!

Need for Interactive Analytics

- interaction with users offers new opportunities
  - progressive answers
    - produce intermediate results
      - iteratively converge to final, correct solution

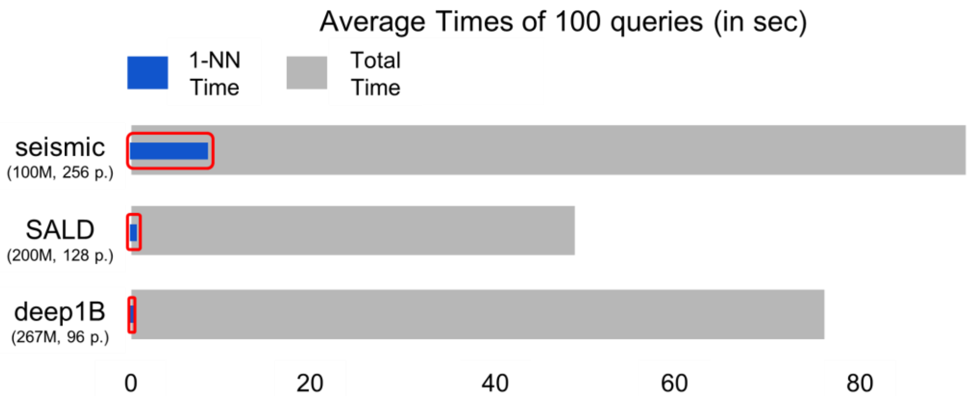
Need for Interactive Analytics

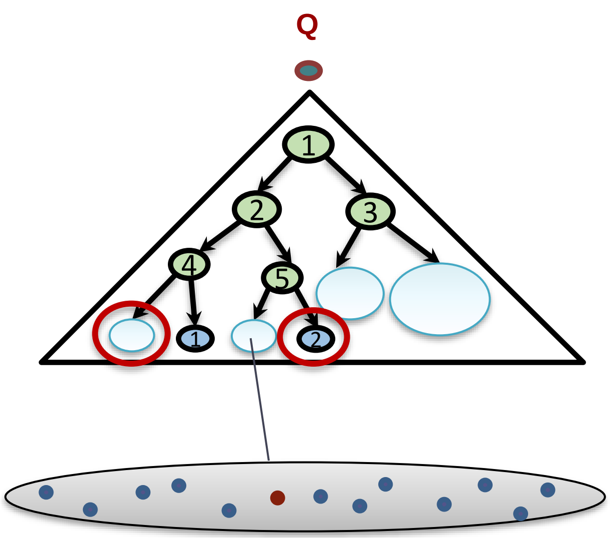
- interaction with users offers new opportunities
  - progressive answers
    - produce intermediate results
      - iteratively converge to final, correct solution
        - Exact or approximate

Need for Interactive Analytics
Exact Search

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

- interaction with users offers new opportunities
  - progressive answers
    - produce intermediate results
      - iteratively converge to final, correct solution
      - Tree-based indexes

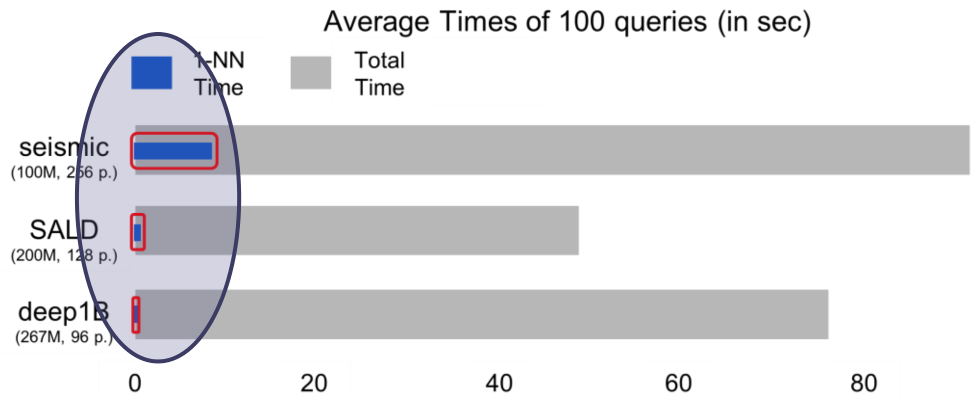


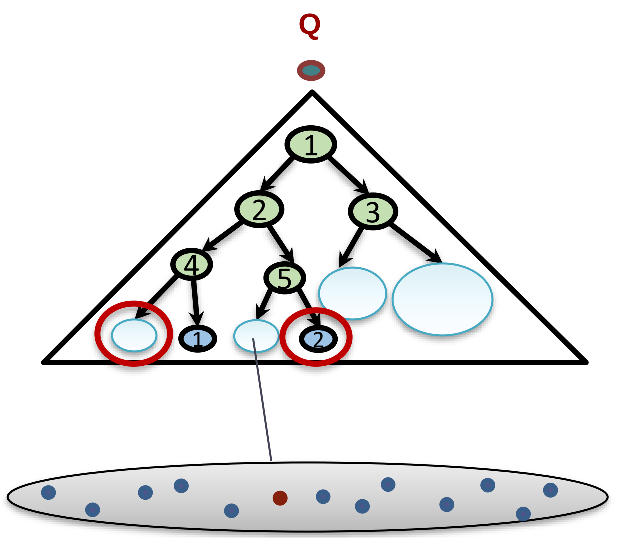


Need for Interactive Analytics
Exact Search

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

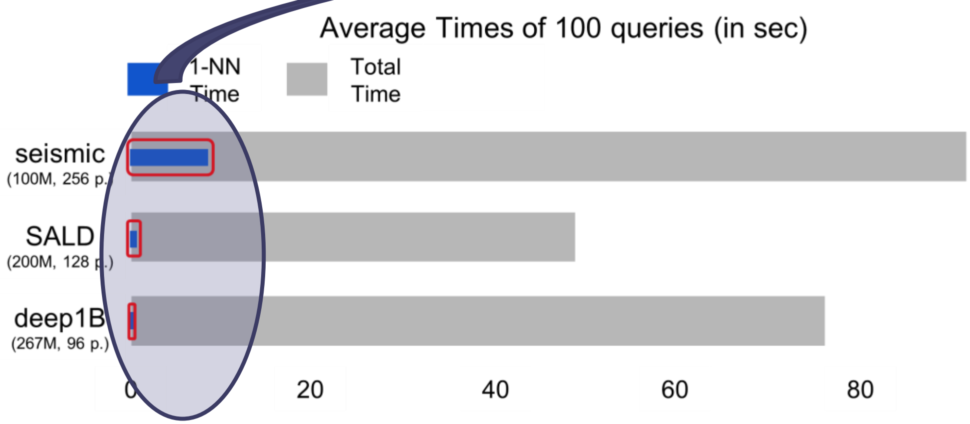
- interaction with users offers new opportunities
  - progressive answers
    - produce intermediate results
      - iteratively converge to final, correct solution
      - Tree-based indexes

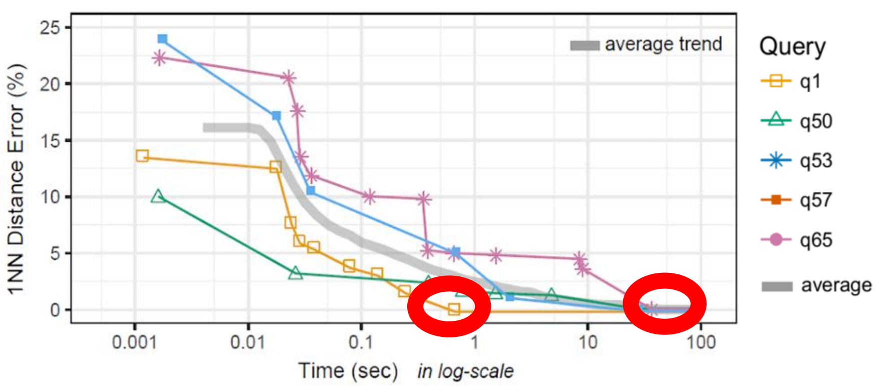




Need for Interactive Analytics
Exact Search

- interaction with users offers new opportunities
  - progressive answers
    - produce intermediate results
      - iteratively converge to final, correct solution



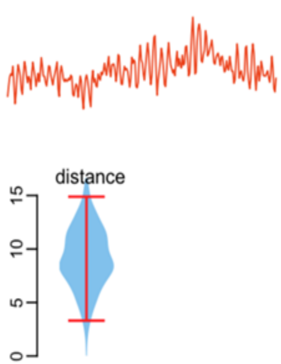


Need for Interactive Analytics
Exact Search

- 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

Publications
- Gogolou-BigVis‘19
- Gogolou-SIGMOD‘20

Query & Initial Estimate



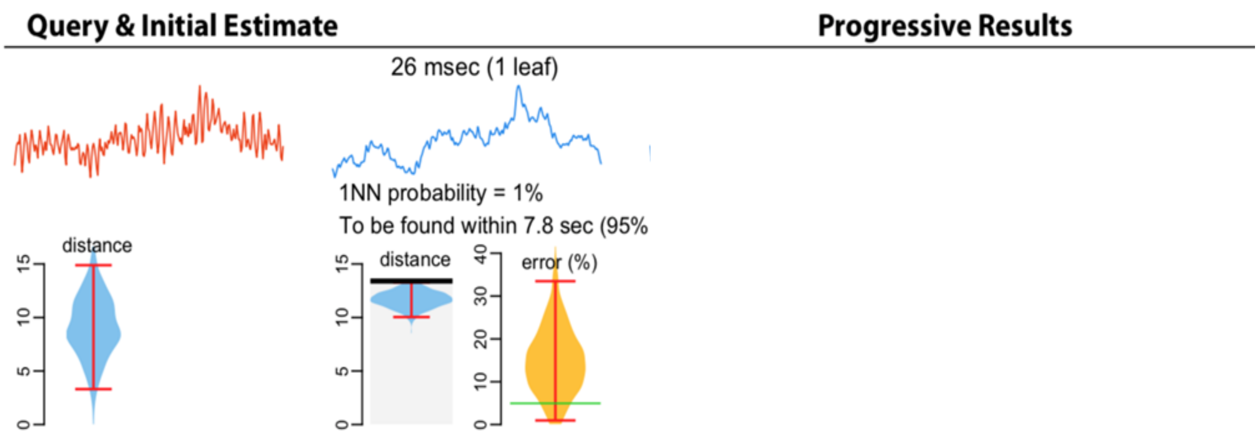
Need for Interactive Analytics
Exact Search

Publications

Gogolou-BigVis'19

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

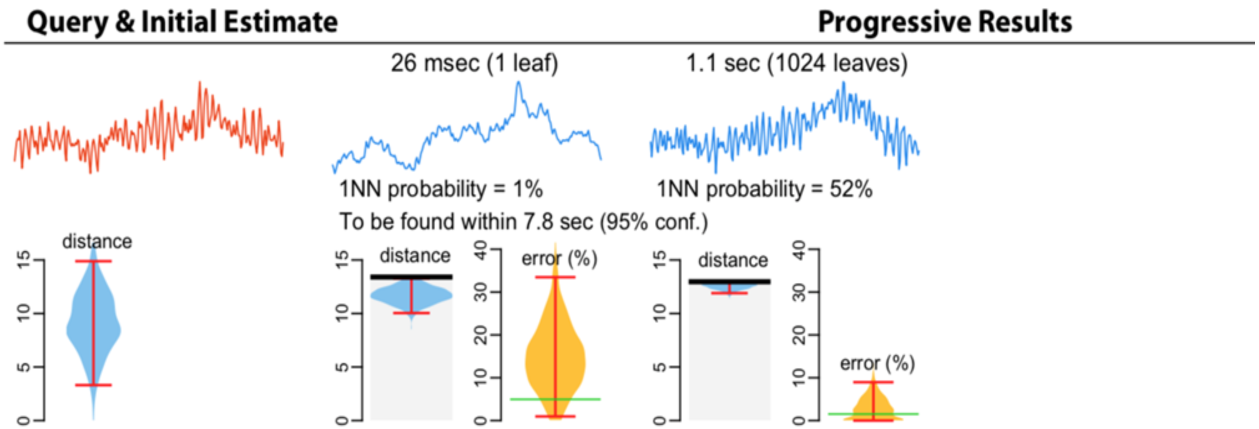


Need for Interactive Analytics

Exact Search

Publications
- Gogolou-BigVis'19
- 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



Need for Interactive Analytics

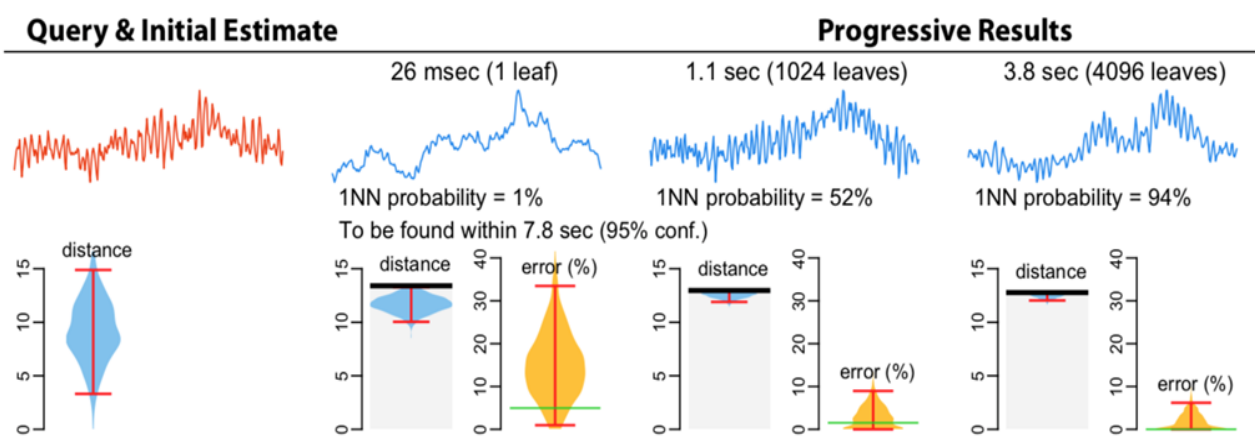
Exact Search

Publications

Gogolou-BigVis'19

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



Need for Interactive Analytics

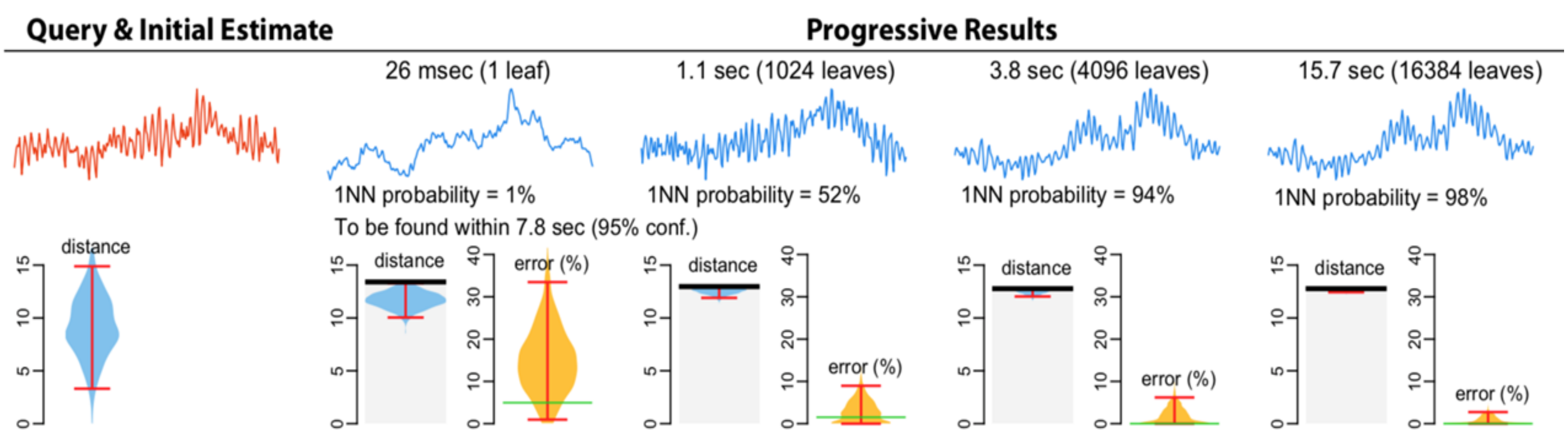
Exact Search

Publications

Gogolou-BigVis'19

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

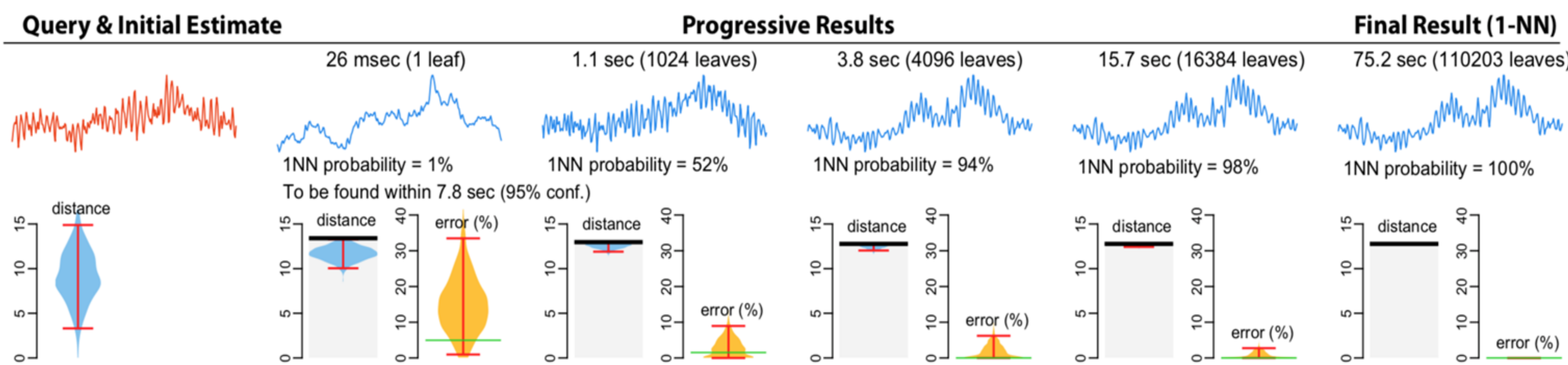


Need for Interactive Analytics

Exact Search

Publications
- Gogolou-BigVis'19
- 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



Need for Interactive Analytics

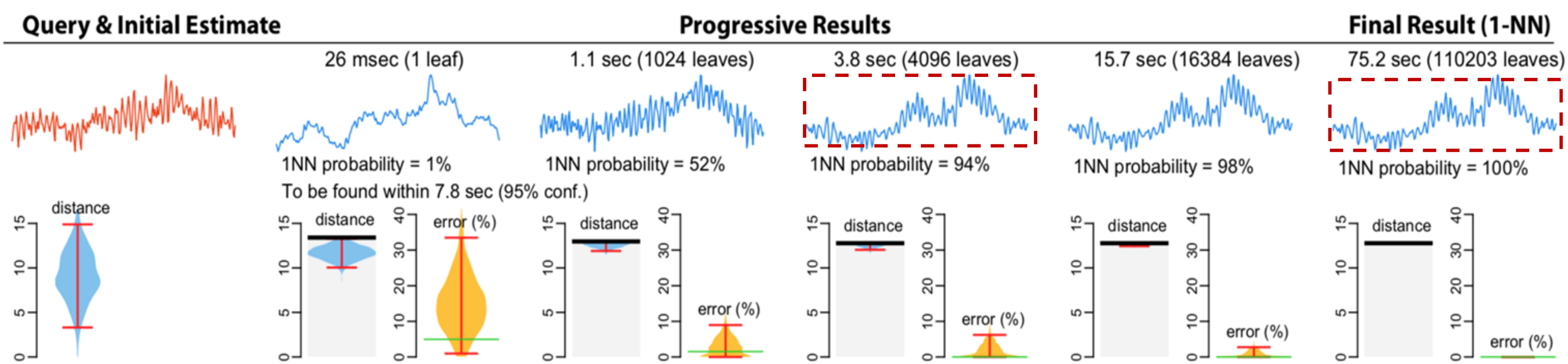
Exact Search

Publications

Gogolou-BigVis'19

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



Need for Interactive Analytics
Exact Search

Publications

Gogolou-BigVis'19

Gogolou-SIGMOD'20

Contributions

Formalize **data series** **progressive similarity search** with **probabilistic quality guarantees** (wrt *exact* answers).

Propose **statistical models** (linear, quantile & logistic regression, and multivariate kernel density estimation) to support **reliable** **progressive estimation** with a **small** **number of training queries**.

Develop **stopping criteria** to stop a search **long before normal query execution ends**.

Need for Interactive Analytics
Exact Search

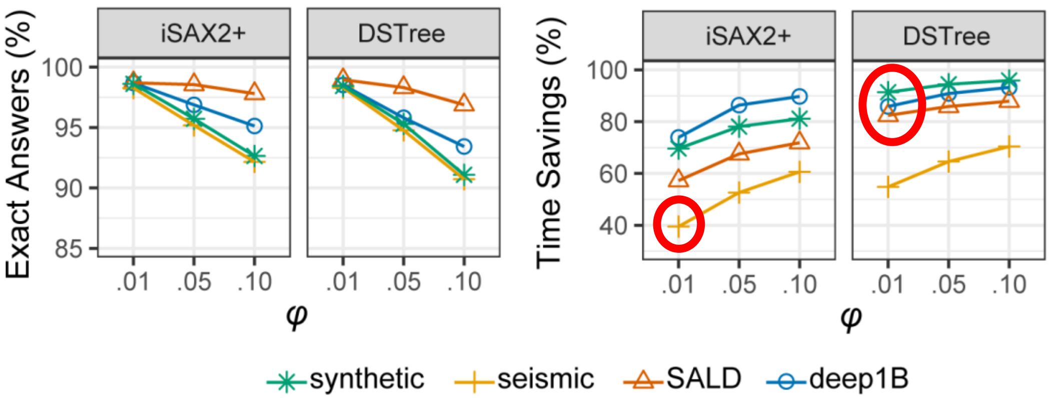
Publications

Gogolou-BigVis'19

Gogolou-SIGMOD'20

Time savings for 1NN queries

Early stopping when predicted **probability** that current answer is exact is higher than $1 - \varphi$



(n_r = 100 training queries)

time savings up to 90%

Need for Interactive Analytics
Exact Search

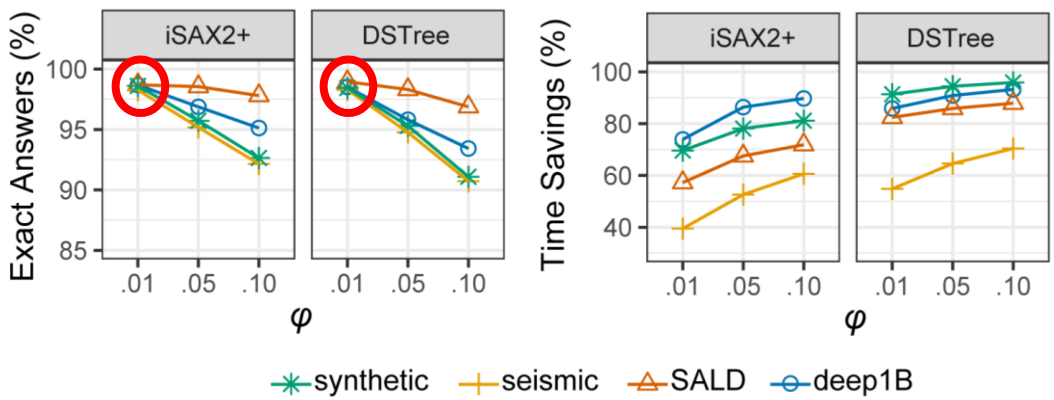
Publications

Gogolou-BigVis'19

Gogolou-SIGMOD'20

Time savings for 1NN queries

Early stopping when predicted **probability** that current answer is exact is higher than *1 - φ*



(n_r = 100 training queries)

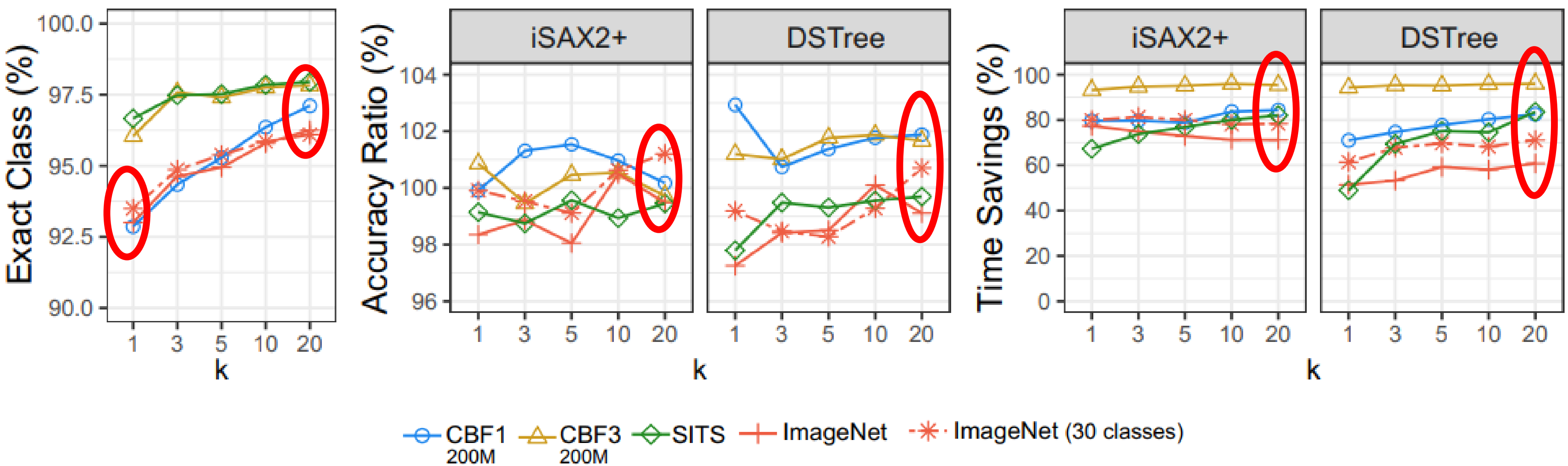
time savings up to 90%, with ~99% of the answers to be exact

Need for Interactive Analytics
Exact Search

Publications

Echihabi-submitted'21

Time savings for kNN classification



time savings up to 95% with ~99% of the answers to be exact

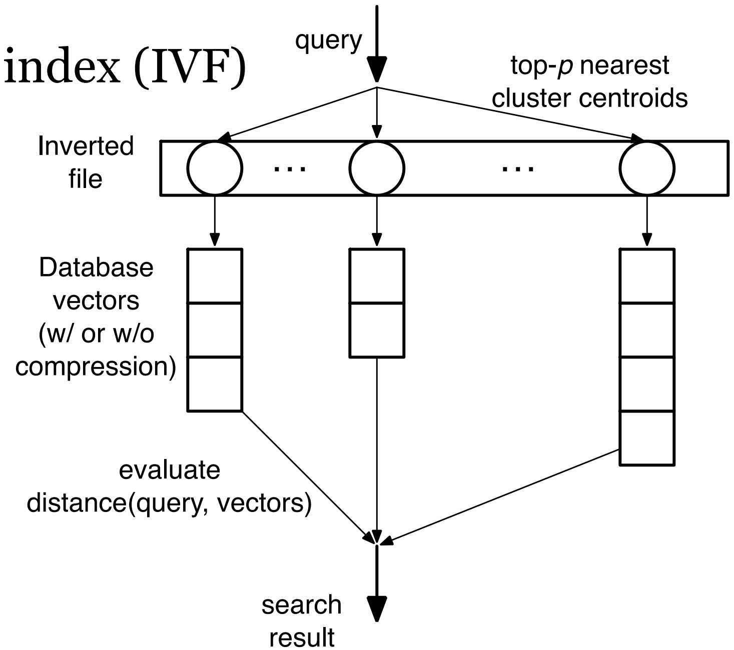
Need for Interactive Analytics
Approximate Search

Publications

Li-SIGMOD'20

- interaction with users offers new opportunities
  - progressive answers
    - produce intermediate results
      - iteratively converge to final, correct solution
      - Inverted files

Inverted file index (IVF)



Slide by C. Li

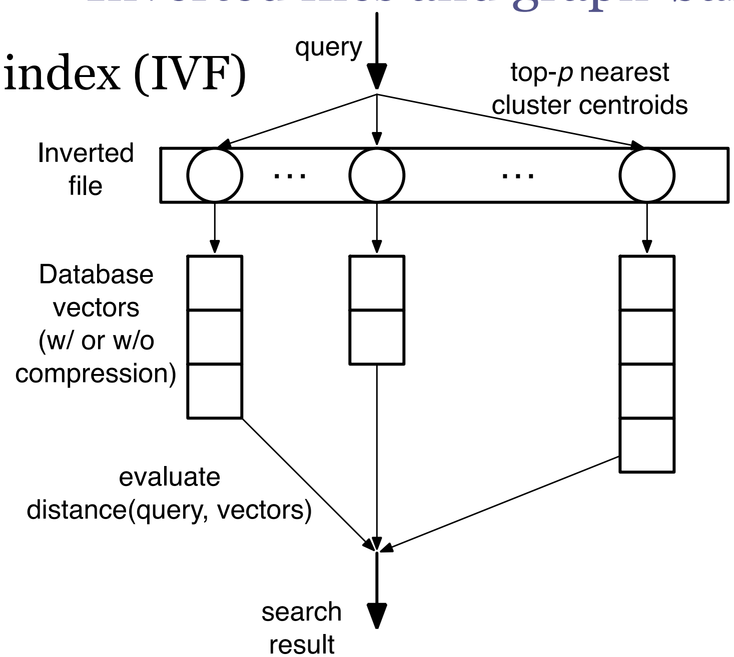
Need for Interactive Analytics
Approximate Search

Publications

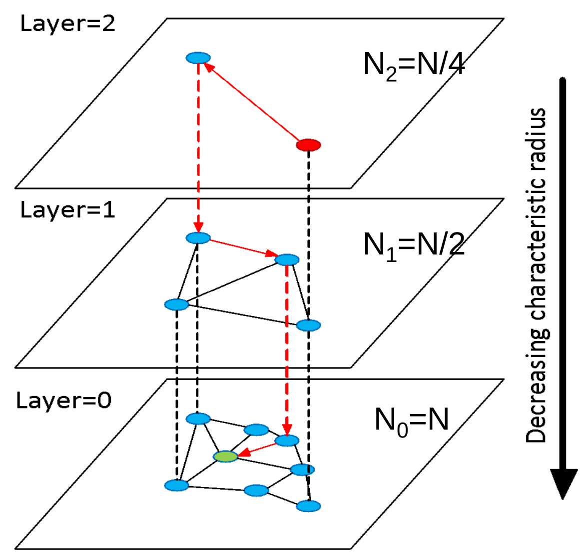
Li-SIGMOD'20

- interaction with users offers new opportunities
  - progressive answers
    - produce intermediate results
      - iteratively converge to final, correct solution
      - Inverted files and graph-based indexes

Inverted file index (IVF)



HNSW



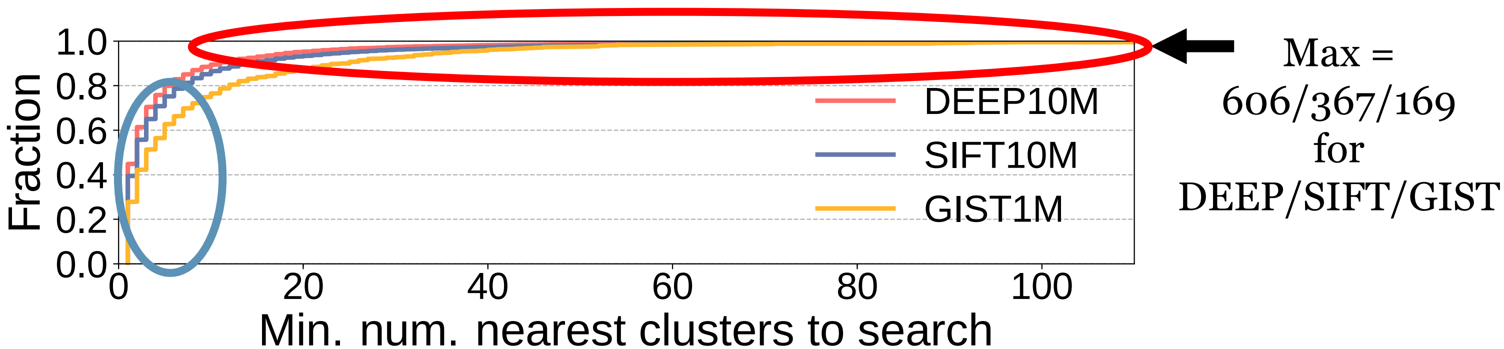
Slide by C. Li

Need for Interactive Analytics
Approximate Search

Publications

Li-SIGMOD'20

- Search termination condition varies greatly
  - Inverted indexes: number of nearest clusters
  - Graph-based indexes: Minimum number of distance evaluations.



IVF index: CDF of min. termination conditions among queries. DEEP10M and SIFT10M have 4000 clusters and GIST1M has 1000 clusters in total.

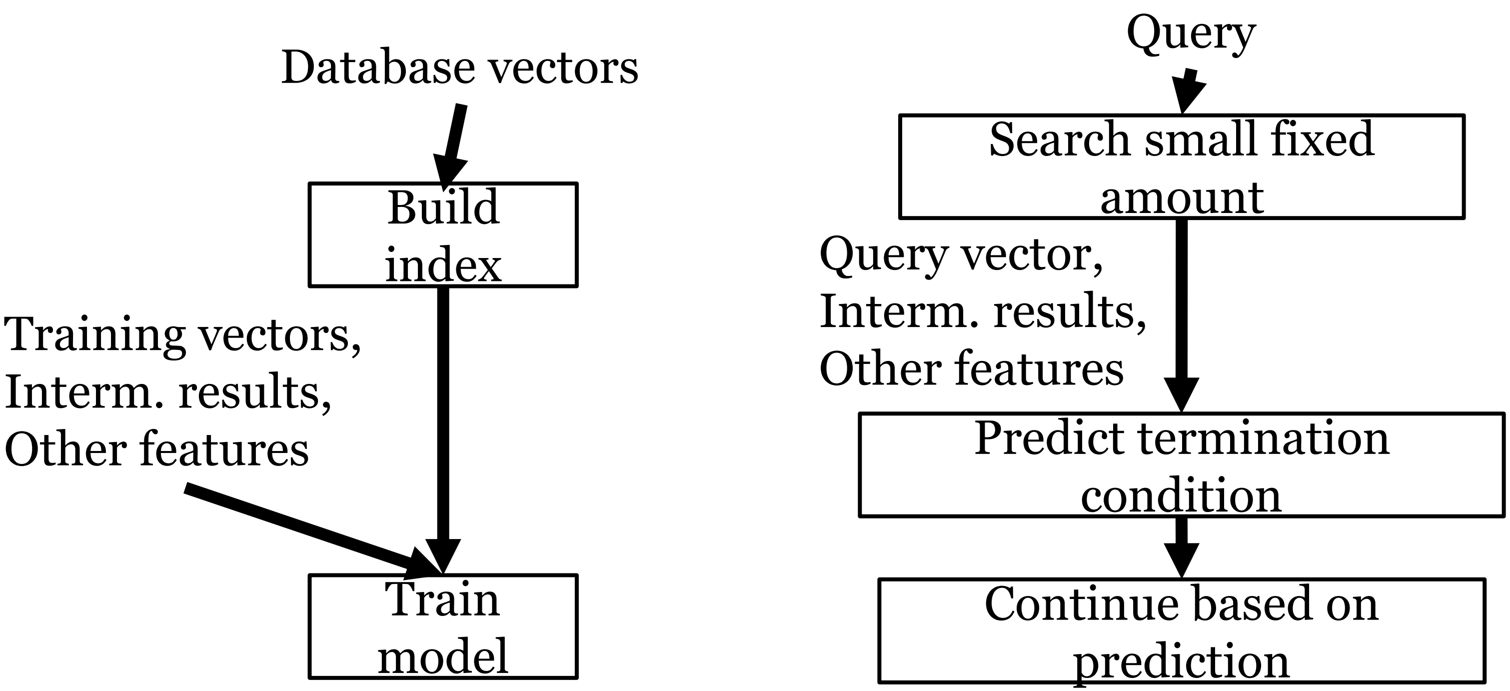
Need for Interactive Analytics
Approximate Search

Publications

Li-SIGMOD'20

- Learned Adaptive Early Termination

At index construction: At online search:



On average, end-to-end latency is improved by up to 7.1x under the same accuracy targets

Slide by C. Li

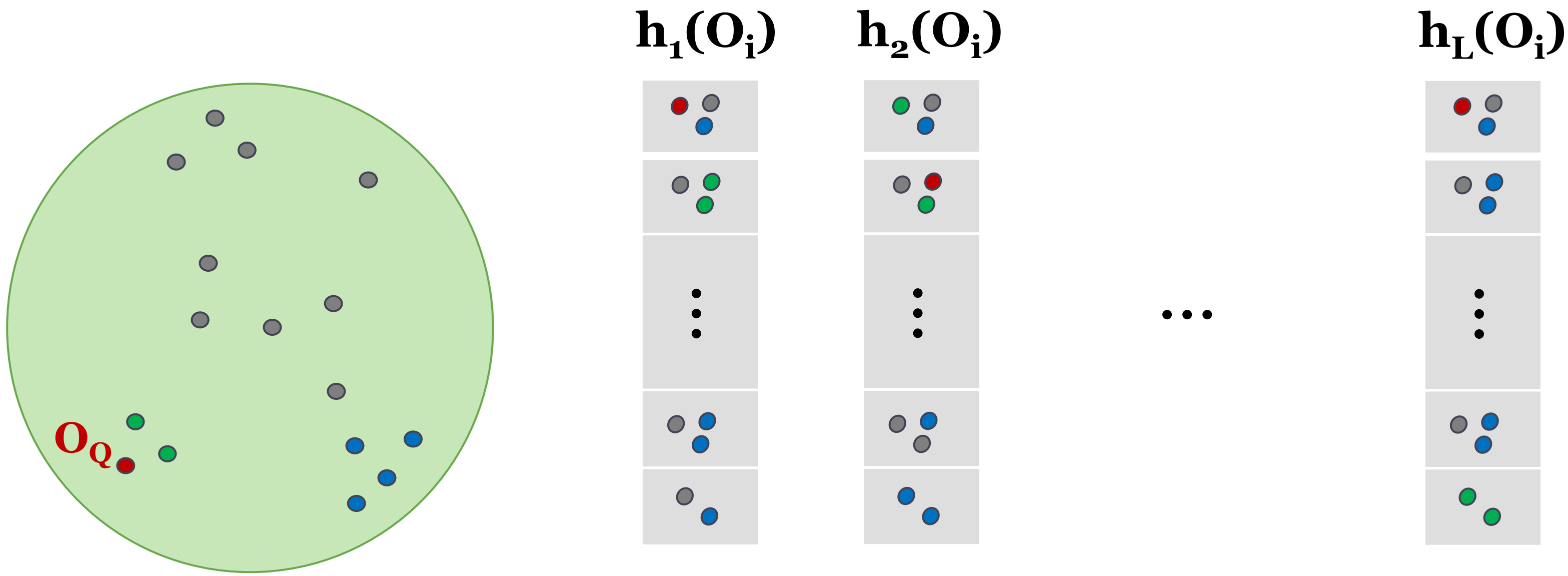
AI and Similarity Search

AI and Similarity Search

- Representation Learning
  - Learned hashing
  - Learned quantization
  - Learned summarizations for data series
- Search and Indexing
  - Learned indexes
  - Similarity search on deep network embeddings

AI and Similarity Search
Representation Learning

- Learned Hashing
  - Prior works:
    - Classical locality sensitive hashing. Typically data insensitive



Each object O is mapped to a single bucket in each of the L hash tables using hash function $h_j(O)$

AI and Similarity Search
Representation Learning

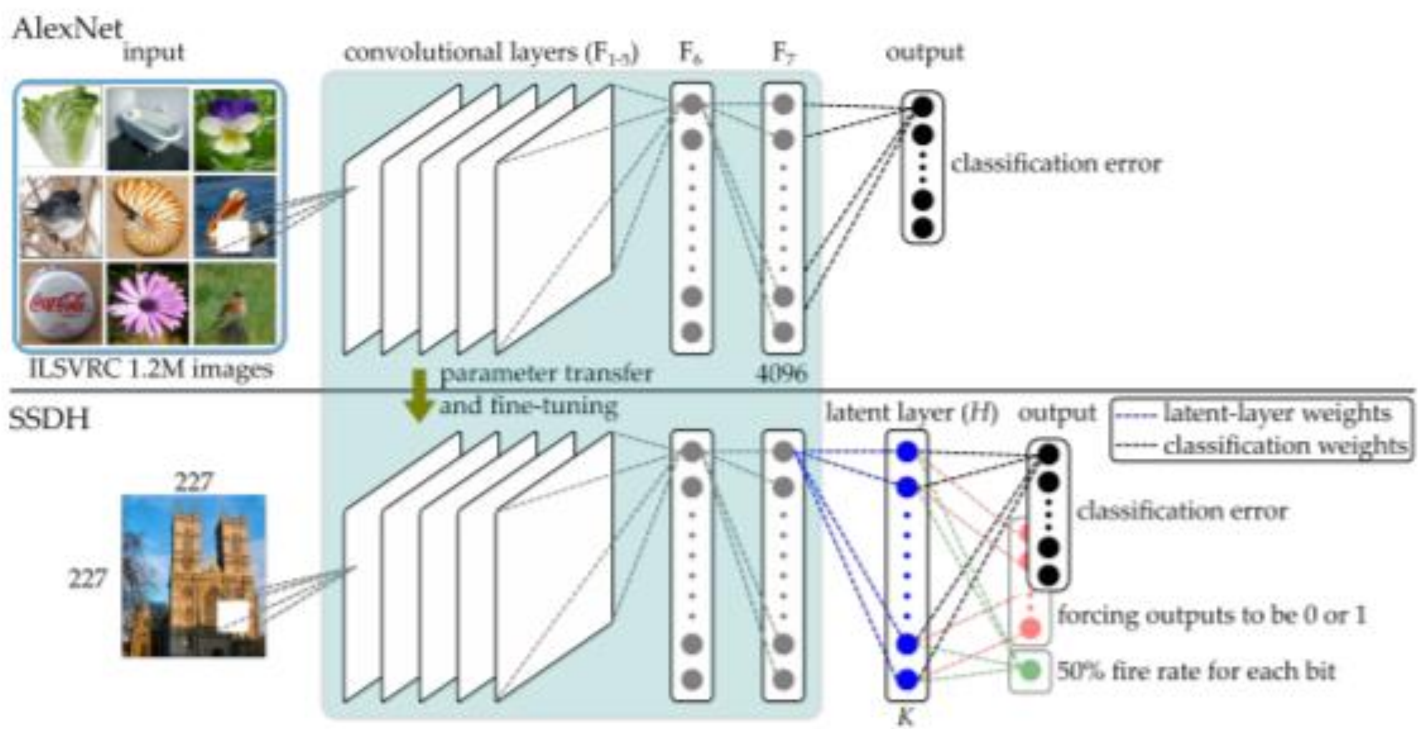
Publications
- Salakhutdinov-IJAR'09
- Weiss-NIPS'09

- Learned Hashing
  - Main goal: learn compact encodings that preserve similarity
  - Early works: semantic hashing, spectral hashing
    - Learn projection vectors instead of the random projections
  - A large body of follow-up work on data-sensitive approaches
    - http://cs.nju.edu.cn/lwj/slides/L2H.pdf
    - https://learning2hash.github.io/
  - Deep-learning approaches

AI and Similarity Search
Representation Learning

Publications
- Yang-TPAMI'18
- Krizhevsky-NIPS'2012

- Deep-Learned Hashing
  - Main Idea:
    - Modify conventional DNN models (eg, AlexNet classification model) by replacing ouput layer with deep hashing modules



AI and Similarity Search

Representation Learning

- Deep-Learned Hashing

Publications

Cai-Arxiv'17

Luo-Arxiv'20

Wang-TPAMI'18

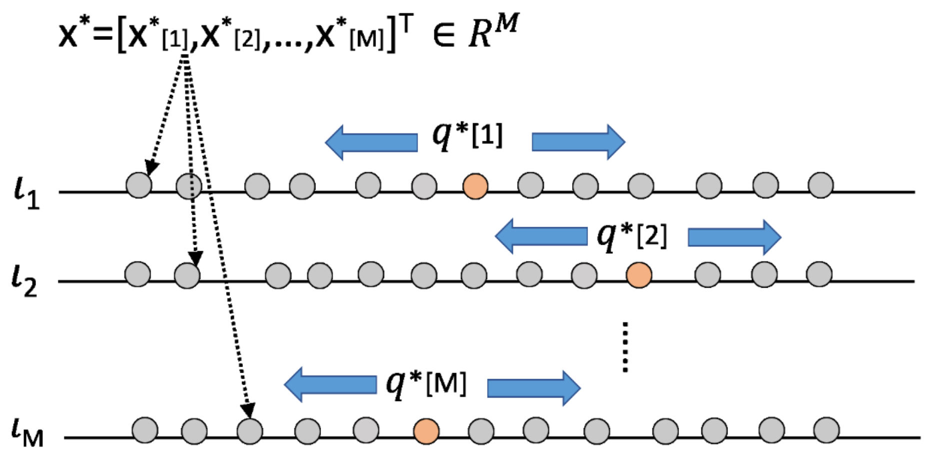
Network	AE, CNN, GAN, Siamese/Triplets, Attention Networks, etc.
Loss Functions	Pair-wise similarity, multi-wise similarity, semantic similarity (label-based), quantization loss, regularization loss, etc.
Optimization	Backpropagation, relaxation, optimizing subproblems, continuation

AI and Similarity Search
Representation Learning

Publications

Li et al. – ICDE'20

- Deep-Learned Hashing
  - OPFA, NeOPFA: approximate NN search for disk-based data
    - learn hashing (i.e., mapping) functions that map vectors to (lower dimensional) embeddings, preserving data locality
    - build indexes (e.g.,. B+-trees) on lists of values of individual dimensions of the embeddings
    - query answering makes bi-directional sequential access to each list, leading to sequential I/O



AI and Similarity Search
Representation Learning

- Deep-Learned Hashing
  - How do they compare?
    - Evaluation Metrics: precision, recall, search time
  - Conflicting results:
    - [Luo-20]: Deep-learned hashing greatly outperforms traditional hashing methods (e.g., SDH and KSH) overall.
    - [Cai17]: Deep-learned hashing is inferior to traditional hashing methods if the later exploit multiple hash tables.
  - [Sablayrolles17]:
    - Need better evaluation criteria: retrieval of unseen classes and transfer learning.

Publications

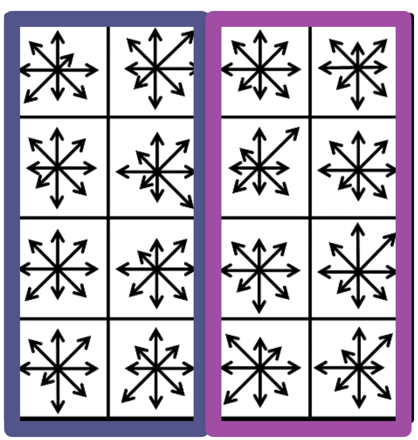
- Cai-Arxiv'17
- Luo-Arxiv'20
- Wang-TPAMI'18
- Sablayrolles-ICASSP'17

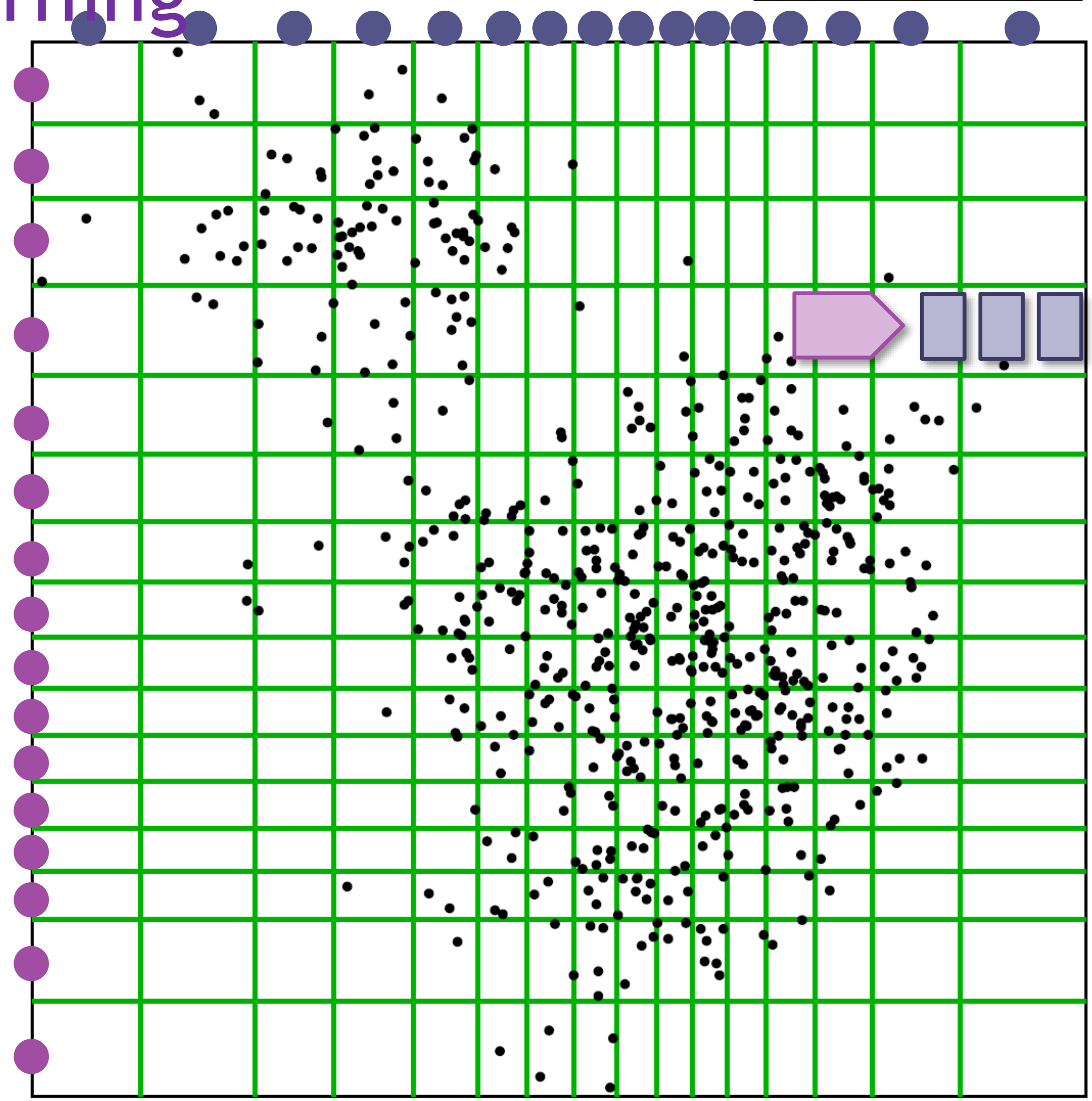
AI and Similarity Search
Representation Learning

Publications

Jegou et al. TPAMI' 11

- Learned Quantization Techniques
  - Prior works:
    - Product Quantization
    - Efficient search with lookup tables





Slide by A. Babenko

AI and Similarity Search
Representation Learning

Publications

Wang-CVPR'16

Cao-AAAI'16

- Learned Quantization
  - Main goal: learn encodings that minimize quantization errors
  - Early works:
    - SQ learns features and quantization separately
      - Exploits Semantic (label-based) loss.
    - DQN learns them simulatenaneously
      - First end-to-end model
      - Combines a similarity-preserving loss and a product quantization loss.
      - But DQN's codebook is trained with k-means clustering.
  - No exhaustive survey

  - we will focus on state-of-the-art deep-learning approaches

AI and Similarity Search
Representation Learning

Publications

Klein-CVPR'19

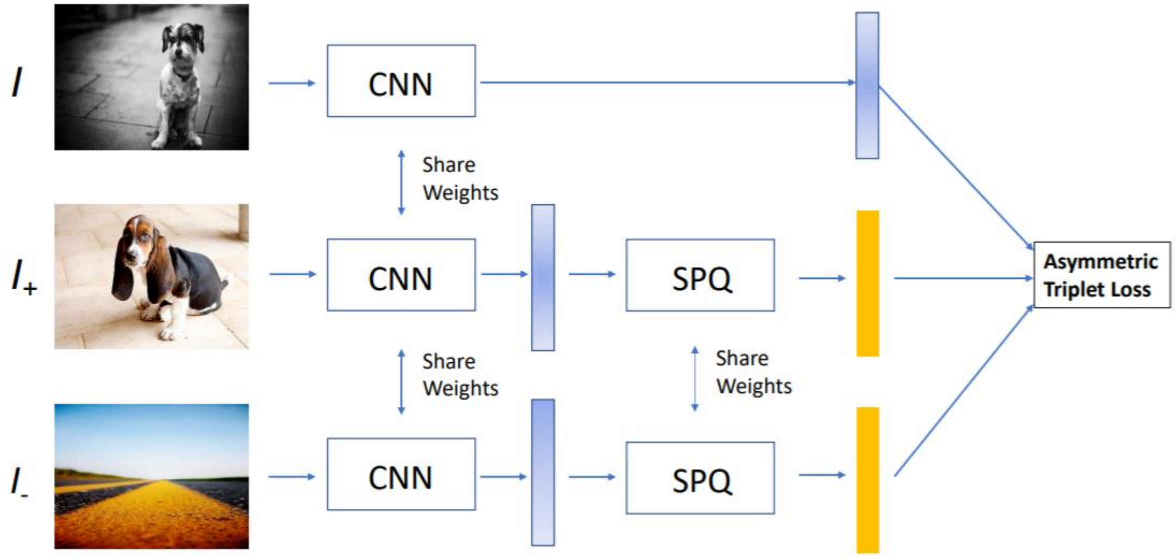
- Supervised Learned Quantization
  - DPQ:
    - Learns centroids and parameters end-to-end
    - Learns a cascade of two fully-connected layers followed by a softmax layer to determine a soft codeword assignment.
    - In contrast to original PQ, codeword assignment is no longer determined by distance between the original feature and codewords.

AI and Similarity Search
Representation Learning

Publications

Yu-ECCV'18

- Supervised Learned Quantization
  - PQN:
    - Codewords are assigned based on similarity between the original features and codewords
    - Less prone to over-fitting compared to DPQ due to the smaller number of parameters.

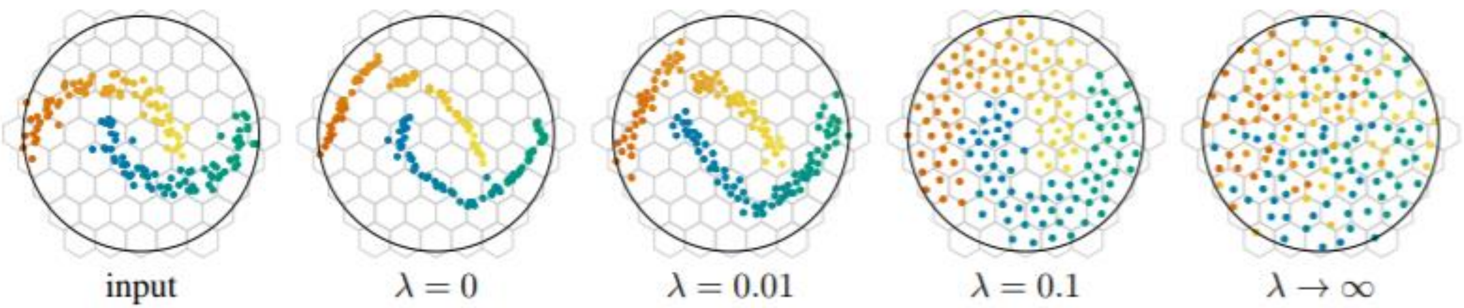


AI and Similarity Search
Representation Learning

Publications

Sablayrolles-ICLR'19

- Unsupervised Learned Quantization
  - Catalyst-Lattice
    - Idea: adapt the data to the quantizer rather than the opposite
      - Train a neural network that maps input features to a uniform output distribution on a unit hypersphere, making high-dimensional indexing more accurate

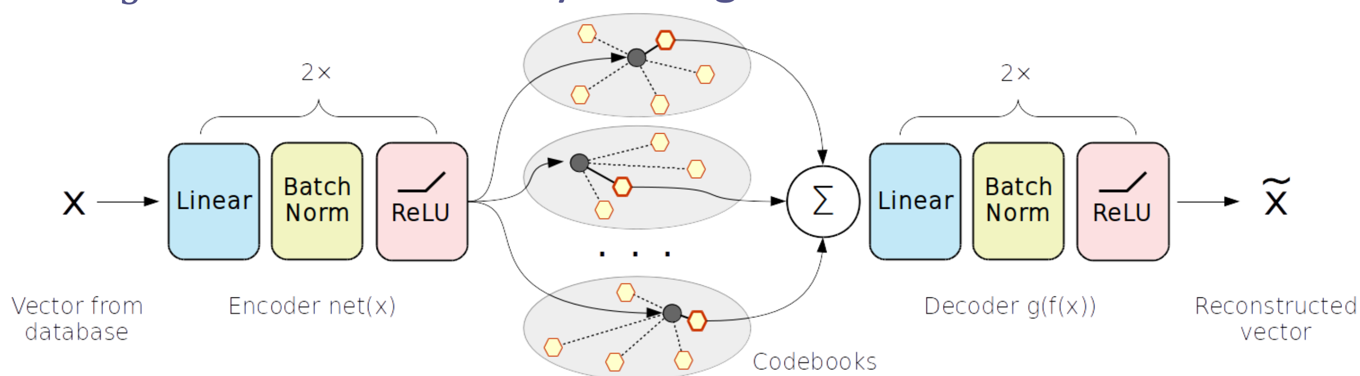


AI and Similarity Search
Representation Learning

Publications

Morozov-ICCV'19

- Unsupervised Learned Quantization
  - Unsupervised Neural Quantization (UNQ)
    - Idea: train multi-layer encoder/decoder in end-to-end fashion in unsupervised setup
  - UNQ Training Loss: $L = L_1 + \alpha L_2 + \beta L_3$
    - L_1 – reconstruction loss
    - L_2 – triplet loss in compressed domain
    - L_3 – enforces diversity among codebooks



AI and Similarity Search
Representation Learning

Publications

Morozov-ICCV'19

Slide by S. Morozov

- Learned Quantization Techniques
  - UNQ vs. Catalyst-Lattice[1] and LSQ[2]

Method	BigANN1B			Deep1B		
	R@1	R@10	R@100	R@1	R@10	R@100
	8 bytes per vector					
Catalyst+Lattice[1]	10.4	37.6	76.6	**16.8**	**38.7**	68.2
LSQ[2]	9.6	35.9	73.3	13.2	32.3	59.9
LSQ+rerank	9.9	36.1	73.8	12.3	31.6	59.7
UNQ	**13.0**	**44.5**	**82.4**	14.5	37.8	**68.5**
	16 bytes per vector					
Catalyst+Lattice	31.1	77.8	98.3	35.3	72.8	95.6
LSQ	38.0	85.6	99.3	30.5	65.0	91.1
LSQ+rerank	37.6	86.0	99.3	30.1	65.8	91.4
UNQ	**38.3**	**86.8**	**99.4**	**35.5**	**74.2**	**96.1**

[1] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid and Hervé Jégou. Spreading vectors for similarity search. ICLR'19
[2] Julieta Martinez, Shobhit Zakhmi, Holger H. Hoos, and James J. Little. LSQ++: lower running time and higher recall in multi-codebook quantization, ECCV'2018

Slide by S. Morozov

AI and Similarity Search
Representation Learning for Data Series

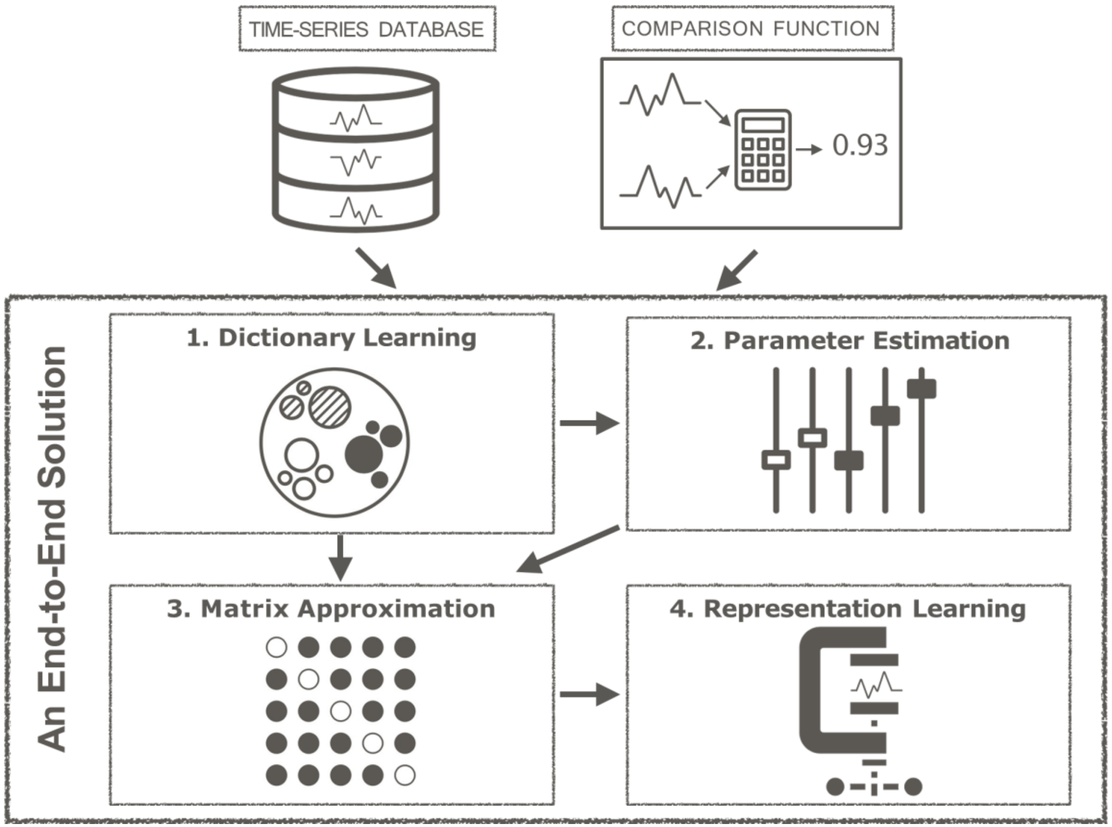
- learn compact similarity-preserving representations
- use those for
  - similarity search
  - classification
  - clustering
  - …

AI and Similarity Search
Representation Learning for Data Series

Publications

Paparrizos - PVLDB'19

- GRAIL
  - learns representations that preserve a user-defined comparison function
  - for a given comparison function:
    - extracts landmark series using clustering
    - optimizes parameters
    - exploits approximations for kernel methods to construct representations by expressing each series as a combination of the landmark series



AI and Similarity Search

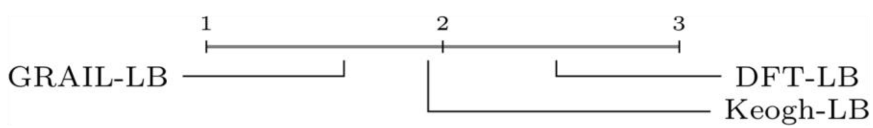
Representation Learning for Data Series

Publications

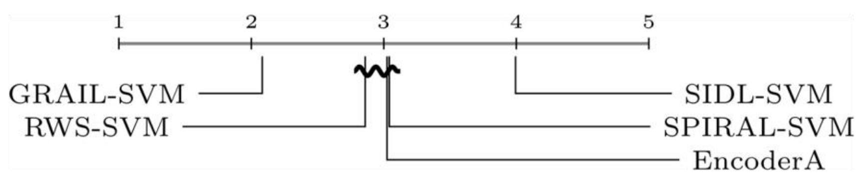
Paparrizos - PVLDB'19

- GRAIL
  - uses the learned representations for querying, classification, clustering, …

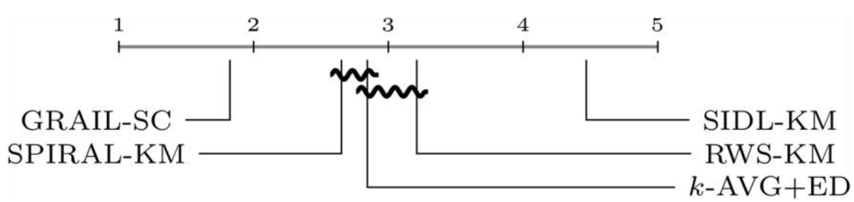
QUERYING: GRAIL Lower Bound vs. Lower Bounds for DFT & DTW



CLASSIFICATION: GRAIL with SVM vs. other Learned Representations



CLUSTERING: GRAIL with Spectral Clustering vs. other Learned Representations



AI and Similarity Search

Representation Learning for Data Series

Publications

Wang - KDD'21

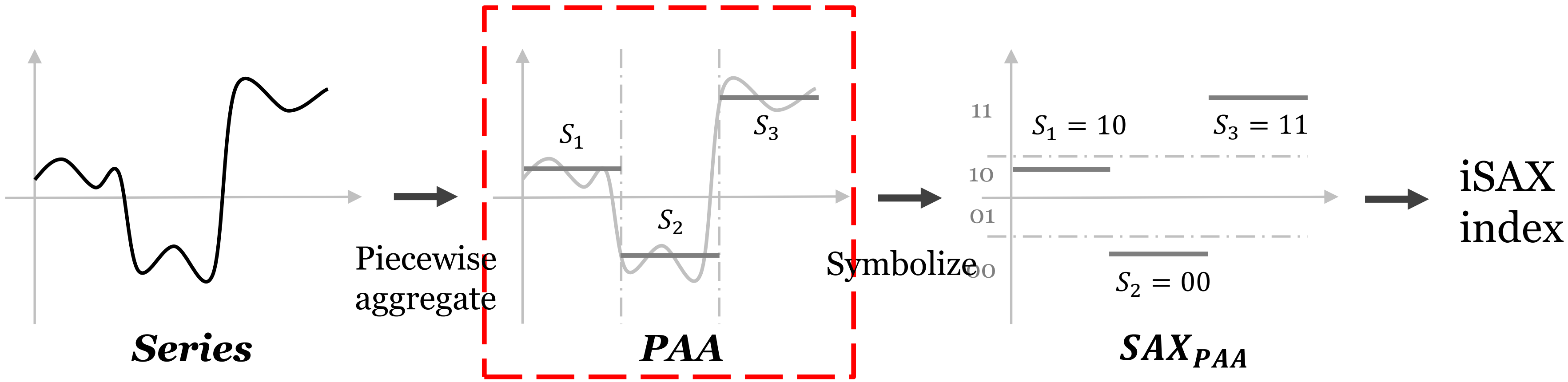
- Series Approximation Network (SEAnet)
  - novel autoencoder architecture
  - learns deep embedding approximations
  - uses those for similarity search

AI and Similarity Search
Representation Learning for Data Series

Publications

Wang - KDD'21

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

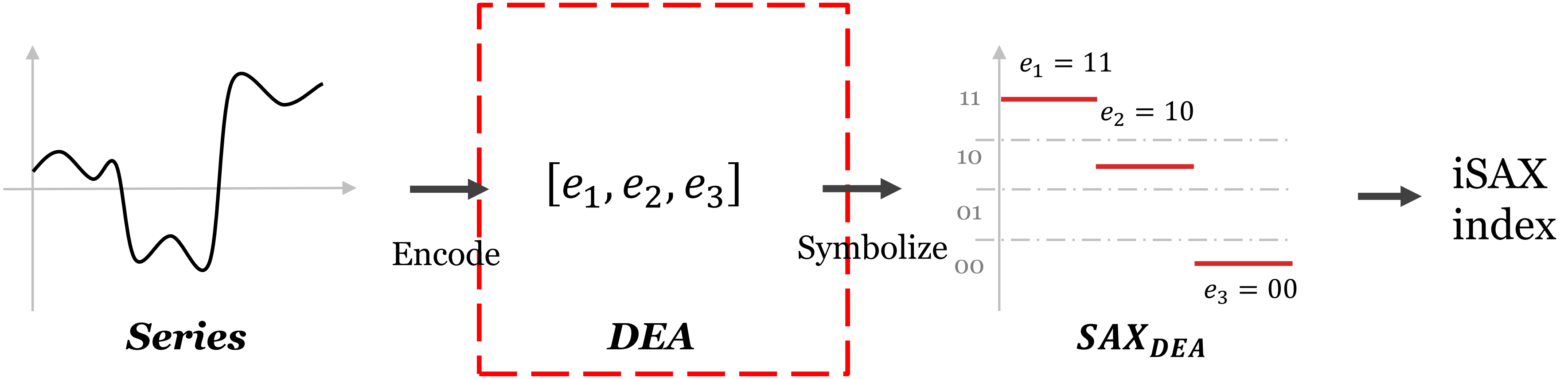


AI and Similarity Search
Representation Learning for Data Series

Publications

Wang - KDD’21

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



AI and Similarity Search

Representation Learning for Data Series

Publications

Wang - KDD'21

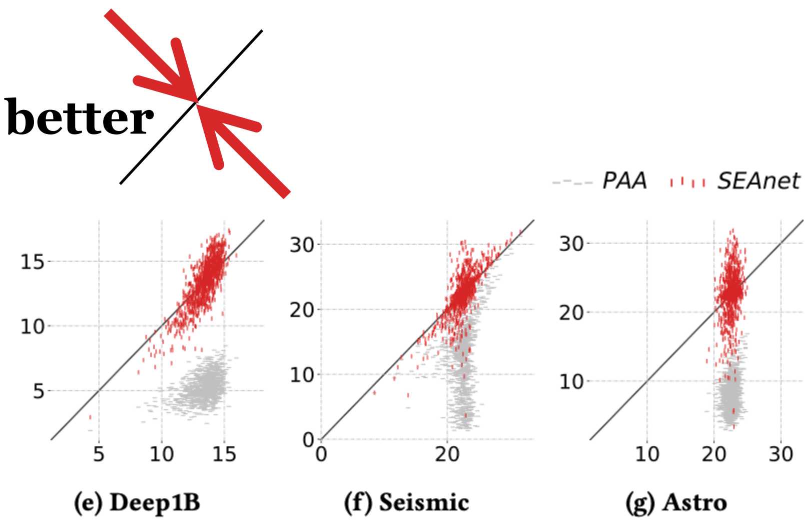
- Series Approximation Network (SEAnet)
  - is an exponentially dilated ResNet architecture + Sum of Squares regularization
  - minimizes
    - reconstruction error
    - difference between distance of two vectors in embedded space and distance in original space

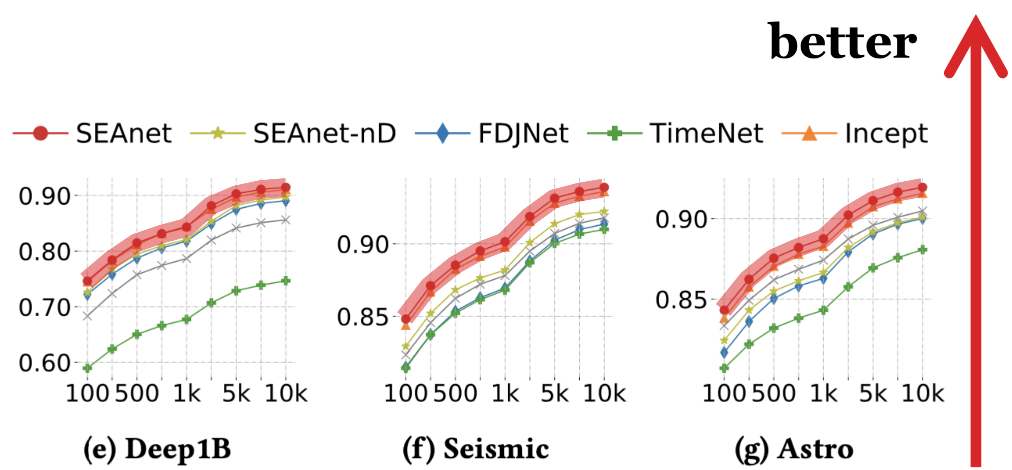
AI and Similarity Search
Representation Learning for Data Series

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





AI and Similarity Search
Search and Indexing

- Search and Indexing
  - Problem:
    - High-d vector similarity search is hard
    - Massive datasets and high dimensionality in 100s-1000s
    - Sophisticated indexing structures and search algorithms
  - Solutions:
    - Learned Indexes
    - Improve search efficiency using deep learning
    - Indexing for learned embeddings

AI and Similarity Search
Search and Indexing

Publications

Kraska-SIGMOD'18

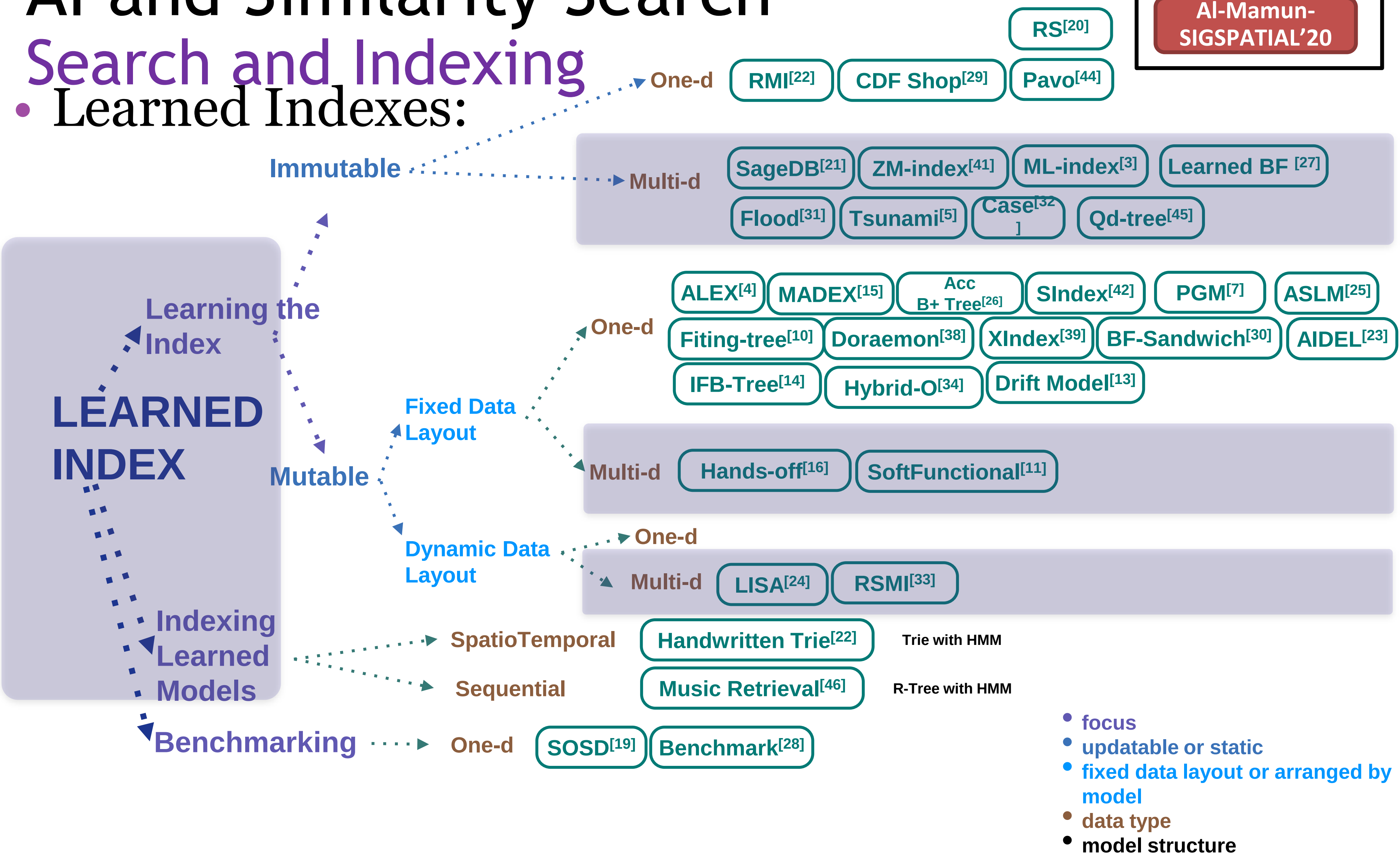
Al-Mamun-SIGSPATIAL'20

- Learned Indexes:
  - Main idea: replace an index with a learned model
    - One-dimensional learned indexes
      - Seminal work: The Case for Learned Indexes
    - Multi-dimensional indexes
      - Exhaustive tutorial on this topic at SIGSPATIAL'20: https://www.cs.purdue.edu/homes/aref/learned-indexes-tutorial.html
    - Some initial attempts for similarity search
  - Main challenges for multi-dimensional indexes:
    - How to sort the data?
    - How to correct prediction errors?
    - Which ML model to choose?
    - How to store the data?

AI and Similarity Search

Search and Indexing

- Learned Indexes:



Slide from W. Aref

AI and Similarity Search
Search and Indexing

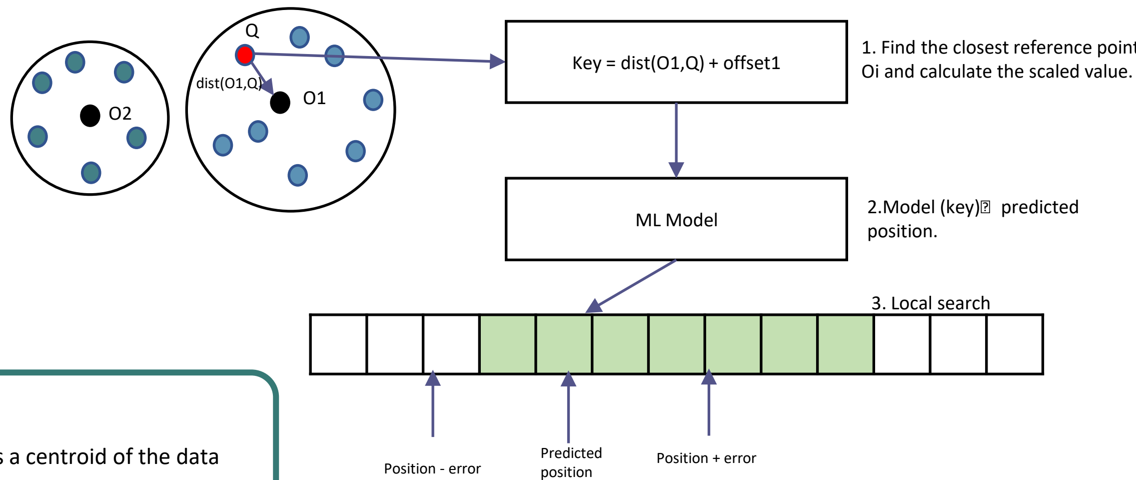
Publications: Davitkova-EDBT'20

- Learned Indexes for similarity search:
  - The ML-Index: A multidimensional, learned index for point, range and NN queries

Core Idea

ML-Index:

- Z/Morton order cannot be easily learned by ML models.
- Multi-dimensional data should be sorted in an order that can be easily learned.
- Partition and transform the data into one-dimensional values based on distribution-aware reference points.
- Combines the scaled ordering with ML models



Query Processing (Point)

Efficient Scaling

Offset Method:

- *m* reference points *Oi* are chosen each can be thought as a centroid of the data partition *Pi*.
- The closest reference points of *Oi* are used to build the partition *Pi*.
- The minimal distance of a point to the reference points is *dl*
- Scaled value = $offset_i + dist(O_i, dl)$
- For reference points *O1, O2,…Om* and their partitions *P1,P2,…Pm*,
- r: The maximal distance from *Oj* to the points in partition *Pj*

Slide from W. Aref

Publications

Davitkova-EDBT'20

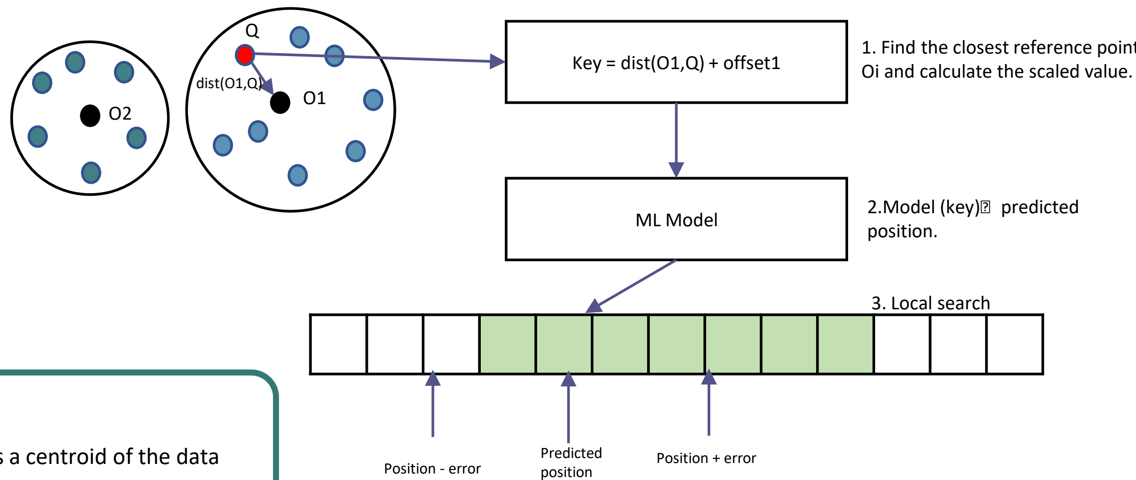
AI and Similarity Search
Search and Indexing

- Learned Indexes for similarity search:
  - The ML-Index: A multidimensional, learned index for point, range and NN queries

Core Idea

ML-Index:

- Z/Morton order cannot be easily learned by ML models.
- Multi-dimensional data should be sorted in an order that can be easily learned.
- Partition and transform the data into one-dimensional values based on distribution-aware reference points.
- Combines the scaled ordering with ML models



Query Processing (Point)

Efficient Scaling

Offset Method:

- *m* reference points *Oi* are chosen each can be thought as a centroid of the data partition *Pi*.
- The closest reference points of *Oi* are used to build the partition *Pi*.
- The minimal distance of a point to the reference points is *dl*
- Scaled value = $offset_i + dist(O_i, dl)$
- For reference points *O1, O2,…Om* and their partitions *P1,P2,…Pm*,
- r: The maximal distance from *Oj* to the points in partition *Pj*

Slide from W. Aref

AI and Similarity Search

Search and Indexing

- Learned Indexes for similarity search:
  - Effectively Learning Spatial Indices

Publications

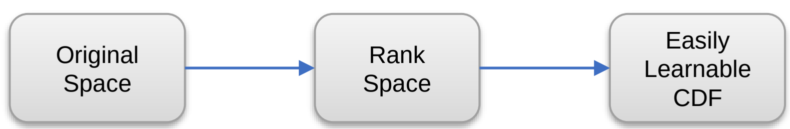
Qi-PVLDB'20

Motivation

- Selecting grid resolution for Z-order for learned multi-dimensional index (e.g. ZM-Index[41]) is difficult:
  - Large cells
    - More false positives due to many points per cell
  - Small cells
    - Hard to learn due to uneven gaps in Cumulative Distribution Function (CDF)

Core Idea

- Spatial index based on ordering the data points by a rank space-based transformation*
  - Simplify the indexing functions to be learned
  - M(search keys) ⟹ disk block Ids (location)



- For scaling to large datasets, proposes:
  - Introduce a Recursive Spatial Model Index (RSMI) (in lieu of RMI)
- Support point, window, and kNN queries
- Support updates

Slide from W. Aref

AI and Similarity Search

Search and Indexing

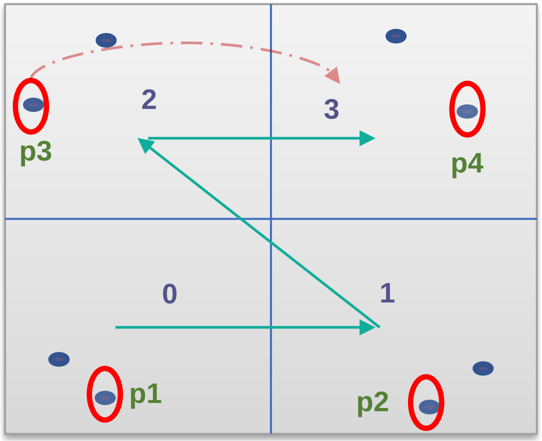
Publications

Qi-PVLDB'20

- Learned Indexes for similarity search:
  - Effectively Learning Spatial Indices

RSMI

- Recursive Spatial Model Index (RSMI):
  - Recursively partitions a dataset
  - Partitioning is learned over the distribution of data
- Steps:
  - Initially distribute the data into equal sized partitions
  - Use a Space Filling Curve (SFC) to assign Ids to partitions
  - Learn the partition Ids using a model $M_{0,0}$
  - Rearrange the data based on the prediction of $M_{0,0}$
  - Recursively repartition
    - Until each partition can be learned with a simple model



Point	p1	p2	p3	p4
Initial partition Id	0	1	2	3
Model predicted Id	0	1	3	3
Learned partition Id	0	1	3	3

Discussion

- Window and kNN query results are highly accurate but not exact.
  - i.e., over 87% across a variety of settings
  - Separate mechanism has been proposed for exact answer.
- Does not support query for spatial objects with non-zero extent

Slide from W. Aref

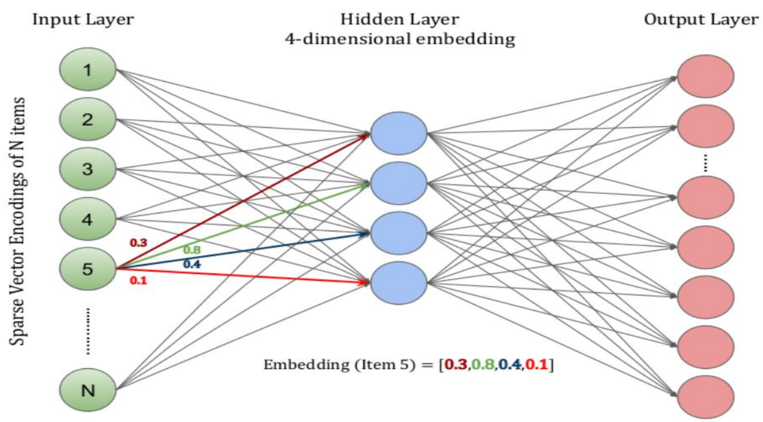
AI and Similarity Search
Search and Indexing

Publications

Echihabi-PVLDB'19

- Indexing Deep Network Embeddings (DNE)

sequences
text
images
video
graphs
…



deep embeddings
high-d vectors learned using a DNN

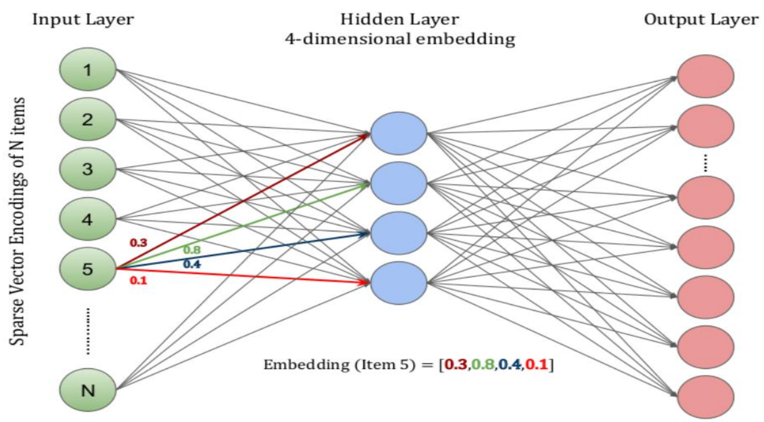
AI and Similarity Search
Search and Indexing

Publications

Echihabi-PVLDB'19

- Indexing Deep Network Embeddings (DNE)

sequences
text
images
video
graphs
...



deep embeddings
high-d vectors learned using a DNN

  - Data series techniques provide effective/scalable similarity search over DNE
  - They outperform hashing-based, quantization-based inverted indexes and kNN graphs on many scenarios

High-d Similarity Search: Challenges and Open Problems

Challenges and Open Problems

- we are still far from having solved the problem
- several challenges remain in terms of
  - usability, ease of use
  - scalability, distribution
  - benchmarking
- these challenges derive from modern data science applications

Challenges and Open Problems
Outline

- benchmarking
- interactive analytics
- parallelization and distribution
- deep learning

Massive High-d Data Collections

Publications

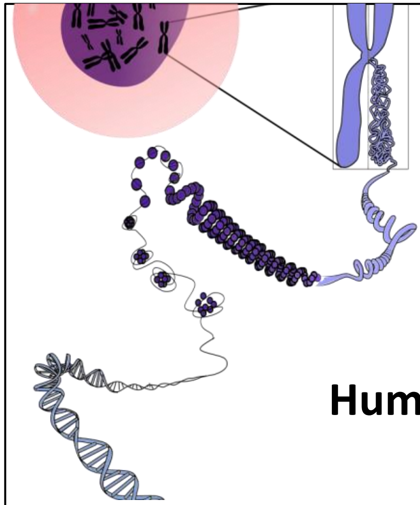
Palpanas-SIGREC'19

NASA's Solar Observatory

1.5 TB per day

Large Synoptic Survey Telescope (2019)

~30 TB per night



Human Genome project

130 TB

passenger aircrafts

20 TB per hour

data center and services monitoring

2B data series

4M points/sec

facebook

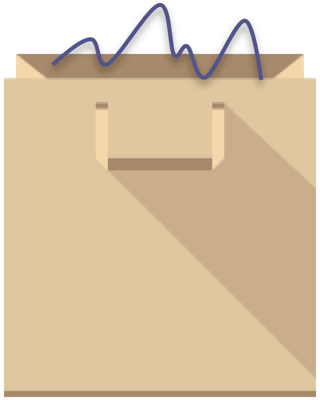
Challenges and Open Problems Outline

- benchmarking
- interactive analytics
- parallelization and distribution
- deep learning

Previous Studies

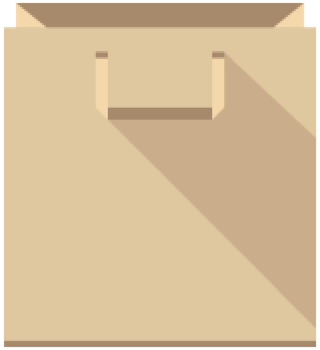
evaluate **performance** of **indexing methods** using **random queries**

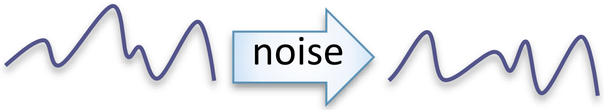
- chosen from the data (with/without noise)



Previous Studies

With or without noise





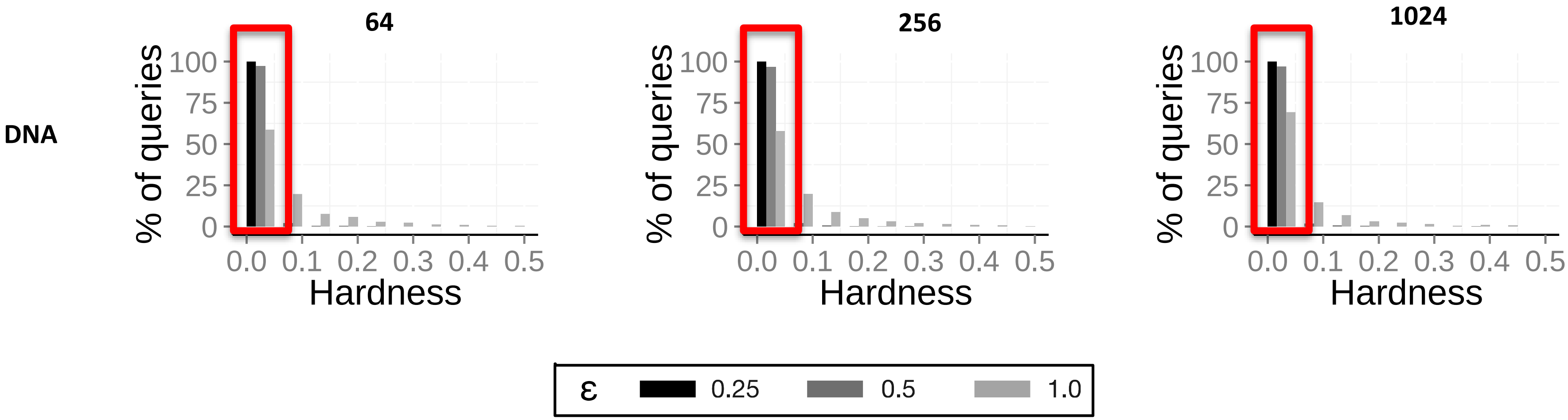
Publications

Zoumbatianos KDD'15

Zoumbatianos TKDE'18

Previous Workloads

Most previous workloads are ***skewed*** to ***easy*** queries



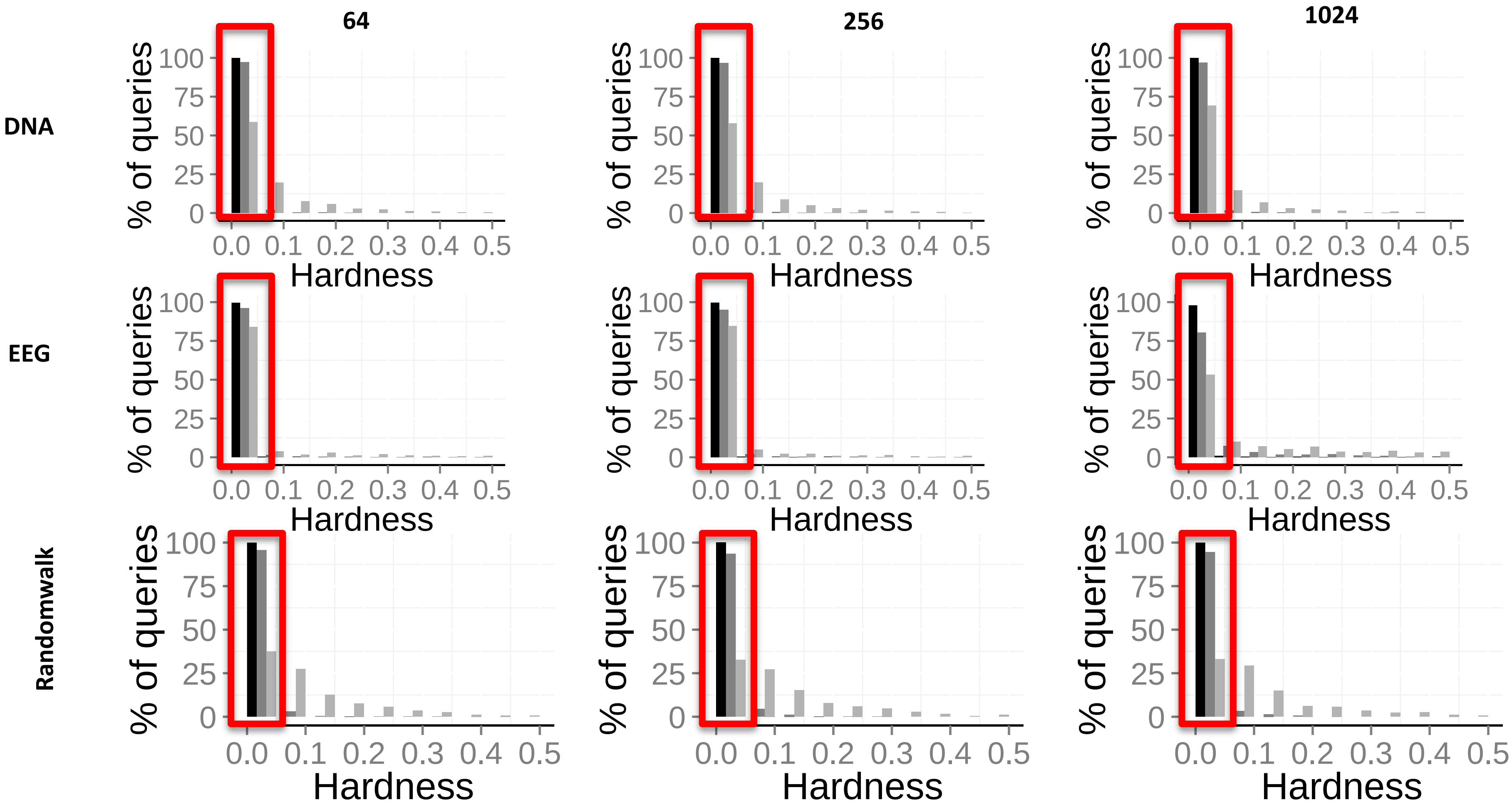
Publications

Zoumbatianos KDD'15

Zoumbatianos TKDE'18

Previous Workloads

Most previous workloads are ***skewed*** to ***easy*** queries



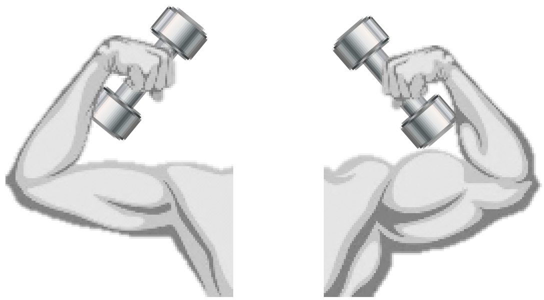
Publications

Zoumbatianos KDD’15

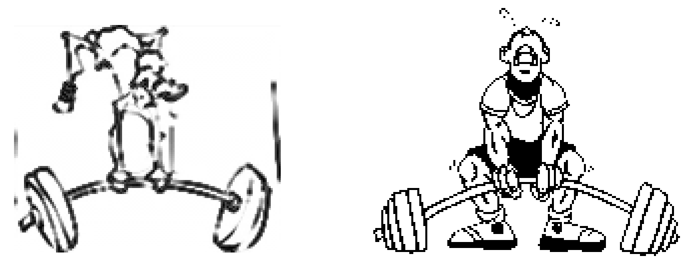
Zoumbatianos TKDE’18

Benchmark Workloads

If all queries are **easy**
all indexes look **good**



If all queries are **hard**
all indexes look **bad**



need **methods** for **generating** queries of **varying hardness**

Publications

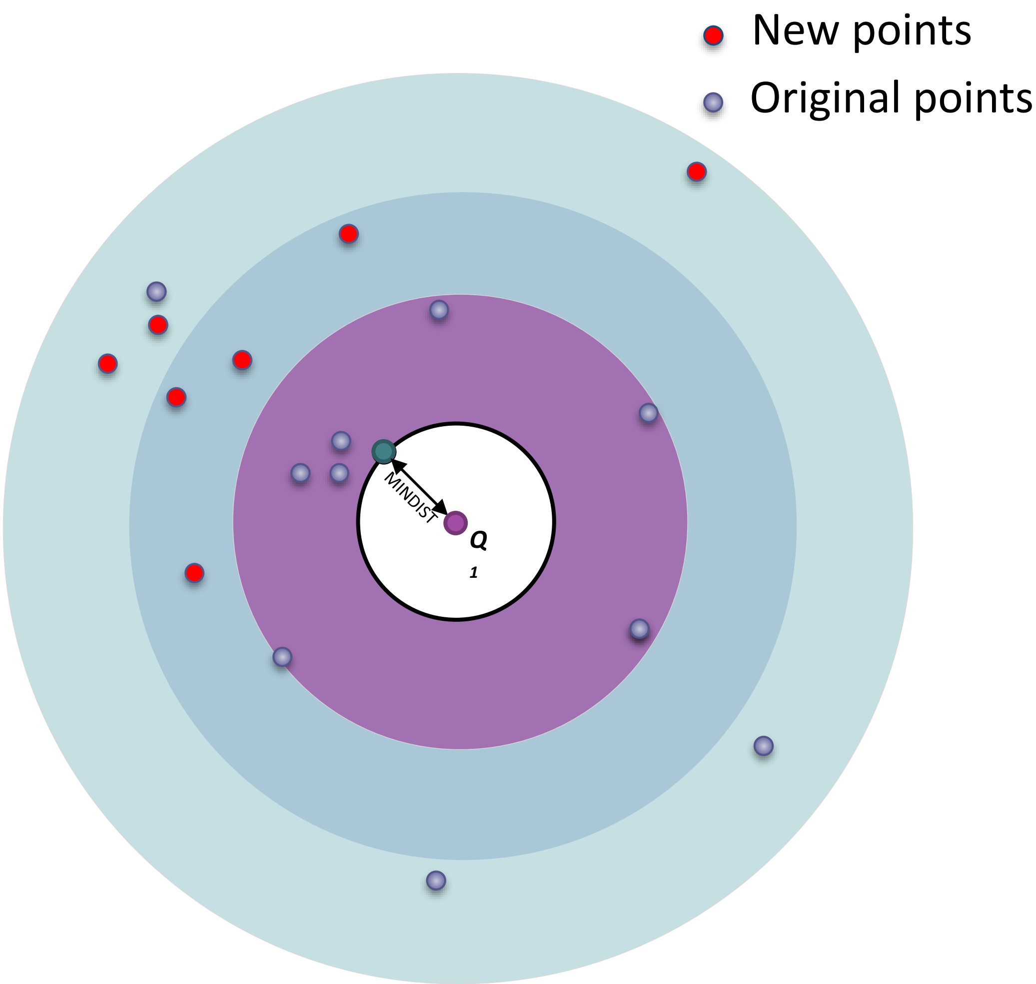
Zoumbatianos KDD’15

Zoumbatianos TKDE’18

Densification Method:
Equi-densification

Distribute points such that:
The **worse** a summarization
the more data it checks

Equal number of points in every “zone”



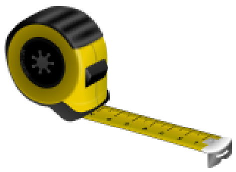
Publications

Zoumbatianos KDD'15

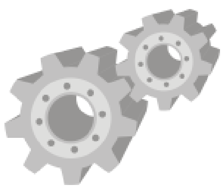
Zoumbatianos TKDE'18

Summary

Pros:



Theoretical background
Methodology for characterizing
NN queries for data series indexes



Nearest neighbor query workload generator
Designed to stress-test data series indexes
at varying levels of difficulty

Cons:

Time complexity
Need new approach to scale to very large datasets

Challenges and Open Problems
Outline

- benchmarking
- interactive analytics
- parallelization and distribution
- deep learning

Need for Interactive Analytics

- 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

- several exciting research problems in intersection of visualization and data management
  - *frontend*: HCI/visualizations for querying/results display
  - *backend*: efficiently supporting these operations

Challenges and Open Problems
Outline

- benchmarking
- interactive analytics
- parallelization and distribution
- deep learning

Need for Parallelization/Distribution

Publications

Palpanas-HPCS'17

- further scale-up and scale-out possible!
  - techniques inherently parallelizable
    - across cores, across machines
- need to
  - propose methods for concurrent query answering
  - combine multi-core and distributed methods
  - examine FPGA and NVM technologies
- more involved solutions required when optimizing for energy
  - reducing execution time is relatively easy
  - minimizing total work (energy) is more challenging

Challenges and Open Problems
Outline

- benchmarking
- interactive analytics
- parallelization and distribution
- deep learning

Connections to Deep Learning

- data series indexing for deep embeddings
  - deep embeddings are high-d vectors
  - data series techniques provide effective/scalable similarity search

- deep learning for summarizing high-d vectors
  - different representations for different high-d data types
  - eg, autoencoders can learn efficient data series summaries

Connections to Deep Learning

- data series indexing for deep embeddings
  - deep embeddings are high-d vectors
  - data series techniques provide effective/scalable similarity search
- deep learning for summarizing high-d vectors
  - different representations for different high-d data types
  - eg, autoencoders can learn efficient data series summaries
- deep learning for designing index data structures
  - learn an index for similarity search

Connections to Deep Learning

- data series indexing for deep embeddings
  - deep embeddings are high-d vectors
  - data series techniques provide effective/scalable similarity search
- deep learning for summarizing high-d vectors
  - different representations for different high-d data types
  - eg, autoencoders can learn efficient data series summaries
- deep learning for designing index data structures
  - learn an index for similarity search
- deep learning for query optimization
  - search space is vast
  - learn optimization function

Connections to Deep Learning

- learning data distributions
  - answer approximate aggregate queries
- learning cardinality estimation
  - estimate query answering cost

Publications

- Thirumuruganathan-ICDE'20
- Sun et al. – SIGMOD'21

- deep learning for designing index data structures
  - learn an index for similarity search
- deep learning for query optimization
  - search space is vast
  - learn optimization function

Connections to Deep Learning

- data series indexing for deep embeddings
  - deep embeddings are high-d vectors
  - data series techniques provide effective/scalable similarity search
- deep learning for summarizing high-d vectors
  - different representations for different high-d data types
  - eg, autoencoders can learn efficient data series summaries
- deep learning for designing index data structures
  - learn an index for similarity search
- deep learning for query optimization
  - search space is vast
  - learn optimization function
- dimensionality of high-d vectors and overall dataset size are major challenges!
  - transfer learning to play an important role

Overall Conclusions

- High-d data is a very common data type
  - across several different domains and applications
- Complex analytics on high-d data are challenging
  - have very high complexity

Overall Conclusions

- High-d data is a very common data type
  - across several different domains and applications
- Complex analytics on high-d data are challenging
  - have very high complexity

- Data series indexing techniques provide state-of-the-art performance for
  - exact similarity search
  - approximate similarity search with quality guarantees
  - disk-based datasets

Overall Conclusions

- High-d data is a very common data type
  - across several different domains and applications
- Complex analytics on high-d data are challenging
  - have very high complexity

- Data series indexing techniques provide state-of-the-art performance for
  - exact similarity search
  - approximate similarity search with quality guarantees
  - disk-based datasets

- Several exciting research opportunities
  - parallel, progressive, and deep learning techniques can lead to further performance improvements

thank you!

google: **Karima Echihabi**
Kostas Zoumpatianos
Themis Palpanas

visit: http://**nestordb.com**

References (chronological order)

- Delaunay, Boris (1934). "Sur la sphère vide". Bulletin de l'Académie des Sciences de l'URSS, Classe des Sciences Mathématiques et Naturelles. **6**: 793–800.
- Ramer, U. (1972). An iterative procedure for the polygonal approximation of planar curves. *Computer Graphics and Image Processing*. 1: pp. 244-256.
- Douglas, D. H. & Peucker, T. K.(1973). Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature. *Canadian Cartographer*, Vol. 10, No. 2, December. pp. 112-122.
- Duda, R. O. and Hart, P. E. 1973. Pattern Classification and Scene Analysis. Wiley, New York.
- Jon Louis Bentley. 1975. Multidimensional binary search trees used for associative searching. Commun. ACM 18, 9 (Sept. 1975), 509–517.
- Pavlidis, T. (1976). Waveform segmentation through functional approximation. *IEEE Transactions on Computers*.
- Godfried T. Toussaint, The relative neighbourhood graph of a finite planar set, Pattern Recognition, Volume 12, Issue 4, 1980, Pages 261-268,
- Ishijima, M., et al. (1983). Scan-Along Polygonal Approximation for Data Compression of Electrocardiograms. *IEEE Transactions on Biomedical Engineering*. BME-30(11):723-729.
- N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: an efficient and robust access method for points and rectangles. In SIGMOD, pages 322–331, 1990.
- C. Faloutsos, M. Ranganathan, & Y. Manolopoulos. Fast Subsequence Matching in Time-Series Databases. In Proc. ACM SIGMOD Int'l Conf. on Management of Data, pp 419–429, 1994.
- McKee, J.J., Evans, N.E., & Owens, F.J. (1994). Efficient implementation of the Fan/SAPA-2 algorithm using fixed point arithmetic. *Automedica*. Vol. 16, pp 109-117.
- Koski, A., Juhola, M. & Meriste, M. (1995). Syntactic Recognition of ECG Signals By Attributed Finite Automata. *Pattern Recognition*, 28 (12), pp. 1927-1940.
- Seshadri P., Livny M. & Ramakrishnan R. (1995): SEQ: A Model for Sequence Databases. ICDE 1995: 232-239
- Shatkay, H. (1995). Approximate Queries and Representations for Large Data Sequences. *Technical Report cs-95-03*, Department of Computer Science, Brown University.
- Shatkay, H., & Zdonik, S. (1996). Approximate queries and representations for large data sequences. *Proceedings of the 12th IEEE International Conference on Data Engineering*. pp 546-553.
- Vullings, H.J.L.M., Verhaegen, M.H.G. & Verbruggen H.B. (1997). ECG Segmentation Using Time-Warping. *Proceedings of the 2nd International Symposium on Intelligent Data Analysis*.

References (chronological order)

- Keogh, E., & Smyth, P. (1997). A probabilistic approach to fast pattern matching in time series databases. *Proceedings of the 3rd International Conference of Knowledge Discovery and Data Mining*. pp 24-20.
- P. Ciaccia, M. Patella, and P. Zezula. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. Proceedings of VLDB'97, pp 426–435.
- Heckbert, P. S. & Garland, M. (1997). Survey of polygonal surface simplification algorithms, Multiresolution Surface Modeling Course. *Proceedings of the 24th International Conference on Computer Graphics and Interactive Techniques*.
- Piotr Indyk, Rajeev Motwani. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. STOC 1998.
- Qu, Y., Wang, C. & Wang, S. (1998). Supporting fast search in time series for movement patterns in multiples scales. *Proceedings of the 7th International Conference on Information and Knowledge Management*.
- Keogh, E., & Pazzani, M. (1998). An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. *Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining*. pp 239-241, AAAI Press.
- Hunter, J. & McIntosh, N. (1999). Knowledge-based event detection in complex time series data. *Artificial Intelligence in Medicine*. pp. 271-280. Springer.
- K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "'nearest neighbor'" meaningful? In *ICDT*, 1999.
- Keogh, E. & Pazzani, M. (1999). Relevance feedback retrieval of time series data. *Proceedings of the 22th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval.*
- P. Ciaccia and M. Patella. PAC Nearest Neighbor Queries: Approximate and Controlled Search in HighDimensional and Metric Spaces. In ICDE, pages 244– 255, 2000.
- H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, and A. El Abbadi. Vector Approximation Based Indexing for Non-uniform High Dimensional Data Sets. In CIKM, pp 202–209, 2000.
- J. Kleinberg. The Small-world Phenomenon: An Algorithmic Perspective. In Proceedings of the Thirty- second Annual ACM Symposium on Theory of Computing, STOC '00, pages 163–170, New York, NY, USA, 2000. ACM

References (chronological order)

- Lavrenko, V., Schmill, M., Lawrie, D., Ogilvie, P., Jensen, D., & Allan, J. (2000). Mining of Concurent Text and Time Series. *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining*. pp. 37-44.
- Wang, C. & Wang, S. (2000). Supporting content-based searches on time Series via approximation. *Proceedings of the 12th International Conference on Scientific and Statistical Database Management*.
- Keogh, E., Chu, S., Hart, D. & Pazzani, M. (2001). An Online Algorithm for Segmenting Time Series. In *Proceedings of IEEE International Conference on Data Mining*. pp 289-296.
- C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *ICDT*, 2001
- Ge, X. & Smyth P. (2001). Segmental Semi-Markov Models for Endpoint Detection in Plasma Etching. To appear in *IEEE Transactions on Semiconductor Engineering*.
- Eamonn J. Keogh, Shruti Kasetty: On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. Data Min. Knowl. Discov. 7(4): 349-371 (2003)
- Sivic and Zisserman, "Video Google: a text retrieval approach to object matching in videos," *Proceedings Ninth IEEE International Conference on Computer Vision*, Nice, France, 2003, pp. 1470-1477 vol.2
- T. Palpanas, M. Vlachos, E. Keogh, D. Gunopulos, W. Truppel (2004). Online Amnesic Approximation of Streaming Time Series. In *ICDE* . Boston, MA, USA, March 2004.
- E. Keogh. Tutorial on Data Mining and Machine Learning in Time Series Databases. KDD 2004.
- Richard Cole, Dennis E. Shasha, Xiaojian Zhao: Fast window correlations over uncooperative time series. KDD 2005: 743-749
- Jessica Lin, Eamonn J. Keogh, Li Wei, Stefano Lonardi: Experiencing SAX: a novel symbolic representation of time series. Data Min. Knowl. Discov. 15(2): 107-144 (2007)
- Jin Shieh, Eamonn J. Keogh: iSAX: indexing and mining terabyte sized time series. KDD 2008: 623-631
- Themis Palpanas, Michail Vlachos, Eamonn J. Keogh, Dimitrios Gunopulos: Streaming Time Series Summarization Using User-Defined Amnesic Functions. IEEE Trans. Knowl. Data Eng. 20(7): 992-1006 (2008)
- Hui Ding, Goce Trajcevski, Peter Scheuermann, Xiaoyue Wang, Eamonn J. Keogh: Querying and mining of time series data: experimental comparison of representations and distance measures. Proc. VLDB Endow. 1(2): 1542-1552 (2008)
- Stephen Blott and Roger Weber. 2008. What's wrong with high-dimensional similarity search? Proc. VLDB Endow. 1, 1 (August 2008), 3.

References (chronological order)

- C. Silpa-Anan and R. Hartley, "Optimised KD-trees for fast image descriptor matching," 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 2008, pp. 1-8
- Alexandr Andoni and Piotr Indyk. 2008. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Commun. ACM 51, 1 (January 2008), 117–122.
- M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In VISAPP International Conference on Computer Vision Theory and Applications, pages 331–340, 2009
- R. Salakhutdinov and G. Hinton, "Semantic hashing," International Journal of Approximate Reasoning, vol. 50, no. 7, pp. 969–978, 2009
- Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in Advances in neural information processing systems, 2009, pp. 1753–1760
- P. Haghani, S. Michel, and K. Aberer. Distributed similarity search in high dimensions using locality sensitive hashing. In EDBT, pages 744–755, 2009.
- Alessandro Camerra, Themis Palpanas, Jin Shieh, Eamonn J. Keogh: iSAX 2.0: Indexing and Mining One Billion Time Series. ICDM 2010: 58-67
- S. Kashyap and P. Karras. Scalable kNN search on vertically stored time series. In KDD, pages 1334–1342 (2011)
- Hervé Jégou, Matthijs Douze, Cordelia Schmid: Product Quantization for Nearest Neighbor Search. IEEE Trans. Pattern Anal. Mach. Intell. 33(1): 117-128 (2011)
- Wei Dong, Charikar Moses, and Kai Li. 2011. Efficient k-nearest neighbor graph construction for generic similarity measures. In Proceedings of the 20th international conference on World wide web (WWW '11).
- P. Schafer and M. Hogvist. Sfa: A symbolic fourier approximation and index for similarity search in high dimensional datasets. ICDE Conference 2012: 516–527
- T. Rakthanmanon, B. J. L. Campana, A. Mueen, G. E. A. P. A. Batista, M. B. Westover, Q. Zhu, J. Zakaria, and E. J. Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In KDD, pages 262–270. ACM, 2012.
- J. He, S. Kumar, and S.-F. Chang. On the difficulty of nearest neighbor search. In *ICML*, 2012.

References (chronological order)

- A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in NIPS, 2012, pp. 1106–1114.
- Bahman Bahmani, Ashish Goel, Rajendra Shinde: Efficient distributed locality sensitive hashing, CIKM 2012: 2174-2178
- Junhao Gan, Jianlin Feng, Qiong Fang, and Wilfred Ng. 2012. Locality-sensitive hashing scheme based on dynamic collision counting. In SIGMOD.
- Babenko, Artem & Lempitsky, Victor. (2012). The Inverted Multi-Index. IEEE Transactions on Pattern Analysis and Machine Intelligence. 37. 3069-3076.
- Y. Wang, P. Wang, J. Pei, W. Wang, and S. Huang. A data-adaptive and dynamic segmentation index for whole matching on time series. PVLDB, 6(10):793–804, 2013.
- M. Norouzi and D. J. Fleet. Cartesian K-Means. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '13, pages 3017–3024, 2013
- Narayanan Sundaram, Aizana Turmukhametova, Nadathur Satish, Todd Mostak, Piotr Indyk, Samuel Madden, Pradeep Dubey: Streaming Similarity Search over one Billion Tweets using Parallel Locality-Sensitive Hashing. Proc. VLDB Endow. 6(14): 1930-1941 (2013)
- Alessandro Camerra, Jin Shieh, Themis Palpanas, Thanawin Rakthanmanon, Eamonn J. Keogh: Beyond one billion time series: indexing and mining very large time series collections with iSAX2+. Knowl. Inf. Syst. 39(1): 123-151 (2014)
- Y. Sun, W. Wang, J. Qin, Y. Zhang, and X. Lin. SRS: Solving c-approximate Nearest Neighbor Queries in High Dimensional Euclidean Space with a Tiny Index. PVLDB, 8(1):1–12, 2014
- Kostas Zoumpatianos, Stratos Idreos, Themis Palpanas: Indexing for interactive exploration of big data series. SIGMOD Conference 2014: 1555-1566
- Y. Malkov, A. Ponomarenko, A. Logvinov, and V. Krylov. Approximate nearest neighbor algorithm based on navigable small world graphs. Information Systems (IS), 45:61 – 68, 2014.
- NSW IS'14: Y. Malkov, A. Ponomarenko, A. Logvinov, and V. Krylov: Approximate nearest neighbor algorithm based on navigable small world graphs, Inf. Syst., vol. 45, pp. 61–68, 2014.

References (chronological order)

- T. Ge, K. He, Q. Ke, and J. Sun. Optimized Product Quantization. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI), 36(4):744–755, Apr. 2014
- A. Babenko and V. Lempitsky. The Inverted MultiIndex. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 37(6):1247–1260, June 2015.
- Kostas Zoumpatianos, Stratos Idreos, Themis Palpanas: RINSE: Interactive Data Series Exploration with ADS+. Proc. VLDB Endow. 8(12): 1912-1915 (2015)
- Kostas Zoumpatianos, Yin Lou, Themis Palpanas, Johannes Gehrke: Query Workloads for Data Series Indexes. KDD 2015: 1603-1612
- Q. Huang, J. Feng, Y. Zhang, Q. Fang, and W. Ng. Query-aware Locality-sensitive Hashing for Approximate Nearest Neighbor Search. PVLDB, 9(1):1–12, 2015
- David C. Anastasiu and George Karypis. 2015. L2Knng: Fast Exact K-Nearest Neighbor Graph Construction with L2-Norm Pruning. In CIKM '15.
- Themis Palpanas: Big Sequence Management: A glimpse of the Past, the Present, and the Future. SOFSEM 2016: 63-80
- Kostas Zoumpatianos, Stratos Idreos, Themis Palpanas: ADS: the adaptive data series index. VLDB J. 25(6): 843-866 (2016)
- Y. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. CoRR, abs/1603.09320, 2016
- Xiaojuan Wang, Ting Zhang, Guo-Jun Qi, Jinhui Tang, Jingdong Wang: Supervised Quantization for Similarity Search. CVPR 2016: 2018-2026
- Cao, Y., Long, M., Wang, J., Zhu, H., Wen, Q.: Deep quantization network for efficient image retrieval. In: AAAI (2016)
- H. Liu, R. Wang, S. Shan and X. Chen, "Deep Supervised Hashing for Fast Image Retrieval," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2064-2072.
- Djamel Edine Yagoubi, Reza Akbarinia, Florent Masseglia, Themis Palpanas: DPiSAX: Massively Distributed Partitioned iSAX. ICDM 2017: 1135-1140
- A. Mueen, Y. Zhu, M. Yeh, K. Kamgar, K. Viswanathan, C. Gupta, and E. Keogh. The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance, August 2017. http://www.cs.unm.edu/~mueen/FastestSimilaritySearch.html.

References (chronological order)

- D. Cai, X. Gu, and C. Wang. A revisit on deep hashings for large-scale content based image retrieval, 2017.
- Katsiaryna Mirylenka, Michele Dallachiesa, Themis Palpanas: Correlation-Aware Distance Measures for Data Series. ICDE 2017: 502-505
- Katsiaryna Mirylenka, Michele Dallachiesa, Themis Palpanas: Data Series Similarity Using Correlation-Aware Measures. SSDBM 2017: 11:1-11:12
- Kostas Zoumpatianos, Themis Palpanas: Data Series Management: Fulfilling the Need for Big Sequence Analytics. ICDE 2018: 1677-1678
- A. Arora, S. Sinha, P. Kumar, and A. Bhattacharya. HD-index: Pushing the Scalability-accuracy Boundary for Approximate kNN Search in High-dimensional Spaces. PVLDB, 11(8):906–919, 2018
- J. Wang, T. Zhang, j. song, N. Sebe, and H. T. Shen. 2018. A Survey on Learning to Hash. TPAMI 40, 4 (2018)
- Michele Linardi, Themis Palpanas: ULISSE: ULtra Compact Index for Variable-Length Similarity Search in Data Series. ICDE 2018: 1356-1359
- J. Wang, T. Zhang, j. song, N. Sebe, and H. T. Shen. A survey on learning to hash. TPAMI, 40(4): 769-790 (2018).
- Kostas Zoumpatianos, Yin Lou, Ioana Ileana, Themis Palpanas, Johannes Gehrke: Generating data series query workloads. VLDB J. 27(6): 823-846 (2018)
- Haridimos Kondylakis, Niv Dayan, Kostas Zoumpatianos, Themis Palpanas: Coconut: A Scalable Bottom-Up Approach for Building Data Series Indexes. Proc. VLDB Endow. 11(6): 677-690 (2018)
- Cagatay Turkay, Nicola Pezzotti, Carsten Binnig, Hendrik Strobelt, Barbara Hammer, Daniel A. Keim, Jean-Daniel Fekete, Themis Palpanas, Yunhai Wang, Florin Rusu: Progressive Data Science: Potential and Challenges. CoRR abs/1812.08032 (2018)
- Michele Linardi, Themis Palpanas: Scalable, Variable-Length Similarity Search in Data Series: The ULISSE Approach. Proc. VLDB Endow. 11(13): 2236-2248 (2018)
- Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, Neoklis Polyzotis: The Case for Learned Index Structures. SIGMOD Conference 2018: 489-504.
- H. Yang, K. Lin and C. Chen, "Supervised Learning of Semantics-Preserving Hash via Deep Convolutional Neural Networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 437-451, 1 Feb. 2018.

References (chronological order)

- Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, Houda Benbrahim: The Lernaean Hydra of Data Series Similarity Search: An Experimental Evaluation of the State of the Art. Proc. VLDB Endow. 12(2): 112-127 (2018)
- Botao Peng, Panagiota Fatourou, Themis Palpanas: ParIS: The Next Destination for Fast Data Series Indexing and Query Answering. BigData 2018: 791-800
- Akhil Arora, Sakshi Sinha, Piyush Kumar, Arnab Bhattacharya: HD-Index: Pushing the Scalability-Accuracy Boundary for Approximate kNN Search in High-Dimensional Spaces. PVLDB. 11(8): 906-919 (2018).
- D.E. Yagoubi, R. Akbarinia, B. Kolev, O. Levchenko, F. Masseglia, P. Valduriez, D. Shasha. ParCorr: efficient parallel methods to identify similar time series pairs across sliding windows. Data Mining and Knowledge Discovery (DMKD), 2018
- Tan Yu, et al. "Product quantization network for fast image retrieval." Proceedings of the European Conference on Computer Vision (ECCV). 2018.
- Matsui, Yusuke, et al. "A survey of product quantization." ITE Transactions on Media Technology and Applications 6.1 (2018): 2-10.
- Haridimos Kondylakis, Niv Dayan, Kostas Zoumpatianos, Themis Palpanas: Coconut Palm: Static and Streaming Data Series Exploration Now in your Palm. SIGMOD Conference 2019: 1941-1944
- Themis Palpanas, Volker Beckmann: Report on the First and Second Interdisciplinary Time Series Analysis Workshop (ITISA). SIGMOD Rec. 48(3): 36-40 (2019)
- S. Morozov and A. Babenko. Unsupervised neural quantization for compressed-domain similarity search. In *ICCV*, 2019.
- Benjamin Klein, Lior Wolf: End-To-End Supervised Product Quantization for Image Search and Retrieval. CVPR 2019: 5041-5050
- Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Herve Jegou. Spreading vectors for similarity search., ICLR, 2019
- Osman Durmaz and Hasan Sakir Bilge. 2019. Fast image similarity search by distributed locality sensitive hashing. Pattern Recognition Letters 128 (2019), 361–369

References (chronological order)

- Oleksandra Levchenko, Boyan Kolev, Djamel Edine Yagoubi, Dennis E. Shasha, Themis Palpanas, Patrick Valduriez, Reza Akbarinia, Florent Masseglia: Distributed Algorithms to Find Similar Time Series. ECML/PKDD (3) 2019: 781-785
- Haridimos Kondylakis, Niv Dayan, Kostas Zoumpatianos, Themis Palpanas: Coconut: sortable summarizations for scalable indexes over static and streaming data series. VLDB J. 28(6): 847-869 (2019)
- Danila Piatov, Sven Helmer, Anton Dignös, Johann Gamper: Interactive and space-efficient multi-dimensional time series subsequence matching. Inf. Syst. 82: 121-135 (2019)
- Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, Houda Benbrahim: Return of the Lernaean Hydra: Experimental Evaluation of Data Series Approximate Similarity Search. Proc. VLDB Endow. 13(3): 403-420 (2019)
- Anna Gogolou, Theophanis Tsandilas, Themis Palpanas, Anastasia Bezerianos: Comparing Similarity Perception in Time Series Visualizations. IEEE Trans. Vis. Comput. Graph. 25(1): 523-533 (2019)
- John Paparrizos, Michael J. Franklin: GRAIL: Efficient Time-Series Representation Learning. Proc. VLDB Endow. 12(11): 1762-1777 (2019)
- Cong Fu, Chao Xiang, Changxu Wang, and Deng Cai. 2019. Fast approximate nearest neighbor search with the navigating spreading-out graph. Proc. VLDB Endow. 12, 5 (January 2019), 461–474.
- Jiaye Wu, Peng Wang, Ningting Pan, Chen Wang, Wei Wang, Jianmin Wang: KV-Match: A Subsequence Matching Approach Supporting Normalization and Time Warping. ICDE 2019: 866-877
- Liang Zhang, Noura Alghamdi, Mohamed Y. Eltabakh, Elke A. Rundensteiner: TARDIS: Distributed Indexing Framework for Big Time Series Data. ICDE 2019: 1202-1213

References (chronological order)

- Conglong Li, Minjia Zhang, David G. Andersen, and Yuxiong He.Improving Approximate Nearest Neighbor Search through Learned Adaptive Early Termination. SIGMOD Conference 2020. 2539–2554. Anna Gogolou, Theophanis Tsandilas, Karima Echihabi, Anastasia Bezerianos, Themis Palpanas: Data Series Progressive Similarity Search with Probabilistic Quality Guarantees. SIGMOD Conference 2020: 1857-1873
- Themis Palpanas. Evolution of a Data Series Index - The iSAX Family of Data Series Indexes. CCIS, 1197 (2020)
- Xiao Luo, Chong Chen, Huasong Zhong, Hao Zhang, Minghua Deng, Jianqiang Huang, and Xiansheng Hua. 2020. A Survey on Deep Hashing Methods.
- Abdullah Al-Mamun, Hao Wu, Walid G. Aref:A Tutorial on Learned Multi-dimensional Indexes. SIGSPATIAL/GIS 2020: 1-4
- Angjela Davitkova, Evica Milchevski, and Sebastian Michel. 2020. The ML-Index:A Multidimensional, Learned Index for Point, Range, and Nearest-Neighbor Queries.. In EDBT. 407–410.
- Jianzhong Qi, Guanli Liu, Christian S Jensen, and Lars Kulik. 2020. Effectively learning spatial indices. Proceedings of the VLDB Endowment13, 12 (2020), 2341–2354.
- Djamel Edine Yagoubi, Reza Akbarinia, Florent Masseglia, Themis Palpanas: Massively Distributed Time Series Indexing and Querying. IEEE Trans. Knowl. Data Eng. 32(1): 108-120 (2020)
- Botao Peng, Panagiota Fatourou, Themis Palpanas: MESSI: In-Memory Data Series Indexing. ICDE 2020: 337-348
- Kefeng Feng, Peng Wang, Jiaye Wu, Wei Wang: L-Match: A Lightweight and Effective Subsequence Matching Approach. IEEE Access 8: 71572-71583 (2020)
- Chen Wang, Xiangdong Huang, Jialin Qiao, Tian Jiang, Lei Rui, Jinrui Zhang, Rong Kang, Julian Feinauer, Kevin Mcgrail, Peng Wang, Diaohan Luo, Jun Yuan, Jianmin Wang, Jiaguang Sun: Apache IoTDB: Time-series database for Internet of Things. Proc. VLDB Endow. 13(12): 2901-2904 (2020)
- Ruiqi Guo, Philip Sun, Erik Lindgren, Quan Geng, David Simcha, Felix Chern, Sanjiv Kumar: Accelerating Large-Scale Inference with Anisotropic Vector Quantization. ICML 2020: 3887-3896.
- Jianbin Qin, Wei Wang, Chuan Xiao, Ying Zhang: Similarity Query Processing for High-Dimensional Data. Proc. VLDB Endow. 13(12): 3437-3440 (2020)

References (chronological order)

- Mingjie Li, Ying Zhang, Yifang Sun, Wei Wang, Ivor W. Tsang, Xuemin Lin: I/O Efficient Approximate Nearest Neighbour Search based on Learned Functions. ICDE 2020: 289-300
- Michele Linardi, Themis Palpanas. Scalable Data Series Subsequence Matching with ULISSE. VLDBJ 2020
- John Paparrizos, Chunwei Liu, Aaron J. Elmore, Michael J. Franklin: Debunking Four Long-Standing Misconceptions of Time-Series Distance Measures. SIGMOD Conference 2020
- Karima Echihabi, Kostas Zoumpatianos, and Themis Palpanas. Scalable Machine Learning on High-Dimensional Vectors: From Data Series to Deep Network Embeddings. In WIMS, 2020
- Oleksandra Levchenko, Boyan Kolev, Djamel-Edine Yagoubi, Reza Akbarinia, Florent Masseflia, Themis Palpanas, Dennis Shasha, Patrick Valduriez. BestNeighbor: Efficient Evaluation of kNN Queries on Large Time Series Databases. Knowledge and Information Systems (KAIS), 2020
- Kejing Lu, Hongya Wang, Wei Wang, Mineichi Kudo. VHP: Approximate Nearest Neighbor Search via Virtual Hypersphere Partitioning. PVLDB, 13(9): 1443-1455, 2020
- Yury A. Malkov, D. A. Yashunin: Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. IEEE Trans. Pattern Anal. Mach. Intell. 42(4): 824-836 (2020)
- Botao Peng, Panagiota Fatourou, Themis Palpanas. Paris+: Data series indexing on multi-core architectures. TKDE, 2021
- Botao Peng, Panagiota Fatourou, Themis Palpanas. SING: Sequence Indexing Using GPUs. ICDE, 2021
- Q. Wang and T. Palpanas. Deep Learning Embeddings for Data Series Similarity Search. In *SIGKDD*, 2021.
- Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas. Big Sequence Management: Scaling Up and Out. EDBT 2021
- Botao Peng, Panagiota Fatourou, Themis Palpanas. Fast Data Series Indexing for In-Memory Data. VLDBJ 2021
- Jeff Johnson, Matthijs Douze, Hervé Jégou: Billion-Scale Similarity Search with GPUs. IEEE Trans. Big Data 7(3): 535-547 (2021)
- Mengzhao Wang, Xiaoliang Xu, Qiang Yue, Yuxiang Wang: A Comprehensive Survey and Experimental Comparison of Graph Based Approximate Nearest Neighbor Search. PVLDB, 2021.
- Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas. High-Dimensional Similarity Search for Scalable Data Science. ICDE 2021
- Ji Sun, Guoliang Li, Nan Tang: Learned Cardinality Estimation for Similarity Queries. SIGMOD Conference 2021