Big Sequence Management: on Scalability

Karima Echihabi

Mohammed VI Polytechnic University

Kostas Zoumpatianos

Harvard University & Université de Paris

Themis Palpanas

Université de Paris & French University Institute (IUF)

IEEE International Conference on Big Data (IEEE BigData), December 2020

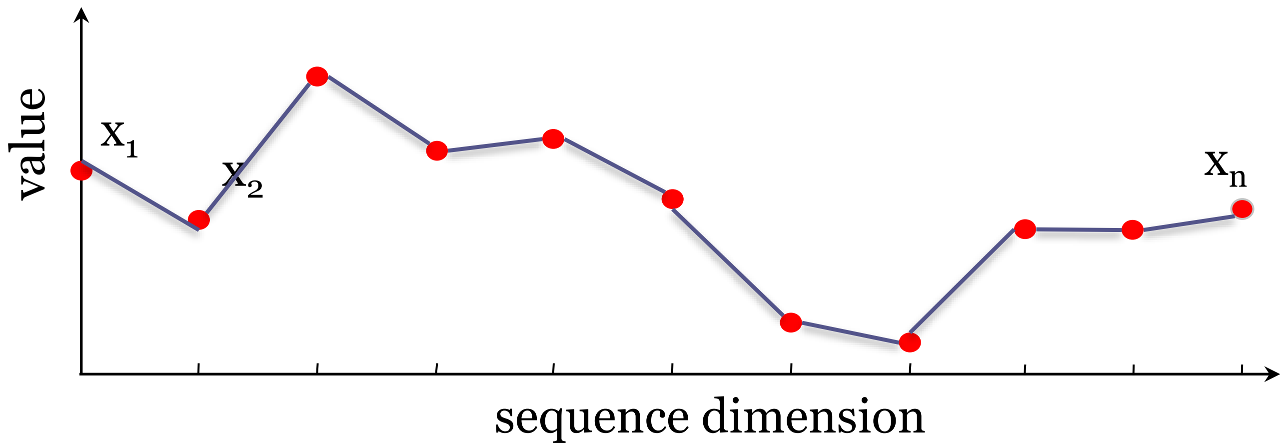
Acknowledgements

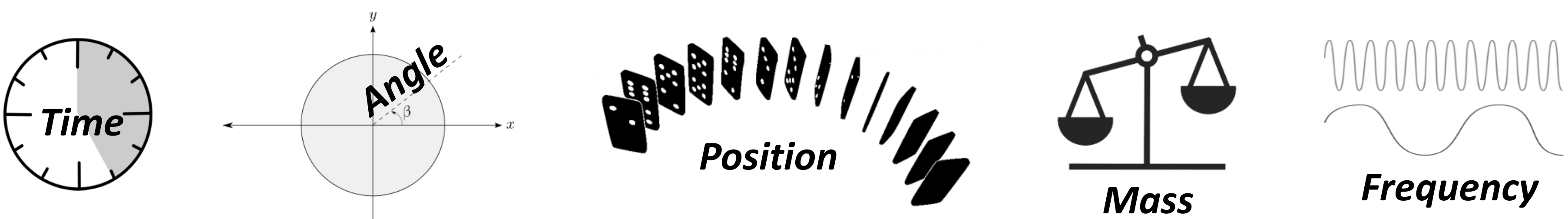
- thanks for slides to
  - Michail Vlachos
  - Eamonn Keogh
  - Panagiotis Papapetrou
  - George Kollios
  - Dimitrios Gunopulos
  - Christos Faloutsos
  - Panos Karras

Introduction, Motivation

Data series

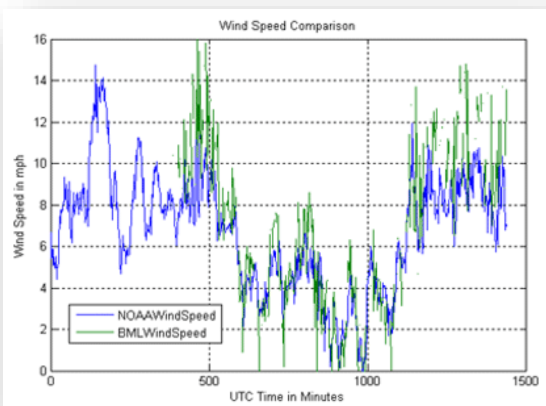
- Sequence of points ordered along some dimension





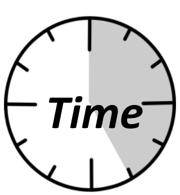
Scientific Monitoring

- meteorology, oceanography, astronomy, finance, sociology, …



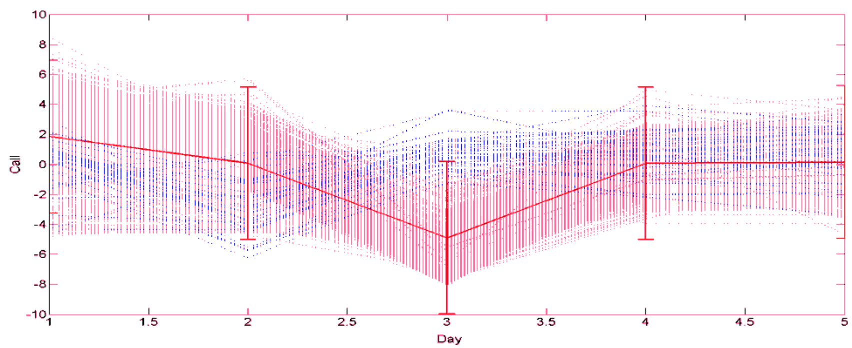
Wind speed
From ocean observing node project
http://bml.ucdavis.edu/boon/wind.html

Historical stock quotes
http://money.cnn.com/2012/04/23/markets/walmart_stock/index.htm

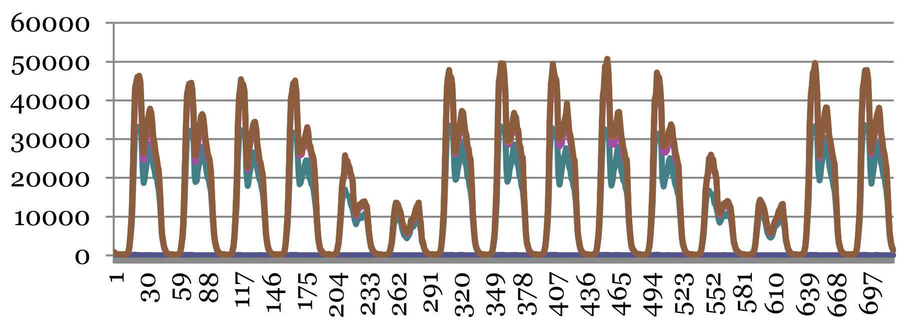


Telecommunications

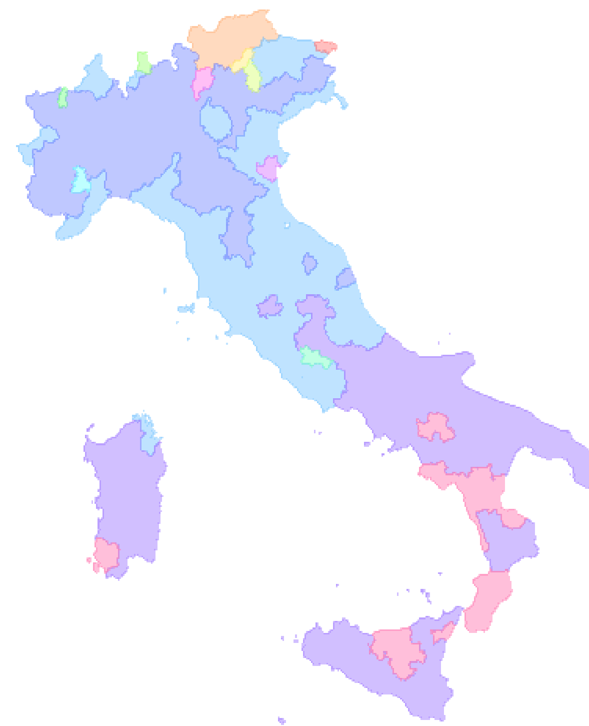
- analysis of call activity patterns
  - Telecom Italia



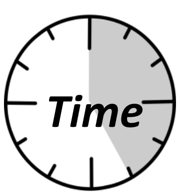
call activity for Easter Monday



average number of calls for 5 smallest clusters

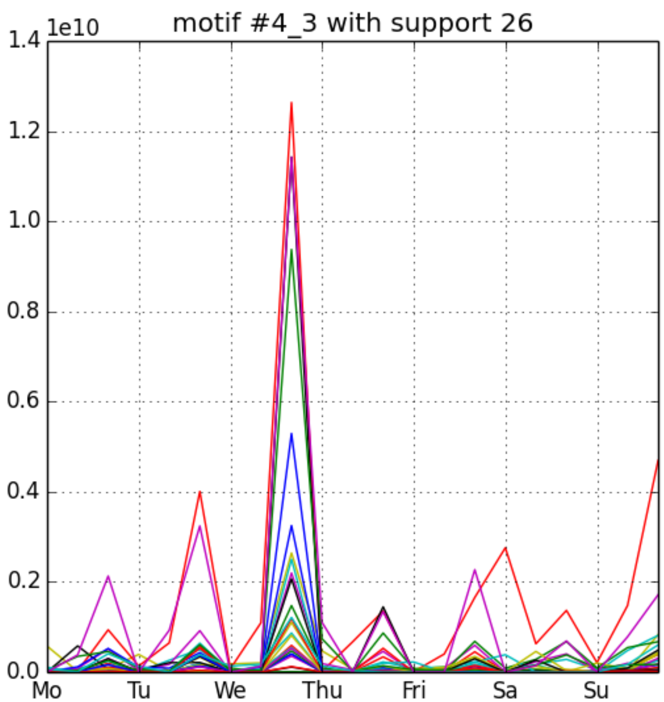


clustermap of incoming calls time series

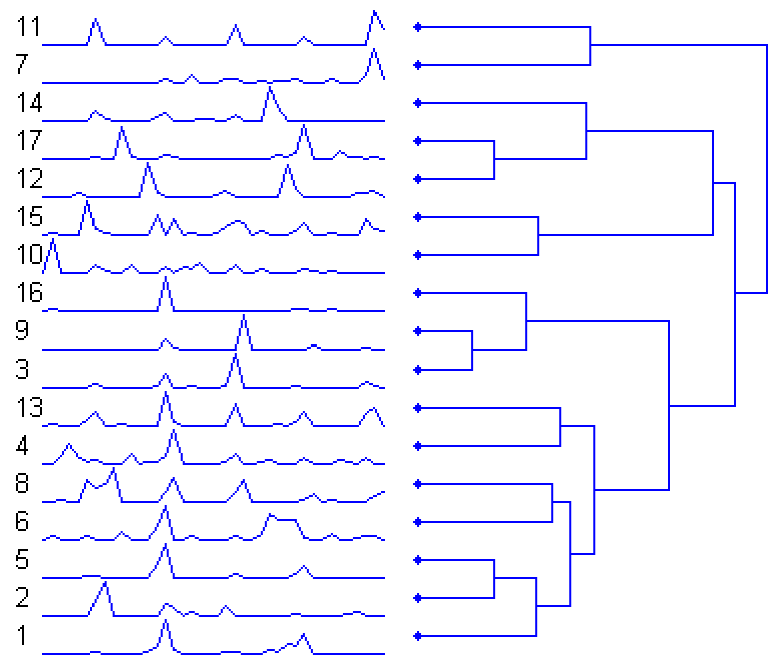


Home Networks

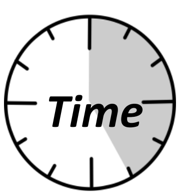
- temporal usage behavior analysis of home networks
    - Portugal Telecom



(previously unknown) frequent behavior pattern

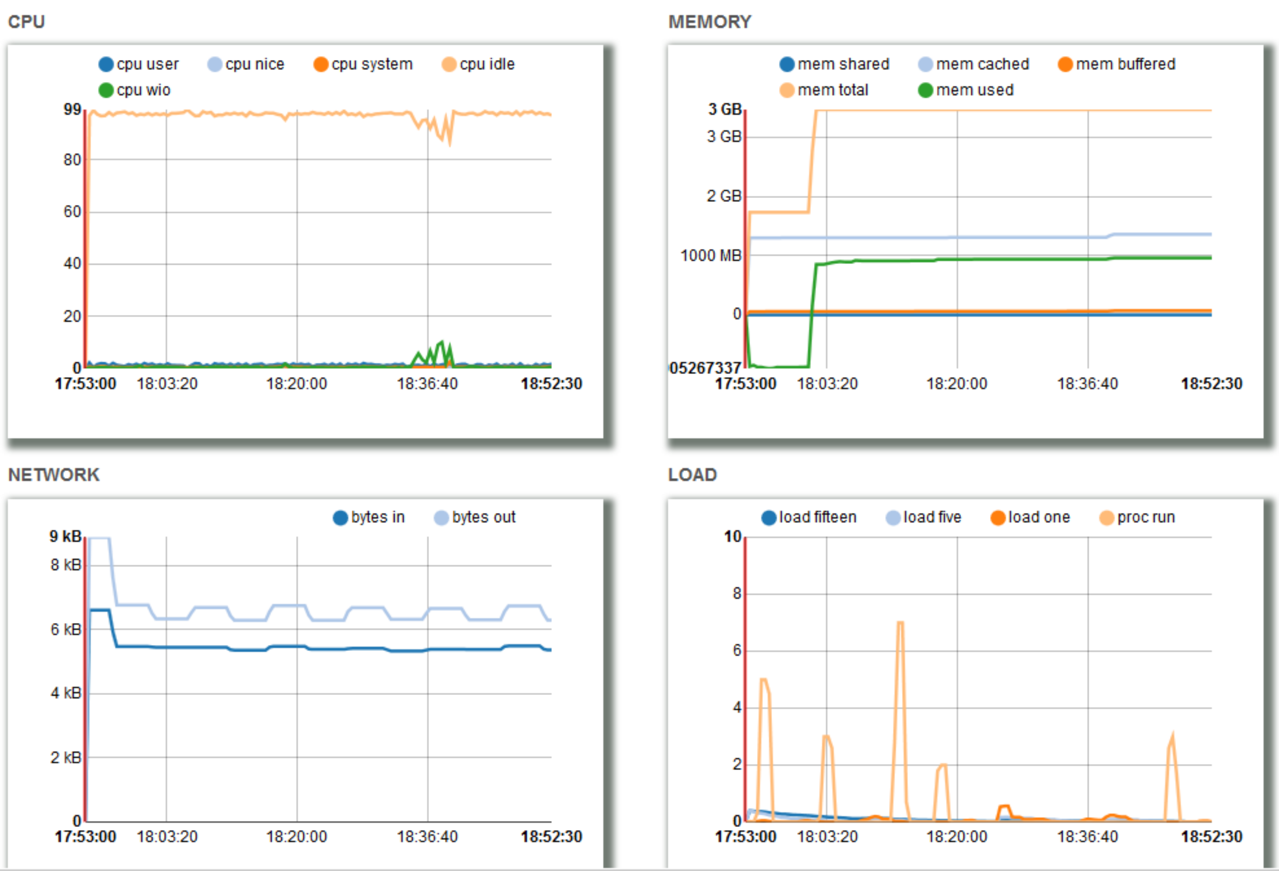


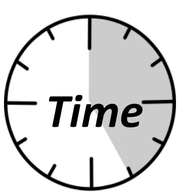
clustering based on user activity patterns



Data Centers

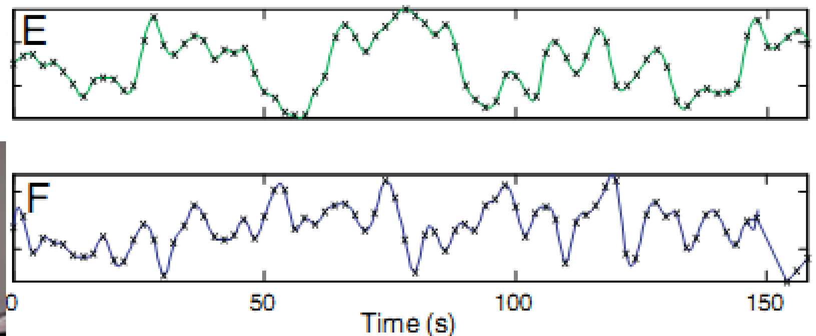
- cloud utilization/operation/health monitoring

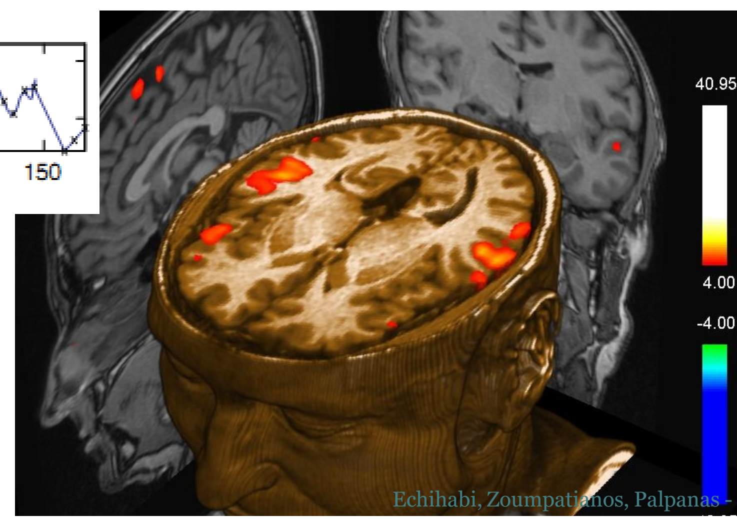


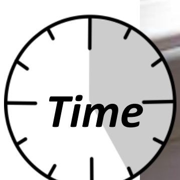


Neuroscience

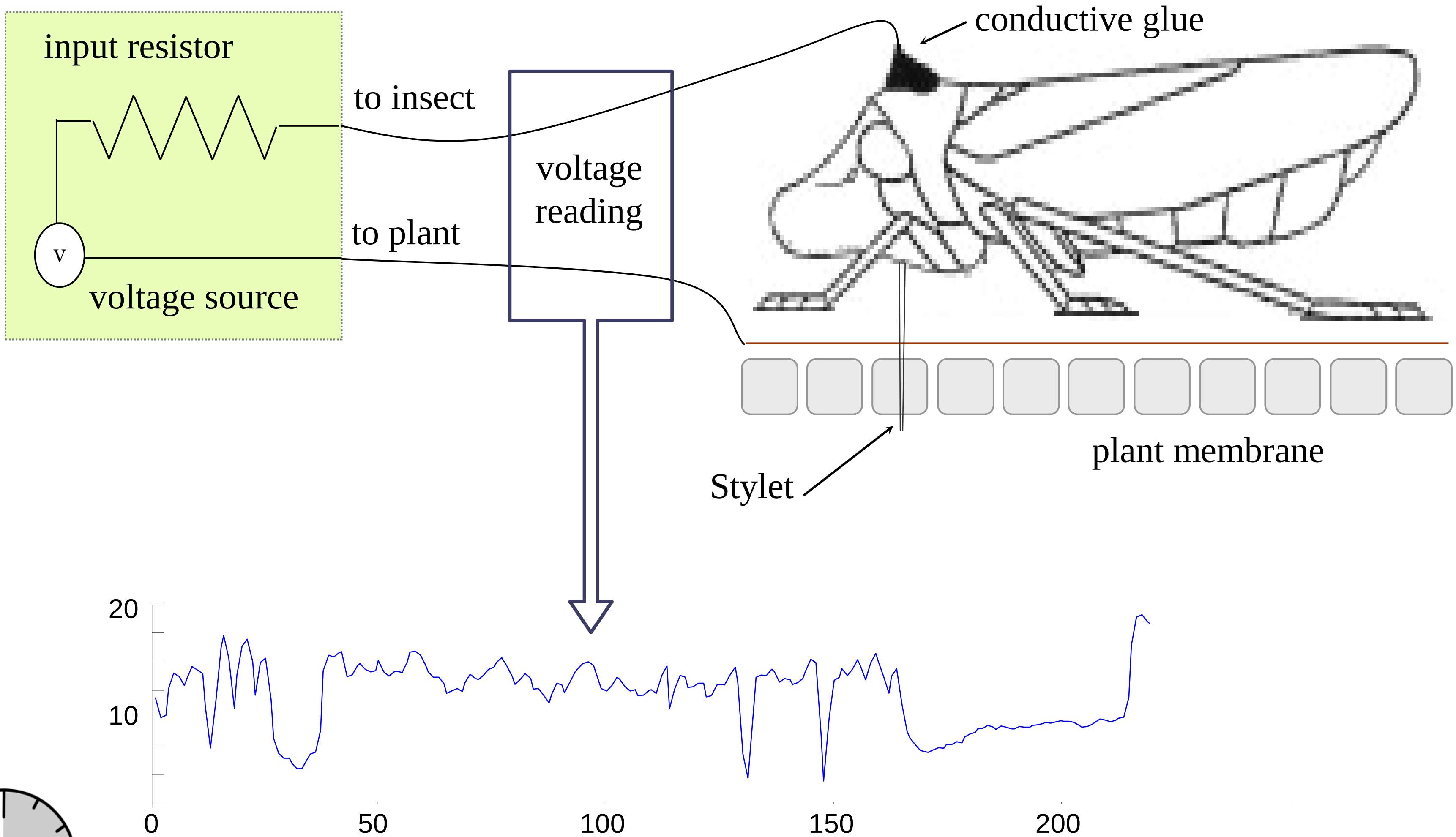
- functional Resonance Magnetic Imaging (fMRI) data
  - primary experimental tool of neuroscientists
  - reveal how different parts of brain respond to stimuli

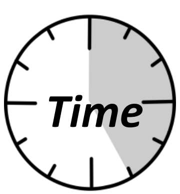




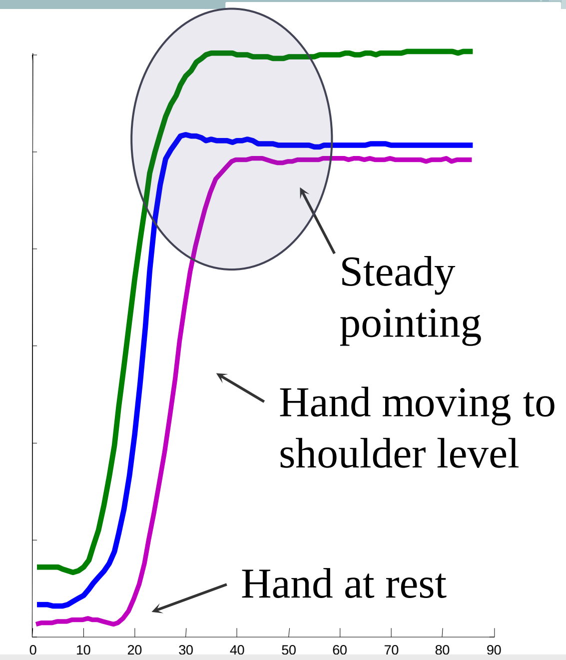


Entomology

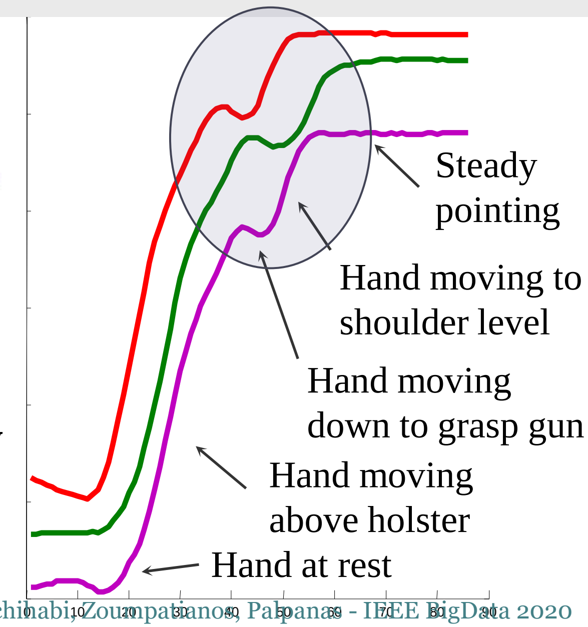

input resistor
to insect
to plant
v
voltage source
voltage reading
conductive glue
plant membrane
Stylet
20
10
0
50
100
150
200

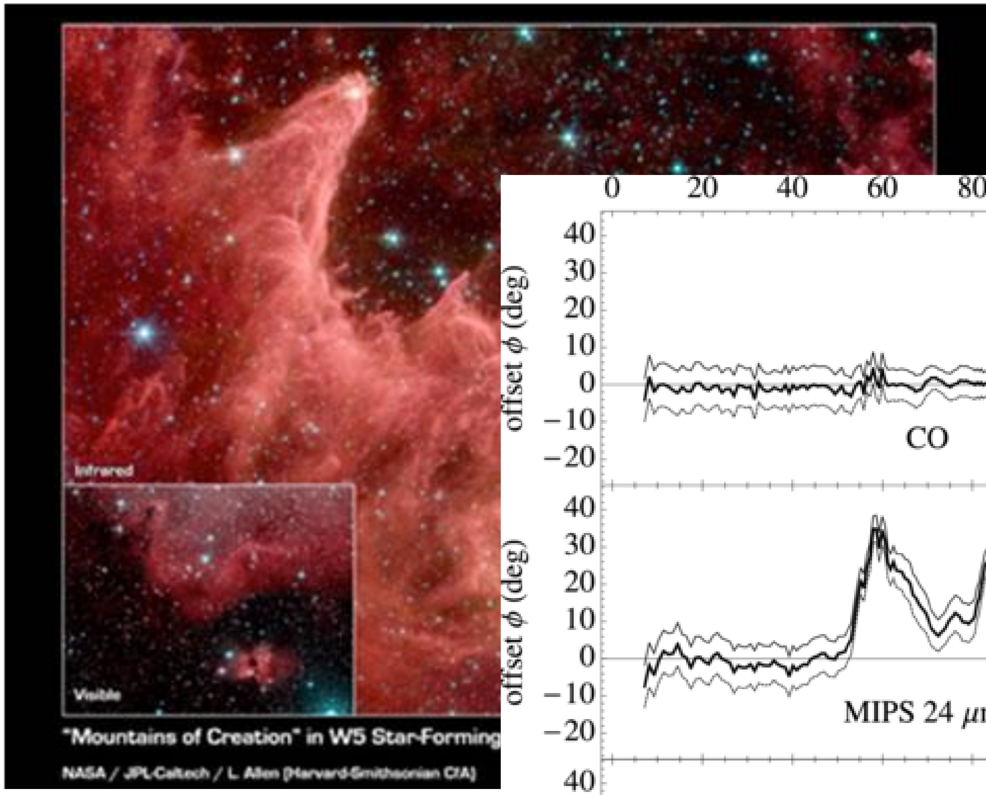

Time

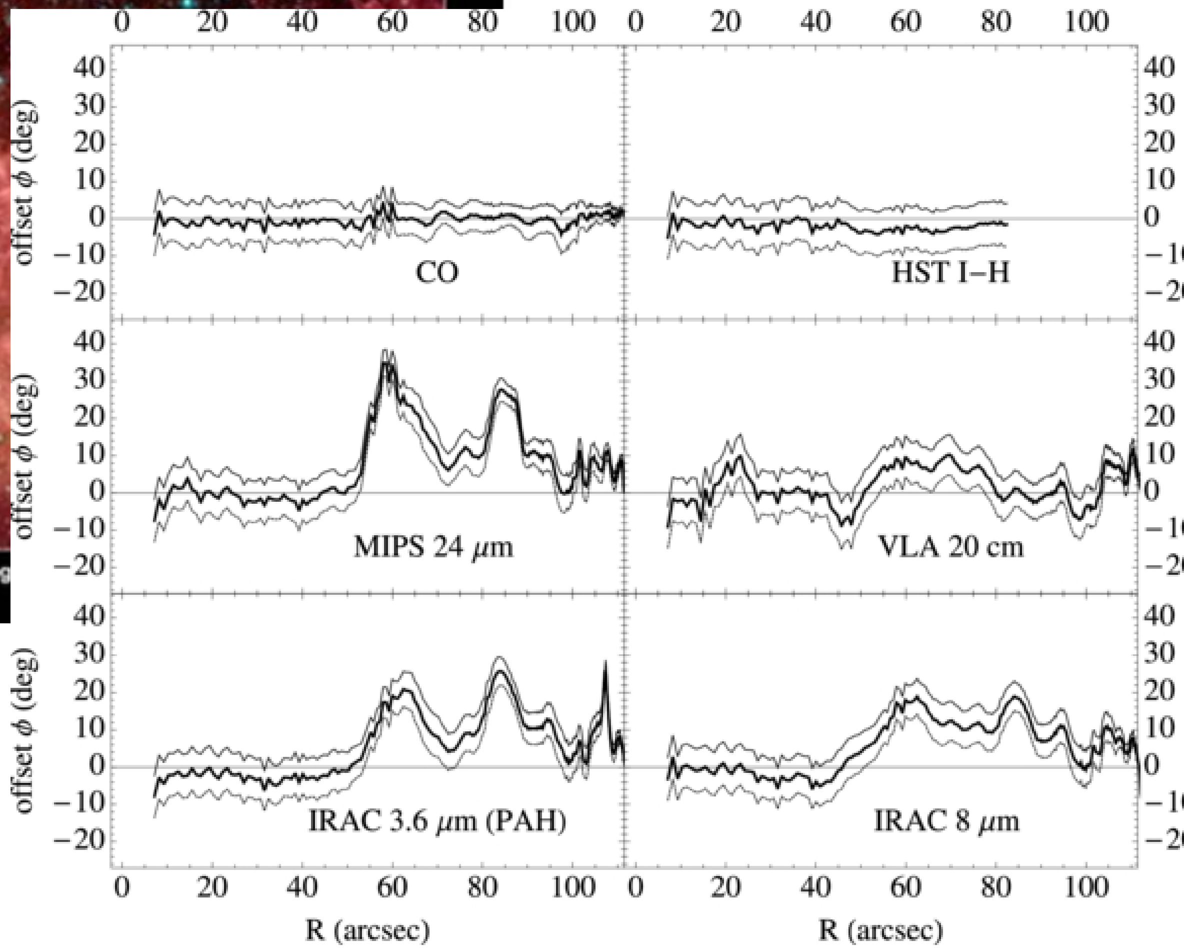
Point

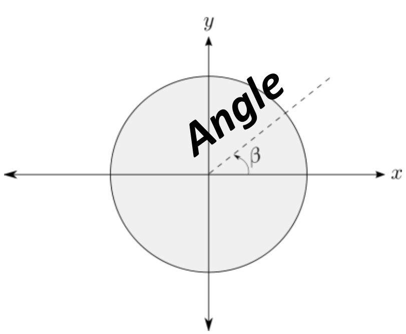


Gun-Draw



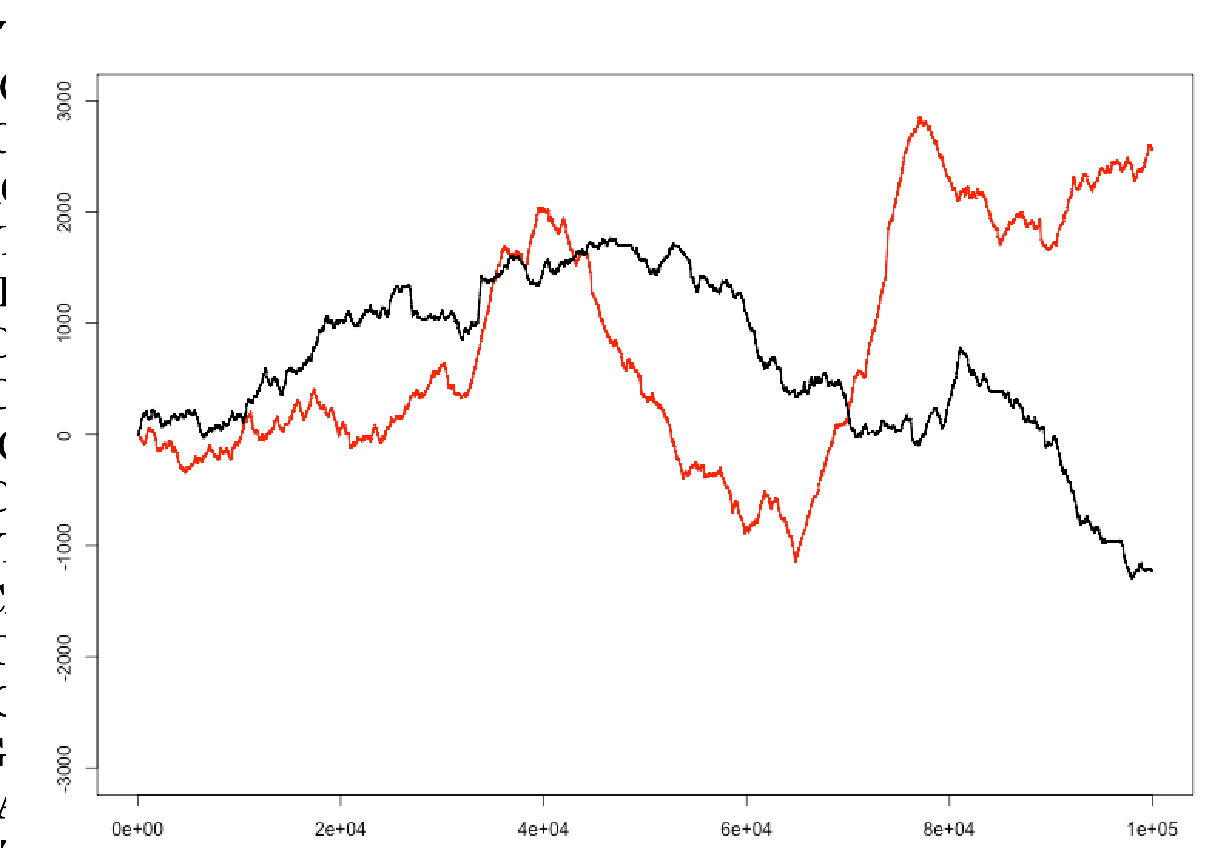




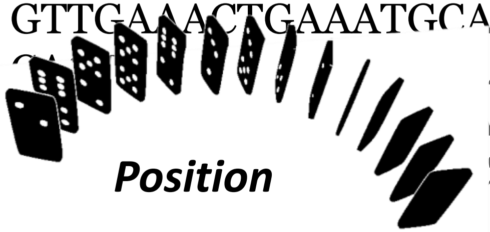


Schinnerer et al.

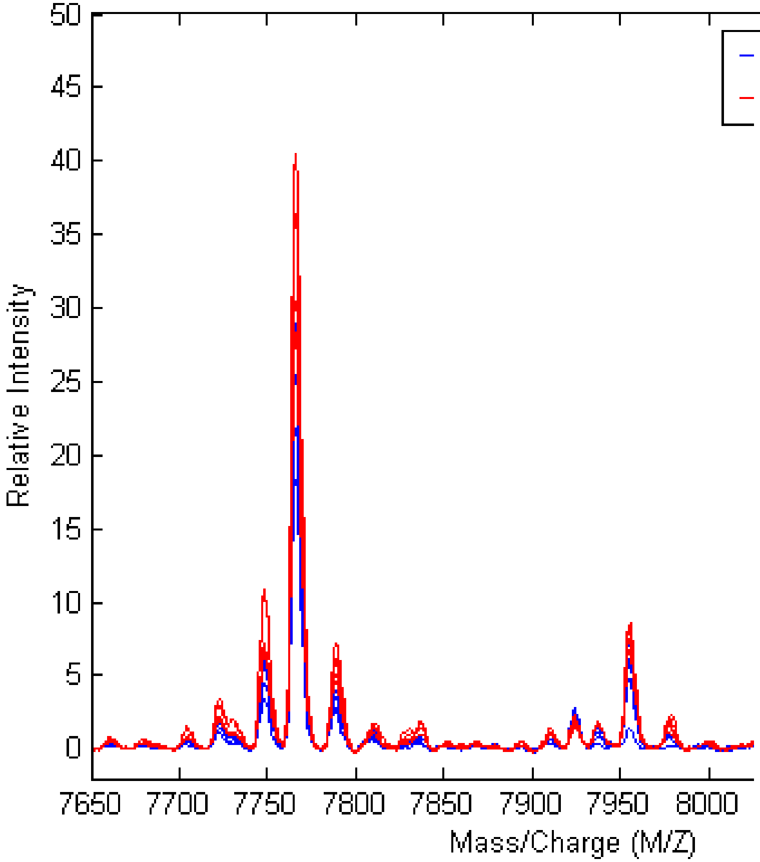
GTCAATGGCCAGGATATTAGAACAGTACTCTGTGAACCCTATTTATGGTGGCACCCCTTAGACTAA
GATAACACAGGGAGCAAGAGGTTGACAGGAAAGCCAGGGGAGCAGGGAAGCCTCCTGTAAAGAG
AGAAGTGCTAACTCTCCTTTCTAAGGCAGATGATGGATTCAAGGCAAAGGCAGATTTGACTAAAGC
CCAAGGATTG AGTCAGATTAA
TATGTGGGTGC TGAGGAGAAA
CTCCTCCACAG GCCAGAGGATC
TGTAAAAGAG ACAAGCAAAA
AAGAGCACTCA GGCCACCAAG
CCAGTTGTCAAT GAGTGTCTTCT
ATTTCTTCCAGC CCTTTACAATT
GTTATTTATGTC CATACTGTCTT
GGCCAAGCATC TGCACGTGGT
TATTTATTAGTC AAGAGCAGAAT
CAATGAGCCGC AGATGAGCATT
GAAGAGGATGC CAGGTTGTTGC
CATGTTTGTTTT TTAGAACAGTA
CTCTGTGAACC AGAGGTTGACA
GGAAAGCCAGG TTTCTAAGGCA
CATGATGGATC TAATCCGATTC
TTGGCAGAAGA ATAAACAAATA
ATTGAATAATCCCTGGAGGTTTAAGTGAGGAGAAACTCCTCCACACTTGCTACCGAGGCAGAACCG
GTTGAAACTGAAATGCACCCGCTGCCAGATTTATTAGTCACCCAAGCATGTATTTTGCATGTCCAT

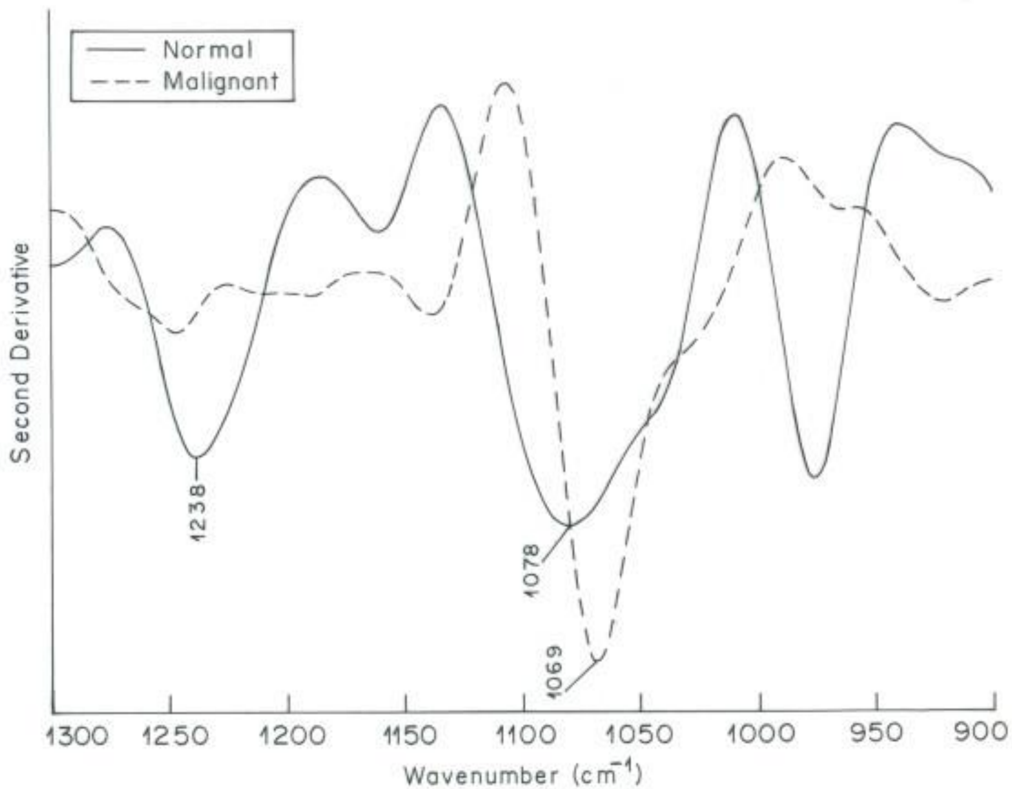


AAGAACAGAATCAATGAGCCGCTGCAGATGCAGACATAGCAGCCCCTTG
GATGAGCATTGAAGAGGATGCACAAGCCCGGTAGCCCGGGAAATGGCA
AGGTTGTTGCCATGTTTGTTTTTGCAACTTGTCTTTTAAACAGATTTGA

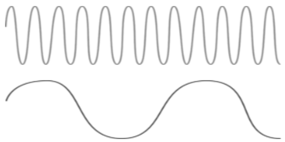


Medicine



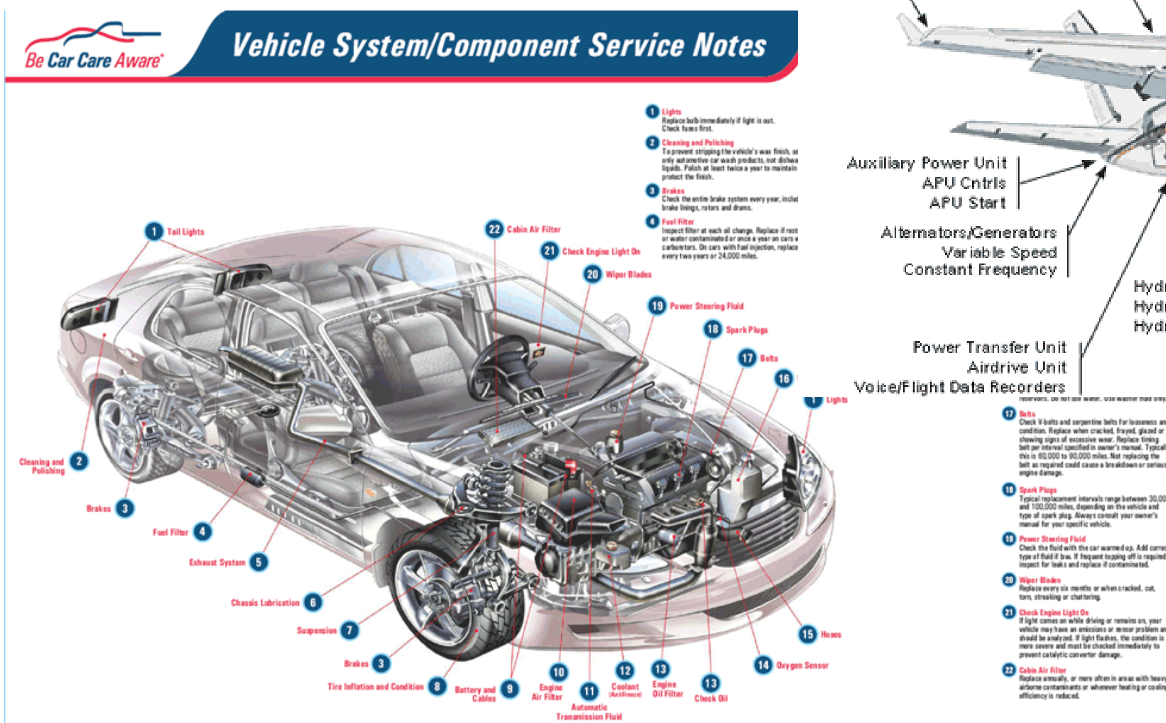


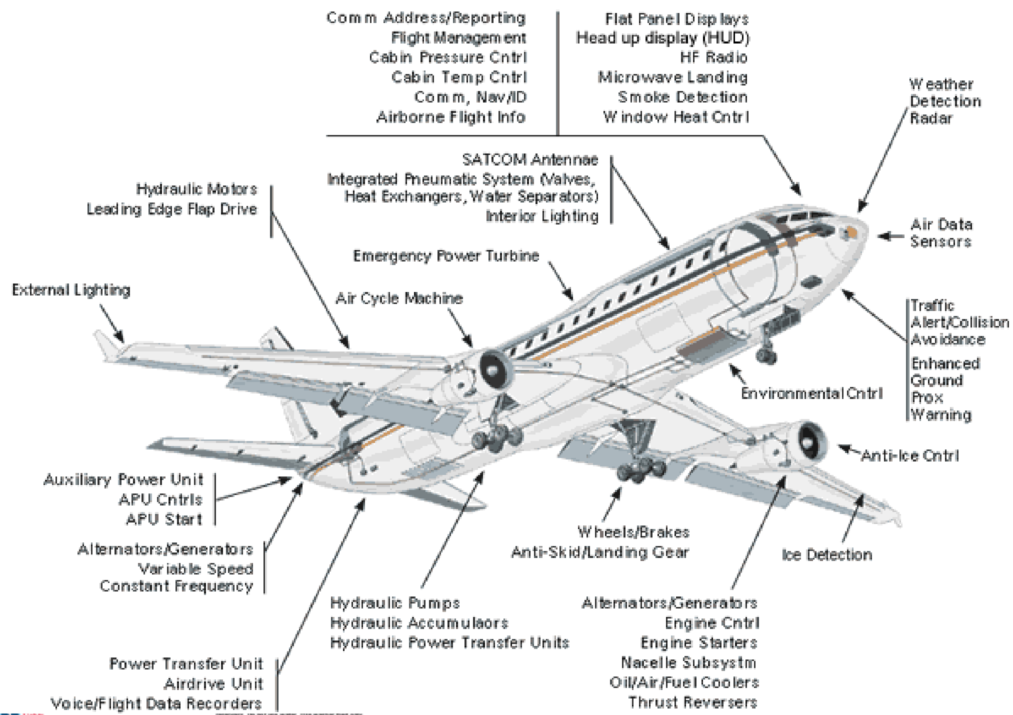
Mass



Frequency

Motivating Examples: Monitoring Vehicle Operation





Data as a Set
Data as a Sequence

- streaming data
  - window of interest
    - landmark window
    - sliding window (shifting window)

- may treat streaming data as a set, or as a sequence
  - depends on whether sequence is important

Data Series (Signal) Processing Data Series Management

- lots of literature on data series processing
  - periodicity detection
  - data series modeling and forecasting
    - ARMA, ARIMA
  - outlier detection
    - focuses on next value
- instead, we will focus on
  - sequences as first class citizens
  - very large collections of data series
  - fast and scalable similarity search

Objectives

- get introduced to the data series data type
  - characteristics, properties, peculiarities
- learn about
  - data series representations
  - data series similarity matching
  - data series indexing
  - systems for data series management
  - challenges and open problems

Data Series Representations

Introduction

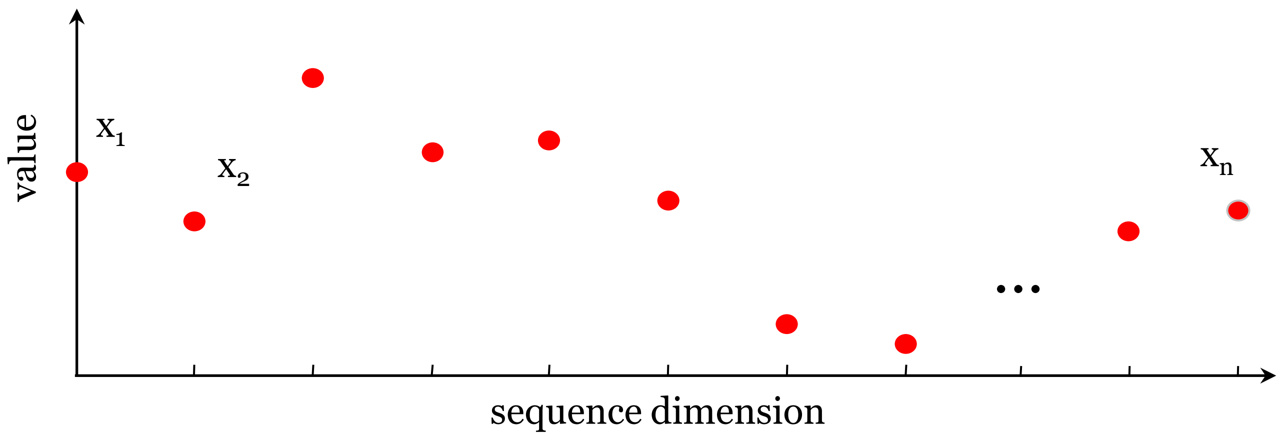
- lots of work on data series representations
  - techniques for representing/storing data series
- main goal
  - summarize data series
  - render subsequent processing more efficient

Outline

- terminology and definitions
- motivation
- pre-processing tasks
- data series representation techniques

Data series

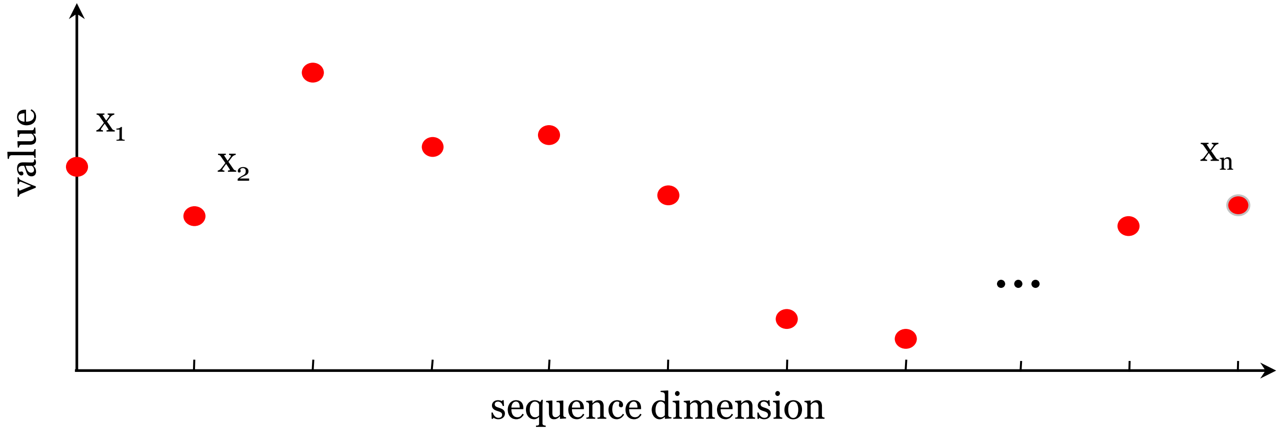
- Sequence of points ordered along some dimension



- terminology: we will use interchangeably
  - data series, time series, data sequence, sequence

Data series

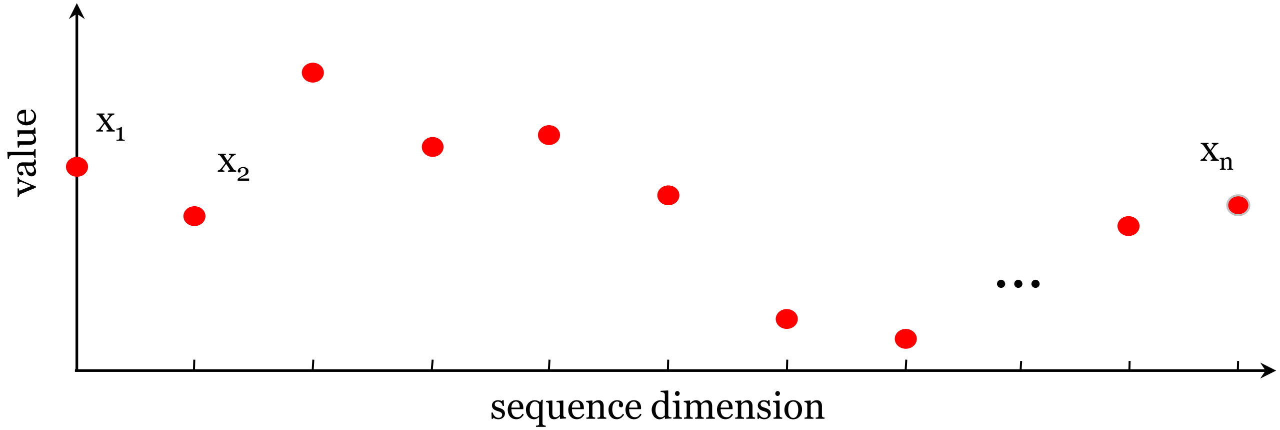
- Sequence of points ordered along some dimension



- number of data series values, n
  - length, or dimensionality

Data series

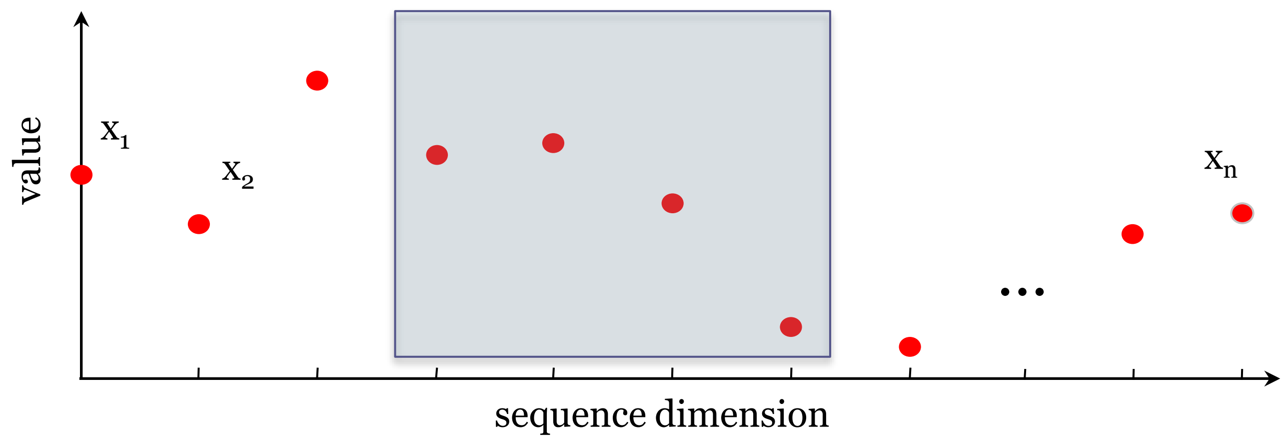
- Sequence of points ordered along some dimension



- subsequence
  - subset of contiguous values

Data series

- Sequence of points ordered along some dimension



- subsequence
  - subset of contiguous values
  - eg, subsequence of length (dimensionality) 4

Outline

- terminology and definitions
- motivation
- pre-processing tasks
- data series representation techniques

Simple Query Answering

select values in time interval

select values in some range

select some data series

combinations of those

Analysis Tasks

Clustering

Outlier Detection

Classification

Frequent Pattern Mining

Analysis Tasks

- analyze evolution of values across x-dimension
- identify trends

- treat data series as a first class citizen
  - analyze each data series as a single object
  - process all n-dimensions at once

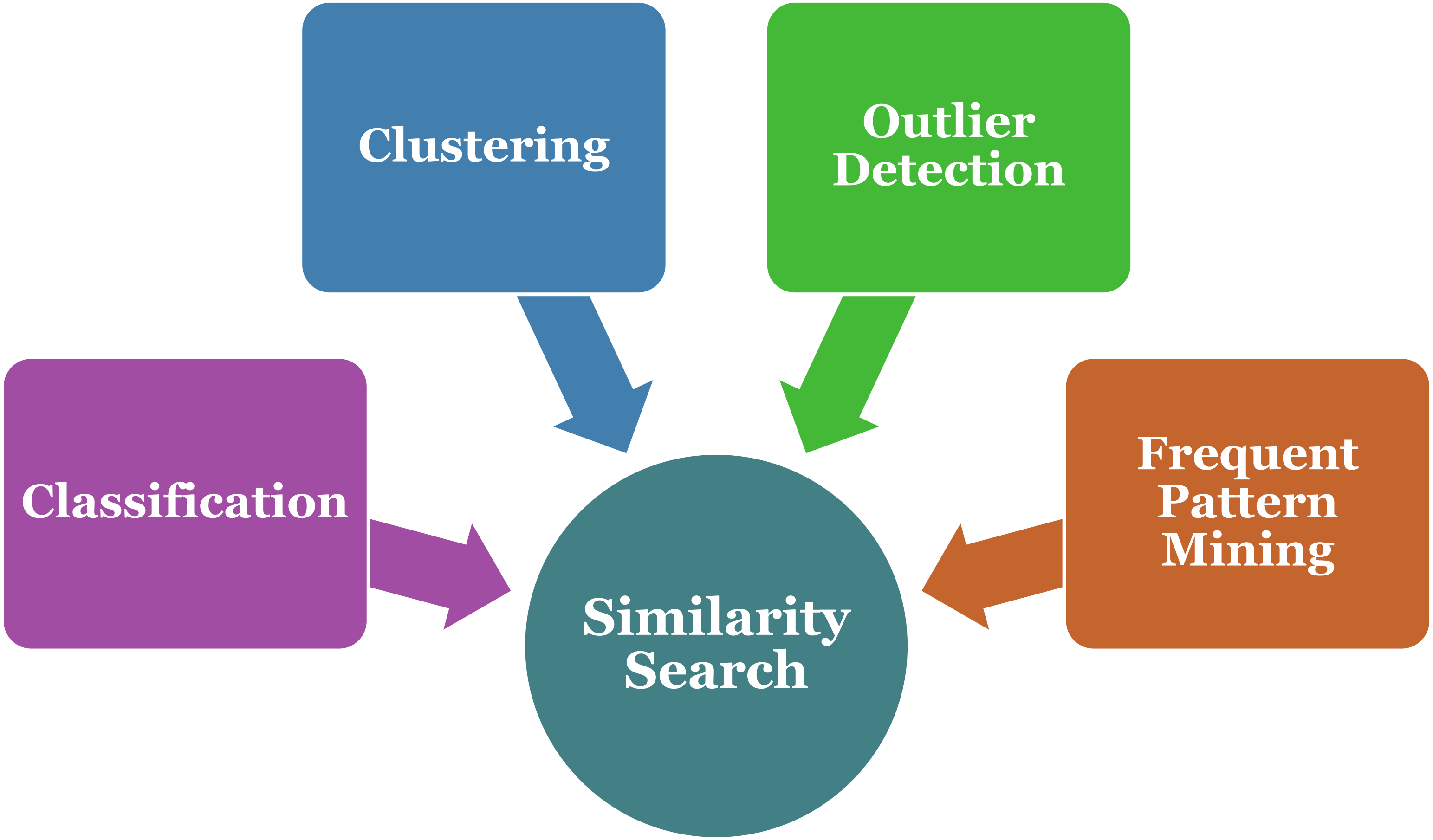
Analysis Tasks
Subsequences

- often times the data series are very long
  - n >> 1
  - streaming data series

Analysis Tasks
Subsequences

- often times the data series are very long
  - n >> 1
  - streaming data series
- we then chop the long sequence in subsequences
  - e.g., using sliding window, or shifting window
  - pick carefully length of subsequence
    - should contain patterns of interest
- and process each subsequence separately

Complex Analytics



Complex Analytics

Clustering

Outlier Detection

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

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

Motivation

- effective representation techniques to the rescue!
  - can significantly reduce the processing time
    - typically much smaller than original/raw data series
- will learn how to compute and use these representations
- these representations can further be used for indexing
- all guarantee correct, exact results!

Outline

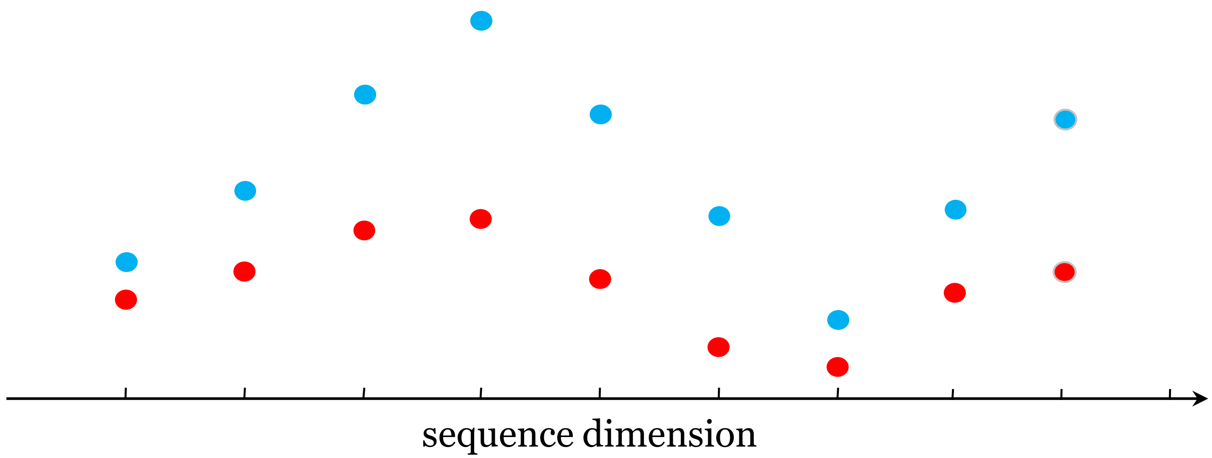
- terminology and definitions
- motivation
- pre-processing tasks
- data series representation techniques

Pre-Processing
z-Normalization

- data series encode trends
- usually interested in identifying similar trends

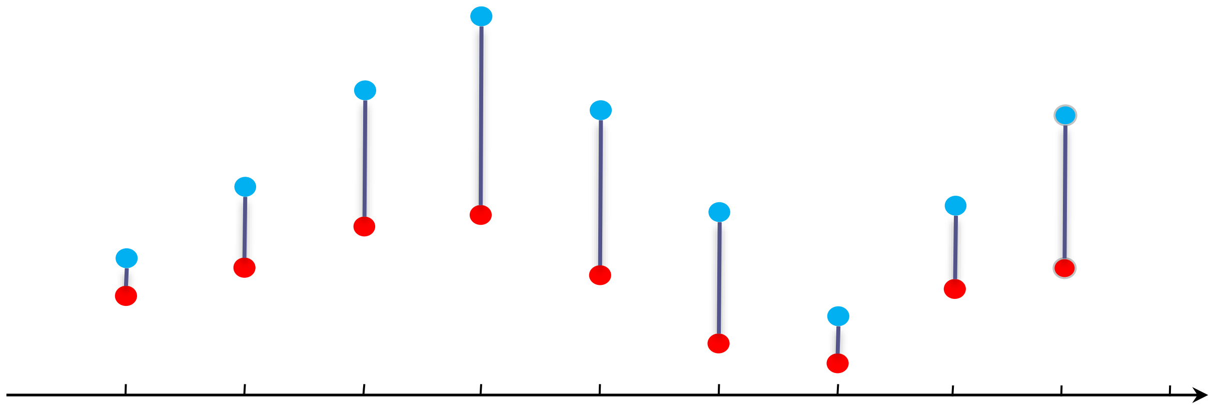
- but absolute values may mask this similarity

Pre-Processing
z-Normalization



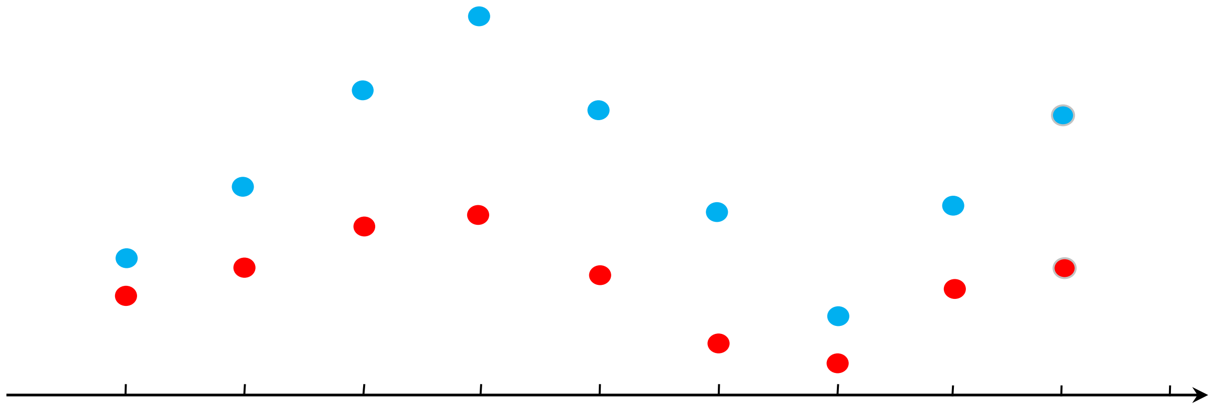
- two data series with similar trends

Pre-Processing
z-Normalization



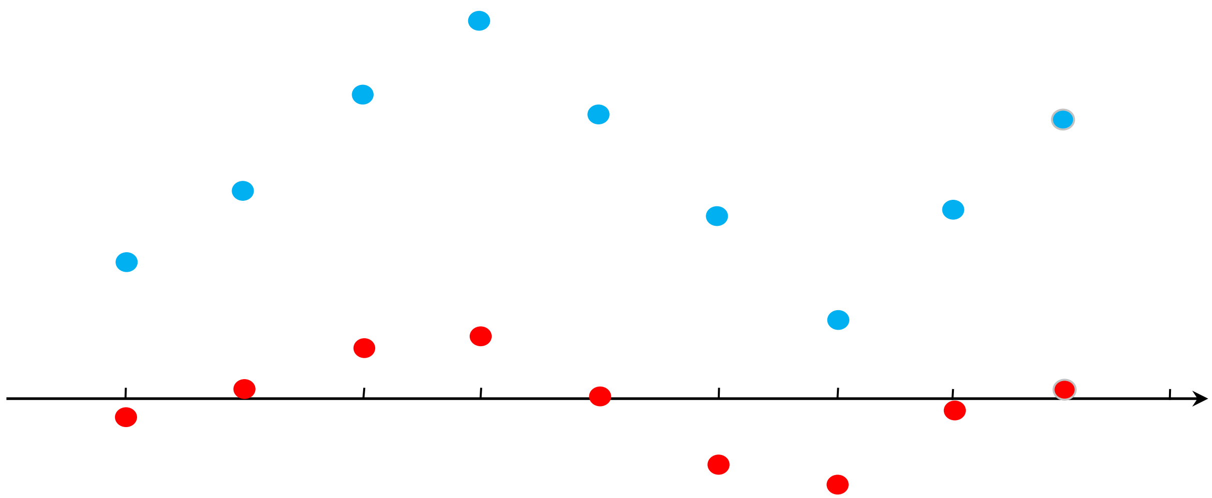
- two data series with similar trends
- but large distance…

Pre-Processing
z-Normalization



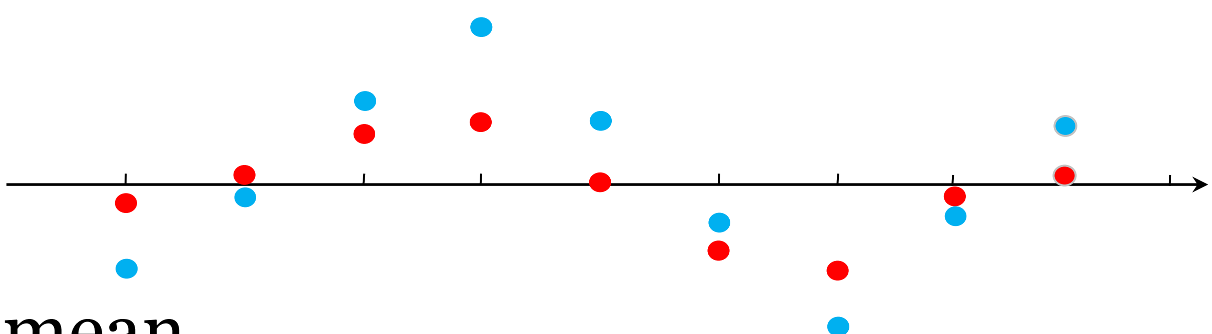
- zero mean
  - compute the mean of the sequence
  - subtract the mean from every value of the sequence

Pre-Processing
z-Normalization



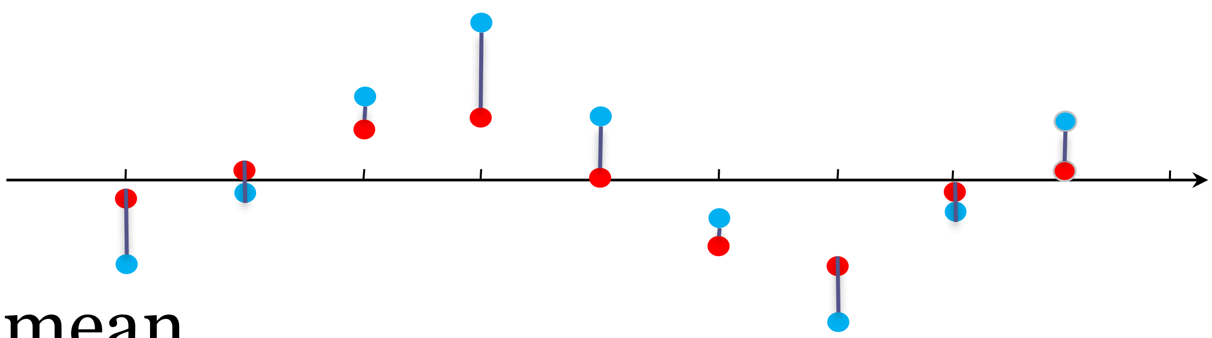
- zero mean
  - compute the mean of the sequence
  - subtract the mean from every value of the sequence

Pre-Processing
z-Normalization



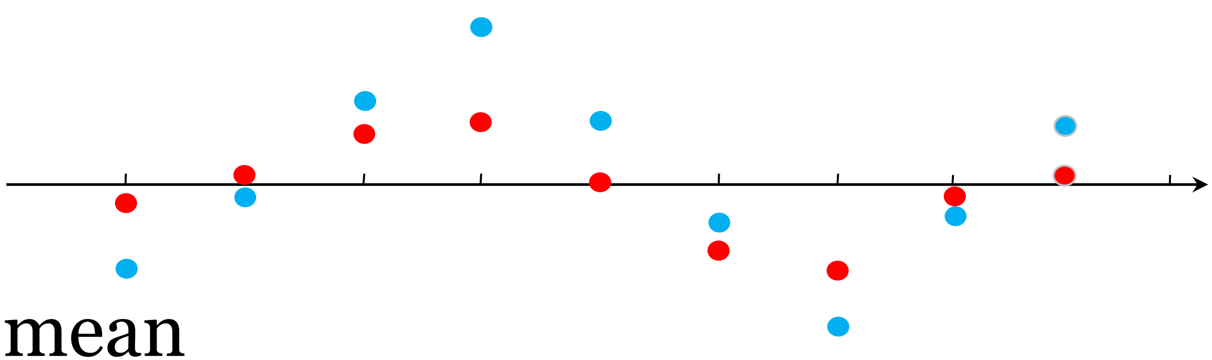
- zero mean
  - compute the mean of the sequence
  - subtract the mean from every value of the sequence

Pre-Processing
z-Normalization



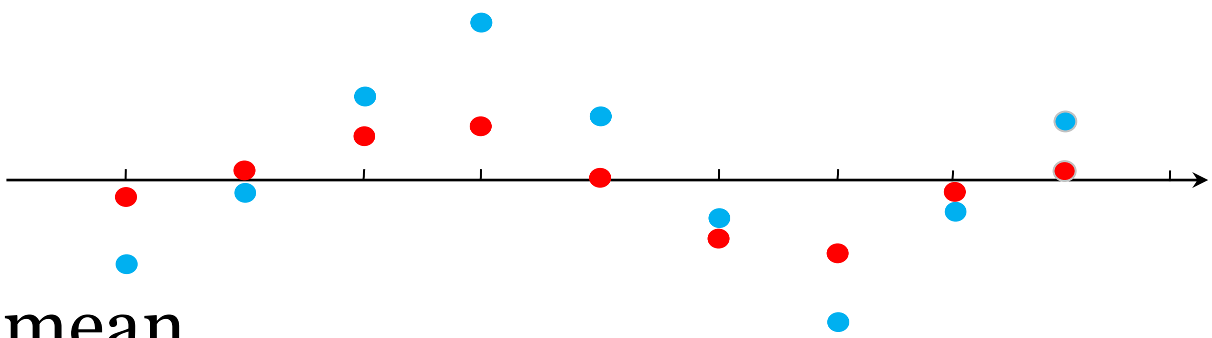
- zero mean
  - compute the mean of the sequence
  - subtract the mean from every value of the sequence

Pre-Processing
z-Normalization



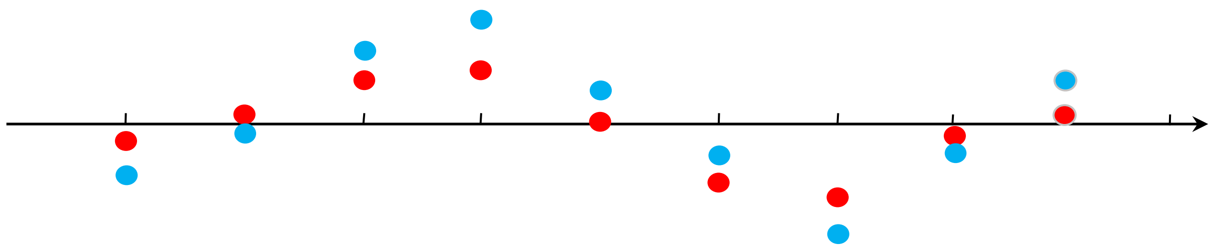
- zero mean
- standard deviation one
  - compute the standard deviation of the sequence
  - divide every value of the sequence by the stddev

Pre-Processing
z-Normalization



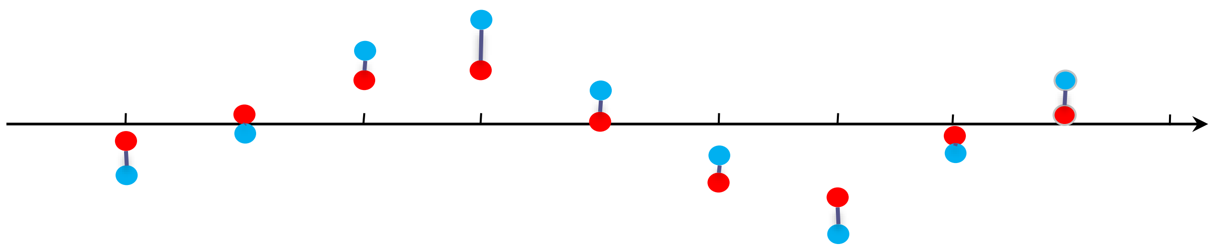
- zero mean
- standard deviation one
    - compute the standard deviation of the sequence
    - divide every value of the sequence by the stddev

Pre-Processing
z-Normalization



- zero mean
- standard deviation one
  - compute the standard deviation of the sequence
  - divide every value of the sequence by the stddev

Pre-Processing
z-Normalization



- zero mean
- standard deviation one

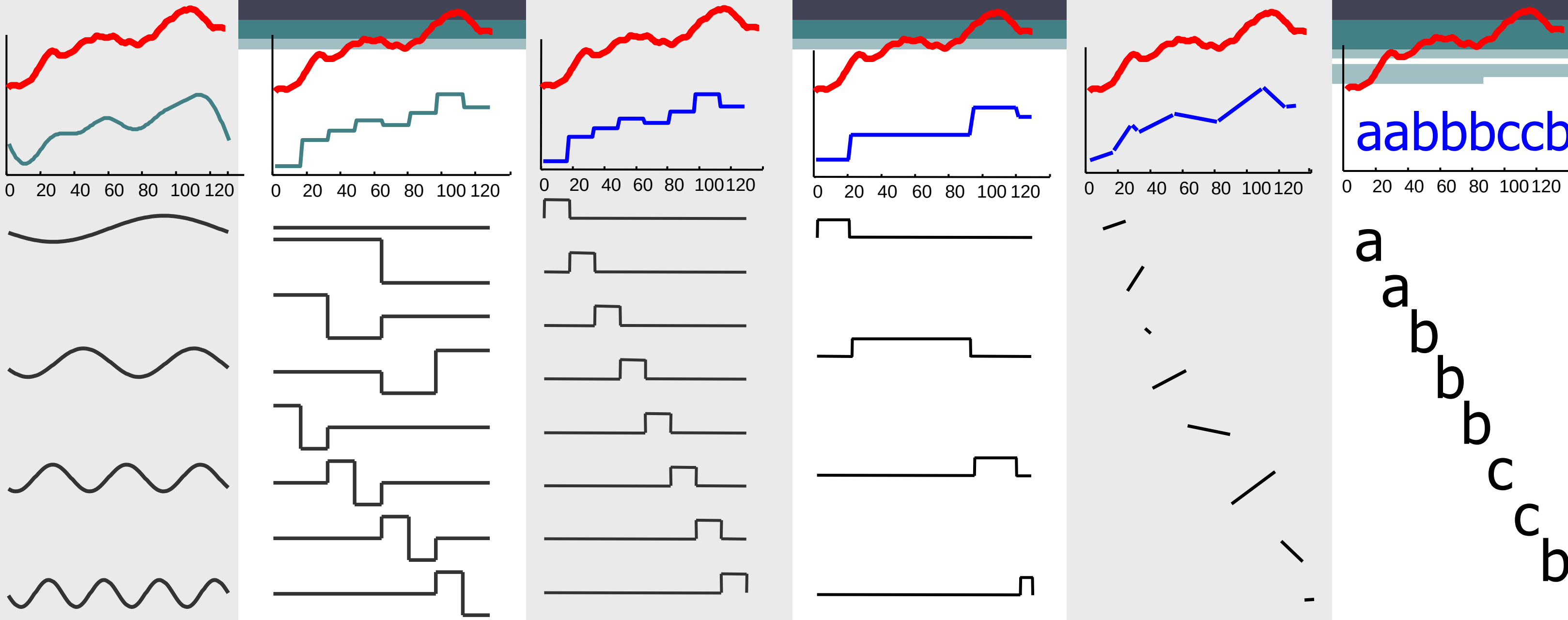
Pre-Processing
z-Normalization

- when to z-normalize
  - interested in trends

- when not to z-normalize
  - interested in absolute values

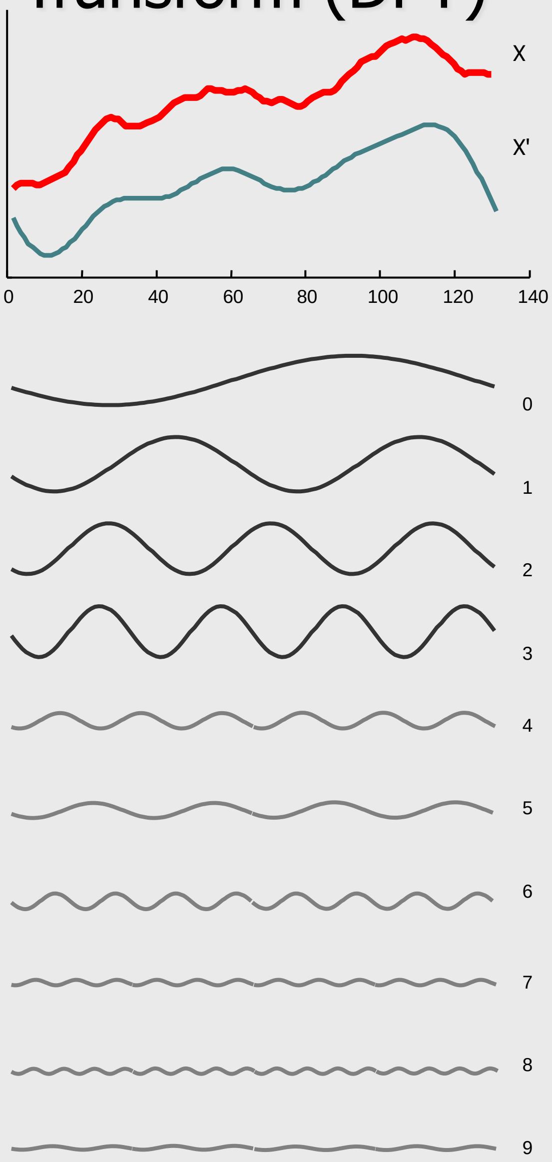
Outline

- terminology and definitions
- motivation
- pre-processing tasks
- data series representation techniques



DFT	DWT	PAA	APCA	PLA	SAX
Agrawal, Faloutsos, &. Swami. FODO 1993 Faloutsos, Ranganathan, & Manolopoulos. SIGMOD 1994	Chan & Fu. ICDE 1999	Keogh, Chakrabarti, Pazzani & Mehrotra KAIS 2000 Yi & Faloutsos VLDB 2000	Keogh, Chakrabarti, Pazzani & Mehrotra SIGMOD 2001	Morinaka, Yoshikawa, Amagasa, & Uemura, PAKDD 2001	

Discrete Fourier Transform (DFT)



Basic Idea: Represent the time series as a linear combination of sines and cosines

Transform the data from the time domain to the frequency domain

Highlight the periodicities but keep only the first *n/2* coefficients

Why *n*/2 coefficients?

✓ Because they are symmetric

Jean Fourier

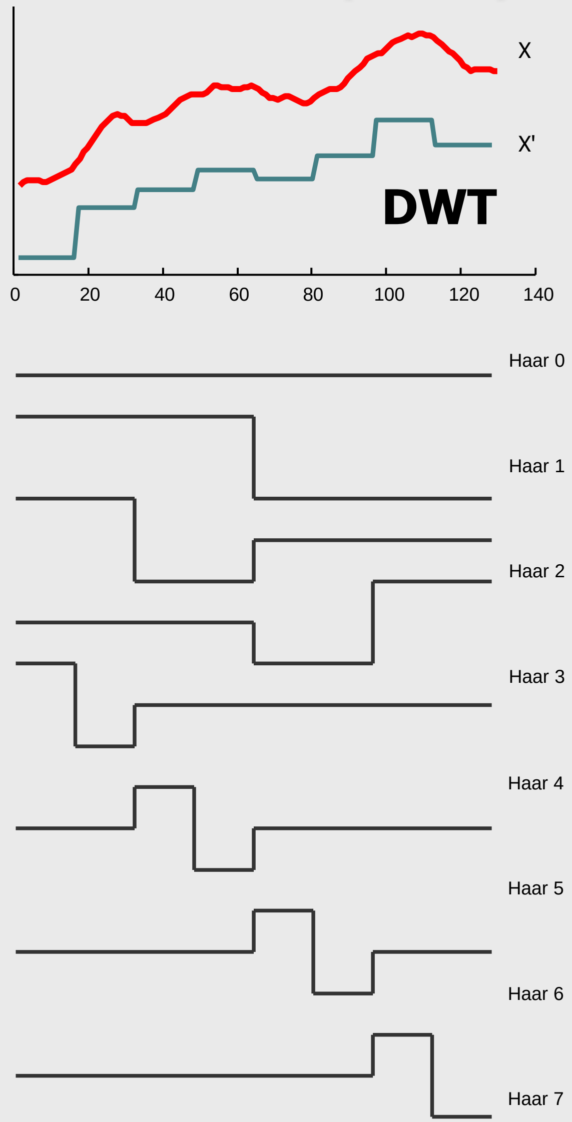
1768-1830

<u>Excellent free Fourier Primer</u>

Hagit Shatkay, The Fourier Transform - a Primer'', Technical Report CS-95-37, Department of Computer Science, Brown University, 1995.

http://www.ncbi.nlm.nih.gov/CBBresearch/Postdocs/Shatkay/

Discrete Wavelet Transform (DWT)



Basic Idea: Represent the time series as a linear combination of Wavelet basis functions, but keep only the first *N* coefficients

Obtained from a single prototype wavelet *ψ(t)* called *mother wavelet* by *dilations* and *shifting:*

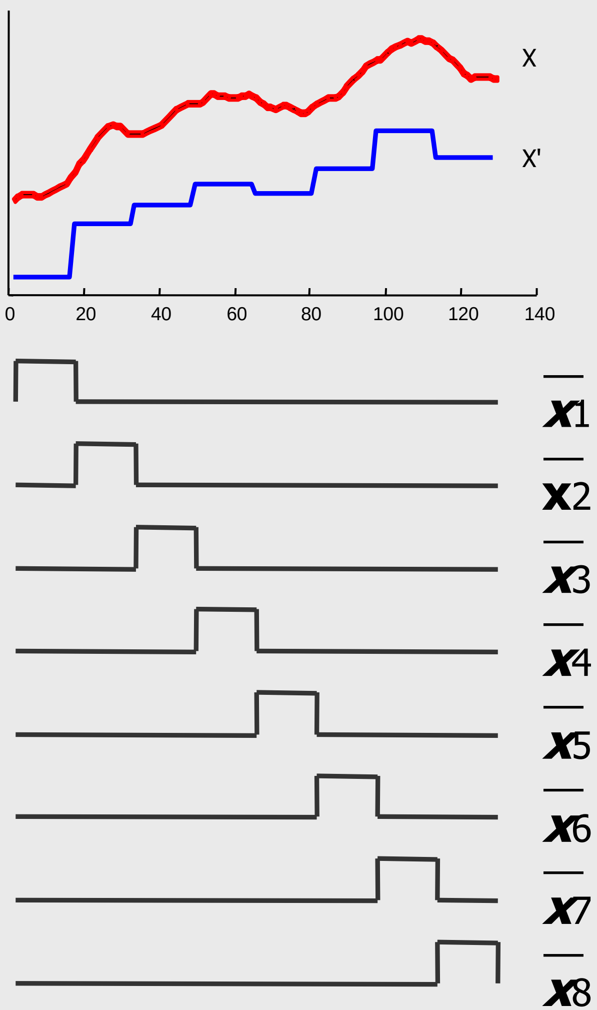
$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)$$

where *a* is the scaling parameter and *b* is the shifting parameter

Excellent free Wavelets Primer

Stollnitz, E., DeRose, T., & Salesin, D. (1995). *Wavelets for computer graphics A primer: IEEE Computer Graphics and Applications.*

Piecewise Aggregate Approximation (PAA)



Basic Idea: Represent the time series as a sequence of box basis functions, each box being of the same length

Computation:

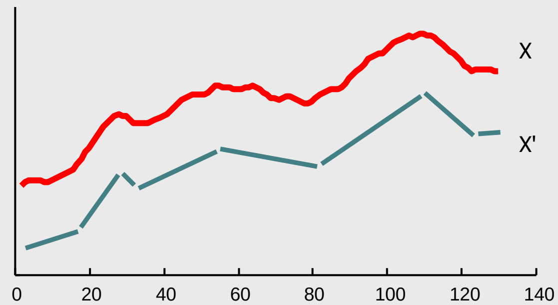
- X: time series of length n
- Can be represented in the N-dimensional space as:

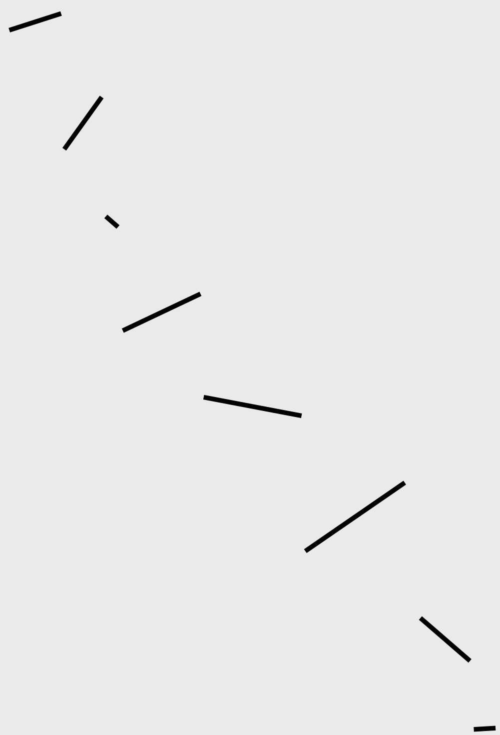
$$\bar{x}_i = \frac{N}{n} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} x_j$$

Keogh, Chakrabarti, Pazzani & Mehrotra, KAIS (2000)

Byoung-Kee Yi, Christos Faloutsos, VLDB (2000)

Piecewise Linear Approximation (PLA)





Basic Idea: Represent the time series (size n) as a sequence of straight lines (size N)

Lines could be **connected** **=> N/2** lines allowed

Lines could be **disconnected** **=> N/3** lines allowed

Empirical evidence on dozens of datasets suggests that **disconnected** is better

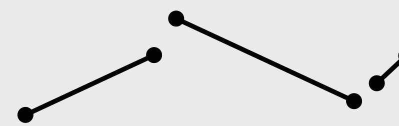
Also only **disconnected** allows a lower bounding Euclidean approximation

Karl Friedrich Gauss

1777 - 1855

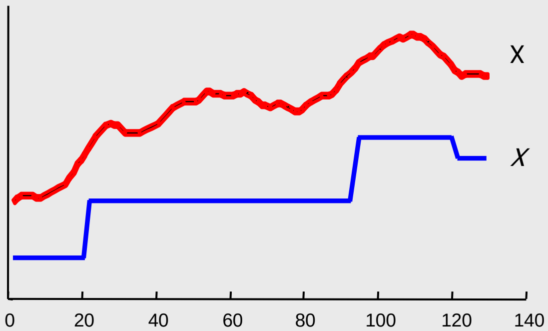
Each line segment has
- `length`
- `left_height`

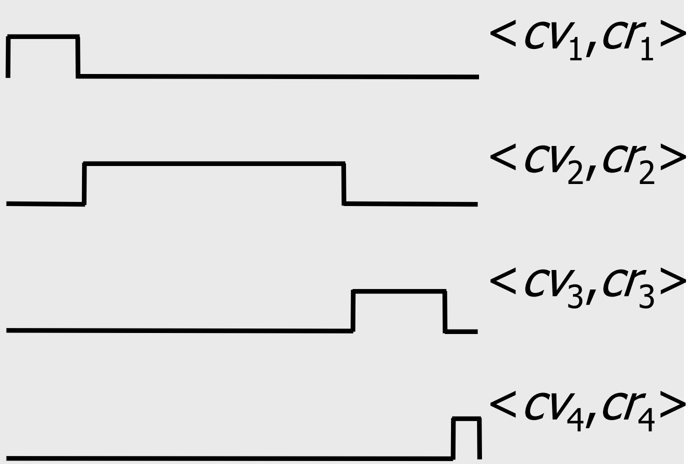
(`right_height` can be inferred by looking at the next segment)



Each line segment has
- `length`
- `left_height`
- `right_height`

Adaptive Piecewise Constant Approximation (APCA)





Basic Idea: Represent the time series as a sequence of box basis functions, each box being of the *different* length

- High quality of APCA noted by many researchers
- Can be indexed*!

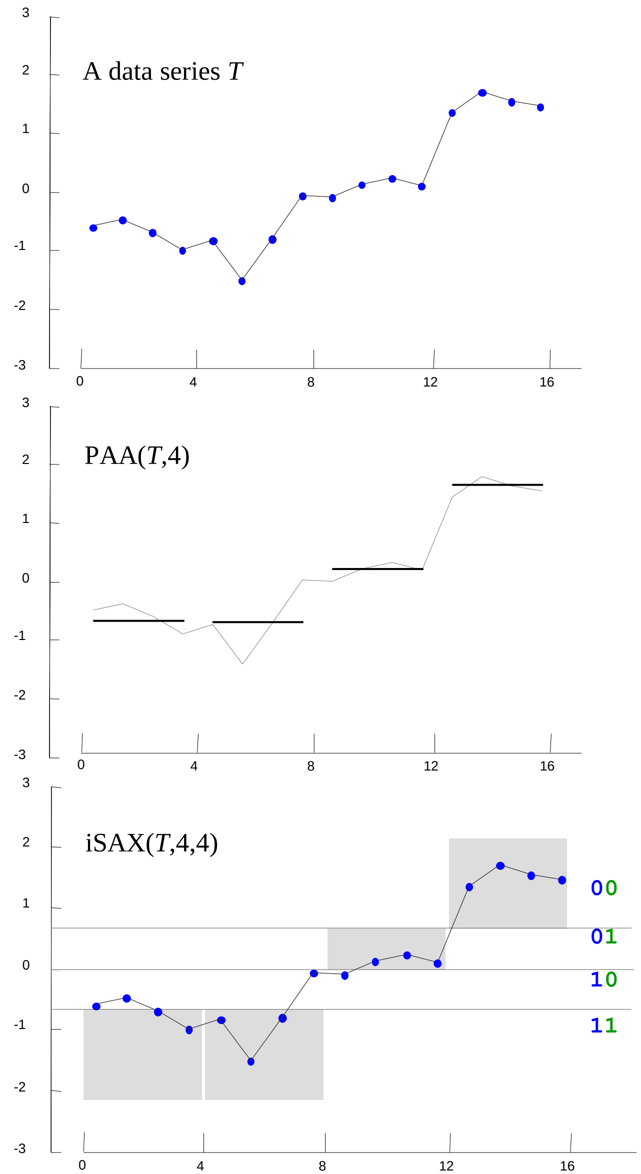
Unfortunately, it is non-trivial to understand and implement and thus has only been re-implemented once or twice

*K. Chakrabarti, E. J. Keogh, S. Mehrotra, M. J. Pazzani: Locally adaptive dimensionality reduction for indexing large time series databases. ACM Trans. Database Syst. 27(2): 188-228 (2002)

SAX Representation

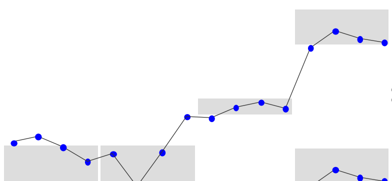
- **S**ymbolic **A**ggregate appro**X**imation (SAX)
  - **(1)** Represent data series T of length $\boldsymbol{n}$ with $\boldsymbol{w}$ segments using Piecewise Aggregate Approximation (PAA)
    - T typically normalized to $\mu = 0, \sigma = 1$
    - PAA(T,w) $= \bar{T} = \bar{t}_1, \ldots, \bar{t}_w$

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

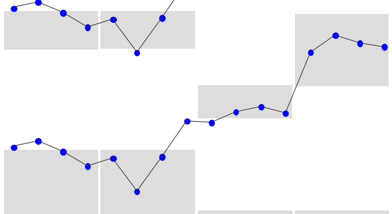


*i*SAX Representation

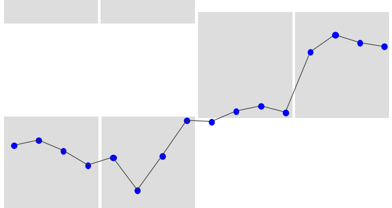
- *i*SAX offers a bit-aware, quantized, multi-resolution representation with variable granularity



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



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



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

Publications

Palpanas-TKDE'08

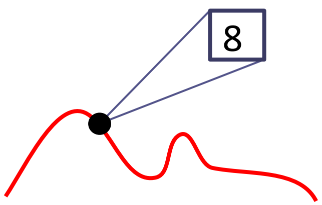
Comparison of Representations

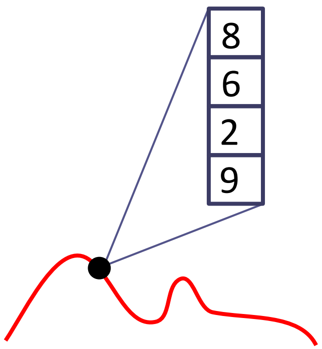
- which representation is the best?
- depends on data characteristics
  - periodic, smooth, spiky, ...
- overall (averaged over many diverse datasets, using same memory budget), when measuring reconstruction error (RMSE)
  - no big differences among methods
  - DFT, PAA, DWT (Haar), iSAX slightly better
- should also take into account other factors
  - visualization, indexable, ...

Data Series Similarity Problem Variations

Problem Variations

Series





<u>Univariate</u>

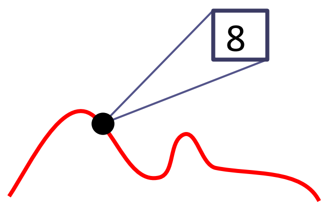
each point represents one value (e.g., temperature)

<u>Multivariate</u>

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

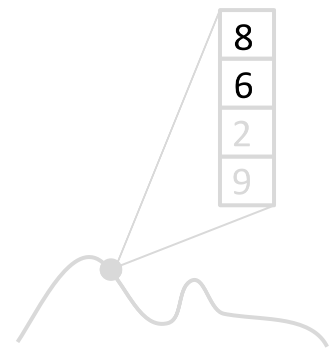
Problem Variations

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

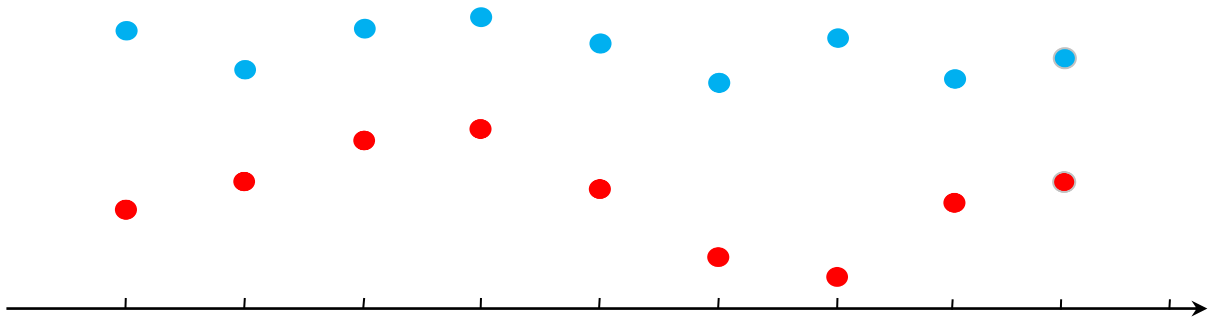
Paparrizos-SIGMOD'20

Problem Variations

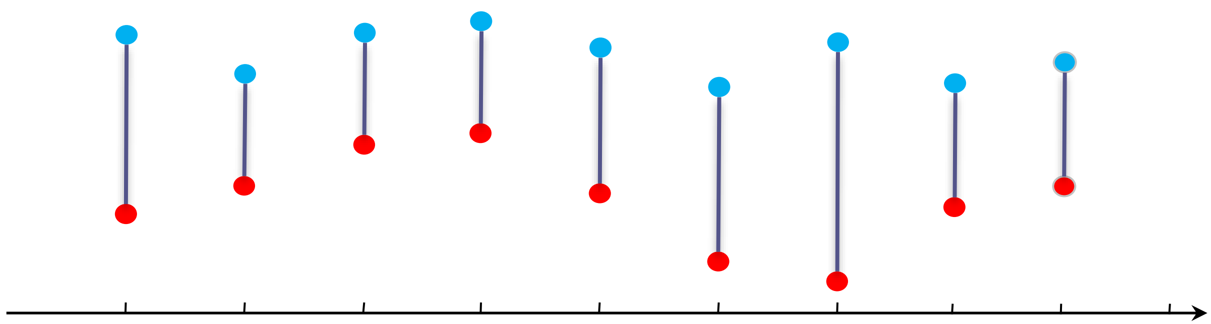
Distance Measures

- 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, ...

Euclidean 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 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 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):

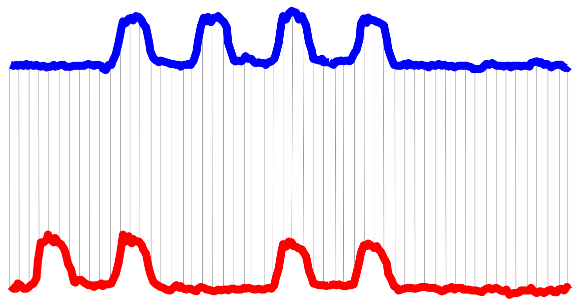
$$PC = \frac{1}{n-1}\sum_{i=1}^{n} x_i \cdot y_i \quad \text{and} \quad ED^2 = 2n(n-1) - 2\sum_{i=1}^{n} x_i y_i$$

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

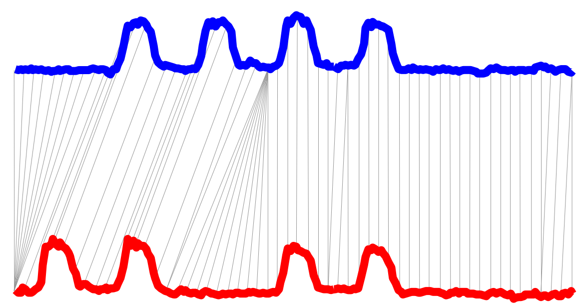
- direct connection between ED and PC for Z-normalized data series
  - if ED is calculated for normalized data series, it can be directly used to calculate the p-value for statistical test of Pearson's correlation instead of actual PC value.

Distance Measures: LCSS against Euclidean, DTW

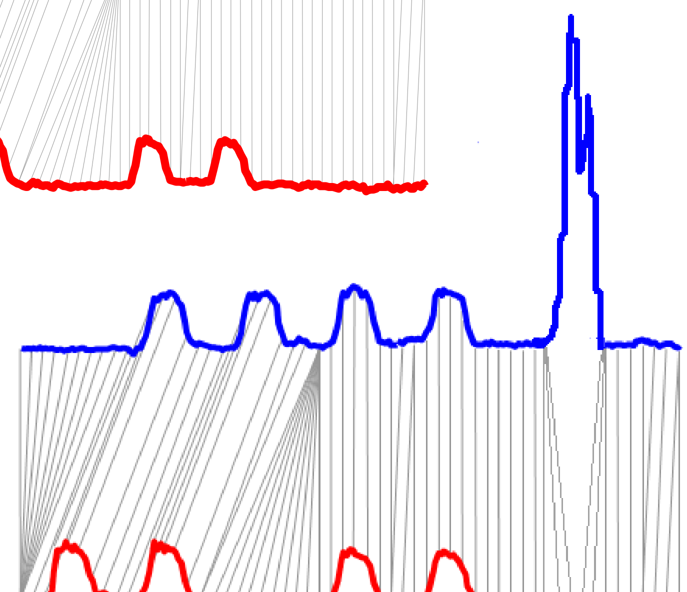
- Euclidean
  - rigid



- Dynamic Time Warping (DTW)
  - allows local scaling

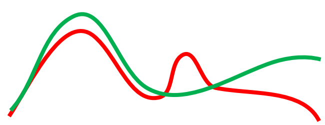


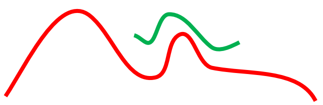
- Longest Common SubSequence (LCSS)
  - allows local scaling
  - ignores outliers



Problem Variations

Queries





<u>Whole matching</u>

Entire query

Entire candidate

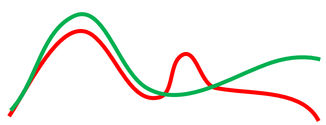
<u>Subsequence matching</u>

Entire query

A subsequence of a candidate

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...

Similarity Matching

- given a data series collection D and a query data series q, return the data series from D that are the most similar to q
  - there exist different flavors of this basic operation

- basis for most data series analysis tasks

Similarity Matching
Nearest Neighbor (NN) Search

- given a data series collection D and a query data series q, return the data series from D that has the smallest distance to q

- result set contains one data series

Similarity Matching
Nearest Neighbor (NN) Search

- serial scan
  - compute the distance between q and every $d_i \in D$
  - return d_i with the smallest distance to q

Similarity Matching
Nearest Neighbor (NN) Search

- serial scan
  - bsf = Inf // best so far distance
  - for every $d_i \in D$
    - compute distance, dist, between d_i and q
    - if this dist less than bsf then bsf=dist
  - return d_i corresponding to bsf

Similarity Matching
k-Nearest Neighbors (kNN) Search

- given a data series collection D and a query data series q, return the k data series from D that have the k smallest distances to q

- result set contains k data series

Similarity Matching
k-Nearest Neighbors (kNN) Search

- serial scan
  - compute the distance between q and every $d_i \in D$
  - return the k d_i with the k smallest distances to q

Similarity Matching
k-Nearest Neighbors (kNN) Search

- serial scan
  - kbsf = Null // best so far max-heap of k elements
  - for every $d_i \in D$
    - compute distance, dist, between d_i and q
    - if this dist less than max of kbsf then insert dist in kbsf
  - return k d_i corresponding to k elements in kbsf

Similarity Matching
ε-Range Search

- given a data series collection D and a query data series q, return all data series from D that are within distance ε from q

- result set contains [?] data series

Similarity Matching
ε-Range Search

- serial scan
  - compute the distance between q and every $d_i \in D$
  - return all d_i with distance less than ε to q

Similarity Matching
ε-Range Search

- serial scan
  - res = {} // empty result set
  - for every $d_i \in D$
    - compute distance, dist, between d_i and q
    - if this dist less than ε then insert dist in res
  - return all d_i corresponding to elements in res

Problem Variations

Queries

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

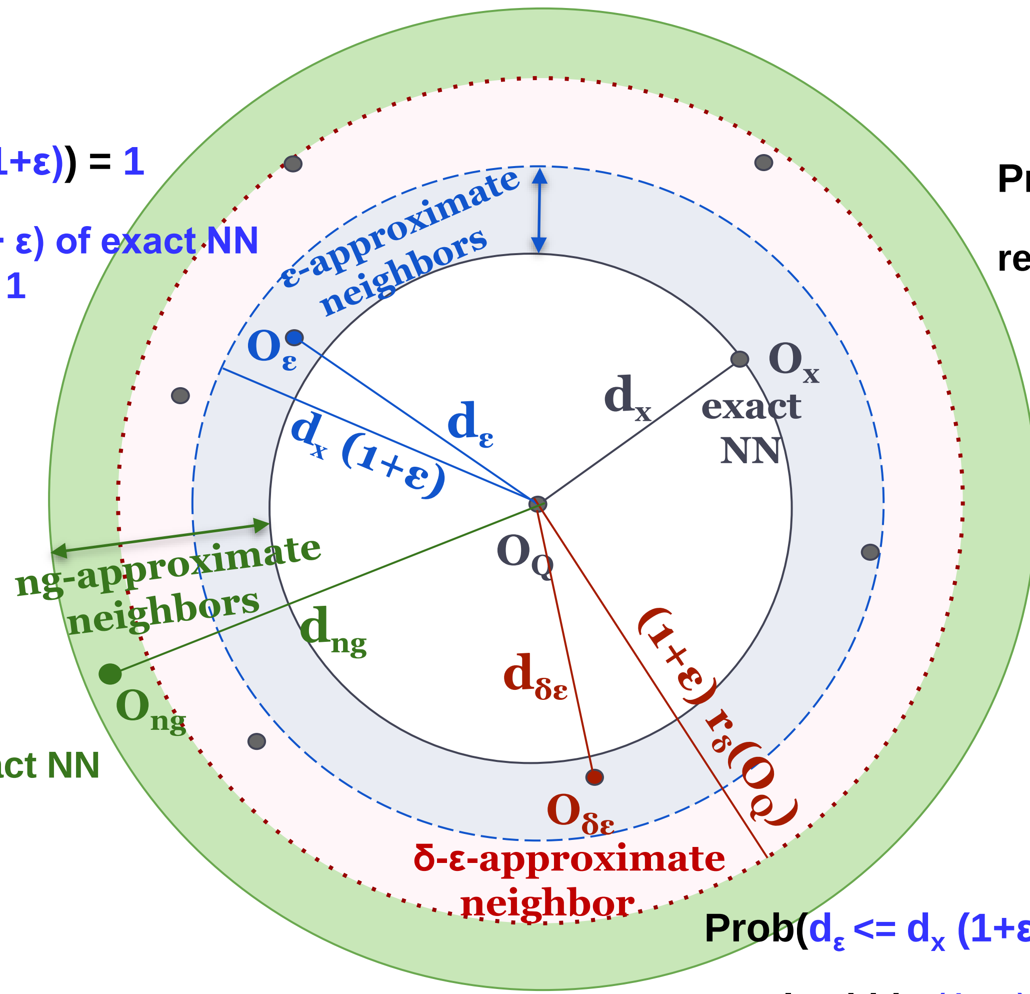
Nearest Neighbor (NN) Queries…

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} <>= ?) = ?

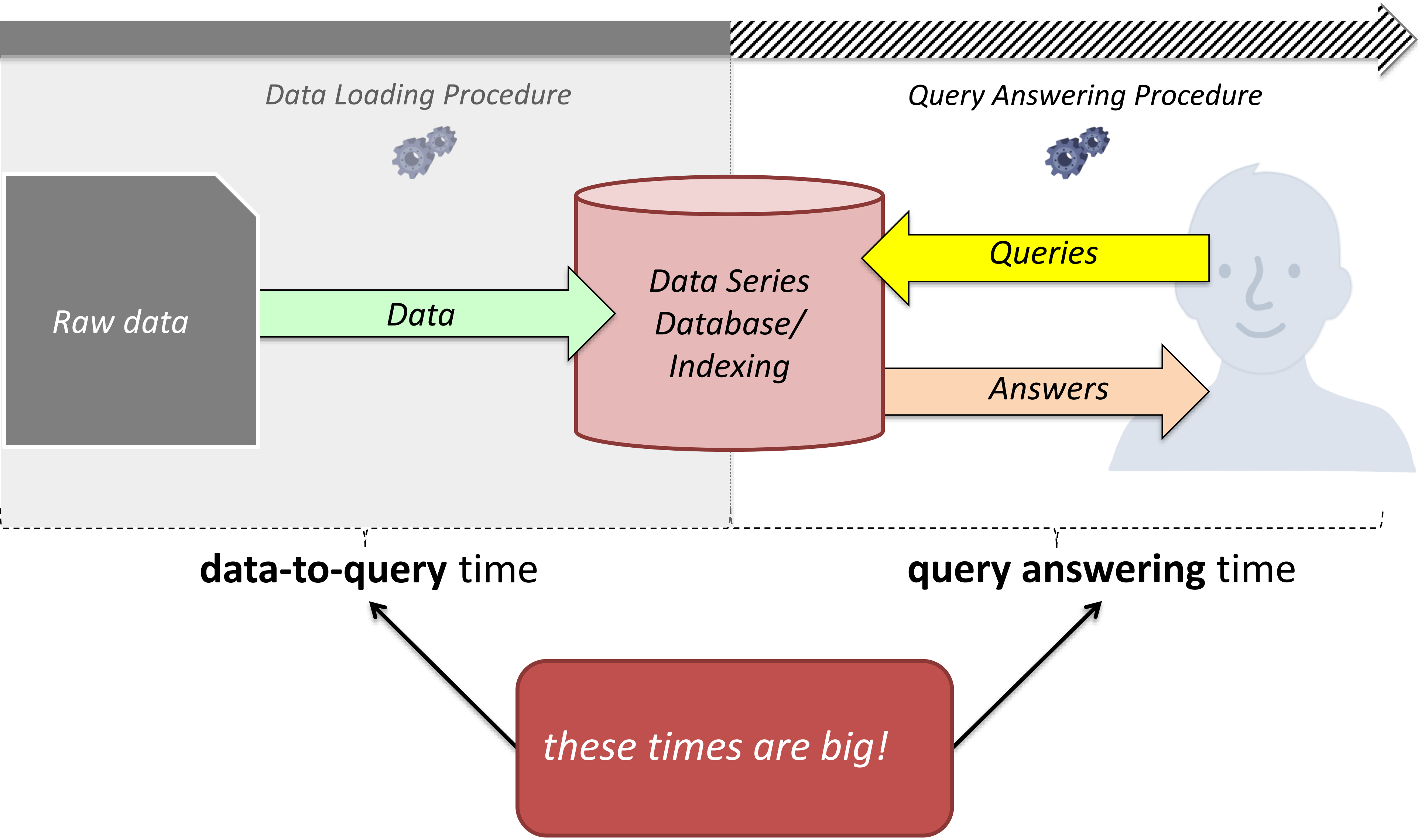
result within ? of exact NN

Prob(d_ε <= d_x (1+ε)) >= δ

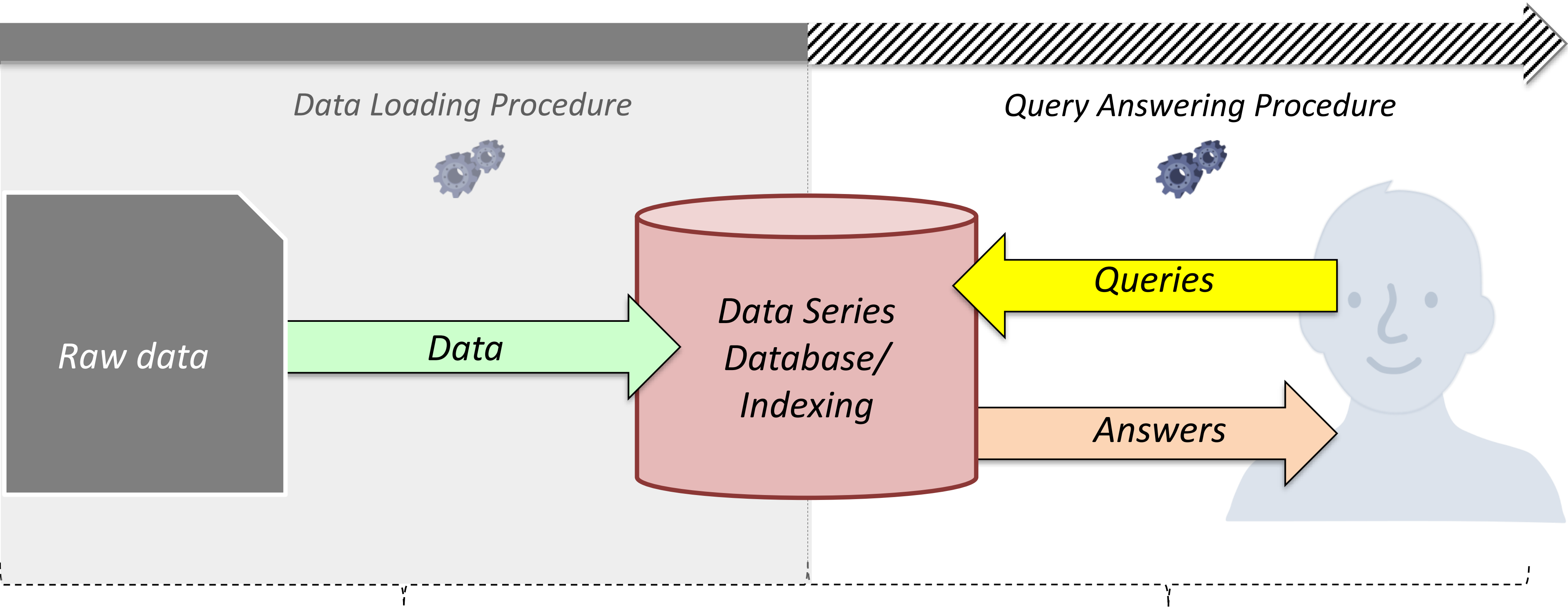
result within (1+ ε) of exact NN
with probability at least δ

Data Series Similarity Query Answering

Query answering process



Query answering process



data-to-query time

query answering time

we need solutions for both problems!

Similarity Matching
Fast Euclidean Distance

- similarity matching requires many distance computations
  - can significantly slow down processing
    - because of large number of data series in the collection
    - because of high dimensionality of each data series

- in case of Euclidean Distance, we can speedup processing by
  - smart implementation of distance function
  - early abandoning

- result in considerable performance improvement

Publications

Keogh-DMKD'03

Similarity Matching
Fast Euclidean Distance

- smart implementation of distance function

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

Publications

Keogh-DMKD'03

Similarity Matching
Fast Euclidean Distance

- smart implementation of distance function
  - do not compute the square root (of the Euclidean Distance)

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

- does not alter the results
- saves precious CPU cycles

Publications

Keogh-DMKD'03

Similarity Matching
Fast Euclidean Distance

- early abandoning
  - stop the distance computation as soon as it exceeds the value of bsf

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

- does not alter the results
- avoids useless computations

GEMINI Framework

Publications

Faloutsos-SIGMOD'94

- Raw data: original full-dimensional space
- Summarization: reduced dimensionality space
- Searching in original space *costly*
- Searching in reduced space *faster*:
  - Less data, indexing techniques available, lower bounding
- Lower bounding enables us to
  - *prune search space:* throw away data series based on reduced dimensionality representation
  - *guarantee correctness* of answer
    - no false negatives
    - false positives filtered out based on raw data

GEMINI Framework

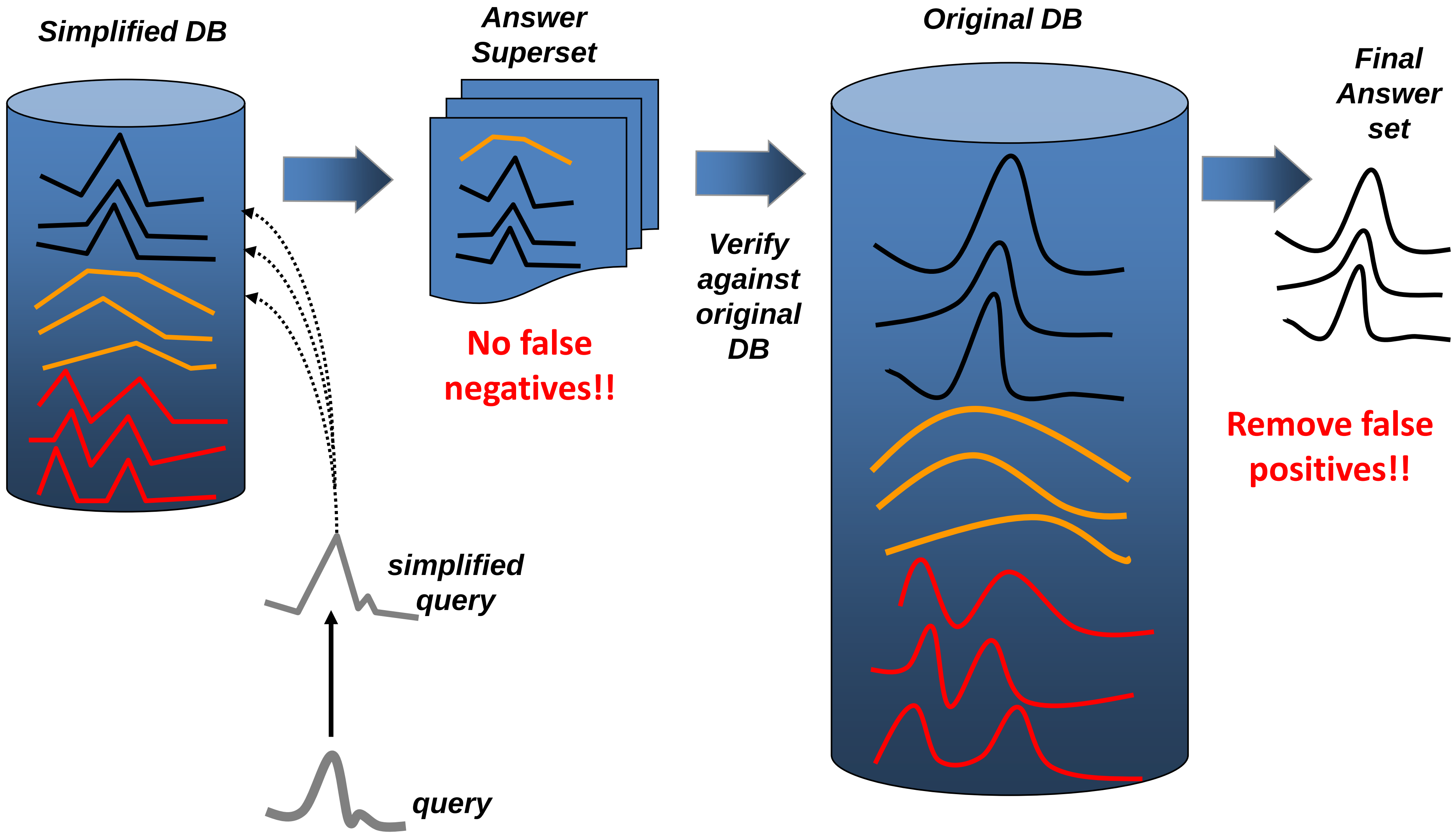
Publications

Faloutsos-SIGMOD'94

GEMINI Solution: Quick filter-and-refine:

- extract *m* features (numbers, e.g., average)
- map to point in *m*-dimensional feature space
- organize points
- retrieve the answer using a NN query
- discard false positives

Generic Search using Lower Bounding



GEMINI: contractiveness

Publications

Faloutsos-SIGMOD'94

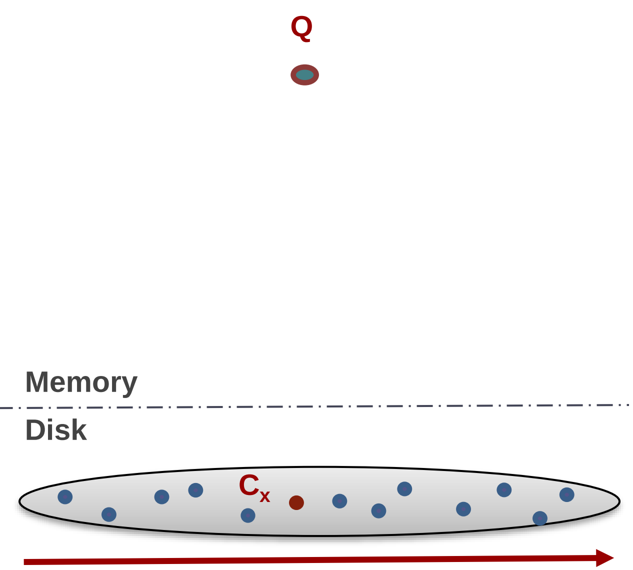
- GEMINI works when:

$$D_{feature}(F(x), F(y)) \;<=\; D(x, y)$$

- *Note that, the closer the feature distance to the actual one, the better*

Similarity Search Classes of Methods

Similarity Matching
Serial Scan

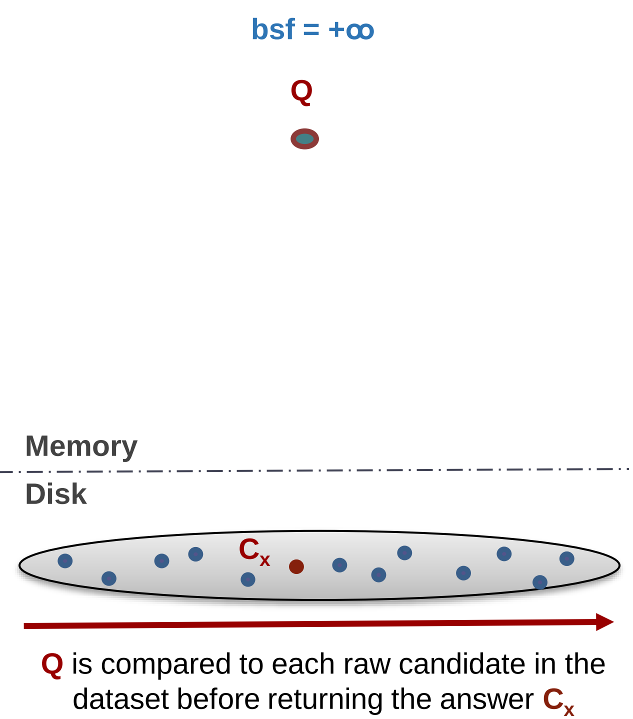


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

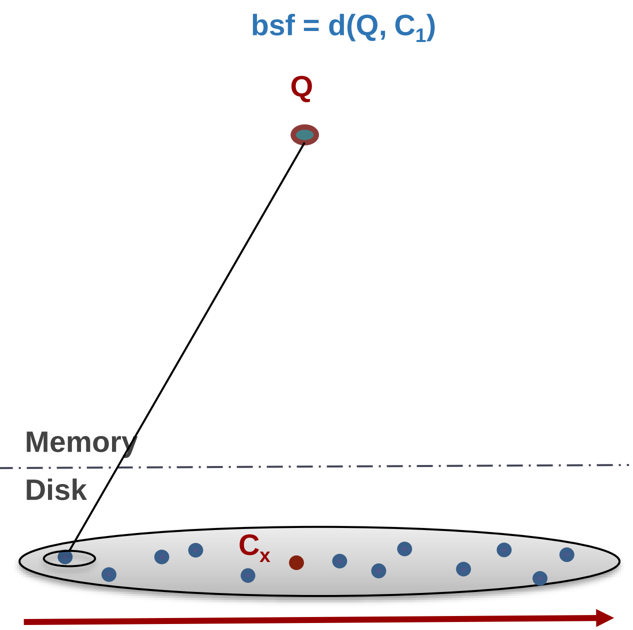
Similarity Matching
Serial Scan



(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching
Serial Scan

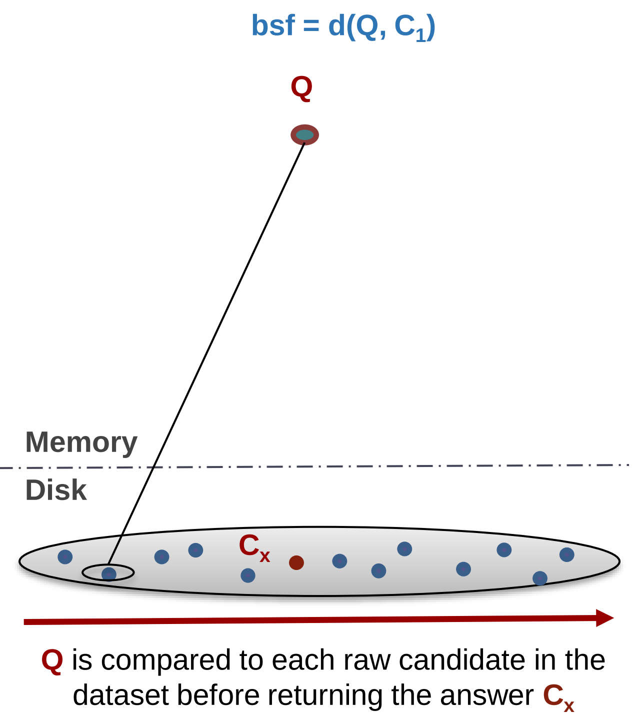


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

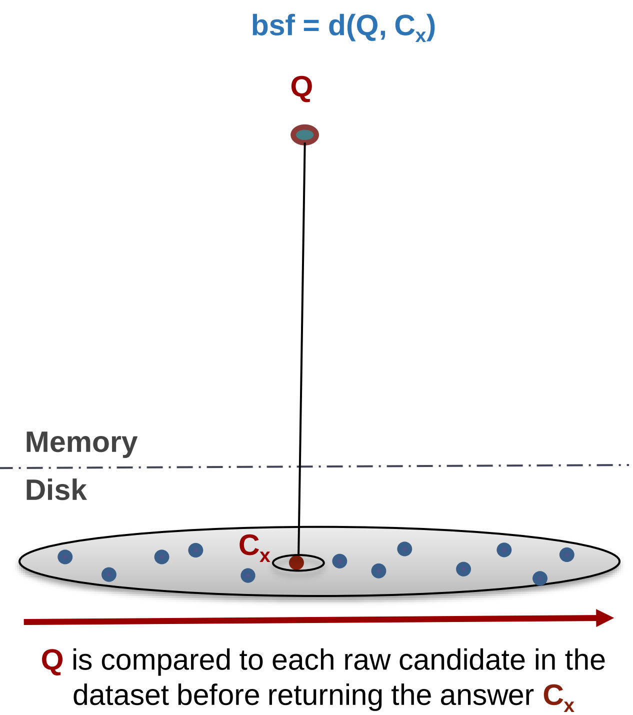
Similarity Matching
Serial Scan



(a) Serial scan

Answering a similarity search query using different access paths

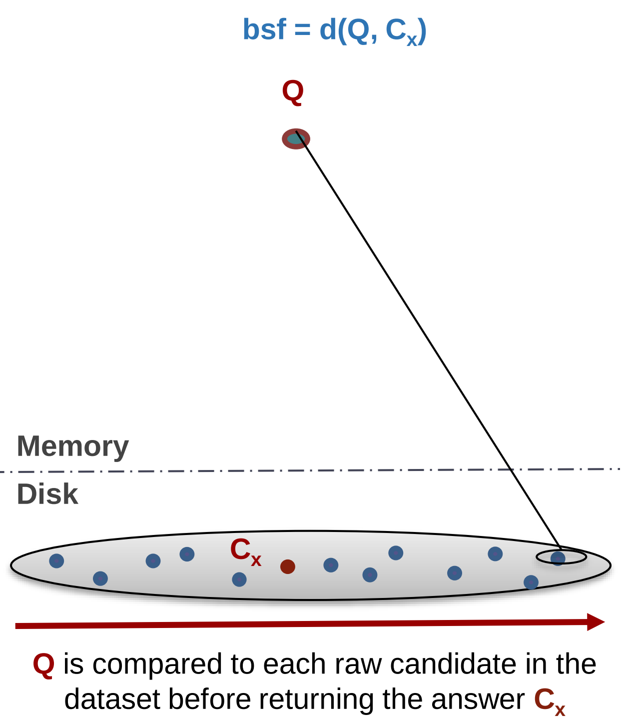
Similarity Matching
Serial Scan



(a) Serial scan

Answering a similarity search query using different access paths

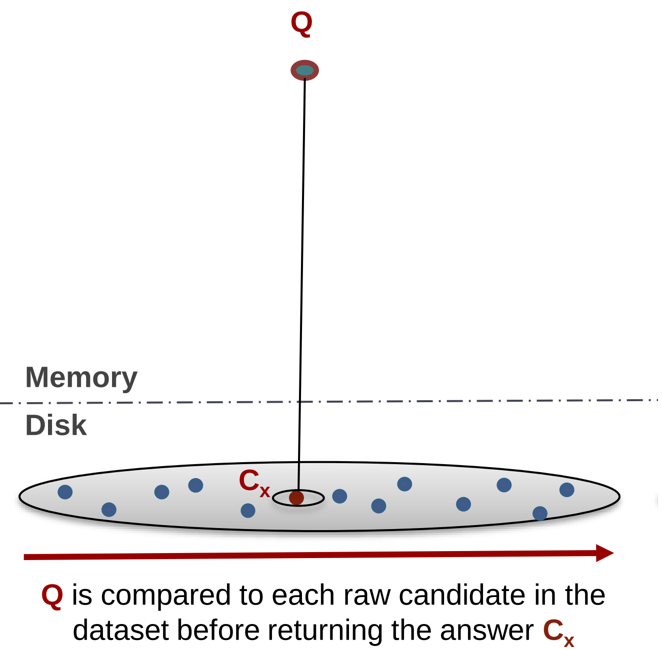
Similarity Matching
Serial Scan



(a) Serial scan

Answering a similarity search query using different access paths

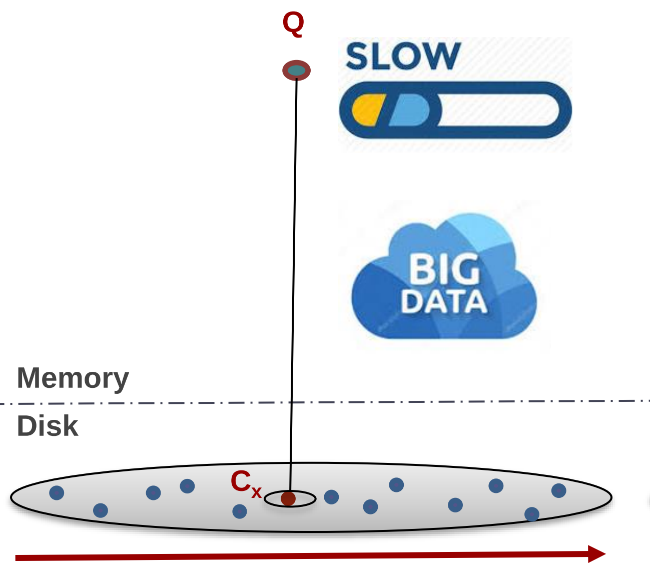
Similarity Matching
Serial Scan



(a) Serial scan

Answering a similarity search query using different access paths

Similarity Matching
Serial Scan

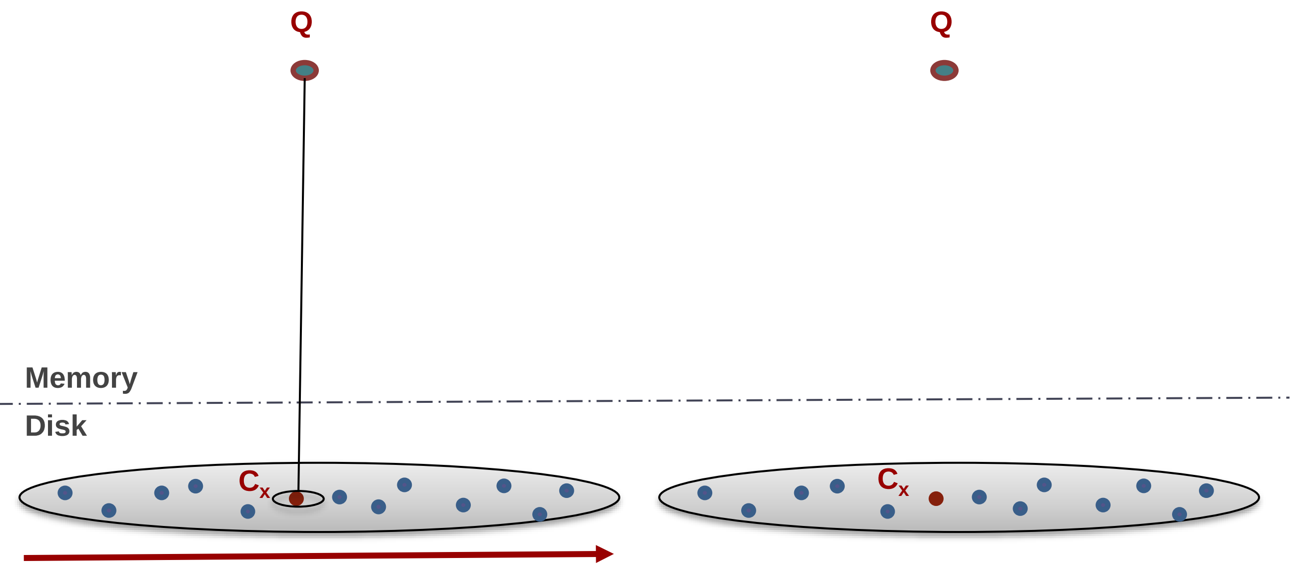


Q is compared to each raw candidate in the dataset before returning the answer C_x

(a) Serial scan

Answering a similarity search query using different access paths

Indexes vs. Scans



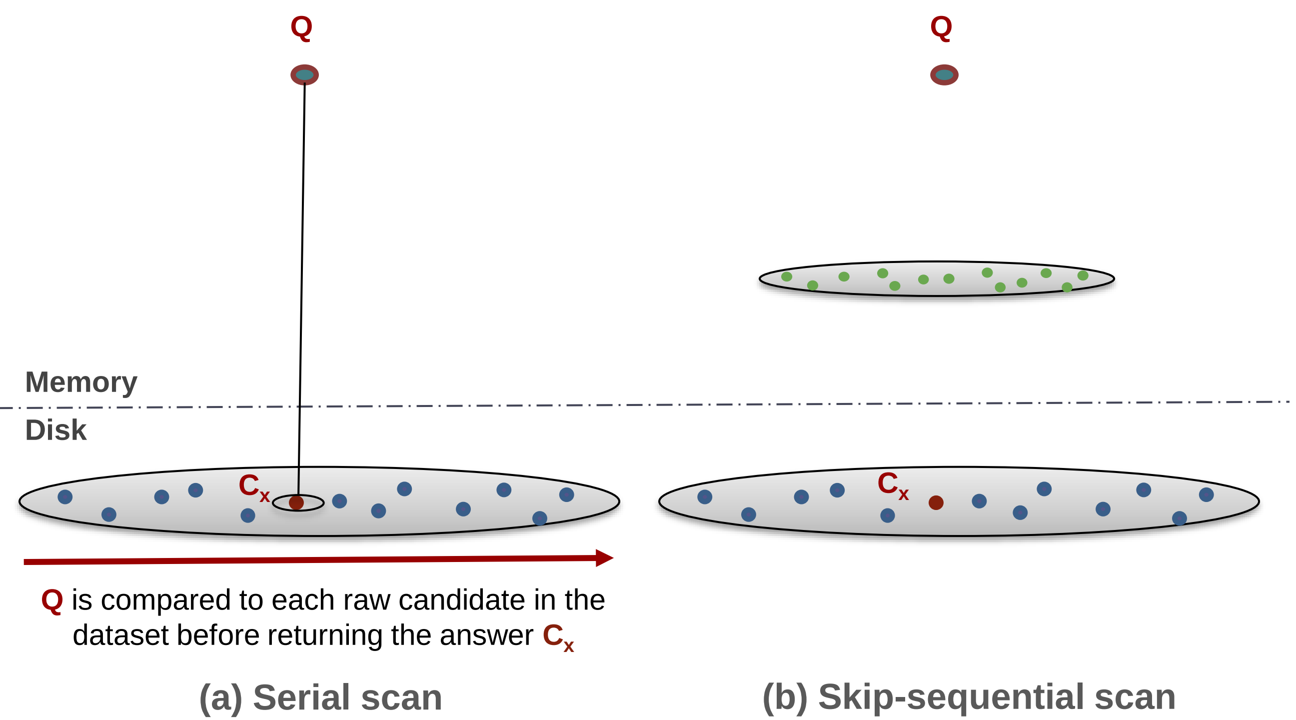
Q is compared to each raw candidate in the dataset before returning the answer **C_x**

(a) Serial scan

(b) Skip-sequential scan

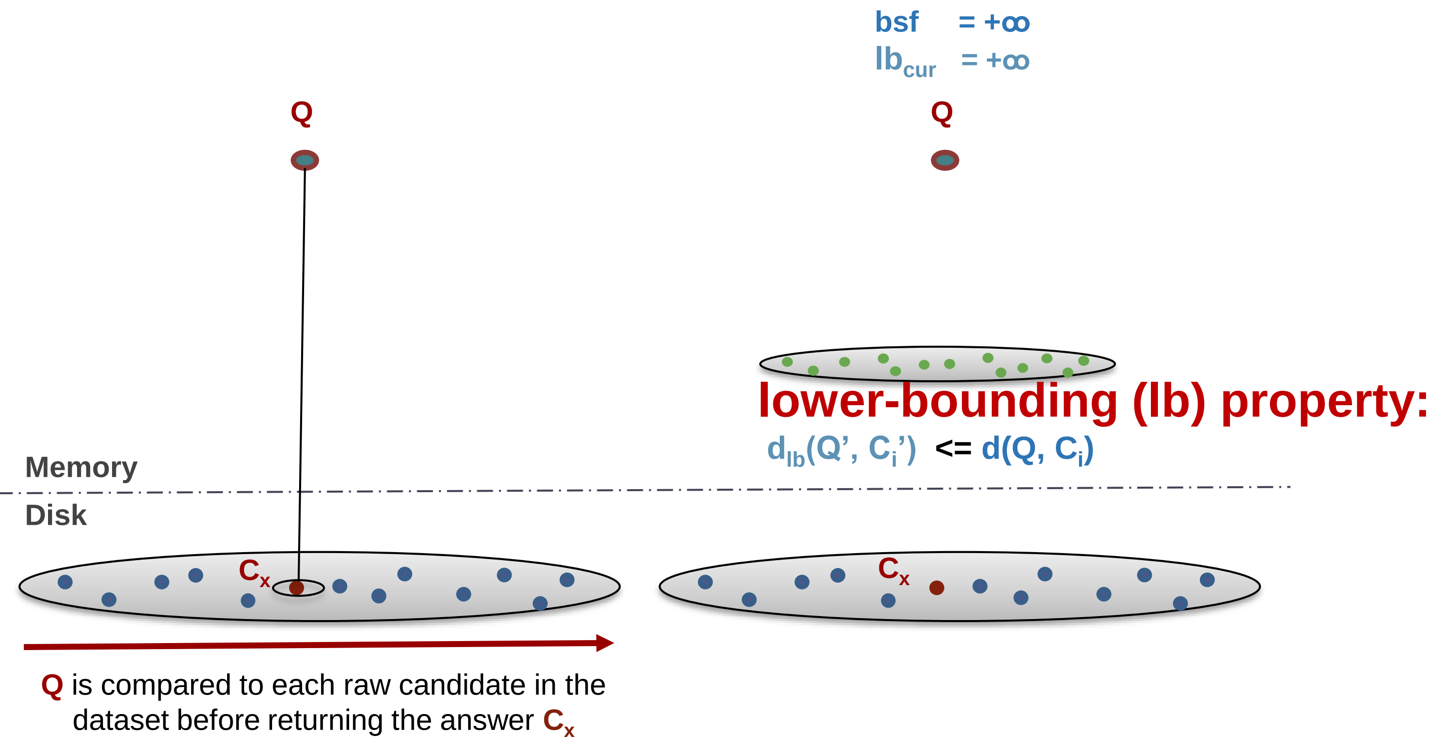
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans

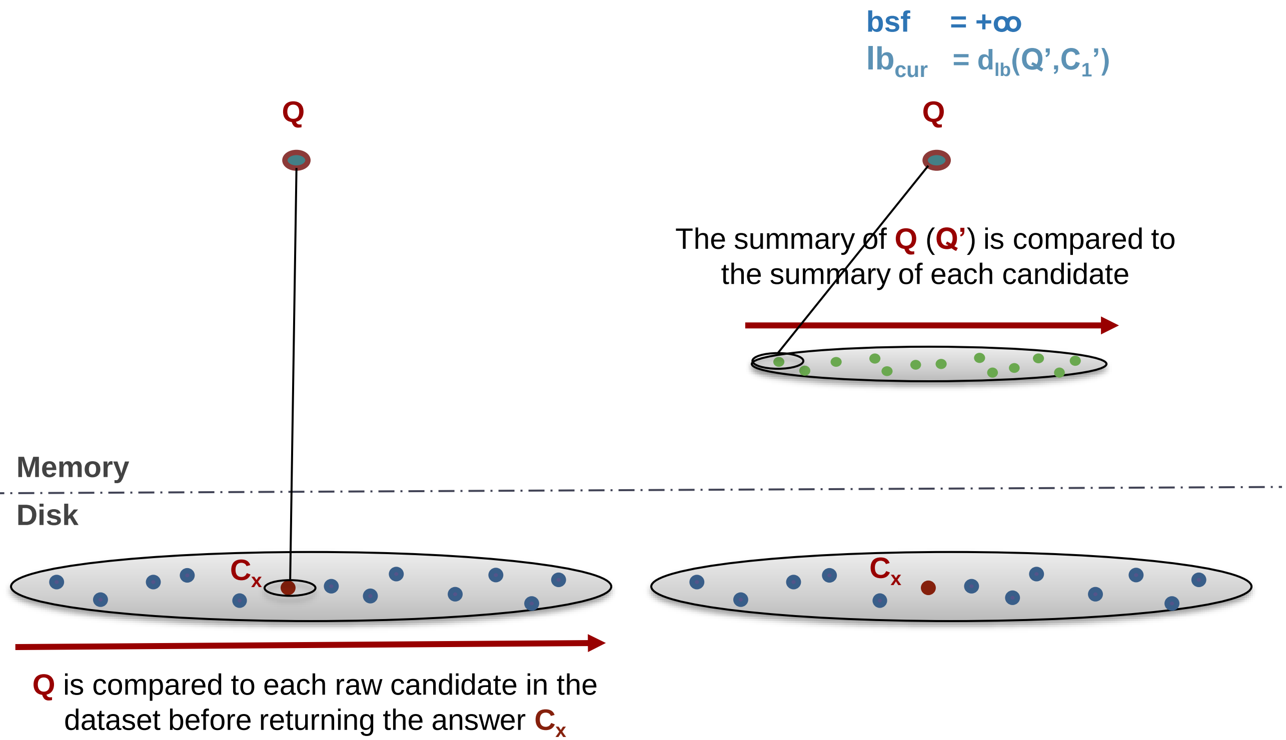


(a) Serial scan

(b) Skip-sequential scan

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

Indexes vs. Scans

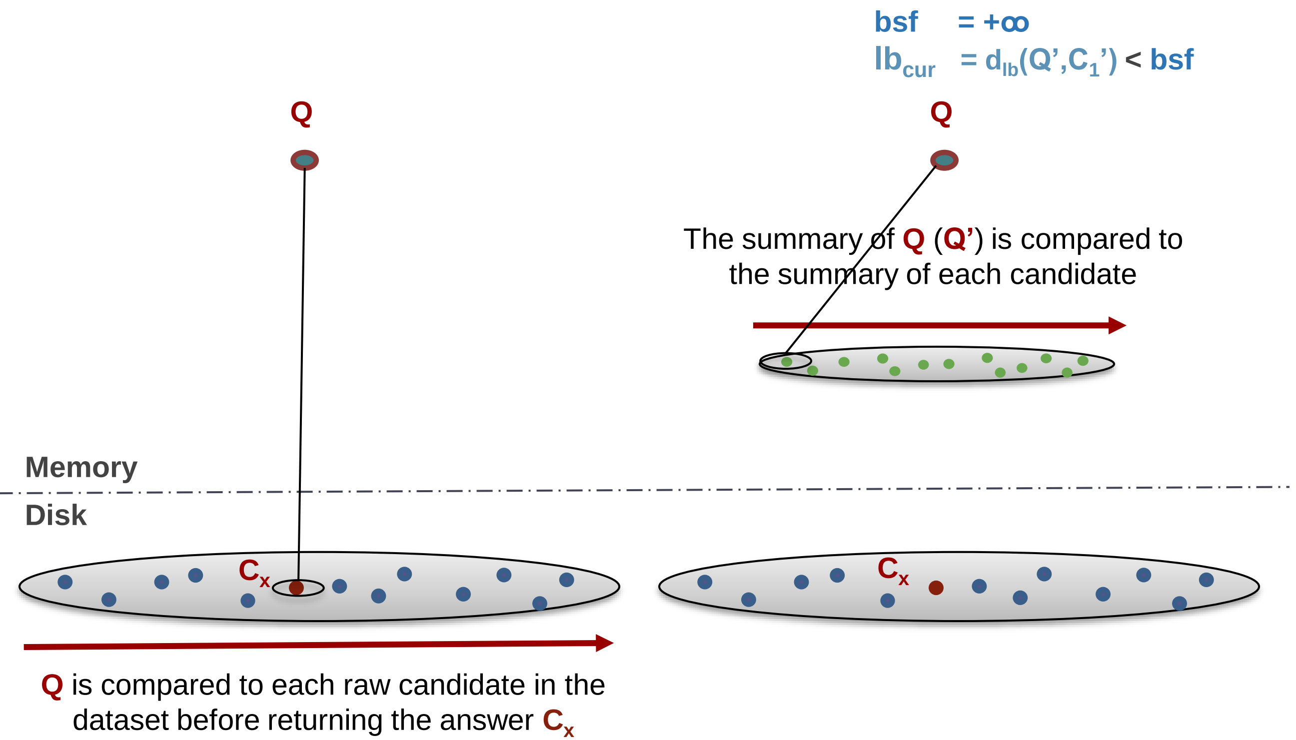


(a) Serial scan

(b) Skip-sequential scan

Answering a similarity search query using different access paths

Indexes vs. Scans

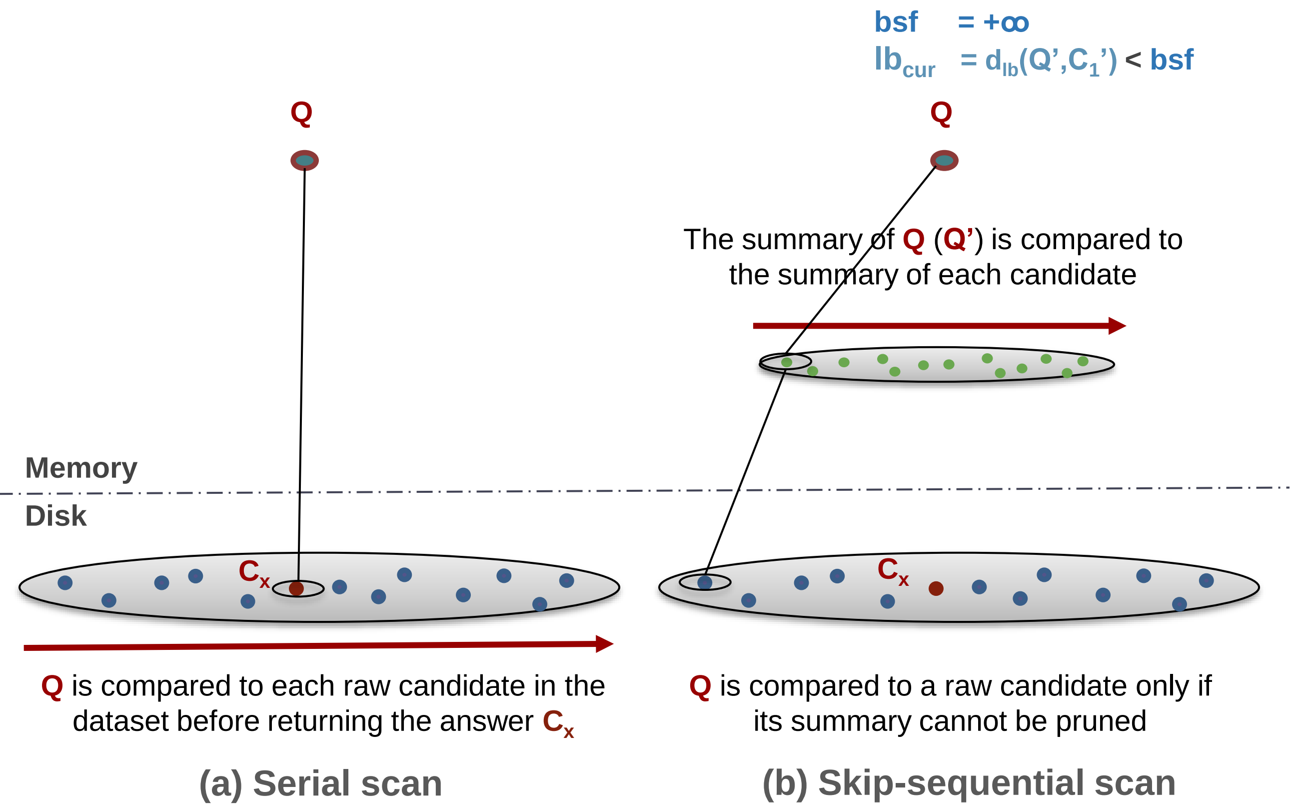


(a) Serial scan

(b) Skip-sequential scan

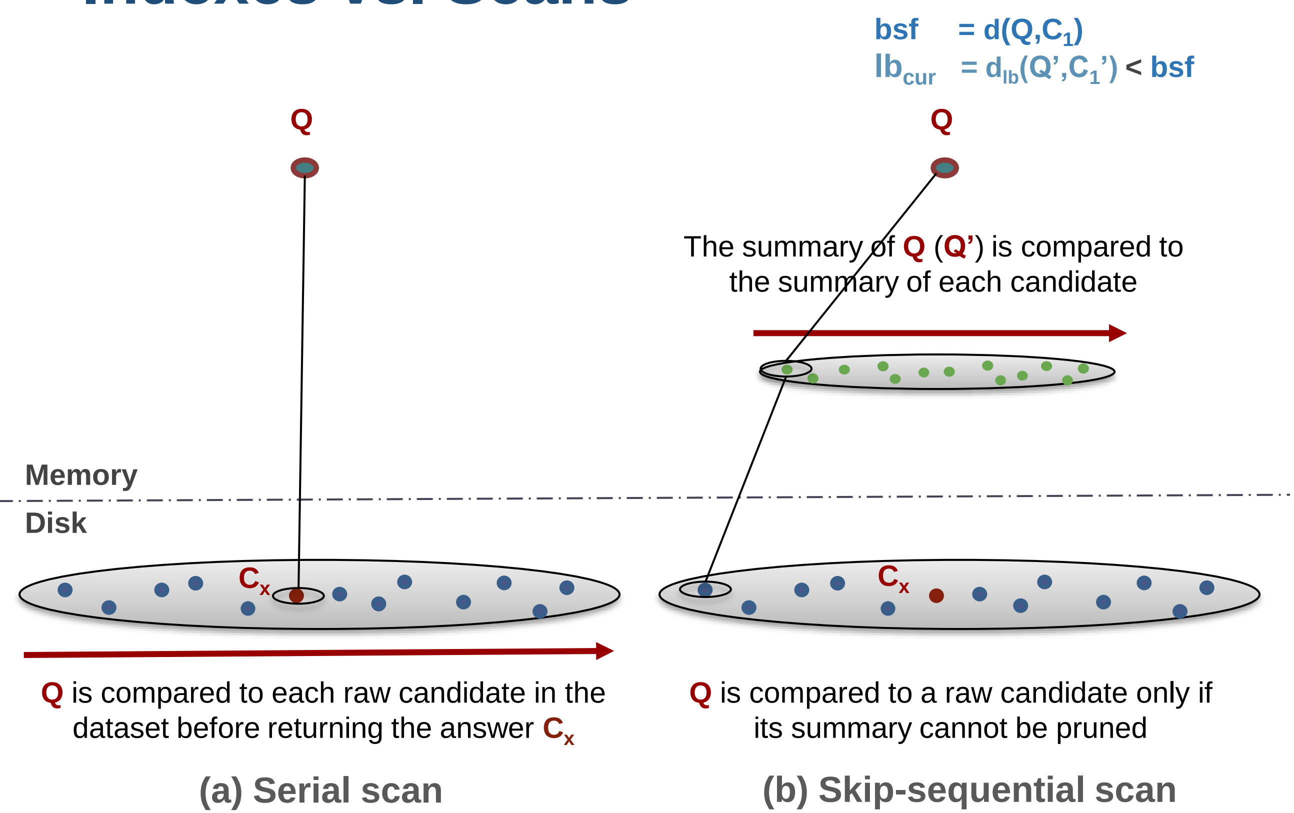
<u>**Answering a similarity search query using different access paths**</u>

Indexes vs. Scans



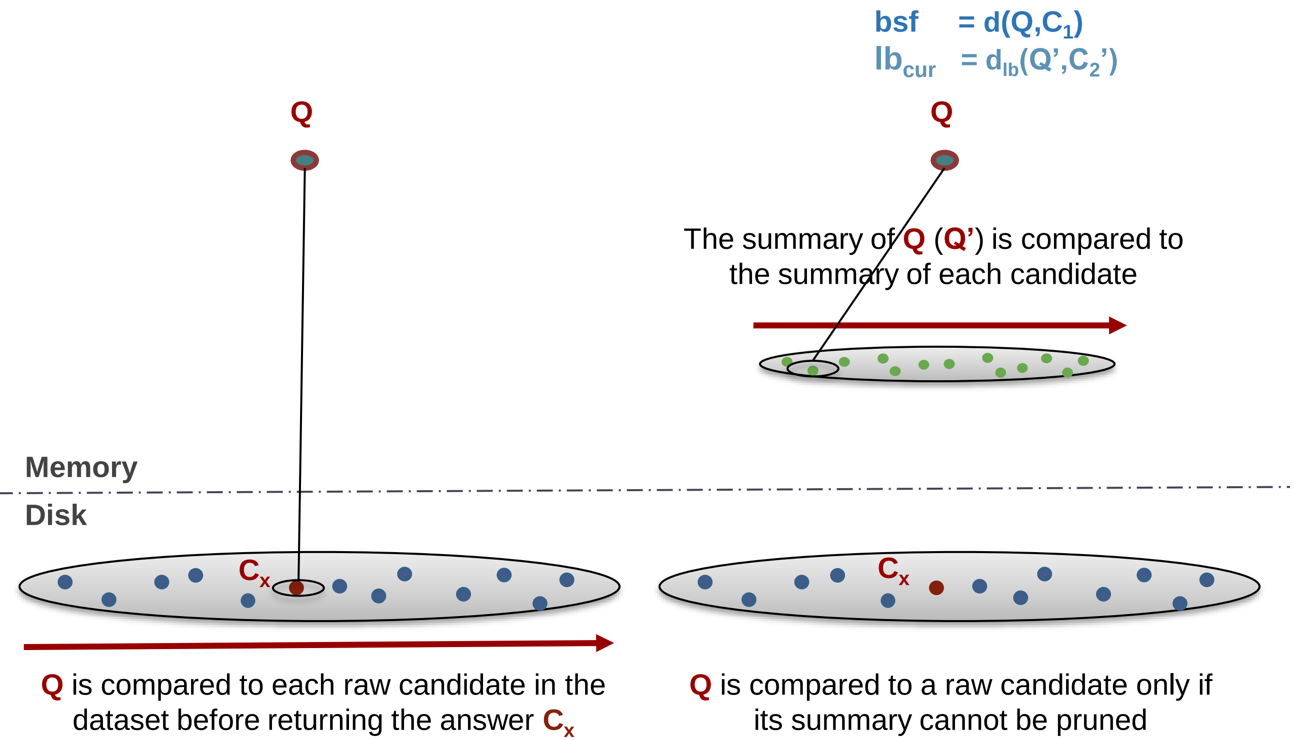
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans



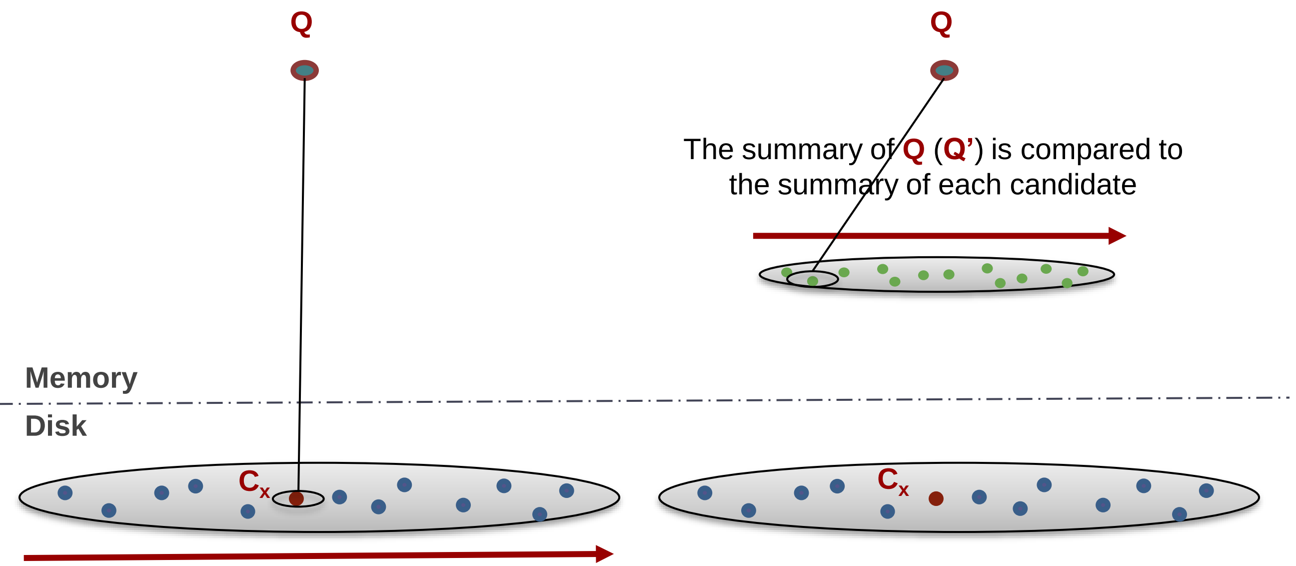
(a) Serial scan

(b) Skip-sequential scan

Answering a similarity search query using different access paths

Indexes vs. Scans

bsf $= d(Q,C_1)$
$lb_{cur} = d_{lb}(Q',C_2') >=$ bsf



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

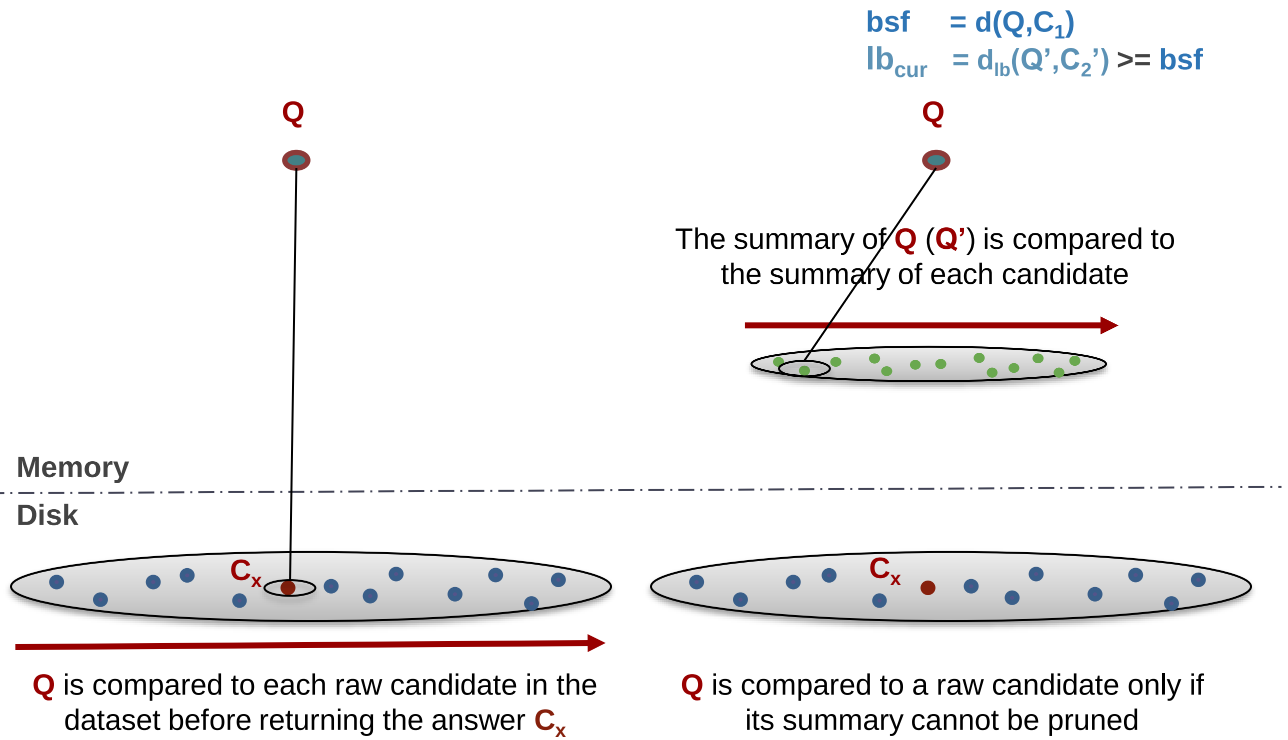
Q is compared to a raw candidate only if its summary cannot be pruned

(a) Serial scan

(b) Skip-sequential scan

Answering a similarity search query using different access paths

Indexes vs. Scans



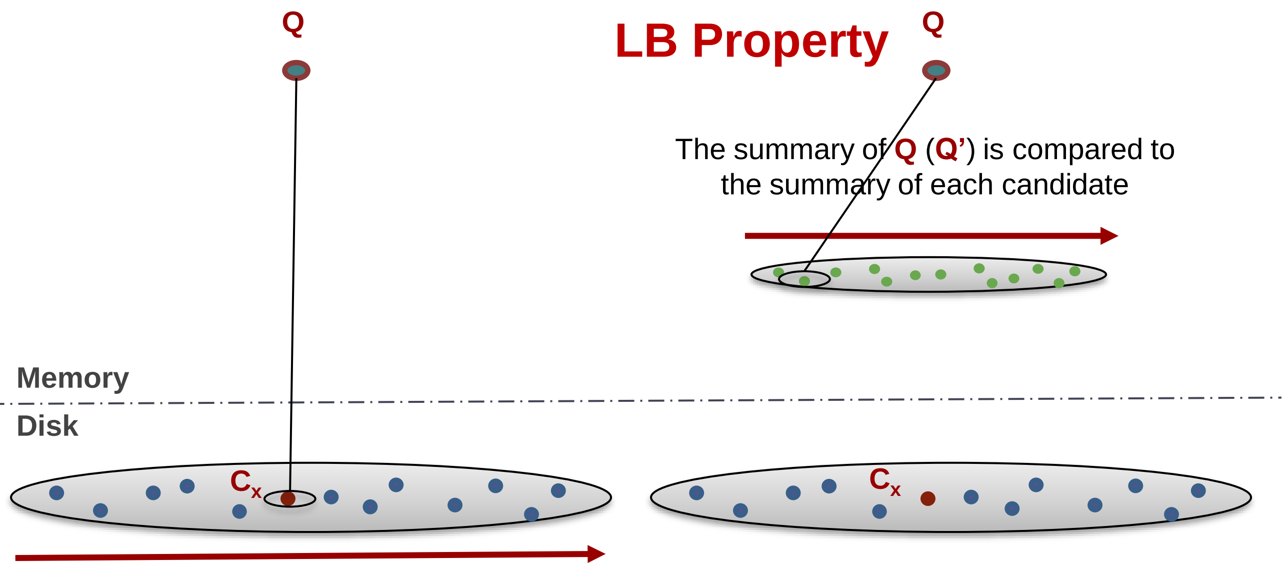
(a) Serial scan

(b) Skip-sequential scan

Answering a similarity search query using different access paths

Indexes vs. Scans

$$bsf = d(Q,C_1)$$
$$d(Q,C_2) >= lb_{cur} = d_{lb}(Q',C_2') >= bsf$$



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

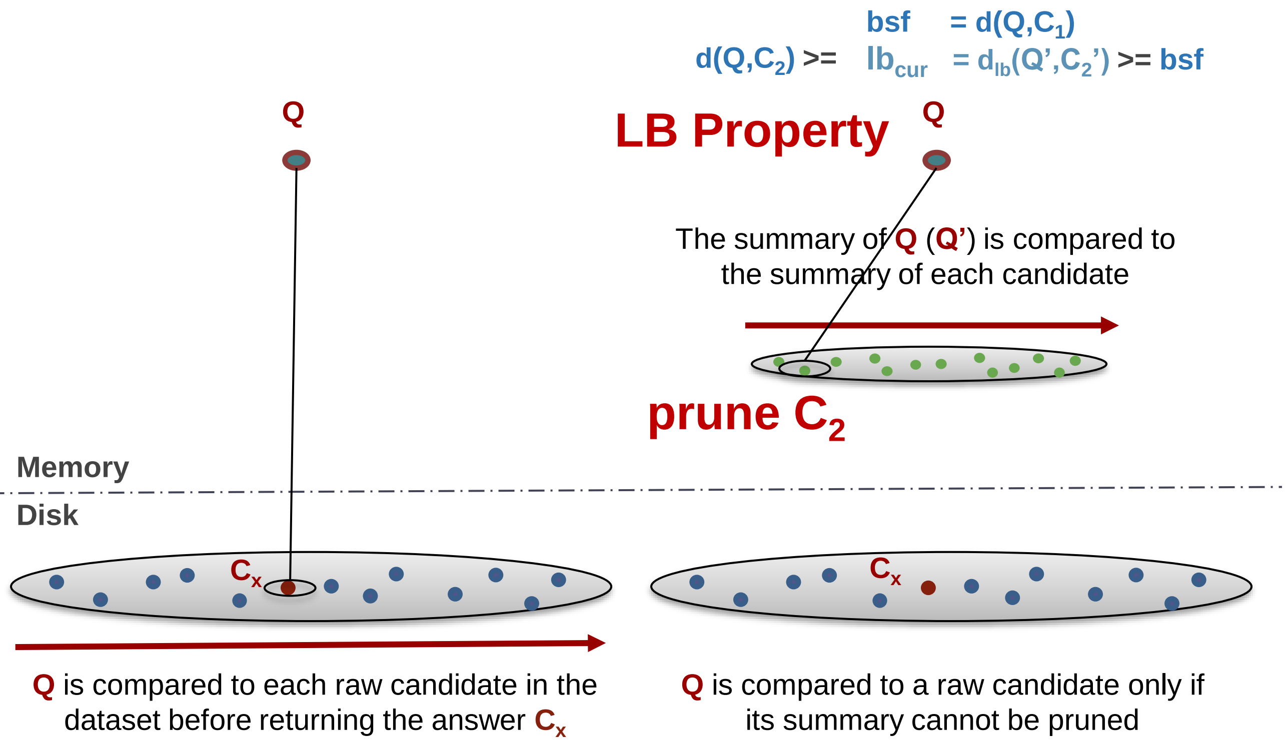
Q is compared to a raw candidate only if its summary cannot be pruned

(a) Serial scan

(b) Skip-sequential scan

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

Indexes vs. Scans

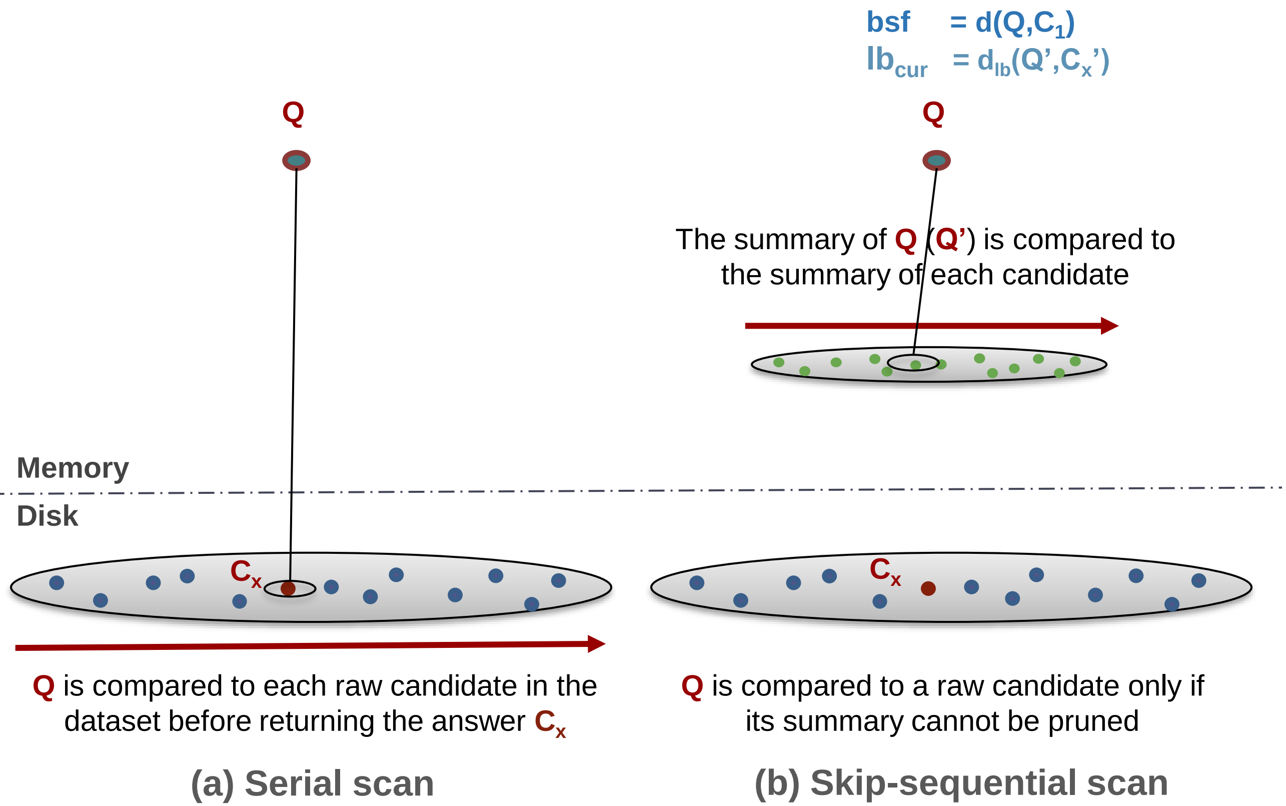


(a) Serial scan

(b) Skip-sequential scan

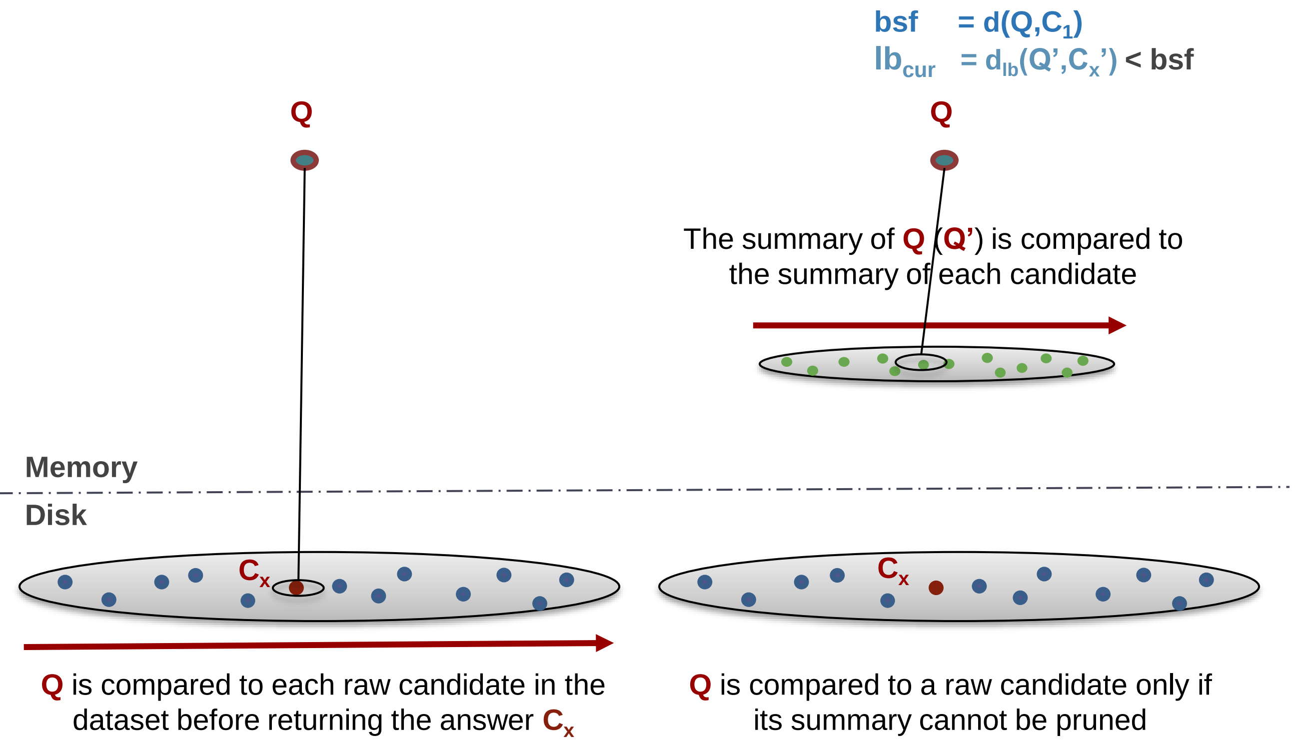
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans

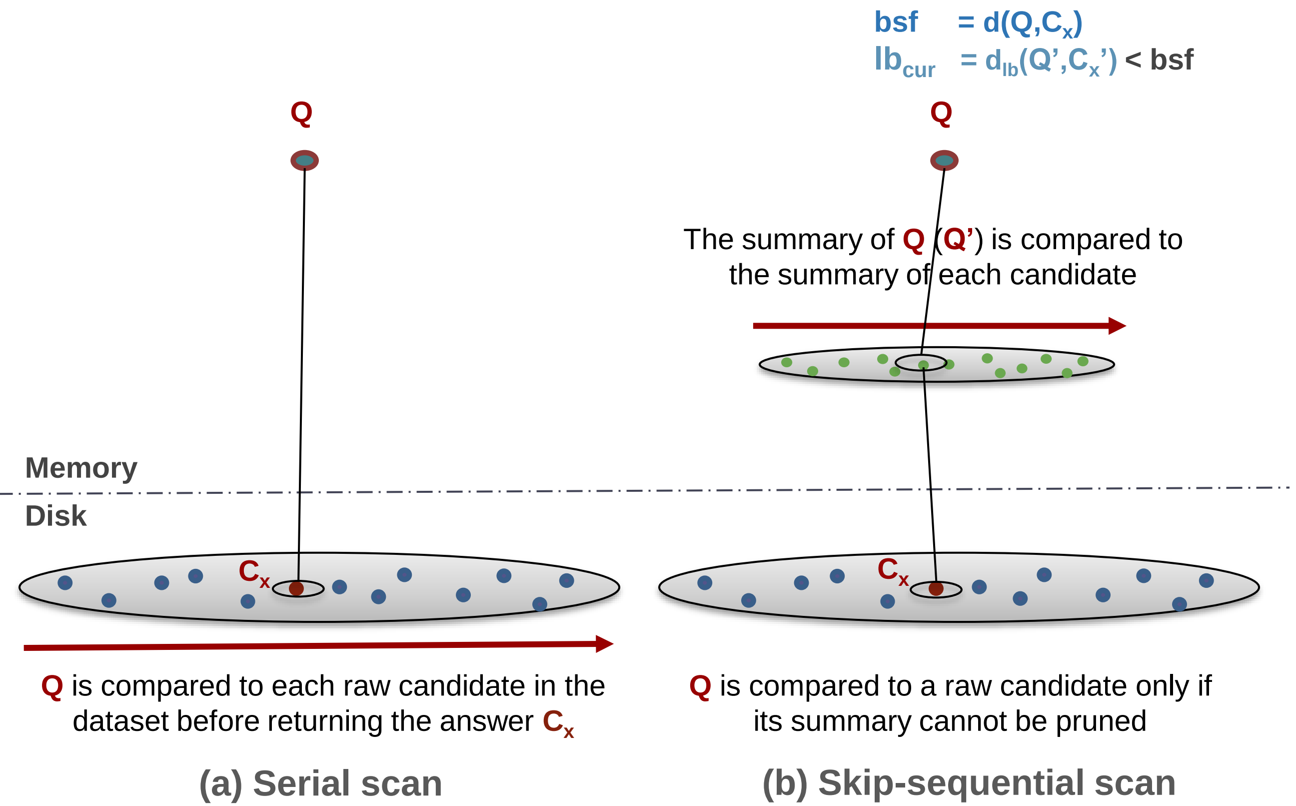


(a) Serial scan

(b) Skip-sequential scan

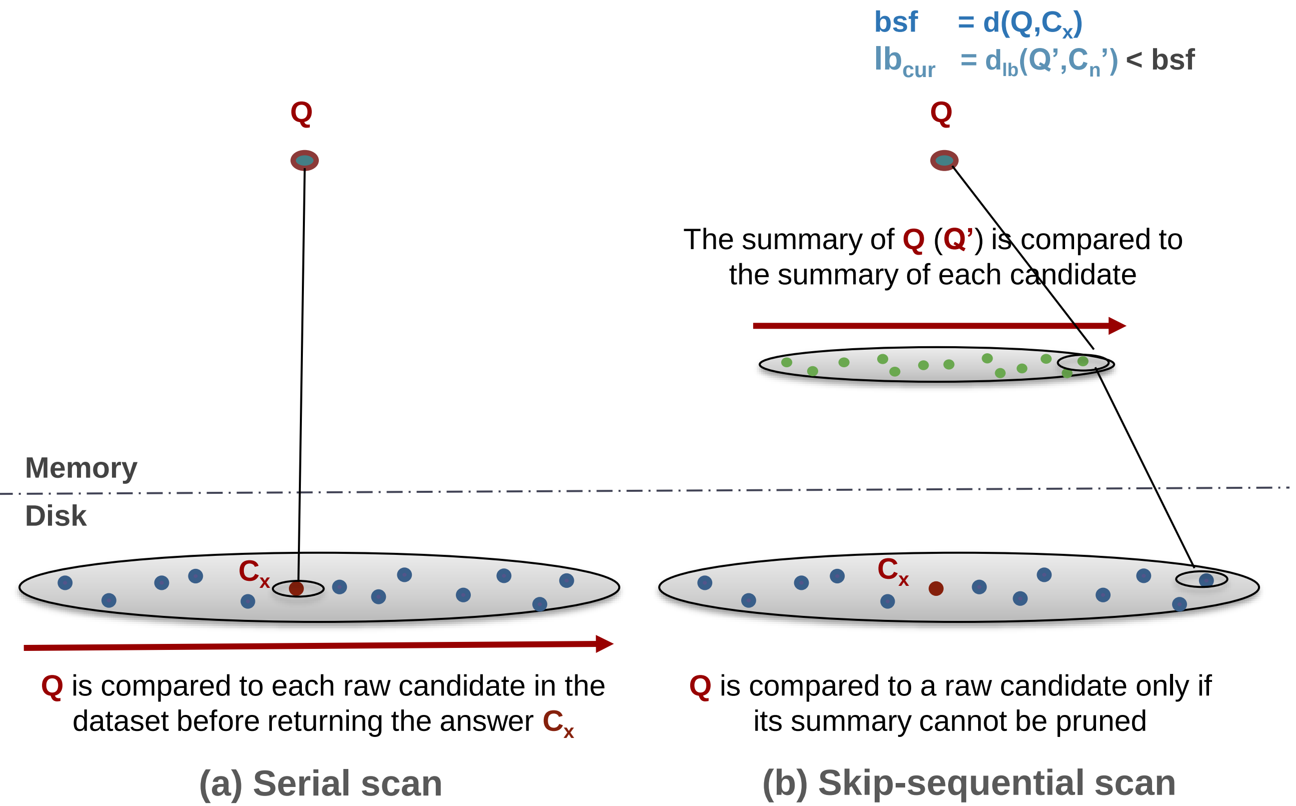
<u>**Answering a similarity search query using different access paths**</u>

Indexes vs. Scans



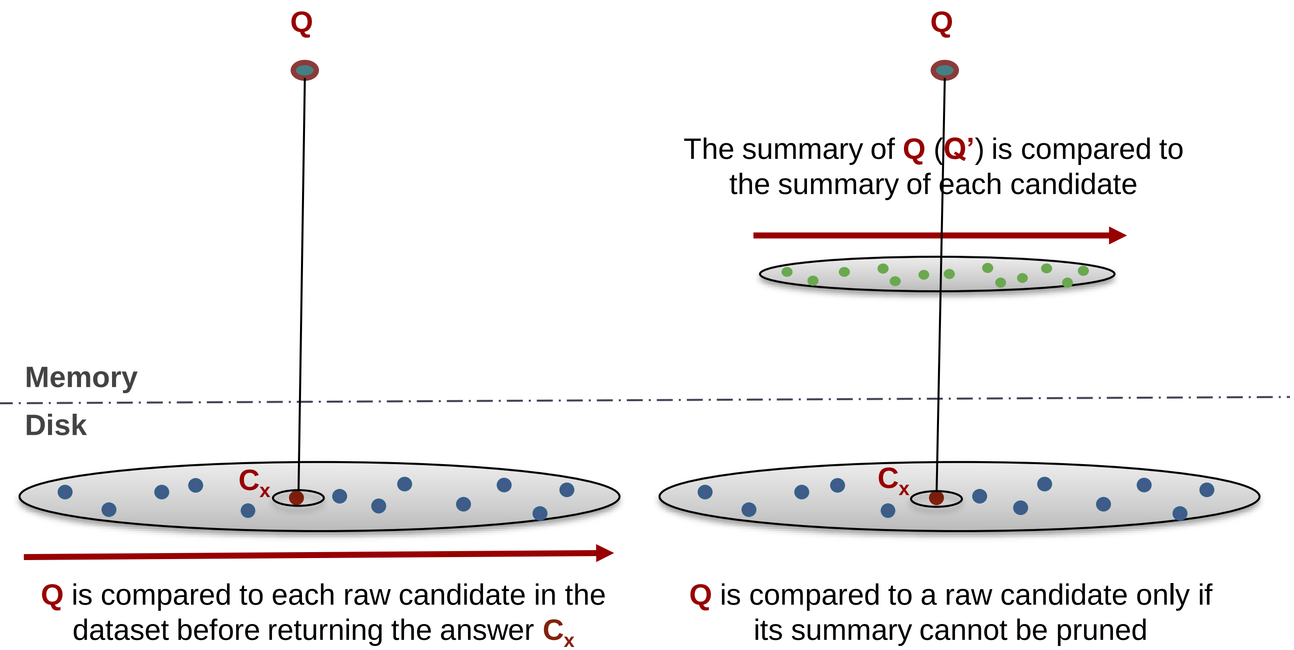
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

Indexes vs. Scans

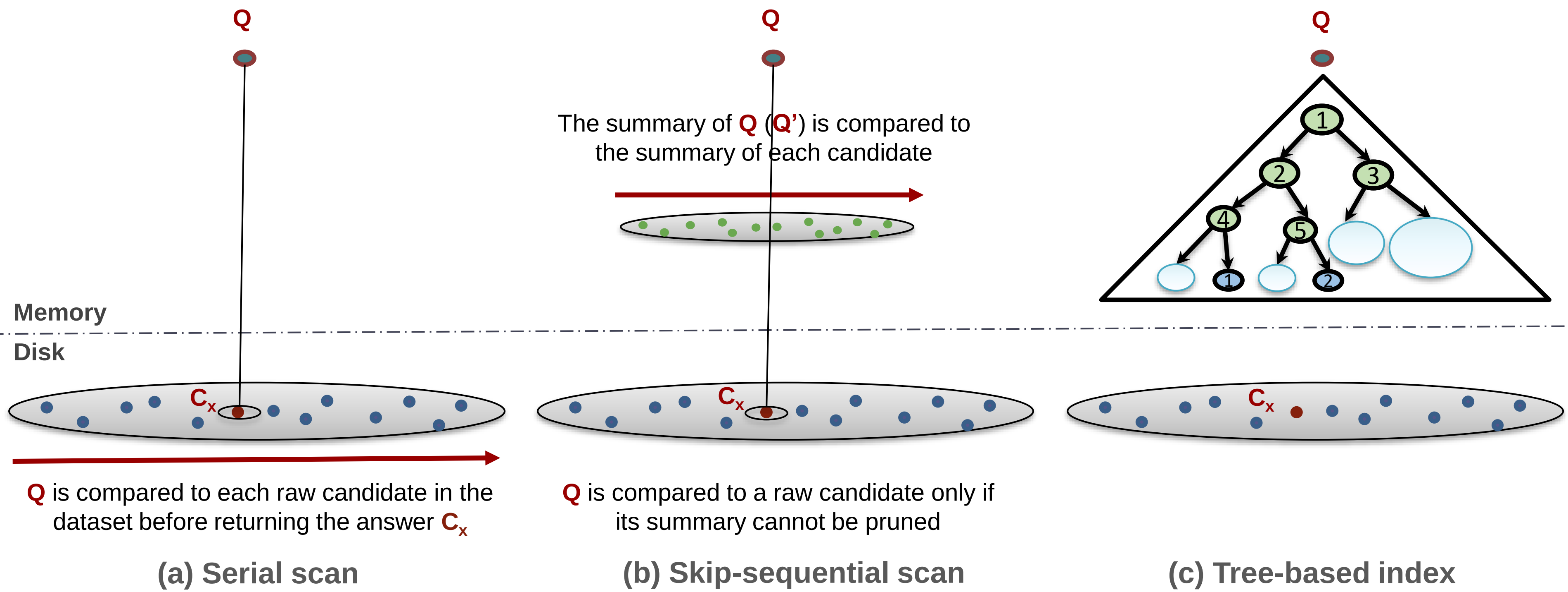


(a) Serial scan

(b) Skip-sequential scan

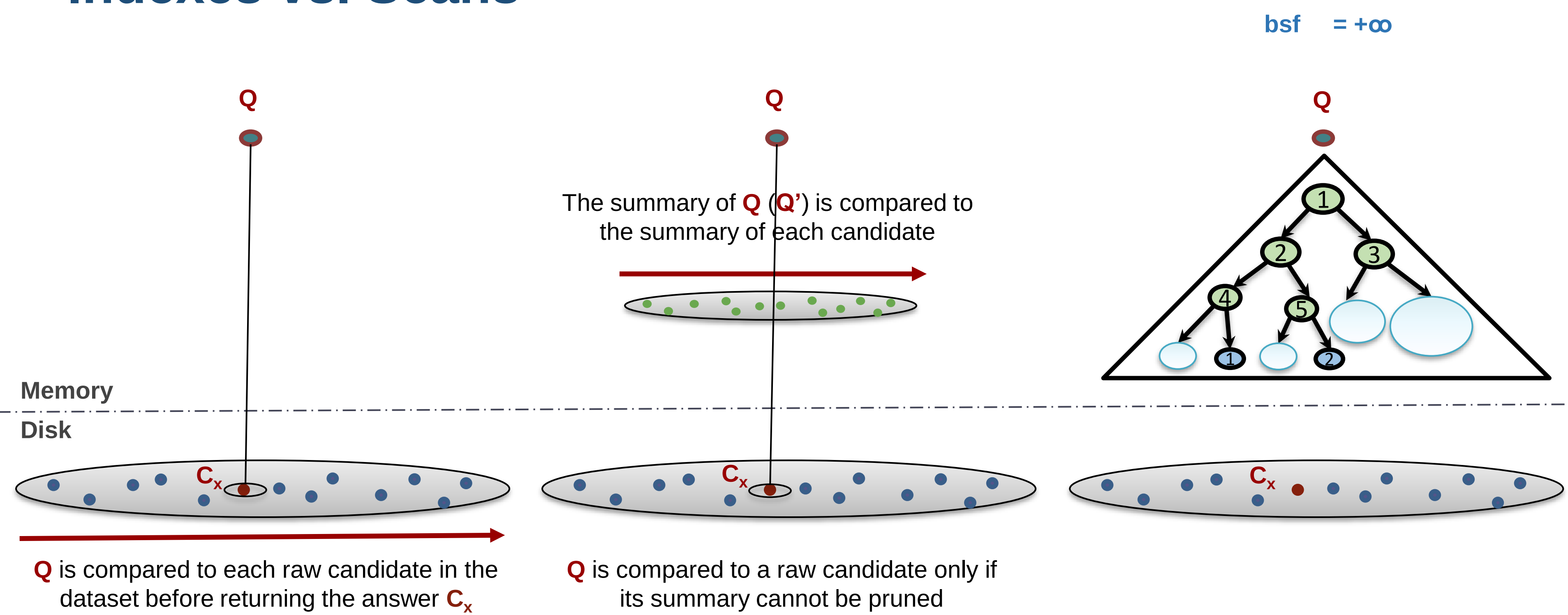
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

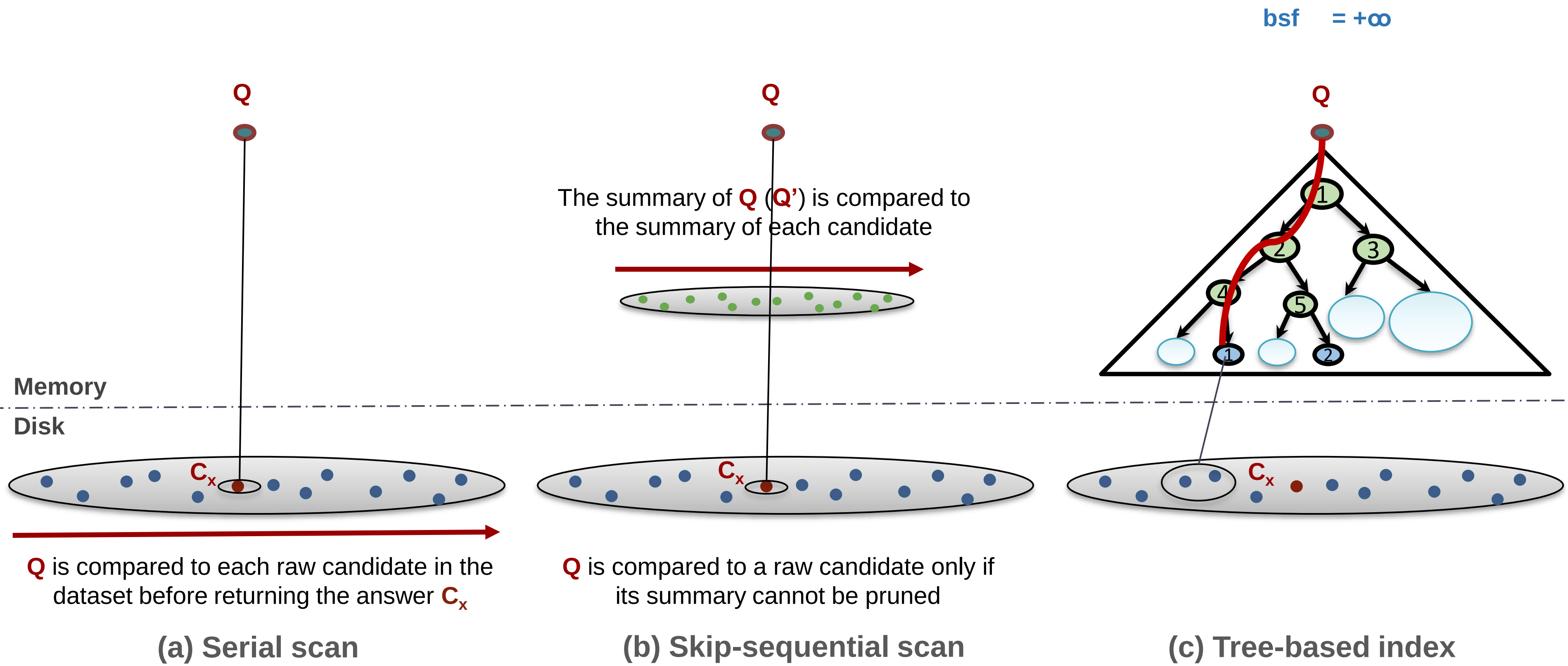
Indexes vs. Scans



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

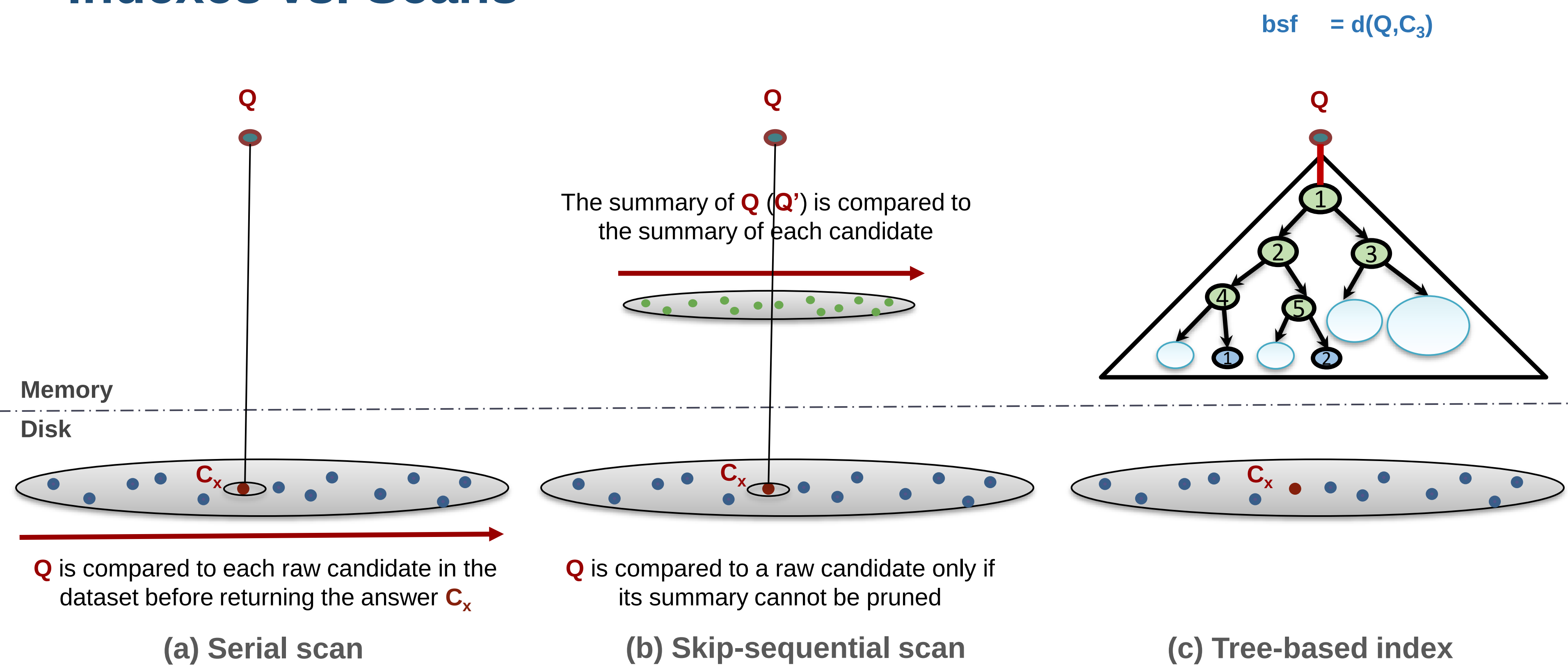
<u>Answering a similarity search query using different access paths</u>

Indexes vs. Scans



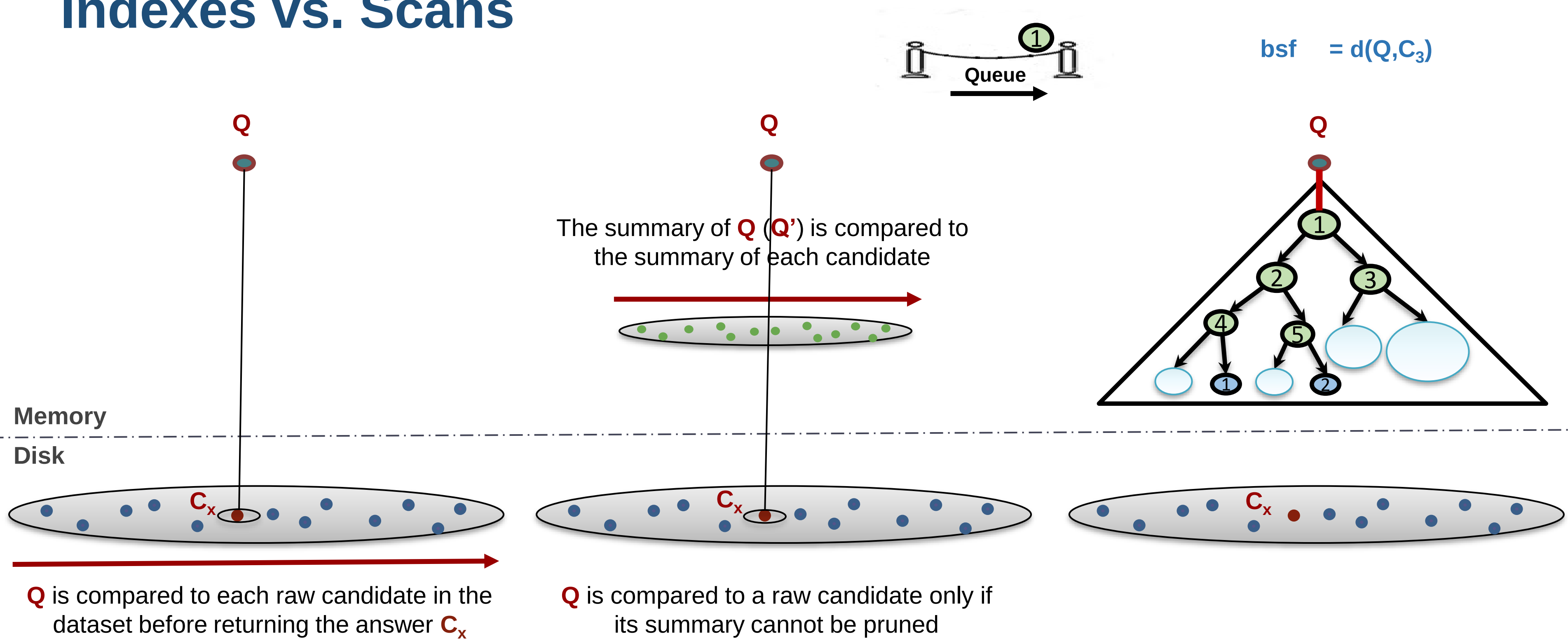
Answering a similarity search query using different access paths

Indexes vs. Scans



Answering a similarity search query using different access paths

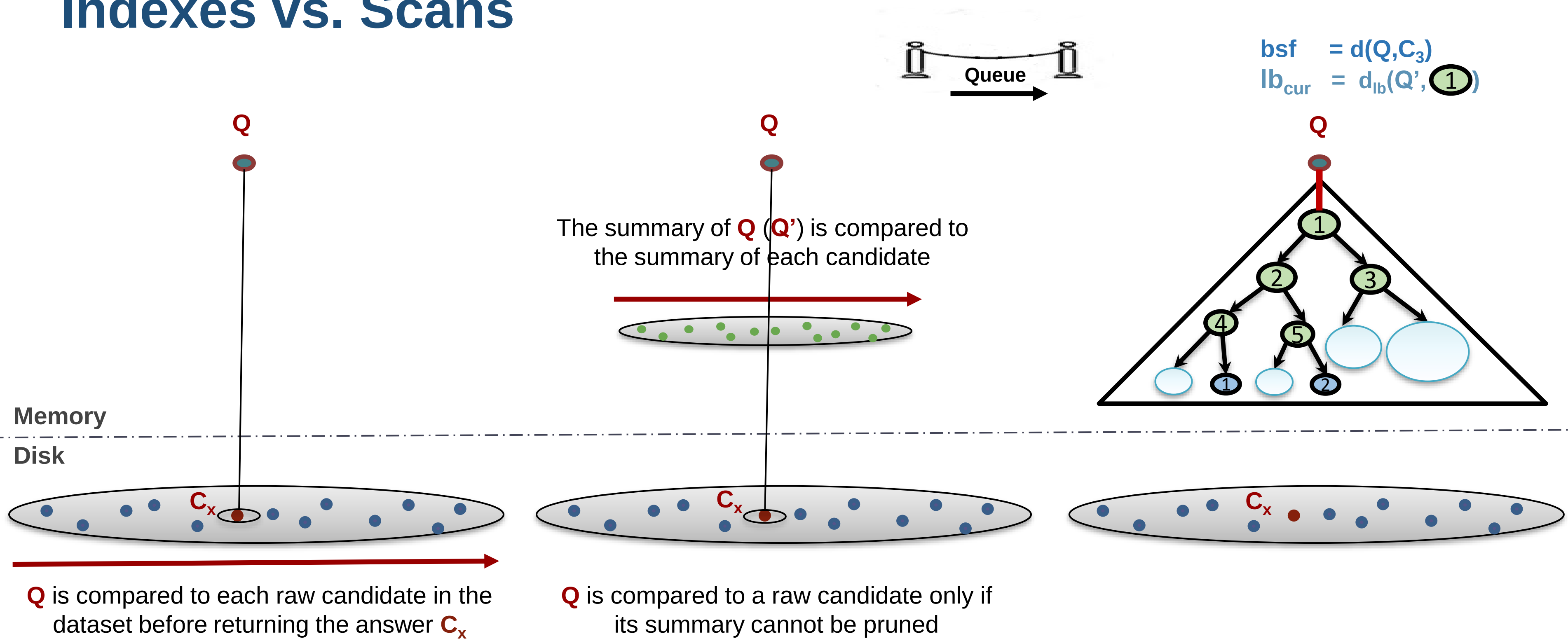
Indexes vs. Scans



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

Answering a similarity search query using different access paths

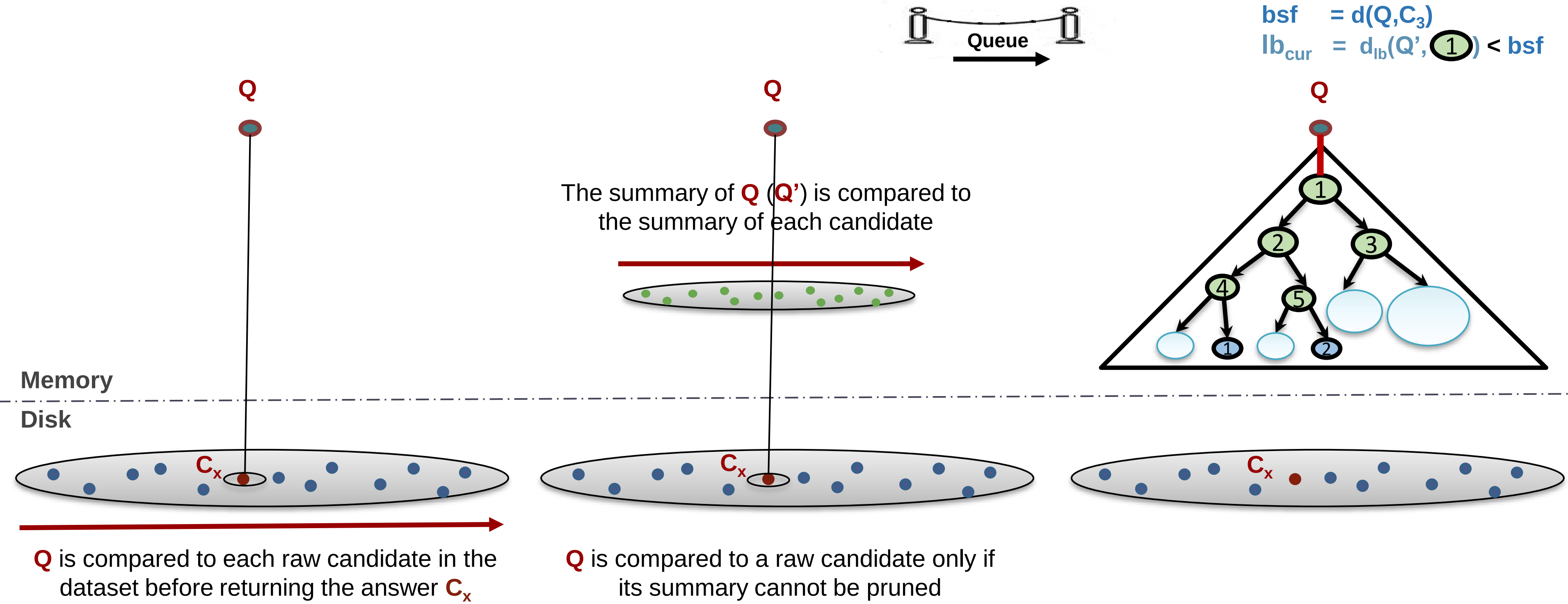
Indexes vs. Scans



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

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

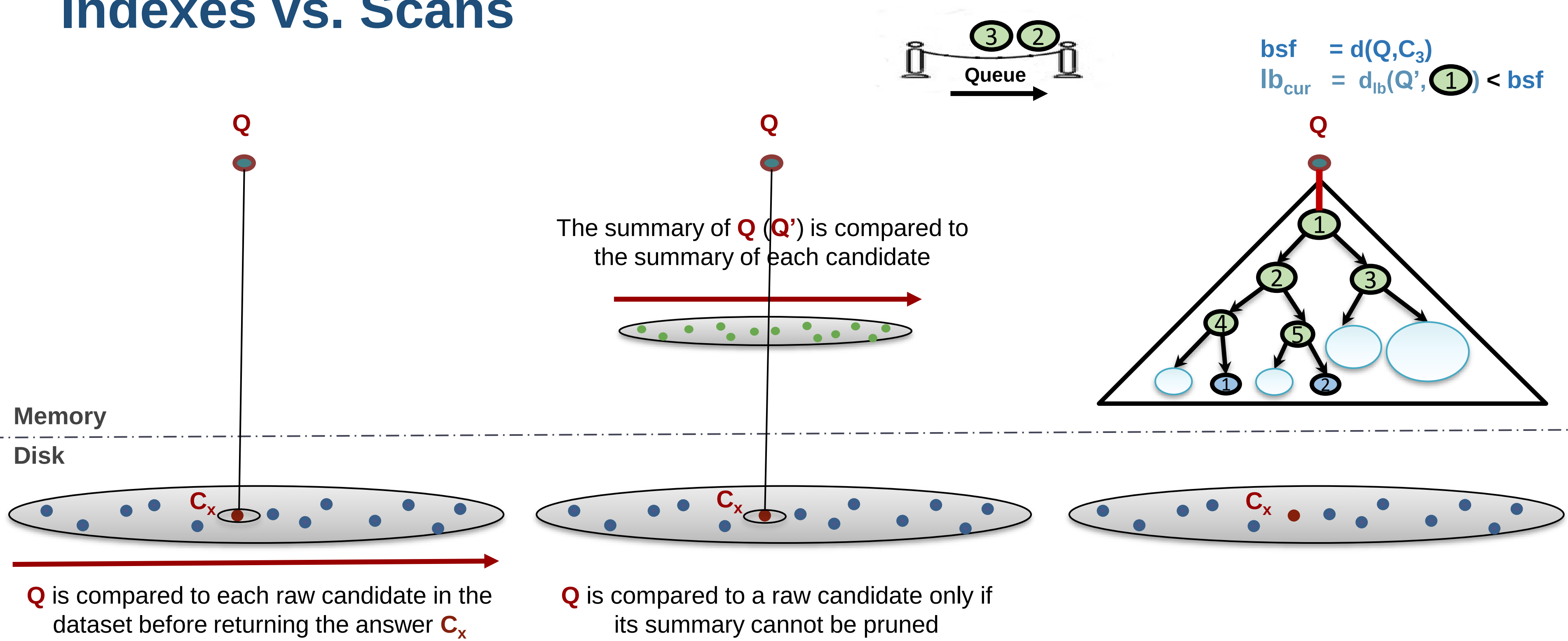
Indexes vs. Scans



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

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

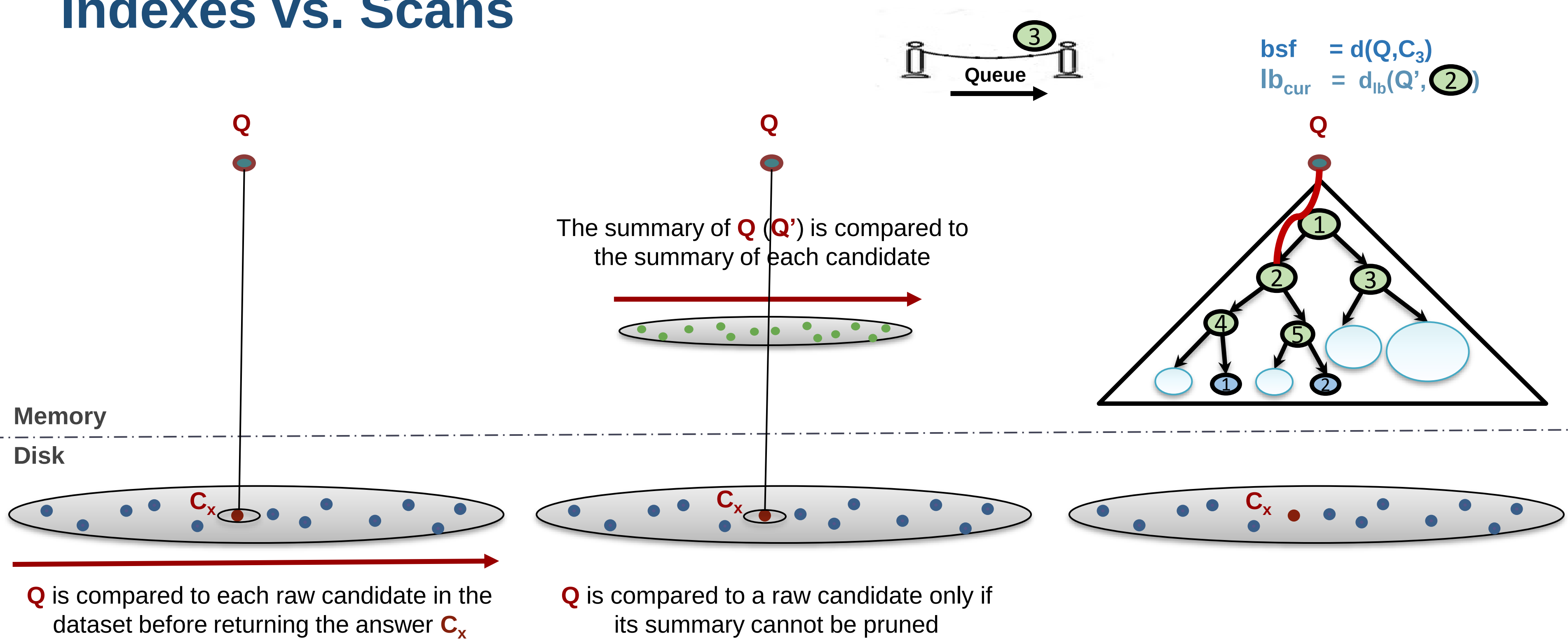
Indexes vs. Scans



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

Answering a similarity search query using different access paths

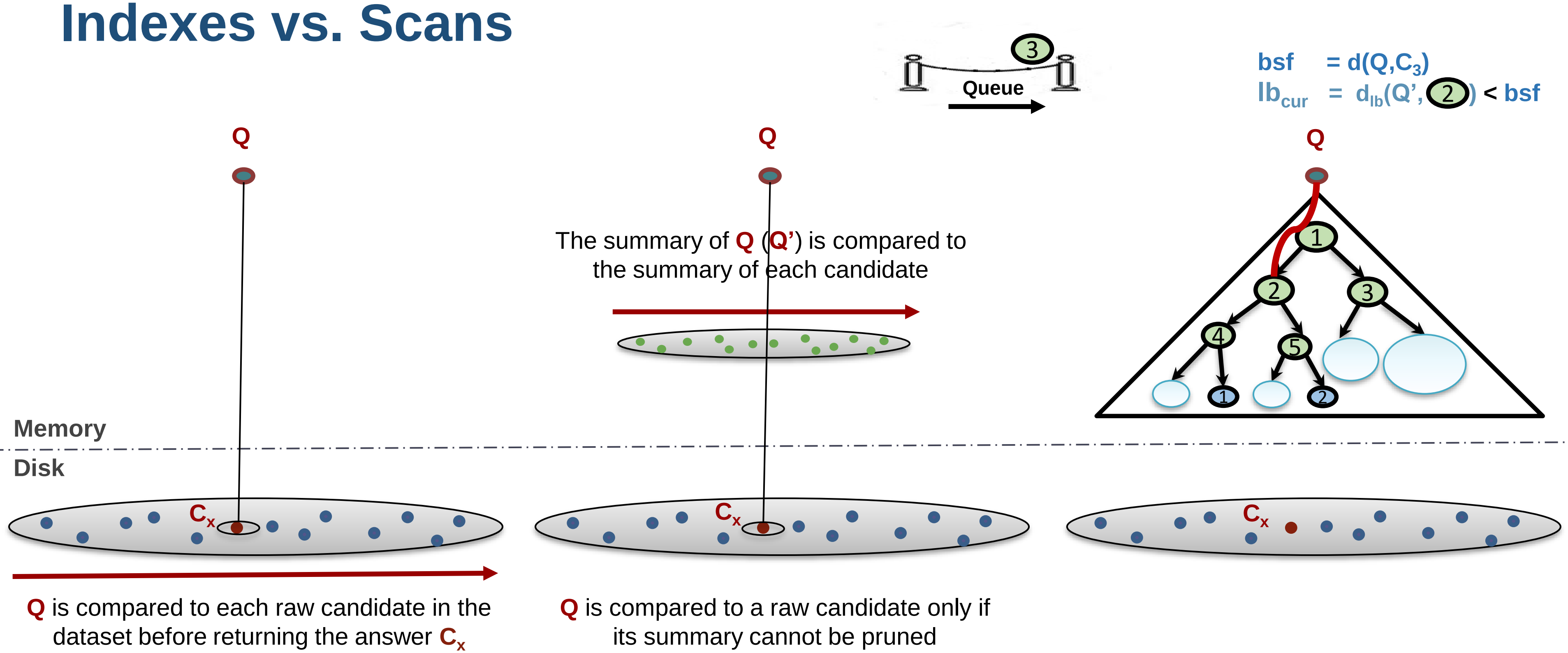
Indexes vs. Scans



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

Answering a similarity search query using different access paths

Indexes vs. Scans



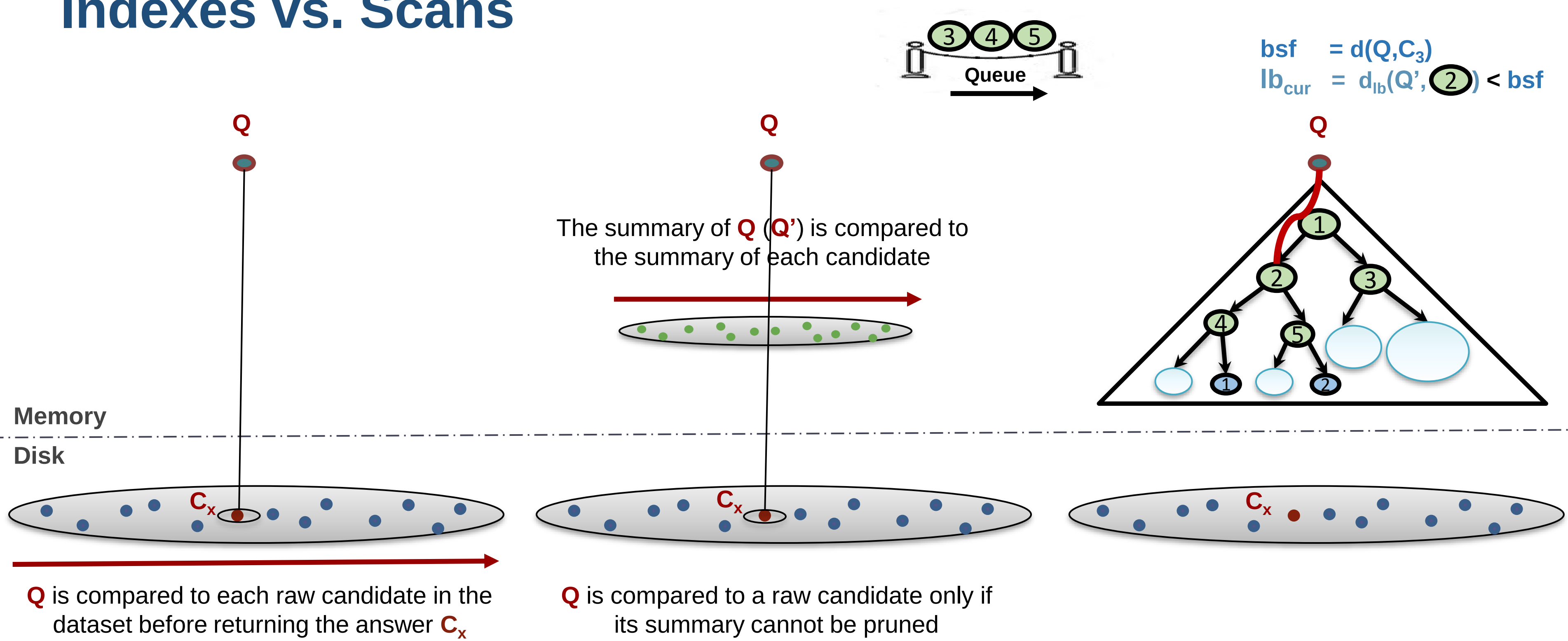
(a) Serial scan

(b) Skip-sequential scan

(c) Tree-based index

Answering a similarity search query using different access paths

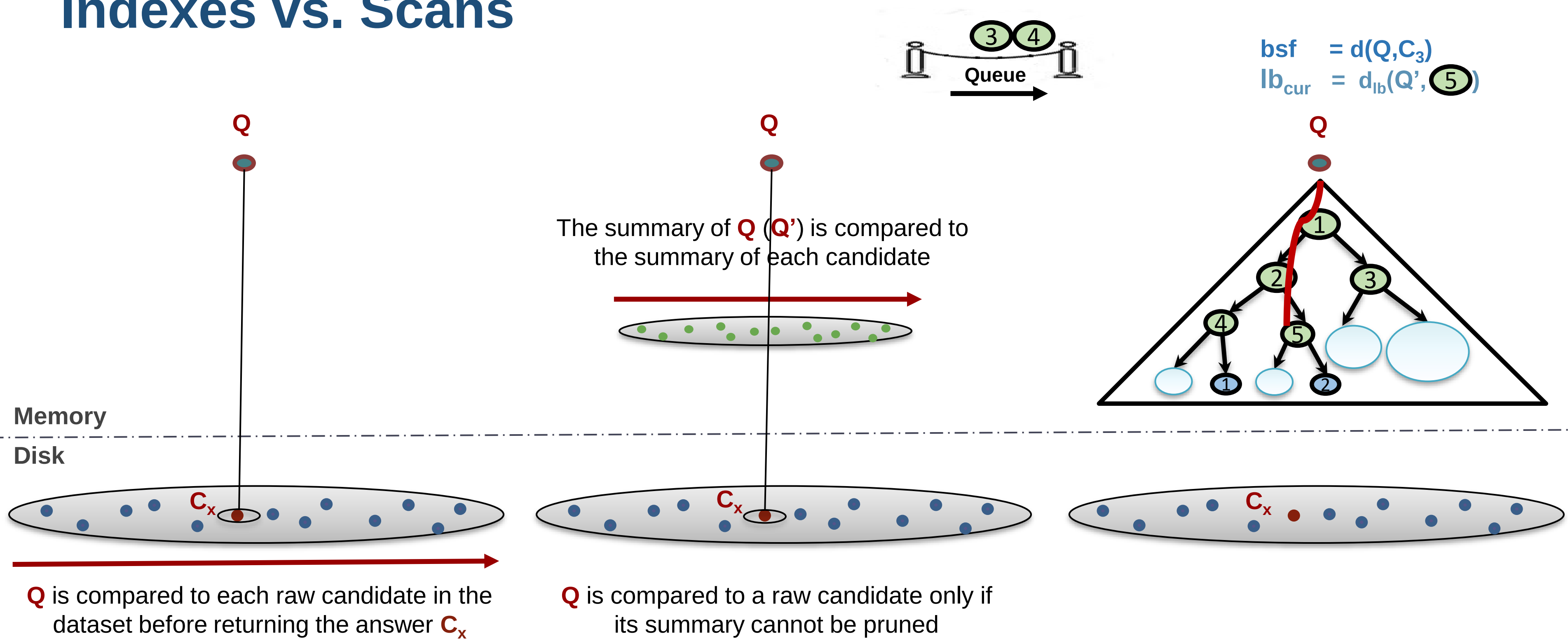
Indexes vs. Scans



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

Answering a similarity search query using different access paths

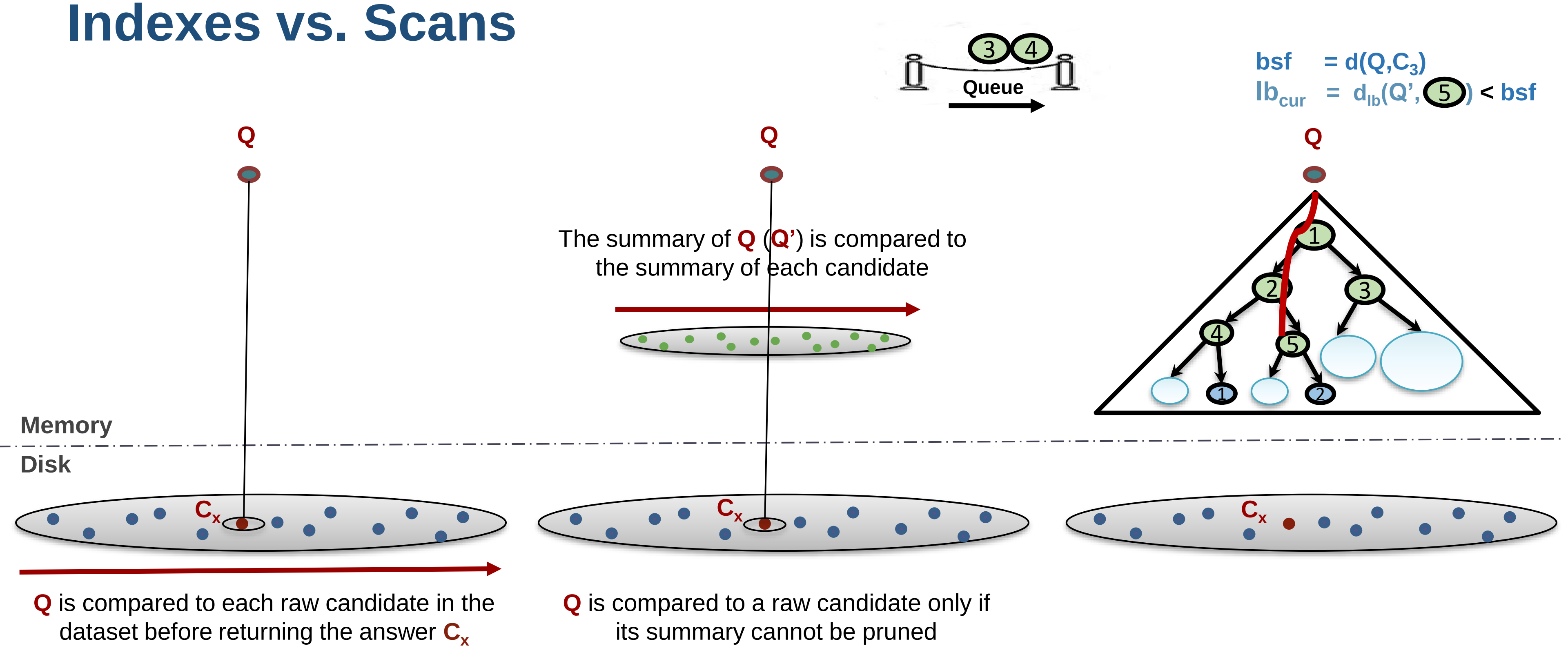
Indexes vs. Scans



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

Answering a similarity search query using different access paths

Indexes vs. Scans



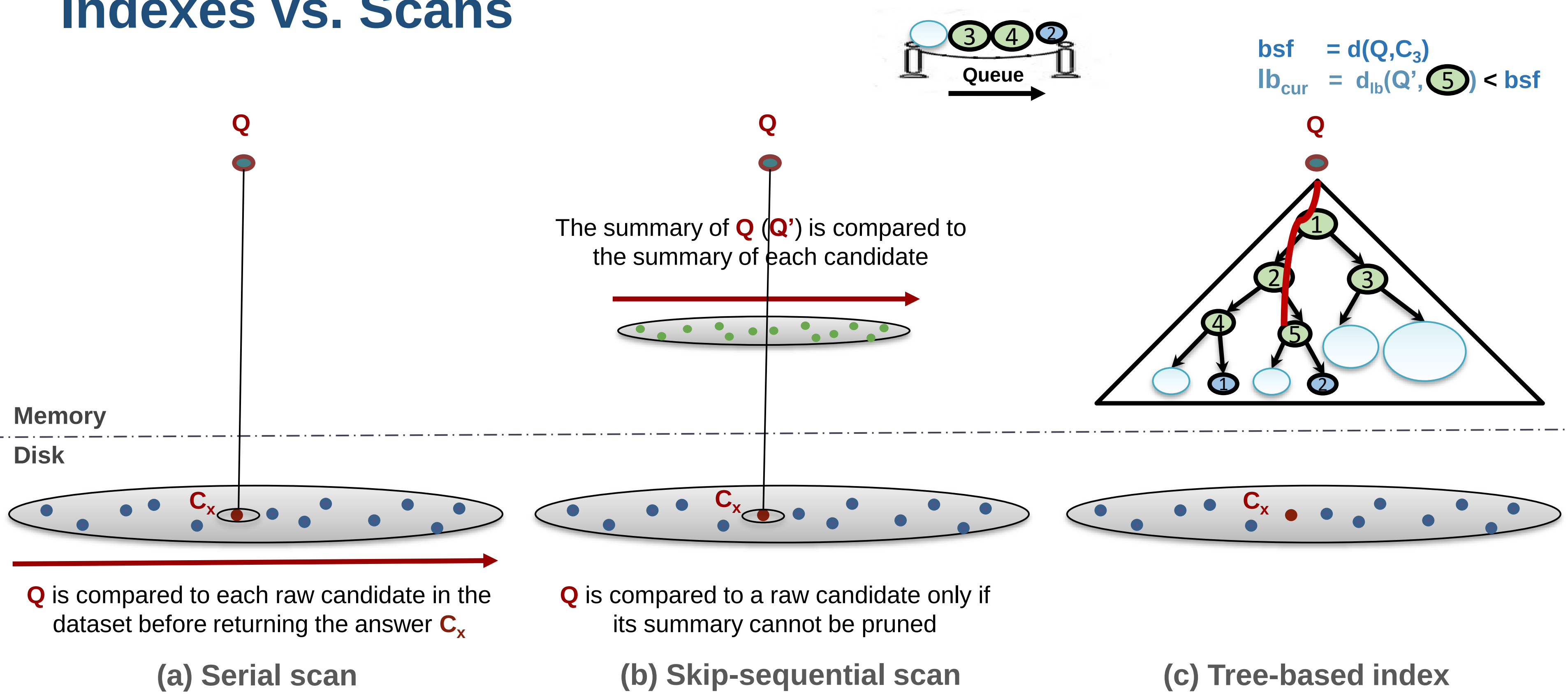
(a) Serial scan

(b) Skip-sequential scan

(c) Tree-based index

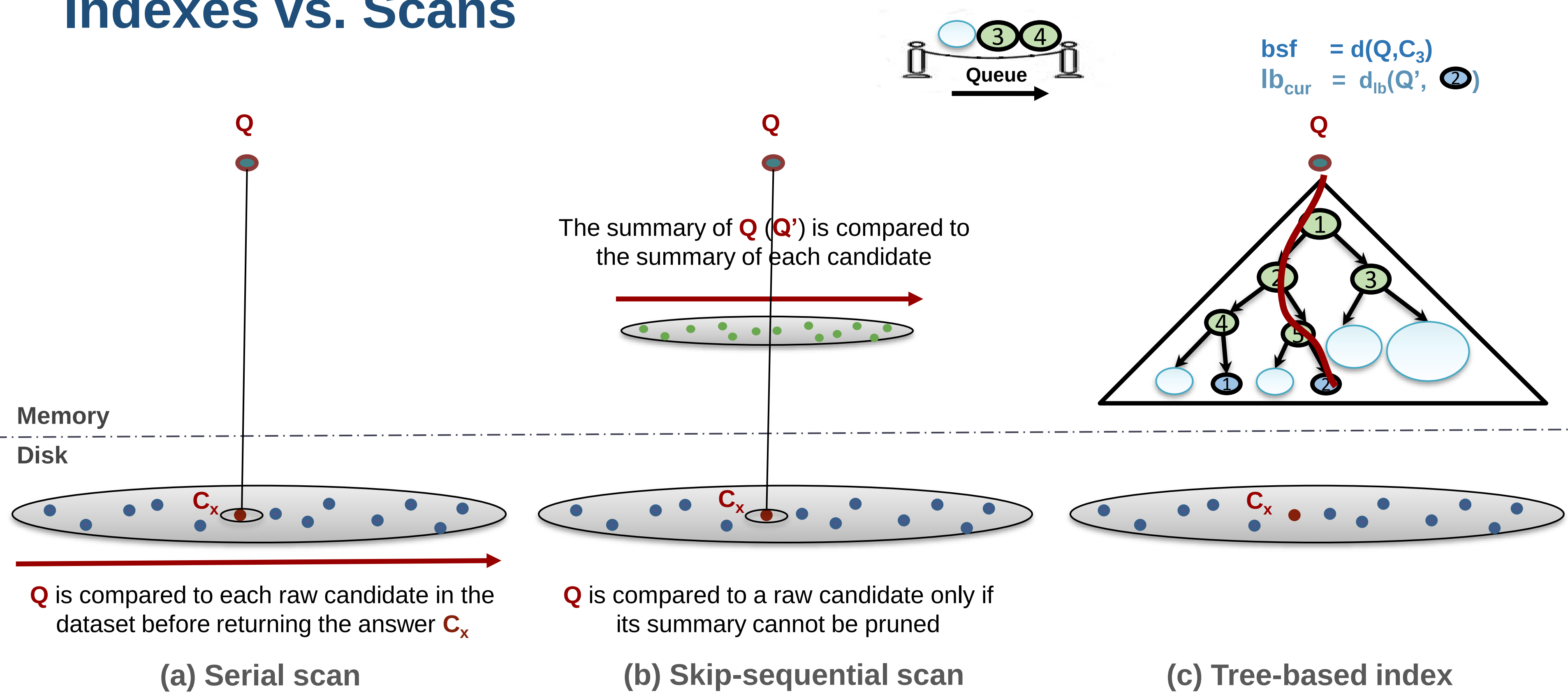
Answering a similarity search query using different access paths

Indexes vs. Scans



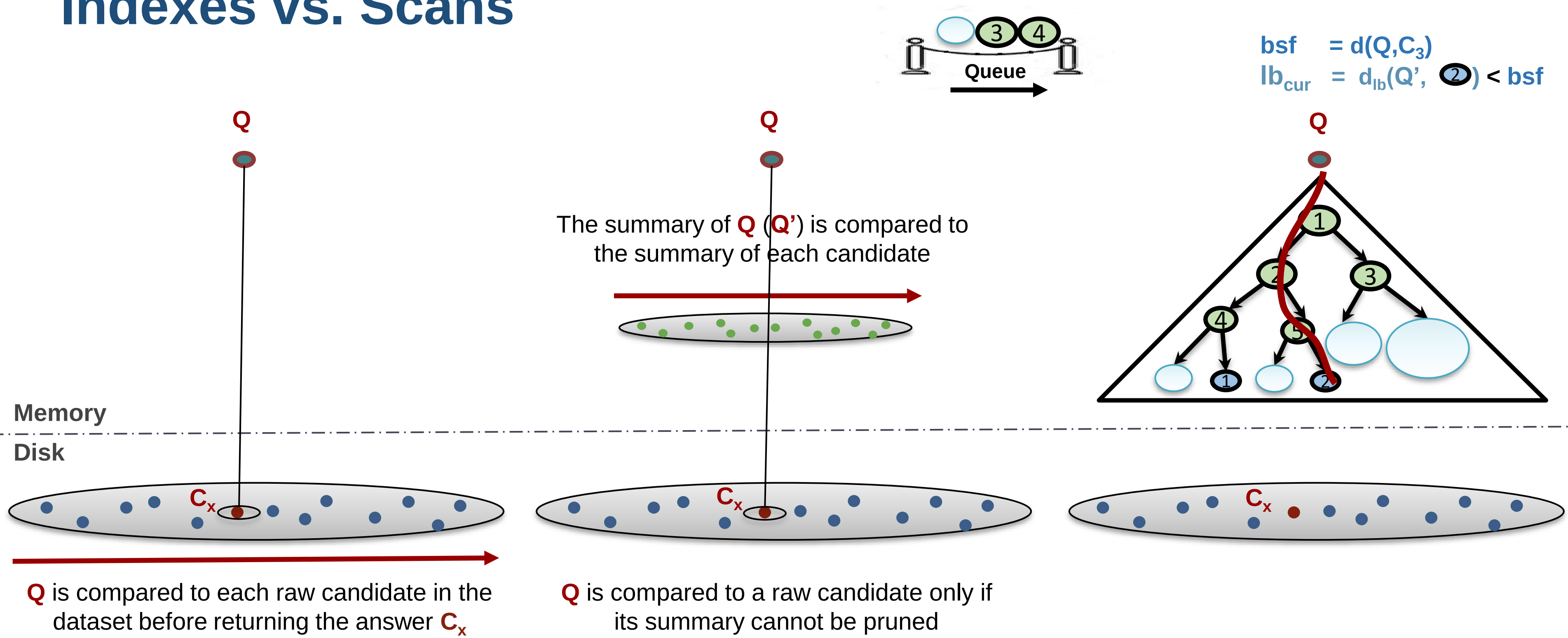
Answering a similarity search query using different access paths

Indexes vs. Scans



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

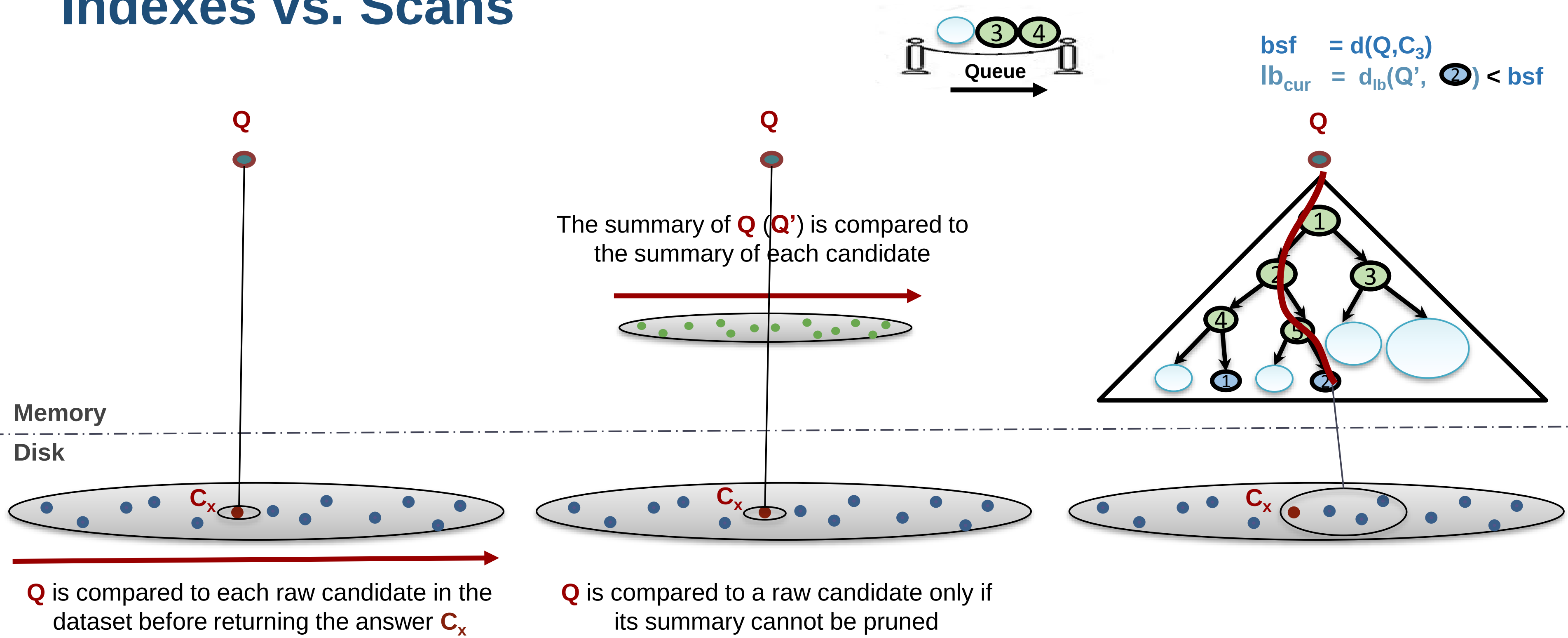
Indexes vs. Scans



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

Answering a similarity search query using different access paths

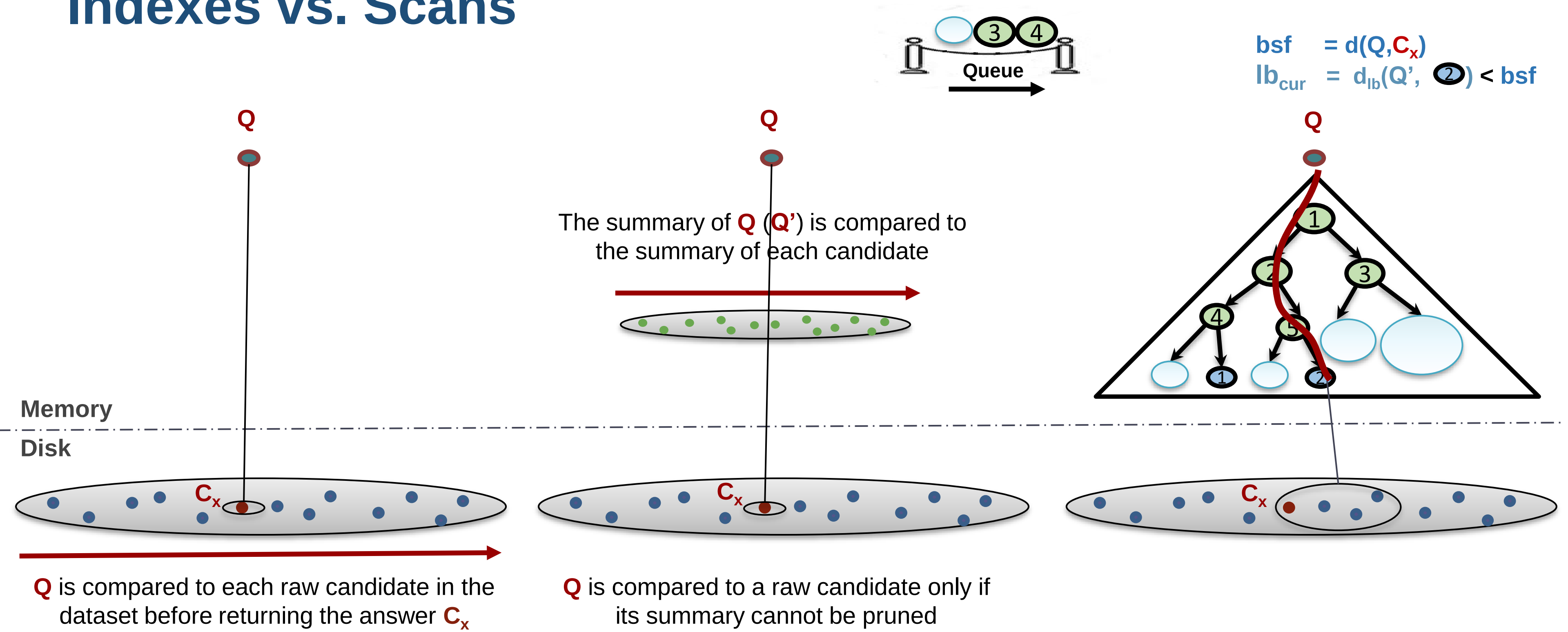
Indexes vs. Scans



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

Answering a similarity search query using different access paths

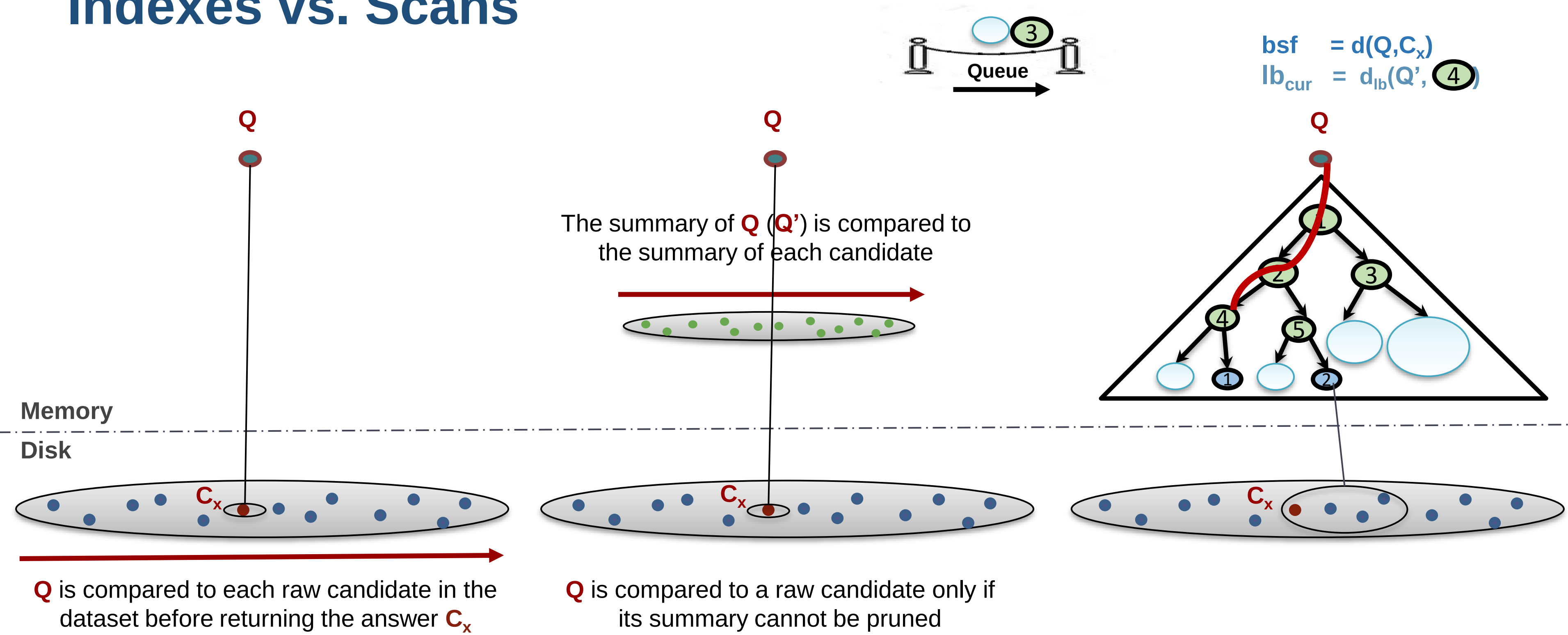
Indexes vs. Scans



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

Answering a similarity search query using different access paths

Indexes vs. Scans



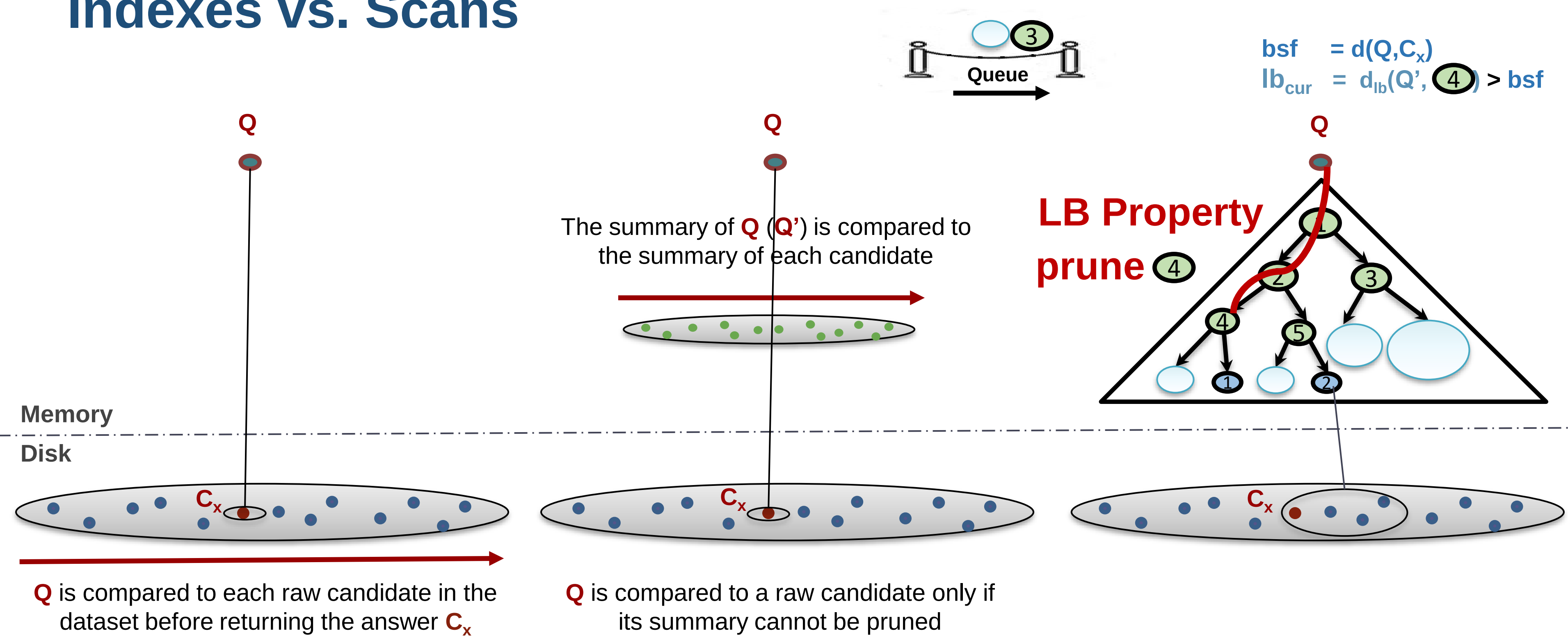
(a) Serial scan

(b) Skip-sequential scan

(c) Tree-based index

Answering a similarity search query using different access paths

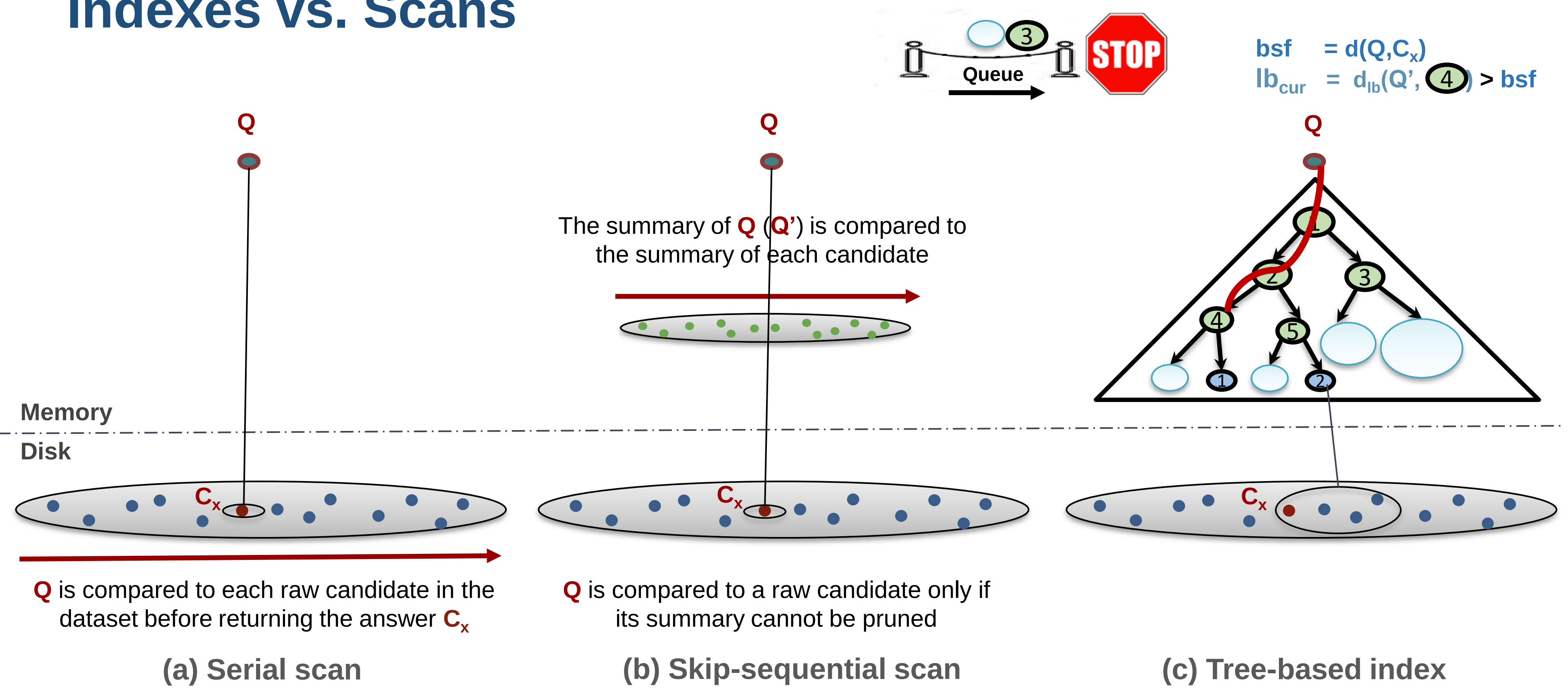
Indexes vs. Scans



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

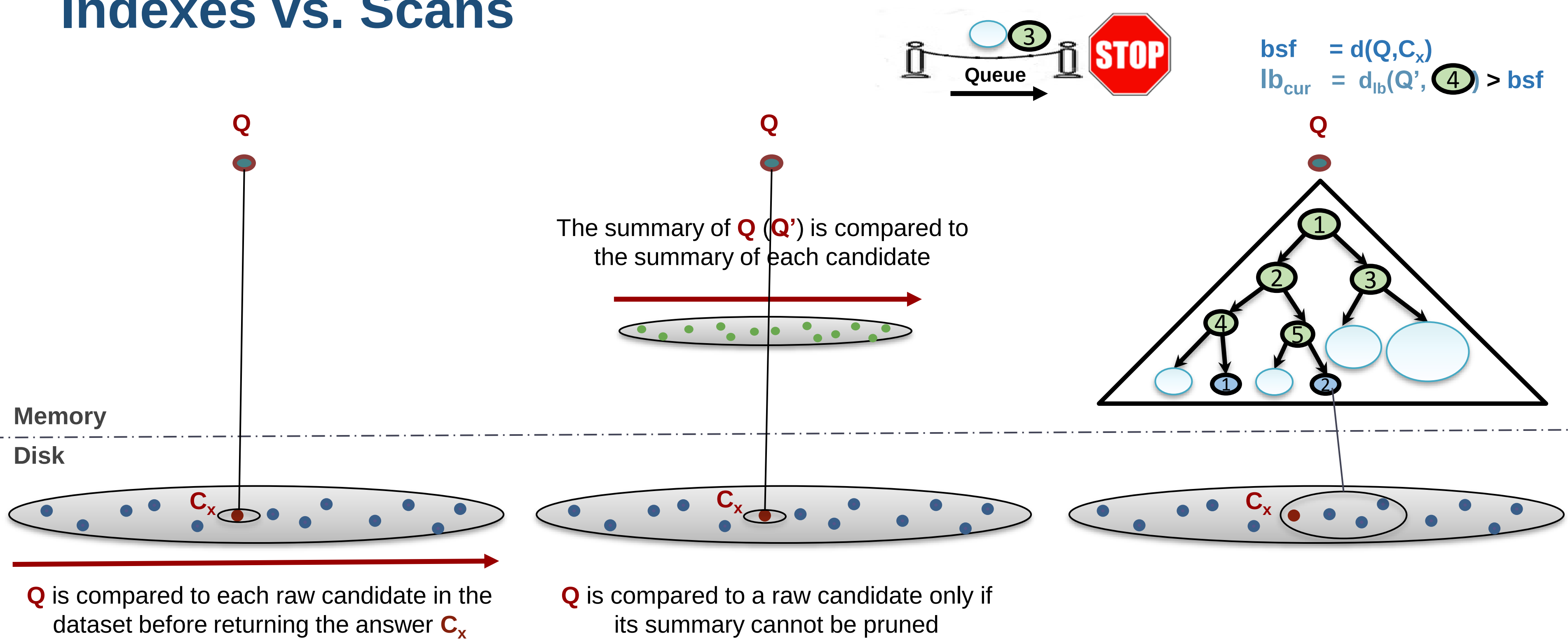
Answering a similarity search query using different access paths

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

Publications

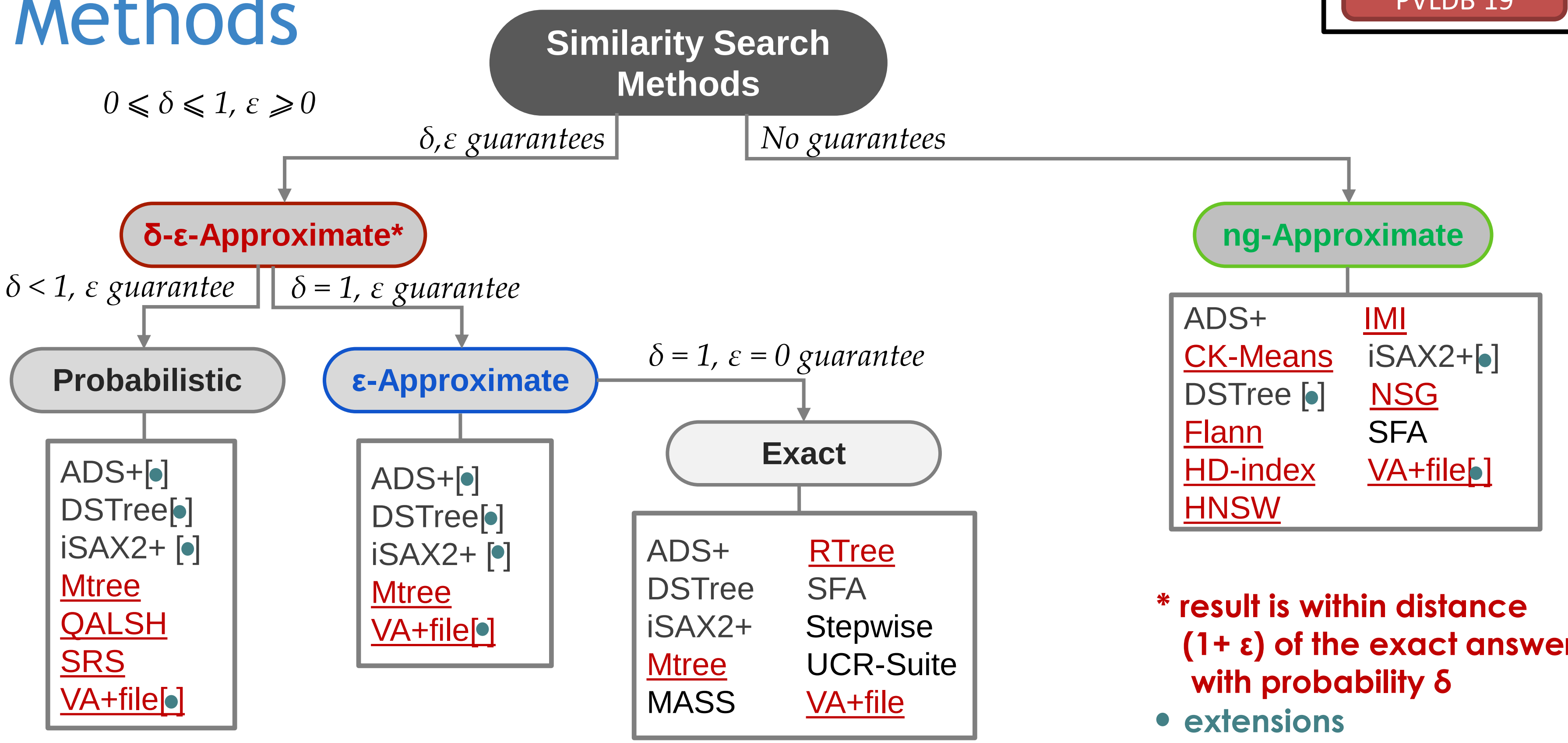
Echihabi-PVLDB‘18

Echihabi-PVLDB‘19

Techniques for data Series

Techniques for High-D vectors

Methods



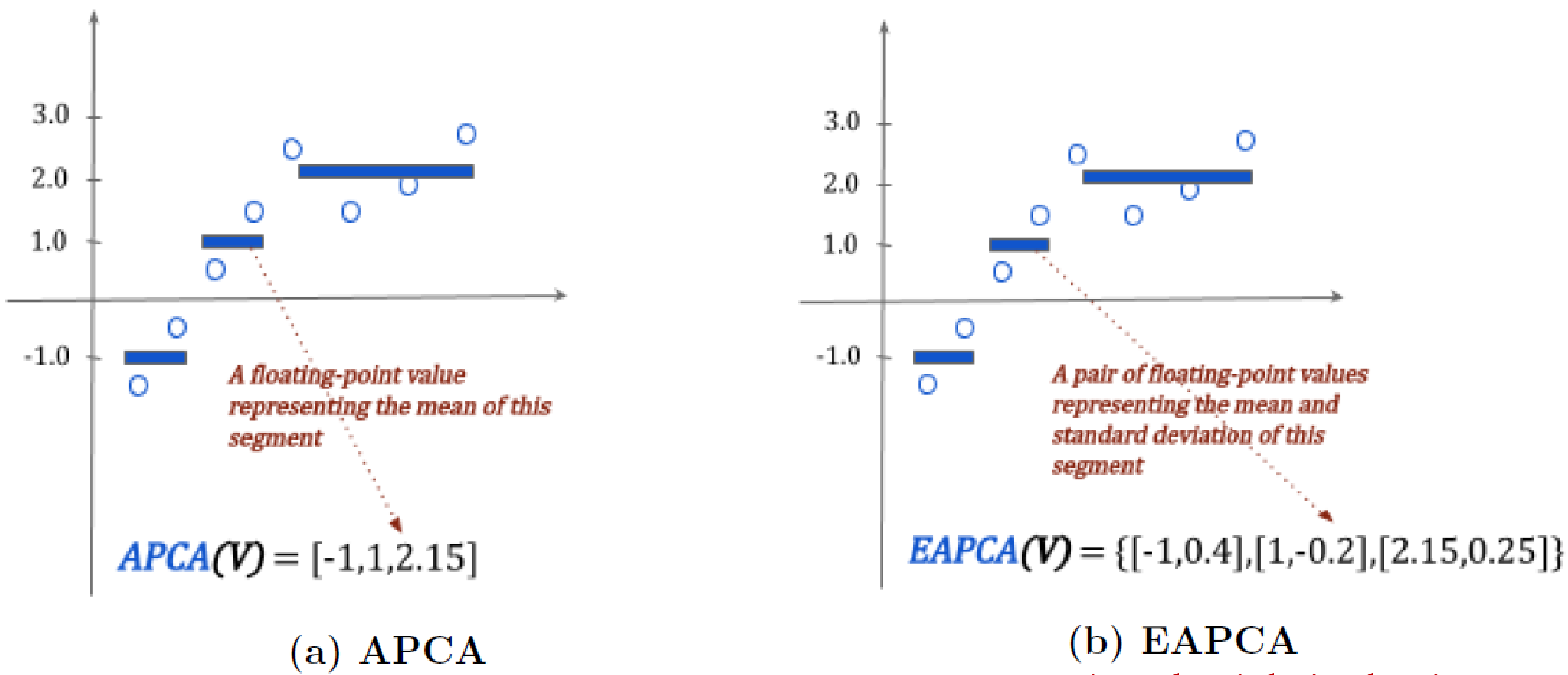
Data Series Indexing

Publications

Wang-PVLDB'13

DSTree
Summarization

V = [-1.5,-0.5, 0.5, 1.5, 2.5, 1.5,2, 2.6]

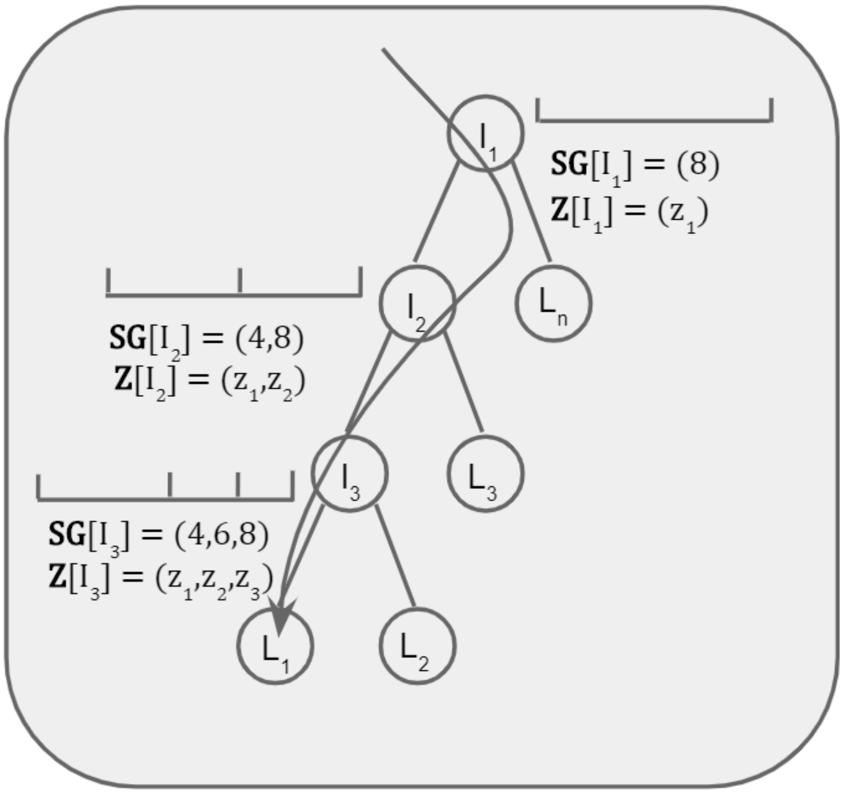


Intertwined with indexing

<u>The APCA and EAPCA representations</u>

DSTree
Indexing

V = [-1.5,-0.5, 0.5, 1.5, 2.5, 1.5,2, 2.6]



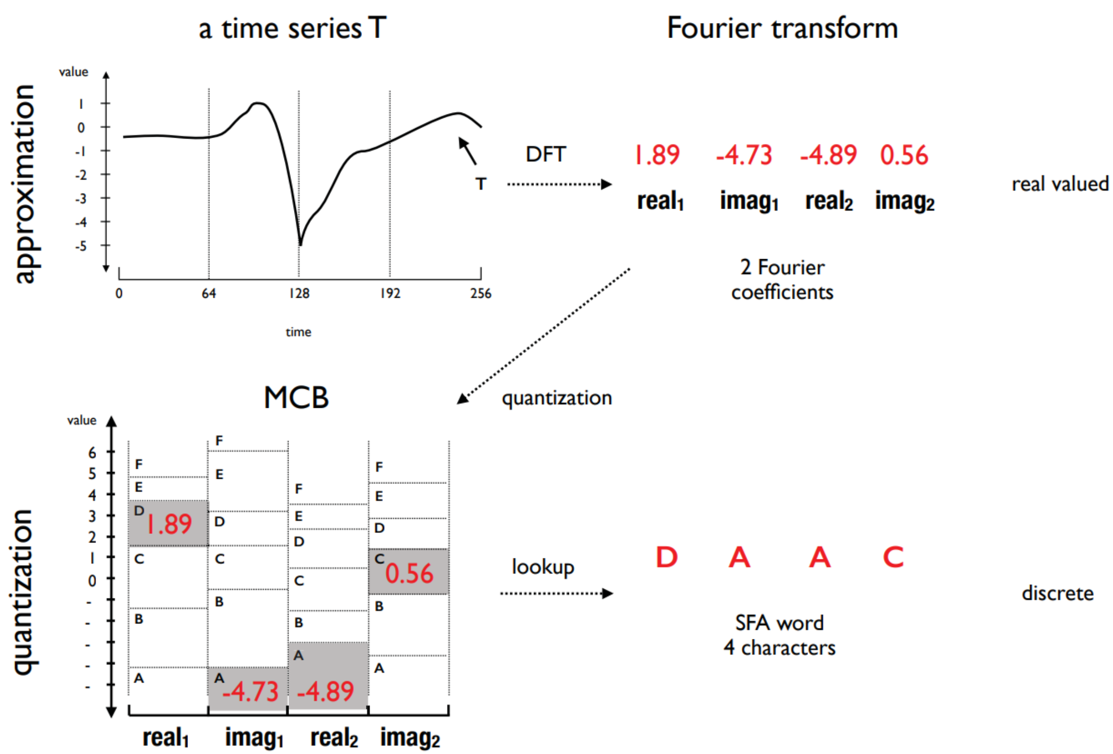
Each node contains
- ❑ # vectors
- ❑ segmentation **SG**
- ❑ synopsis **Z**

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

Symbolic Fourier Approximation (SFA)
Summarization

Publications

Schafer-EDBT'12



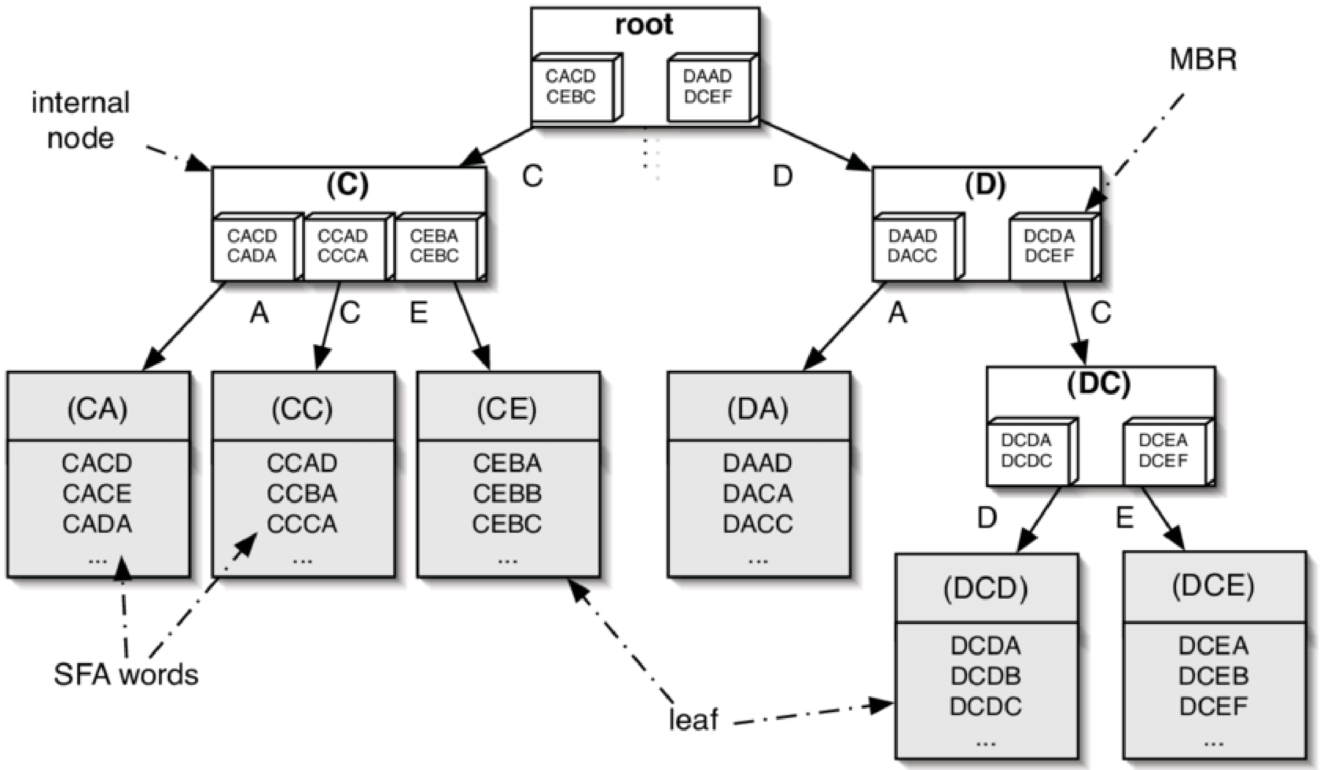
The SFA representation*

*https://www2.informatik.hu-berlin.de/~schaefpa/talks/scalable_classification.pptx

SFA
Indexing

Publications

Schafer-EDBT'12



The SFA Trie*

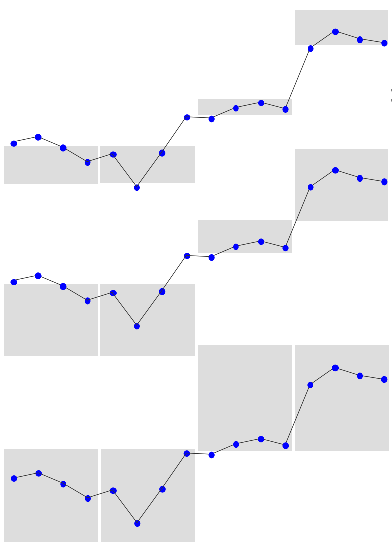
*https://www2.informatik.hu-berlin.de/~schaefpa/talks/scalable_classification.pptx

Publications

Shieh-KDD‘08

iSAX Family
iSAX Summarization

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



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

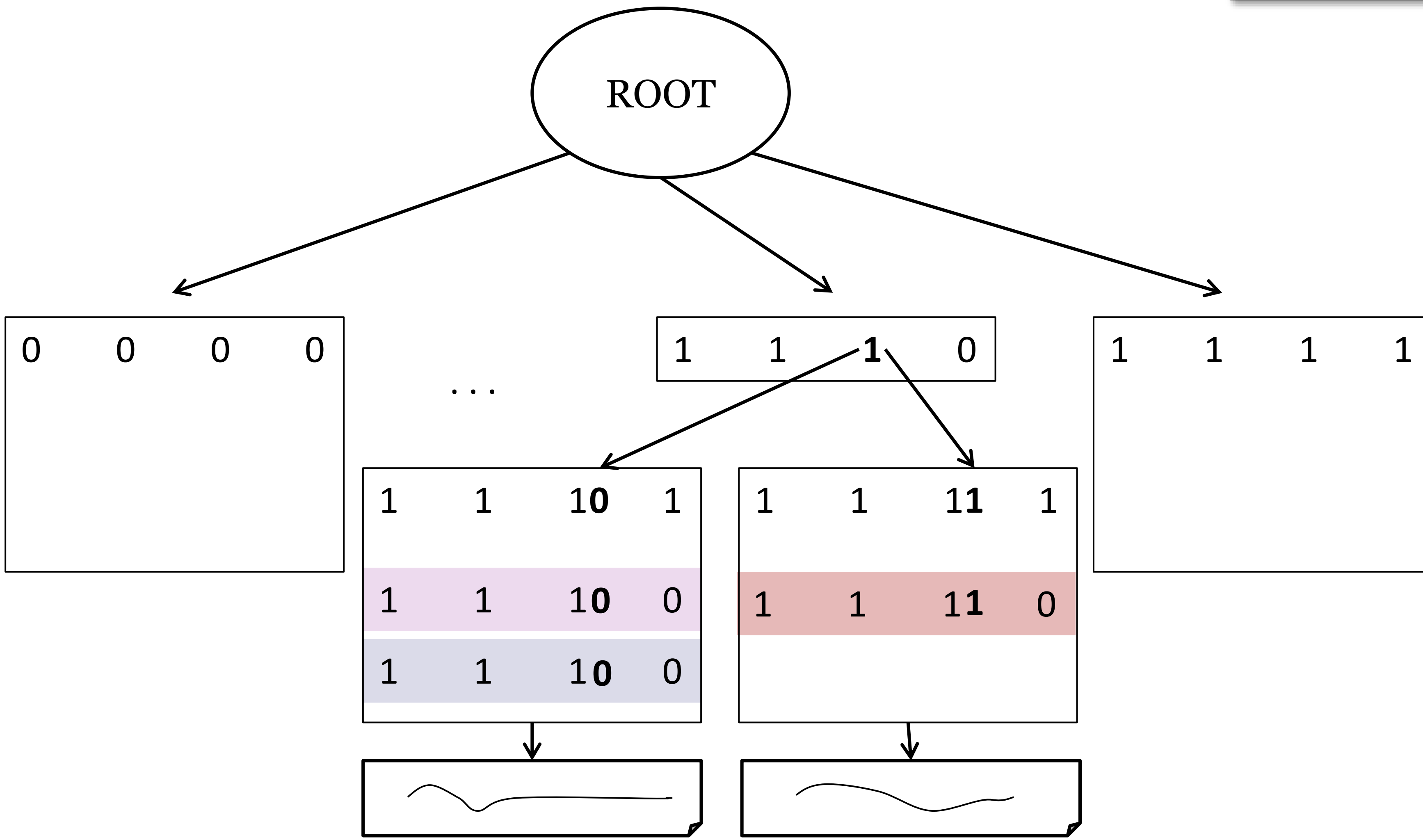
= { 3, 3, 1, 0} = {11 ,11 ,011 ,00 }

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

iSAX-Indexing

Publications

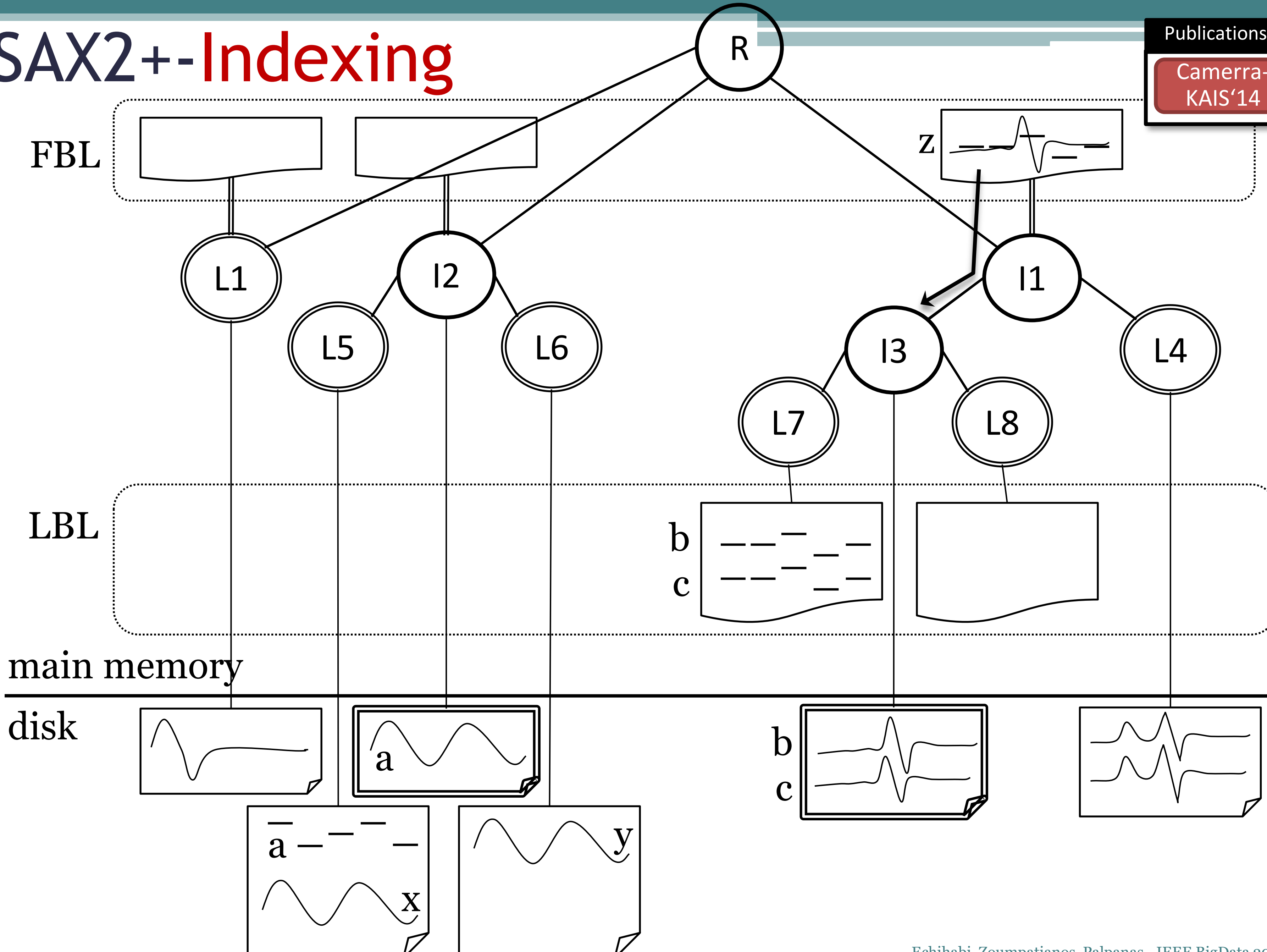
Shieh-KDD'08



iSAX2+-Indexing

Publications

Camerra-KAIS‘14



Publications

Zoumbatianos-SIGMOD'14

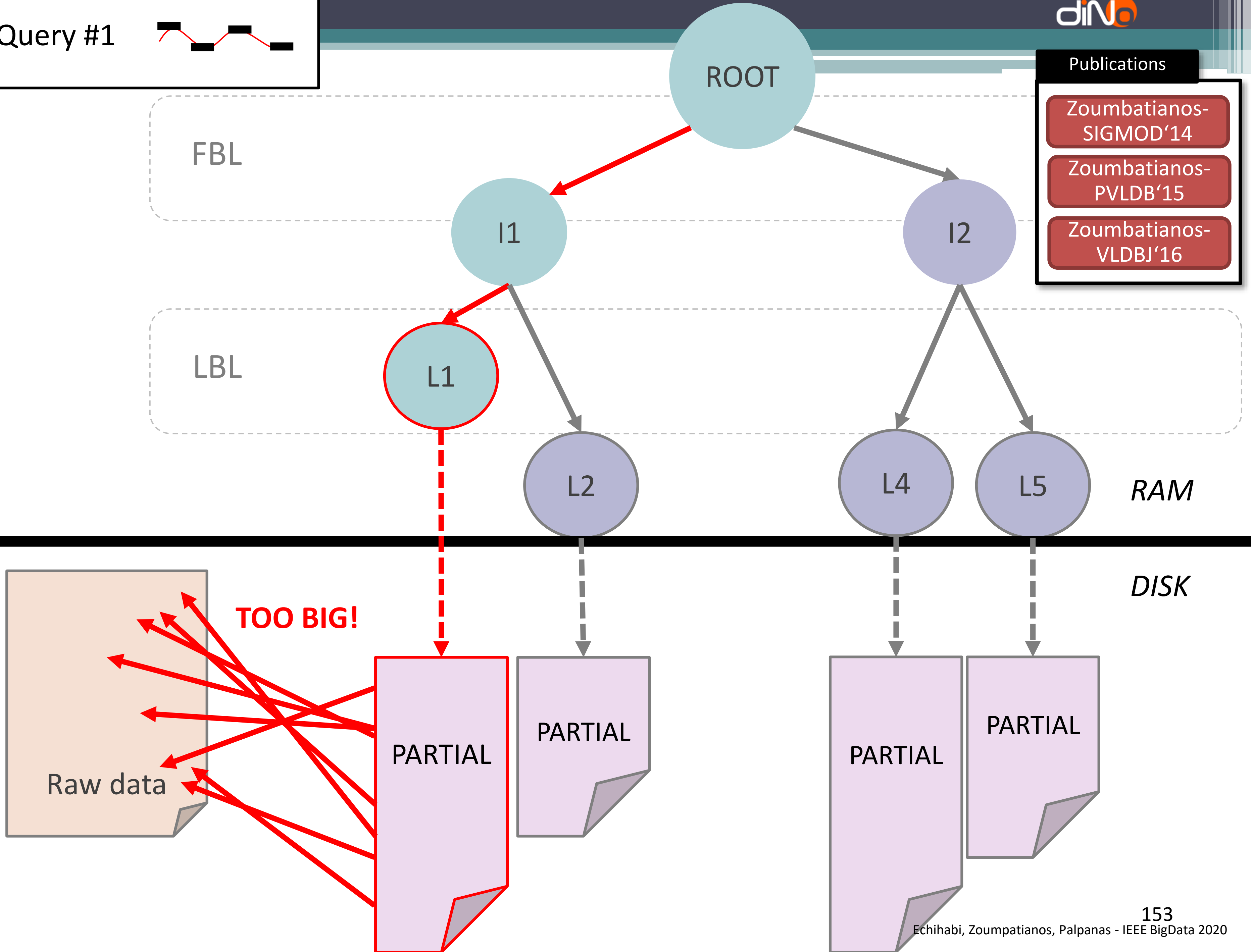
Zoumbatianos-PVLDB'15

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



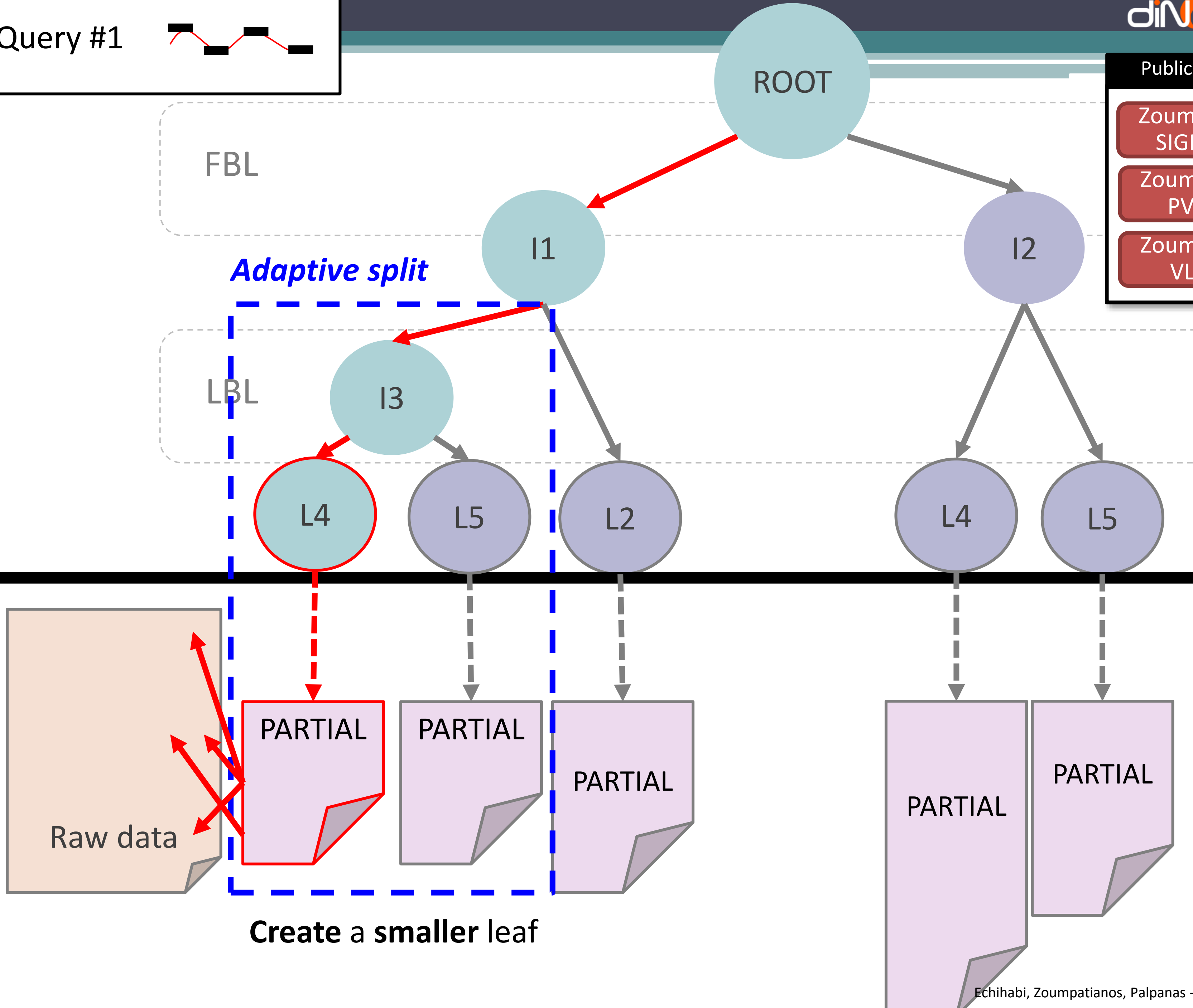
Publications

Zoumbatianos-SIGMOD'14

Zoumbatianos-PVLDB'15

Zoumbatianos-VLDBJ'16

Query #1



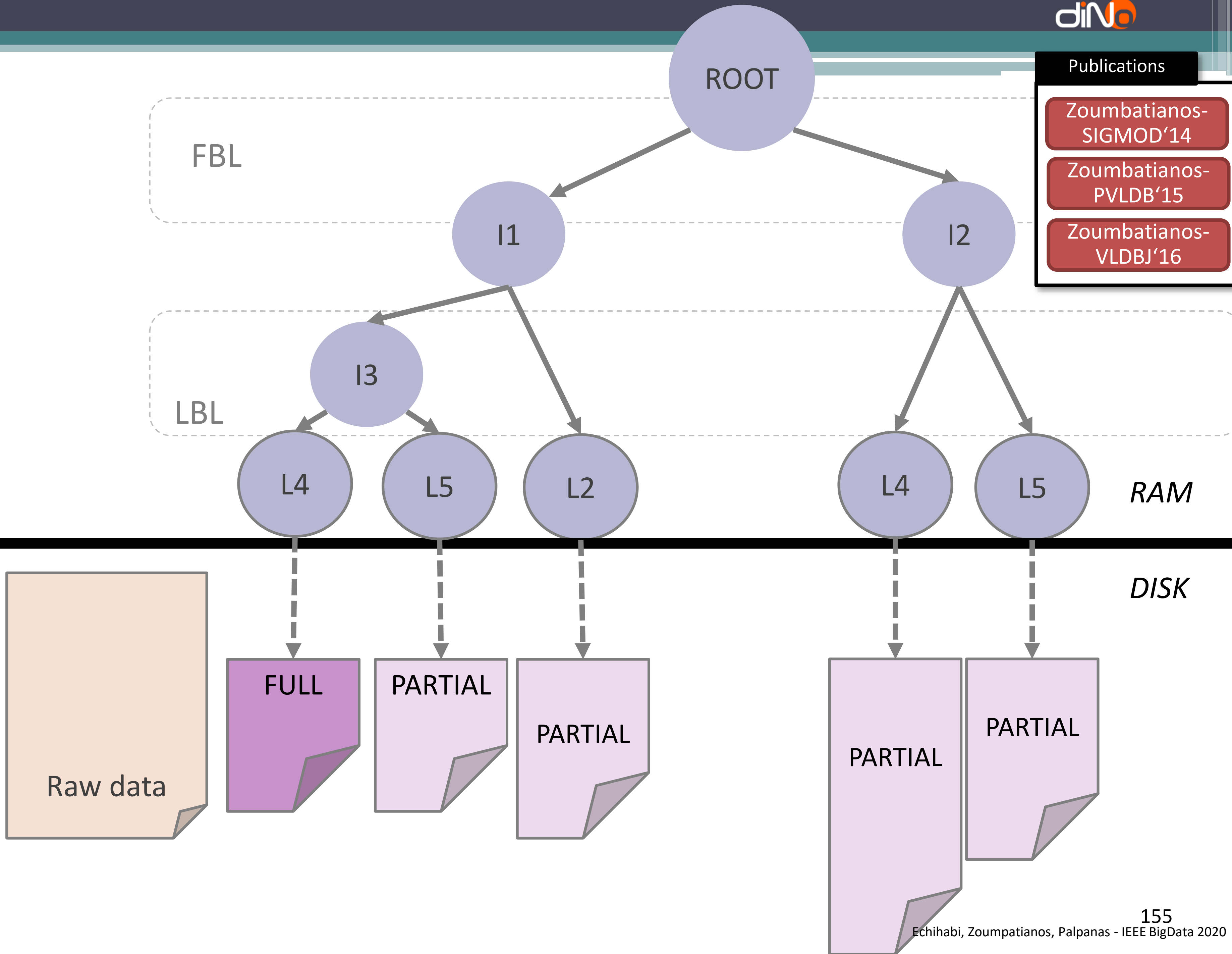
Create a **smaller** leaf

Publications

Zoumbatianos-SIGMOD'14

Zoumbatianos-PVLDB'15

Zoumbatianos-VLDBJ'16



Publications

Zoumbatianos-SIGMOD'14

Zoumbatianos-PVLDB'15

Zoumbatianos-VLDBJ'16

Publications

Kondylakis-PVLDB'18

Kondylakis-SIGMOD'19

Extensions…

- Coconut: current solution for limited memory devices and streaming time series
  - bottom-up, succinct index construction based on sortable summarizations

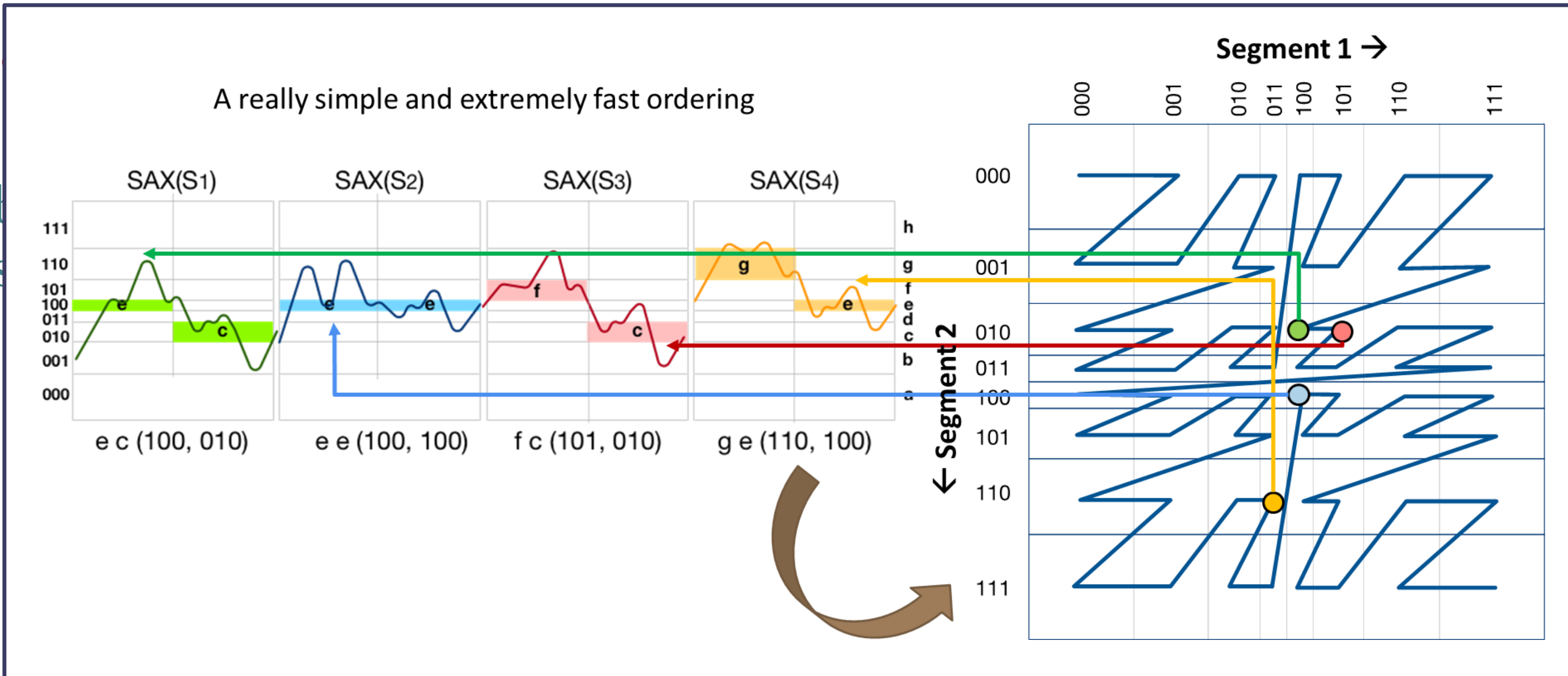
Publications

Kondylakis-PVLDB'18

Kondylakis-SIGMOD'19

Extensions…

- Co
  - b s



Publications

Kondylakis-PVLDB'18

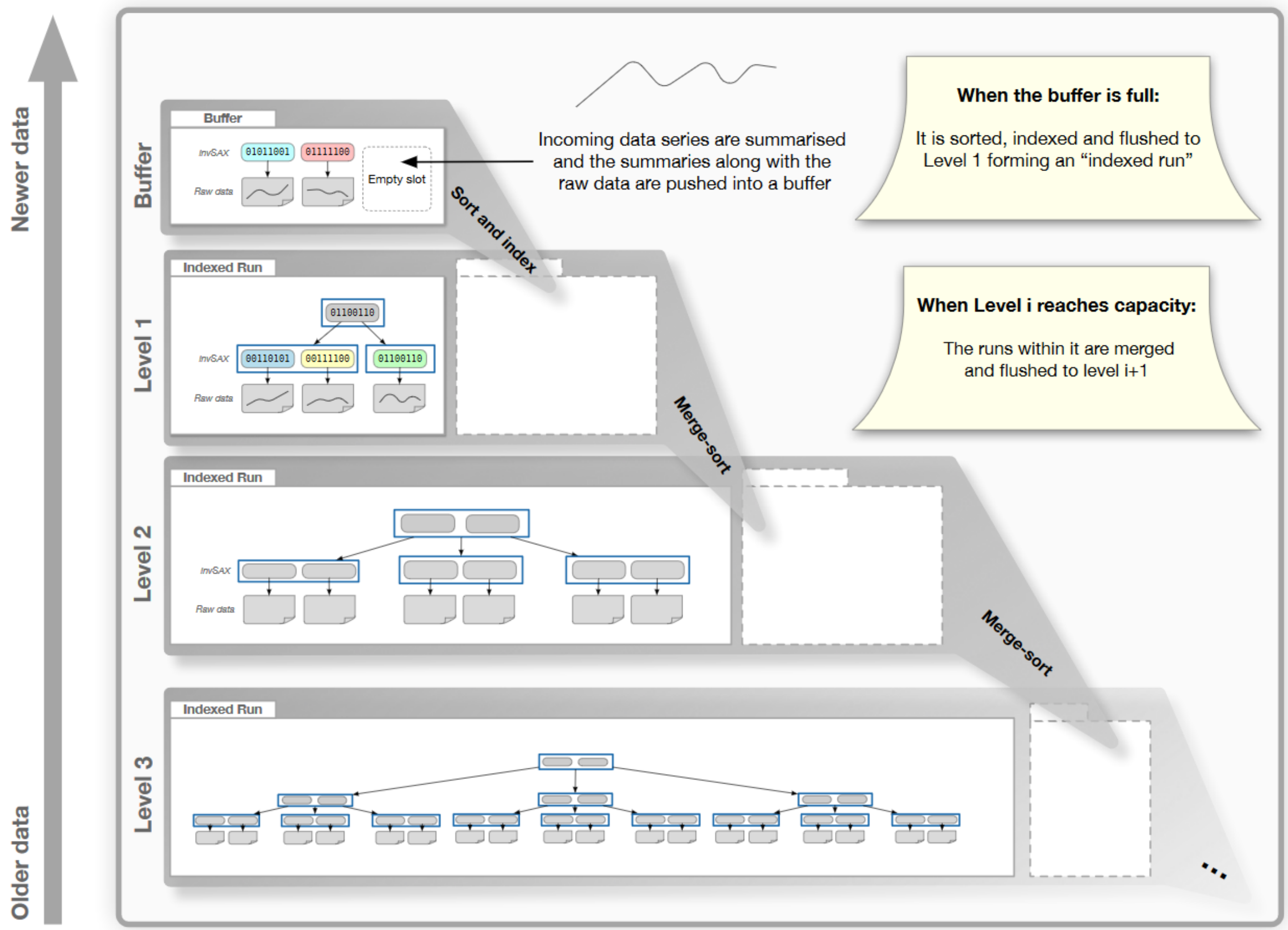
Kondylakis-SIGMOD'19

Extensions…

- Coconut: current solution for limited memory devices and streaming time series
  - bottom-up, succinct index construction based on sortable summarizations
  - outperforms state-of-the-art in terms of index space, index construction time, and query answering time

Coconut-LSM

Extensions…



Publications

Kondylakis-PVLDB'18

Kondylakis-SIGMOD'19

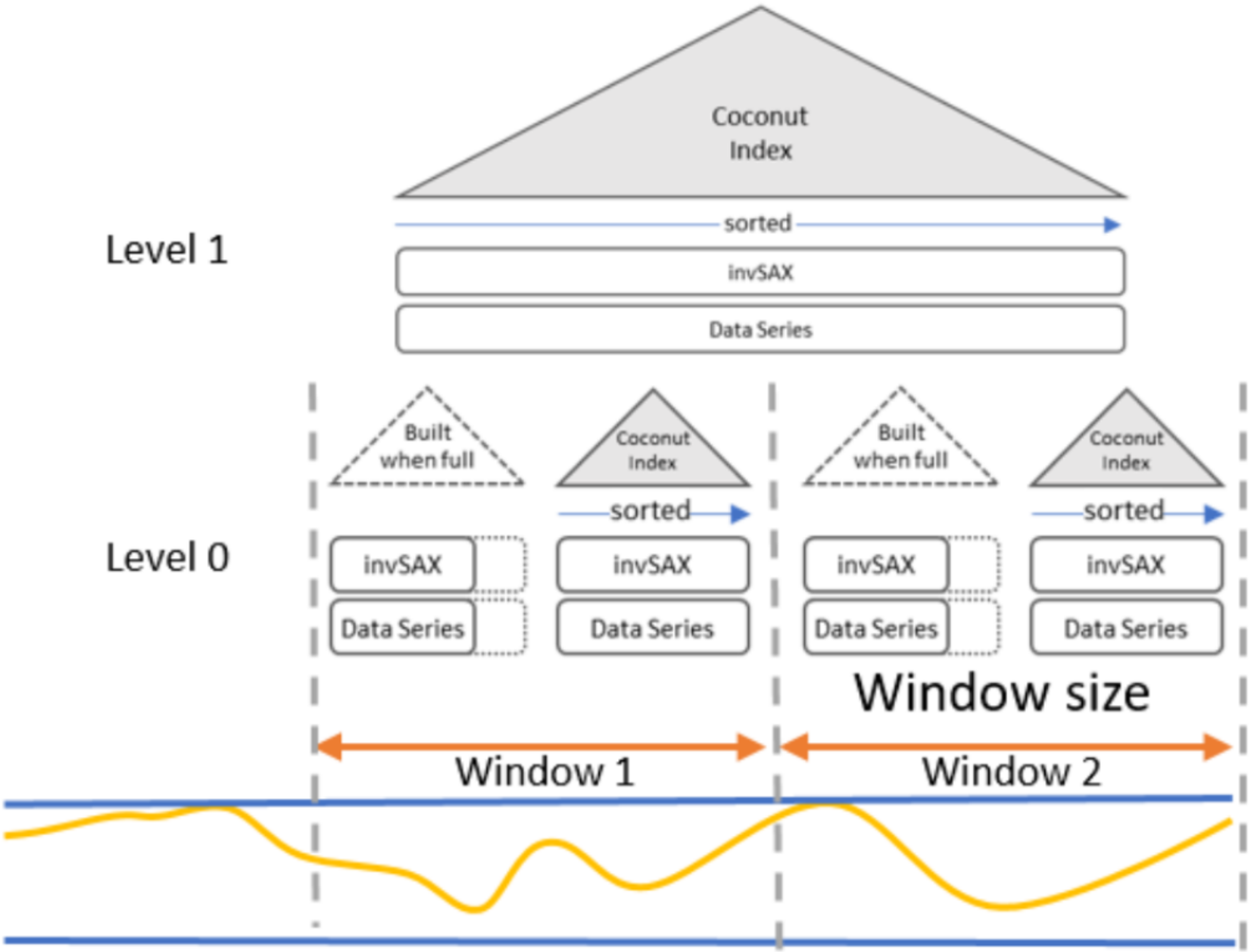
Kondylakis-VLDBJ'20

Coconut-LSM

Extensions…

Publications

- Kondylakis-PVLDB'18
- Kondylakis-SIGMOD'19
- Kondylakis-VLDBJ'20



Publications

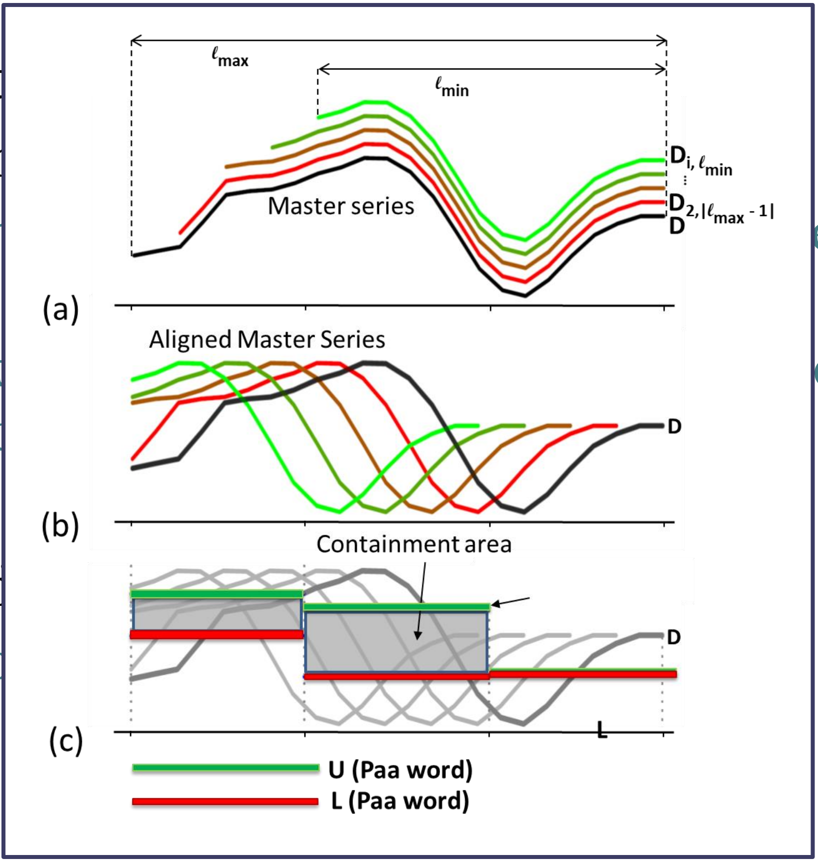
- Kondylakis-PVLDB'18
- Kondylakis-SIGMOD'19
- Kondylakis-VLDBJ'20
- Linardi-ICDE'18
- Linardi-PVLDB'19
- Linardi-VLDBJ'20

Extensions...

- Coconut: current solution for limited memory devices and streaming time series
  - bottom-up, succinct index construction based on sortable summarizations
  - outperforms state-of-the-art in terms of index space, index construction time, and query answering time
- ULISSE: current solution for variable-length queries
  - single-index support of queries of variable lengths

Extensions...

- Coconut: current solut
  and streamin
  - bottom-up, succinct in   e
    summarizations
  - outperforms state-of-t   ex
    construction time, and
- ULISSE: current soluti
  - single-index support o



Publications

- Kondylakis-PVLDB‘18
- Kondylakis-SIGMOD’19
- Kondylakis-VLDBJ’20
- Linardi-ICDE’18
- Linardi-PVLDB‘19
- Linardi-VLDBJ‘20

Extensions…

- Coconut: current solution for limited memory devices and streaming time series
  - bottom-up, succinct index construction based on sortable summarizations
  - outperforms state-of-the-art in terms of index space, index construction time, and query answering time
- ULISSE: current solution for variable-length queries
  - single-index support of queries of variable lengths
  - orders of magnitude faster than competing approaches

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

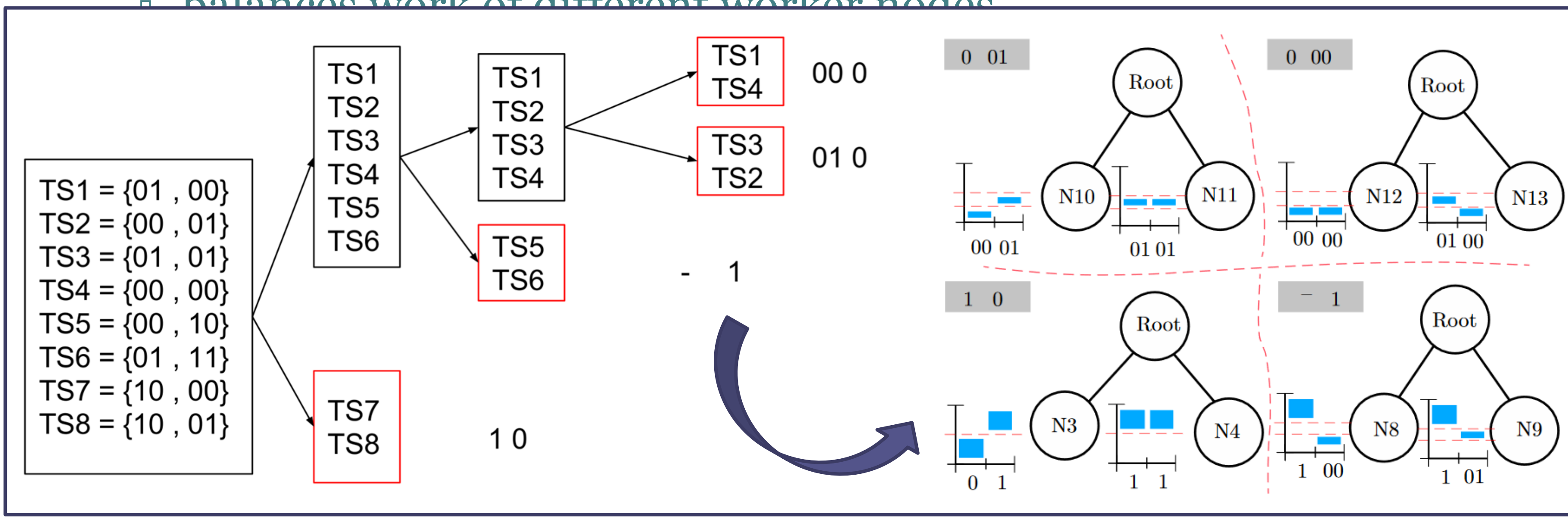
Yagoubi-ICDM'17

Yagoubi-TKDE'18

Lavchenko-KAIS'20

Parallelization/Distribution

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



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
  - performs 2 orders of magnitude faster than centralized solution

Publications

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

Parallelization/Distribution

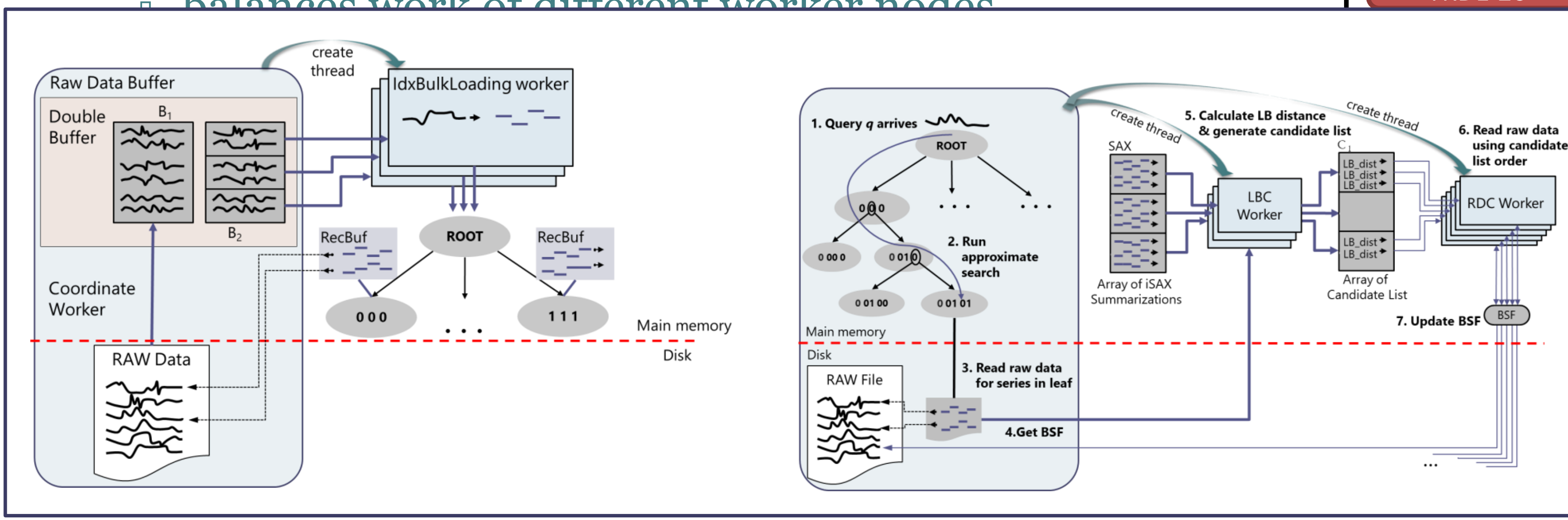
- DPiSAX: current solution for distributed processing (S
  - balances work of different worker nodes
  - performs 2 orders of magnitude faster than centralized solution
- ParIS: current solution for modern hardware
  - completely masks out the CPU cost

Publications

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

Parallelization/Distribution

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



Publications

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

Parallelization/Distribution

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

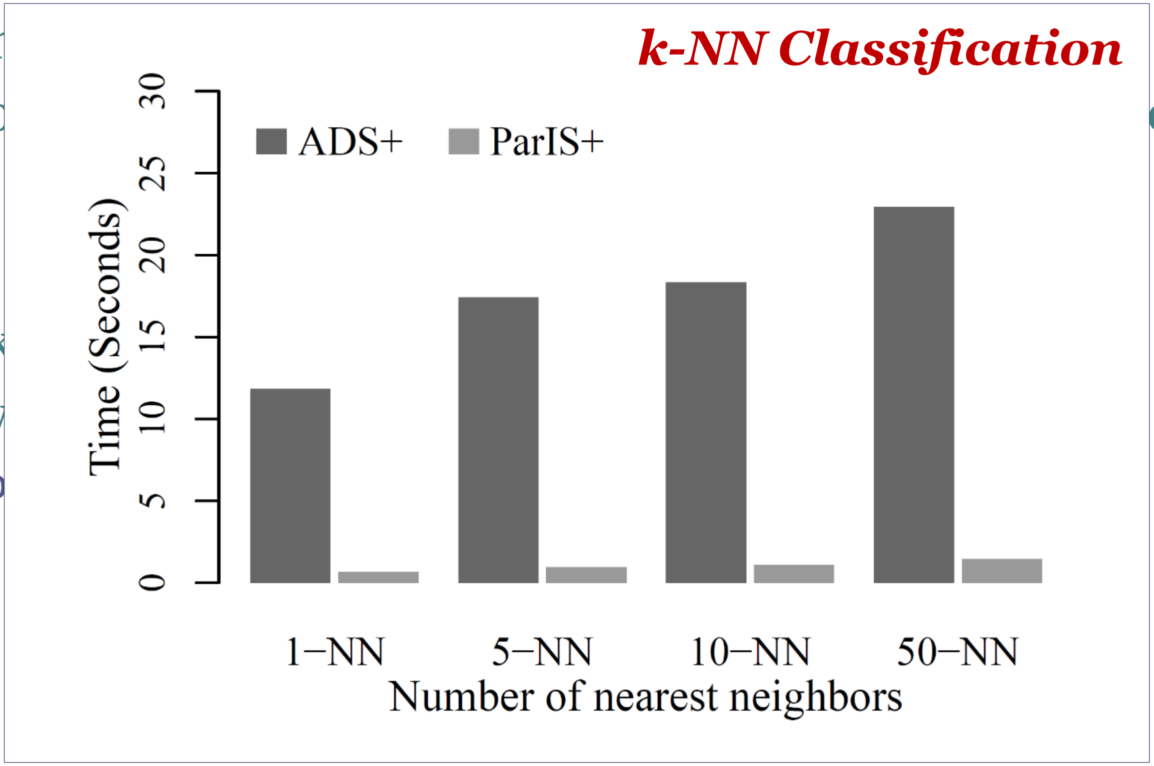
Lavchenko-KAIS'20

Peng-BigData'18

Peng-TKDE'20

Parallelization/Distribution

- DPiSAX: current solution for distributed processing (S
  - balar
  - perfo d solution
- ParIS:
  - mask
  - answ
    - 3 o

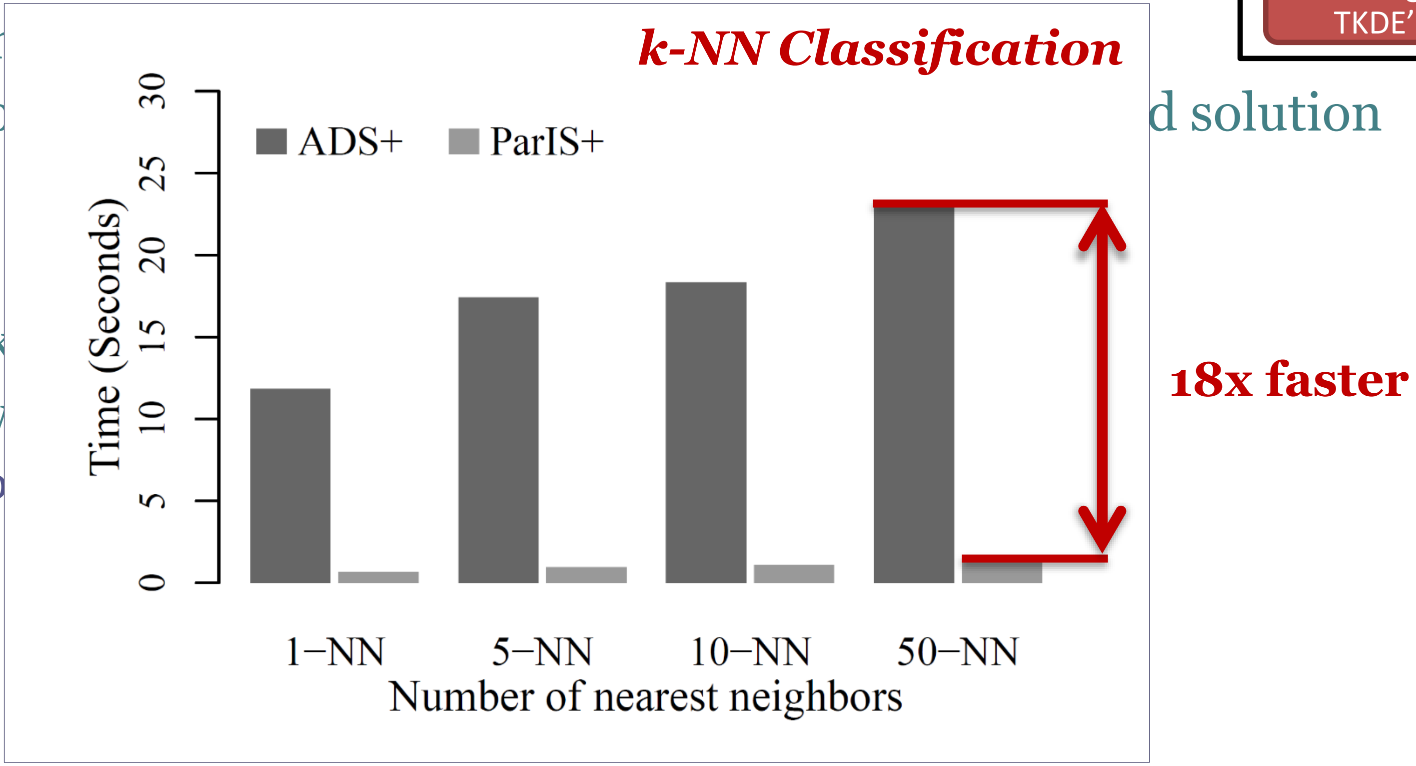


Publications

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

Parallelization/Distribution

- DPiSAX: current solution for distributed processing (S
  - balar
  - perfo ... d solution
- ParIS:
  - mask
  - answ
    - 3 o

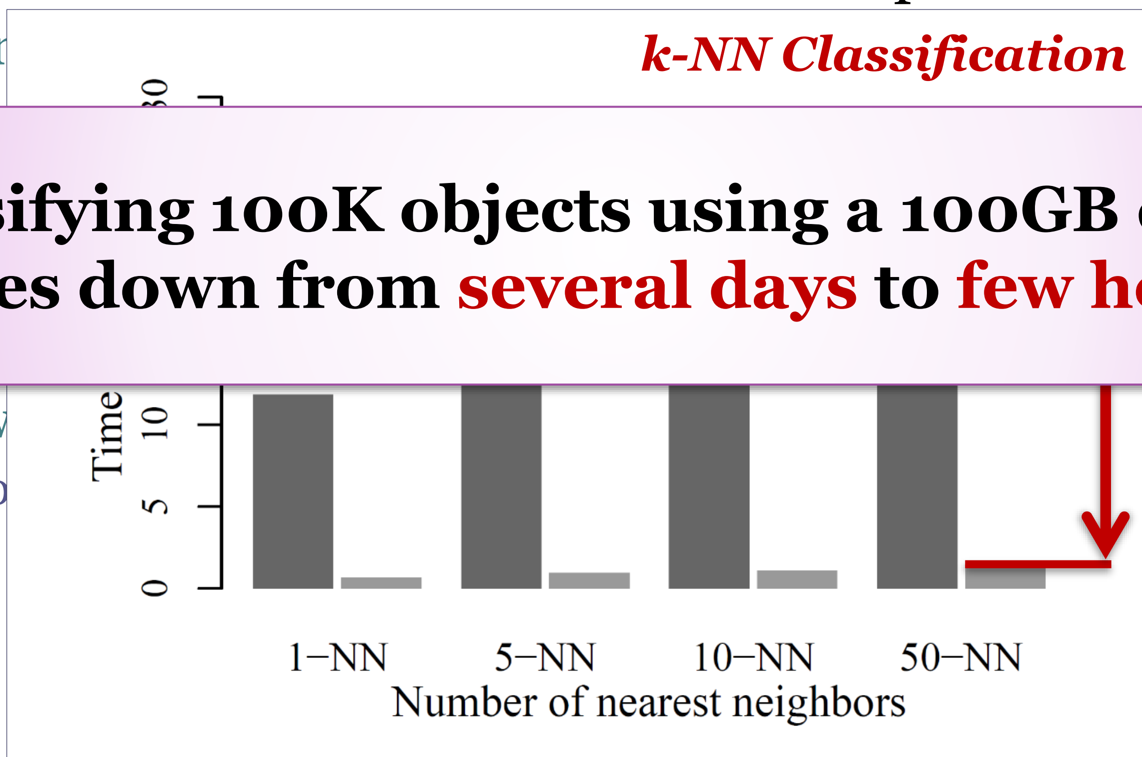


Publications

- Yagoubi-ICDM‘17
- Yagoubi-TKDE’18
- Lavchenko-KAIS’20
- Peng-BigData’18
- Peng-TKDE’20

Parallelization/Distribution

- DPiSAX: current solution for distributed processing (S
  - balar
  - answ
    - 3 o



18x faster

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'20
- Peng-ICDE'20

Parallelization/Distribution

- DPiSAX: current solution for distributed processing (Spar
  - 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

Publications

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

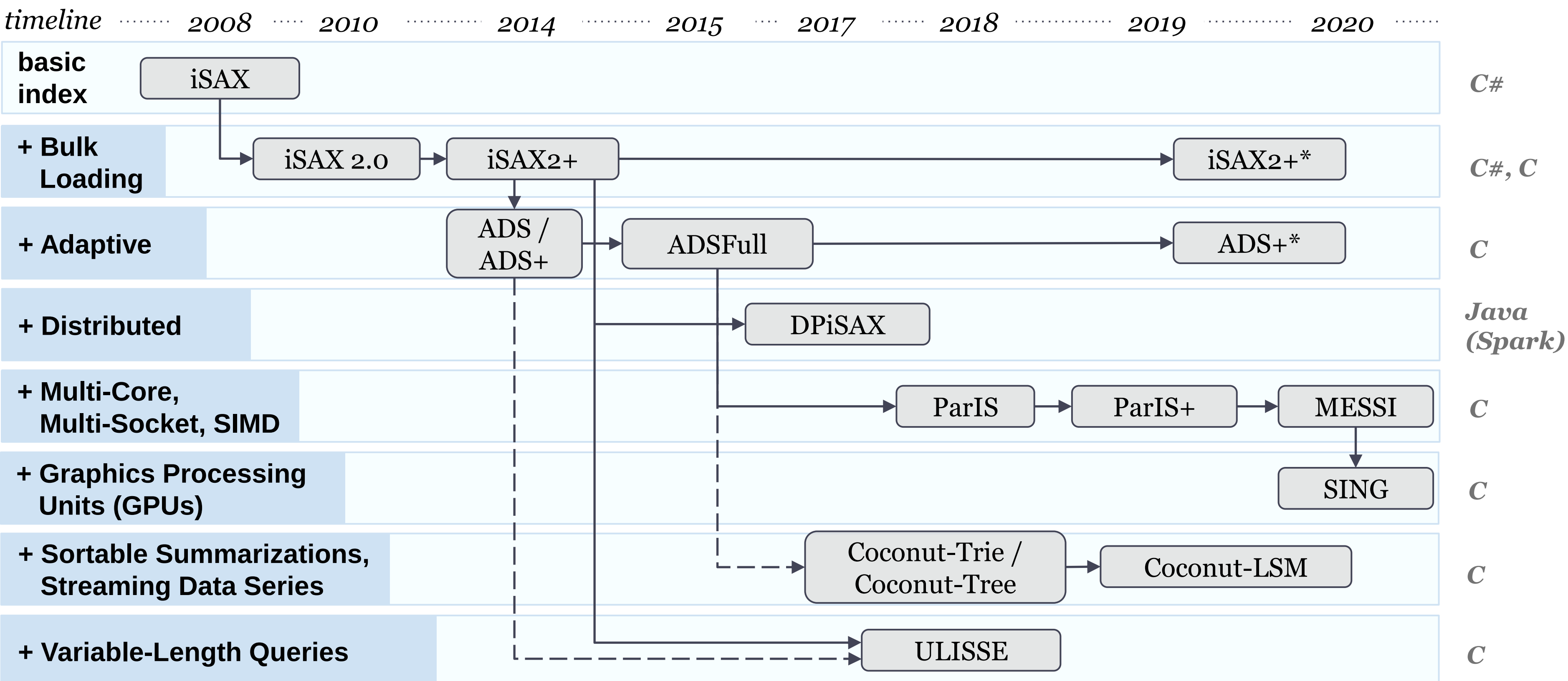
Parallelization/Distribution

- DPiSAX: current solution for distributed processing (Spar
  - 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

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.

Experimental Comparison: Exact Query Answering Methods

How do these methods compare?

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

- we now present results of extensive experimental comparison

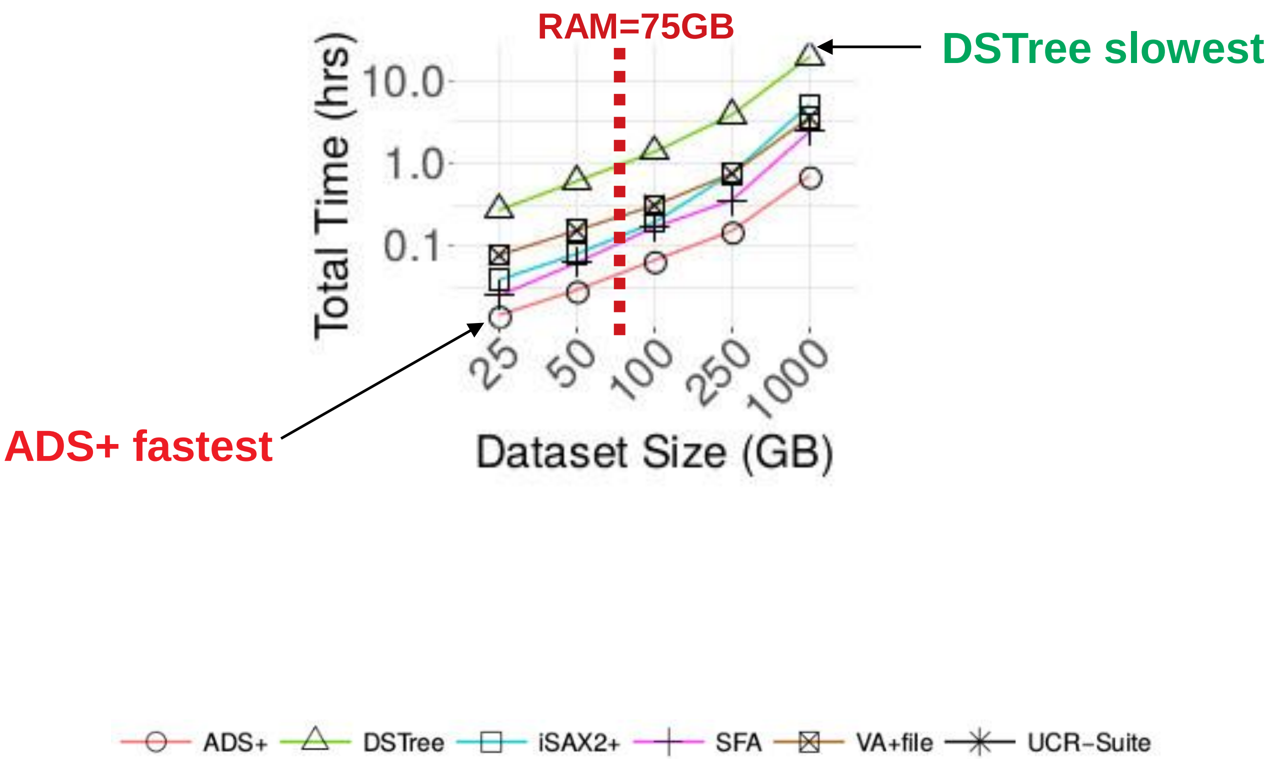
Publications

Echihabi-PVLDB'18

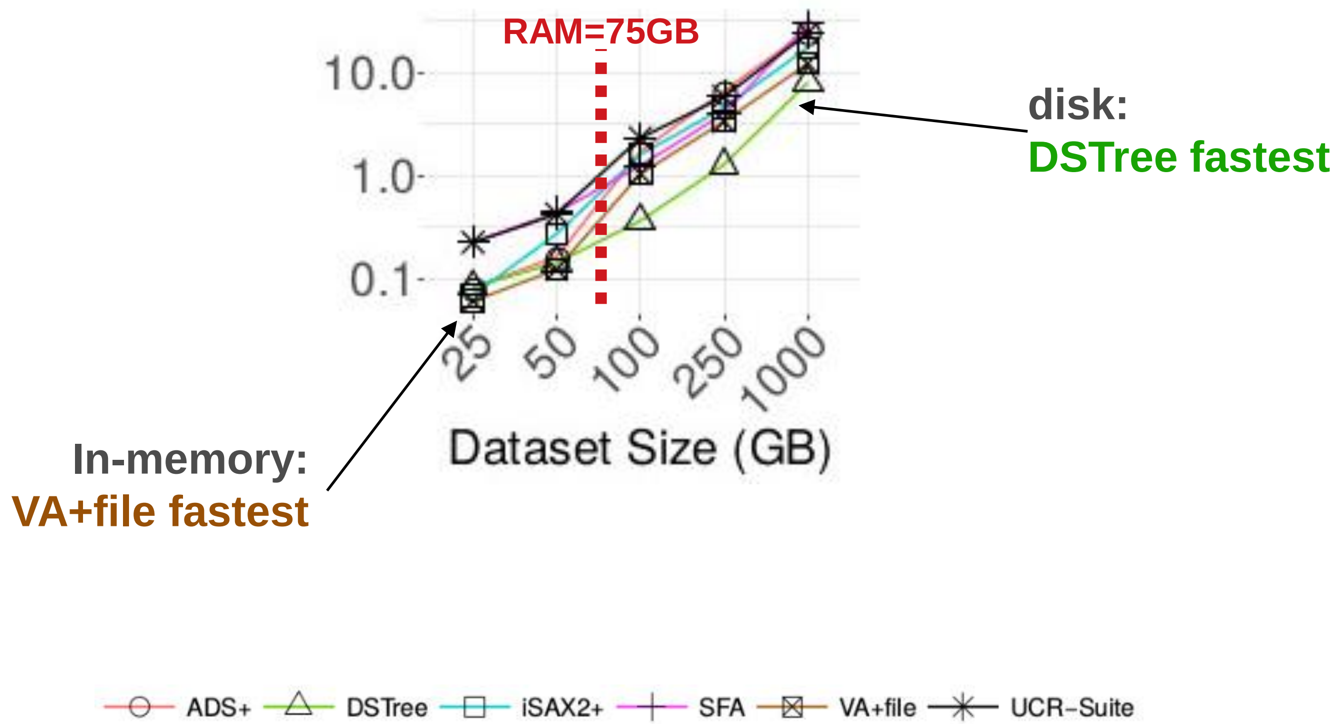
Experimental Framework

- 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

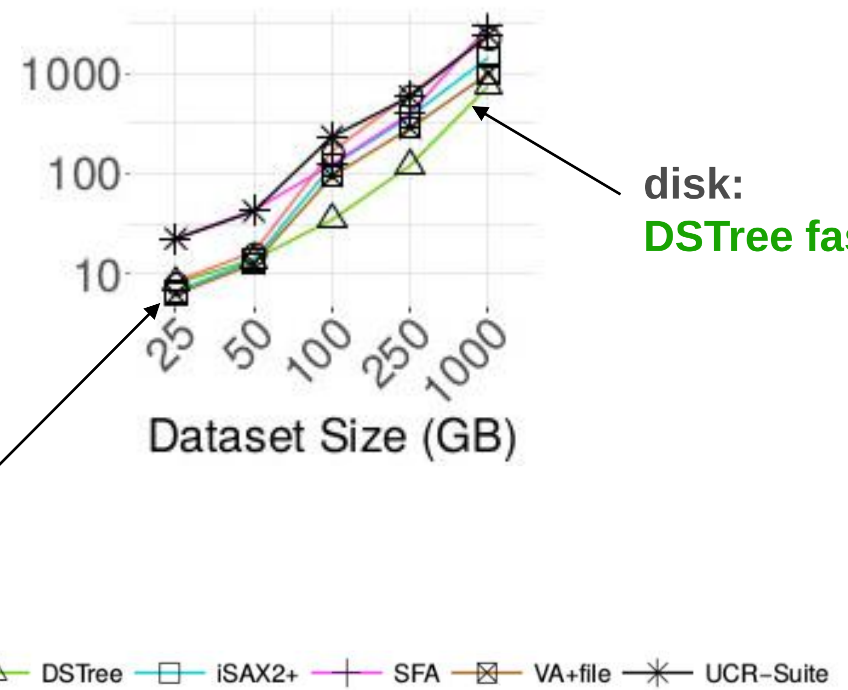
Time for Indexing (Idx) vs. Dataset Size



Time for 100 Exact Queries vs. Dataset size



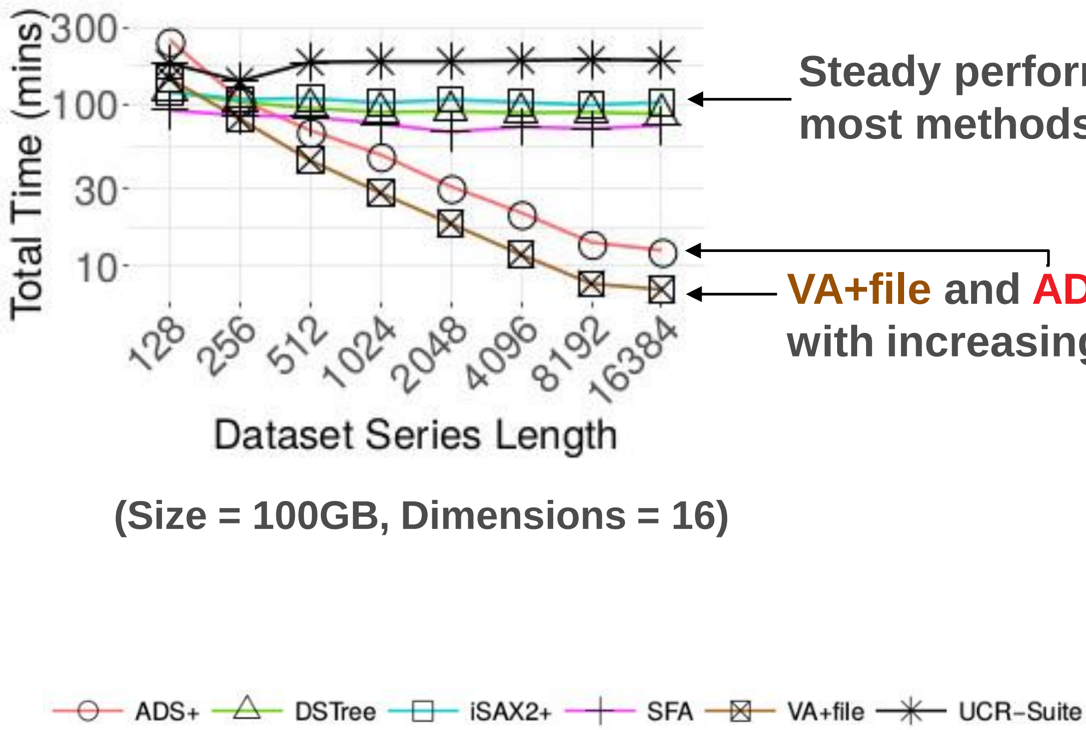
Time for Idx + 10K Exact Queries vs. Dataset size



disk:
DSTree fastest

In-memory:
VA+file fastest

Time for Idx + 10K Exact Queries vs. Series Length



Steady performance for most methods

VA+file and ADS+ get faster with increasing length

(Size = 100GB, Dimensions = 16)

Publications

Echihabi-PVLDB'18

Unexpected Results

- Some methods do not scale as expected (or not at all!)
- Brought back to the spotlight two older methods VA+file and DSTree
  - Our 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

Publications

Echihabi-PVLDB'18

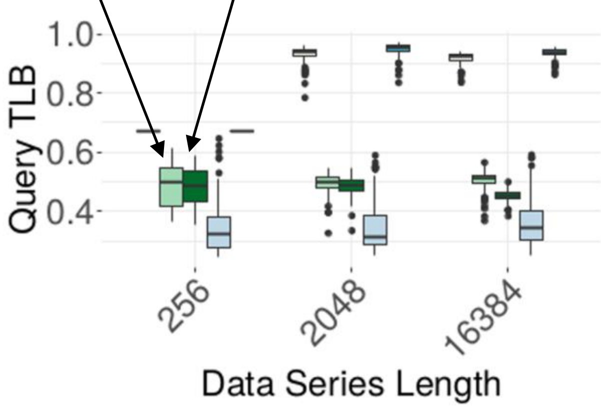
TLB does not always predict performance

The TLB measures the quality of a summarization (higher is better)

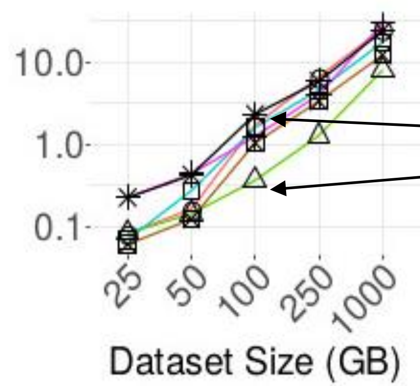
$$0 \leq \quad \text{TLB} = \frac{dist(Query, candidate)\ in\ reduced\ space}{dist(Query, candidate)\ in\ original\ space} \leq 1$$

worst best

DSTree and iSAX2+ have similar TLB



YET



iSAX2+ 5x slower than DSTree

No bias, same data and same implementation framework

Publications

Echihabi-PVLDB‘18

Insights

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

Time Series Management Systems

Storing Time-Series

Multiple options. By popularity:

- File System
- RDBMS
- Specialized Time-Series DBs
- Array DBs

Storing Time-Series:
File-System

Multiple **different formats** implemented for **various applications**

- FITS
- HDF5

Astronomy

- SEED
- MiniSEED
- ASCII
- GeoCSV

Seismology

- BIDS (EEG)
- WFDB (ECG)
- EDF(ECG)
- FASTA (DNA)

Biology

- CSV

Finance

- HDF5
- NetCDF

Engineering

- HDF5
- NetCDF

Physics

- CSV
- TSV
- XLS
- Parquet
- etc.

Data Science

Storing Time-Series: **DBMS**

Commercial System

Illustra (1993) → IBM Informix (Time-Series DataBlade):
- Users need to define a time-series sub-type, which have a datetime as the first column in the definition
- Can encode both regular and irregular time-series (fixed of variable intervals)
- Can describe meta-data
- Supports: running aggregates, prev, next value reasoning, horizontal and vertical mathematical operations, lags, etc.

Academic System

Publications

Seshadri-ICDE'95

Shore → SEQ
- Custom Time-Series Data Type
- Various time-series operators (order, correlation, etc.)

Commercial System

Oracle:
- Introduced Time-Series functionality in Oracle8
- Now merged into the main product.
- It is in the form of time-series analytics functions (e.g., forecasting)

Storing Time-Series: **DBMS**

Commercial System

Illustra (1993) → IBM Inform... ...es DataBlade):

- Users need to define a ... which have a datetime as the first column in the d...
- Can encode ... ...d of variable interv...
- Ca...
- Supp... ...zontal and vertica...

Most people use DBMSs merely for storing and retrieving time-series.

All analysis is performed externally.

Academic System

Shore → SE...

- Custom Ti...
- Various time...

Commercial System

Oracle:

- Introduced Ti... ...nality ...
- Now merged into the main product.
- It is in the form of time-series analytics functions (e.g., forecasting)

Storing Time-Series: **Specialized Time-Series DBs**

InfluxDB
- Storage: Custom (TSM-Tree)

TimeScaleDB
- Storage: PostgreSQL

Beringei
- Storage: Compressed Arrays on Disk

Druid
- Metadata Storage: DBMS
- Data Storage: HDFS, S3

Prometheus
- Storage: Custom (TSDB Format)

CrateDB
- Storage: Custom (Column-oriented)

IoTDB
- Storage: Custom: (TsFile – compression + stats)

OpenTSDB
- Storage: HBase

QuasarDB
- Storage: RocksDB

Amazon TimeStream
- Storage: Unknown

Storing Time-Series: **ArrayDBs**

TileDB

Custom Log-Structured storage

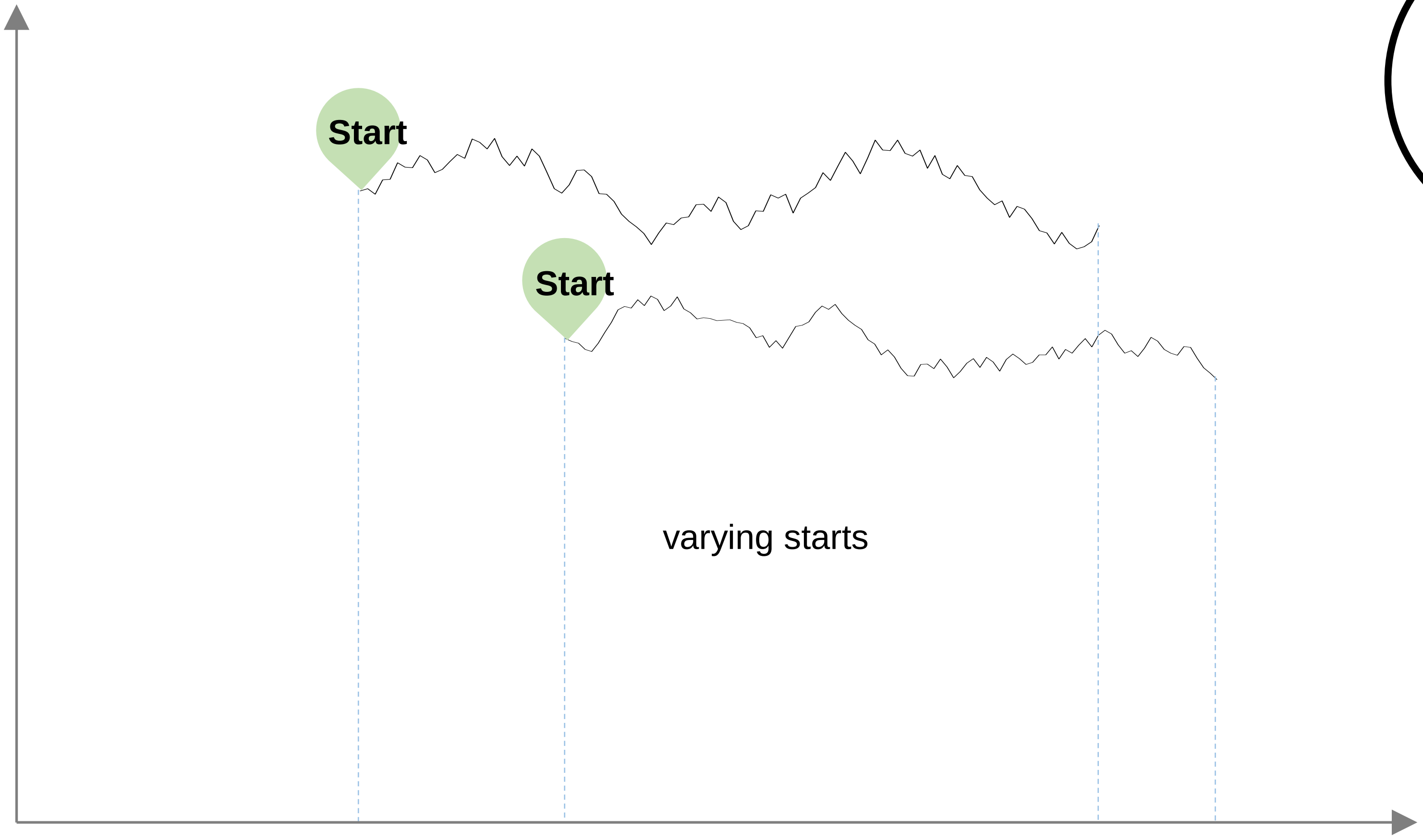
Rasdaman

Sits on top of existing DBMSs

SciDB

Custom storage

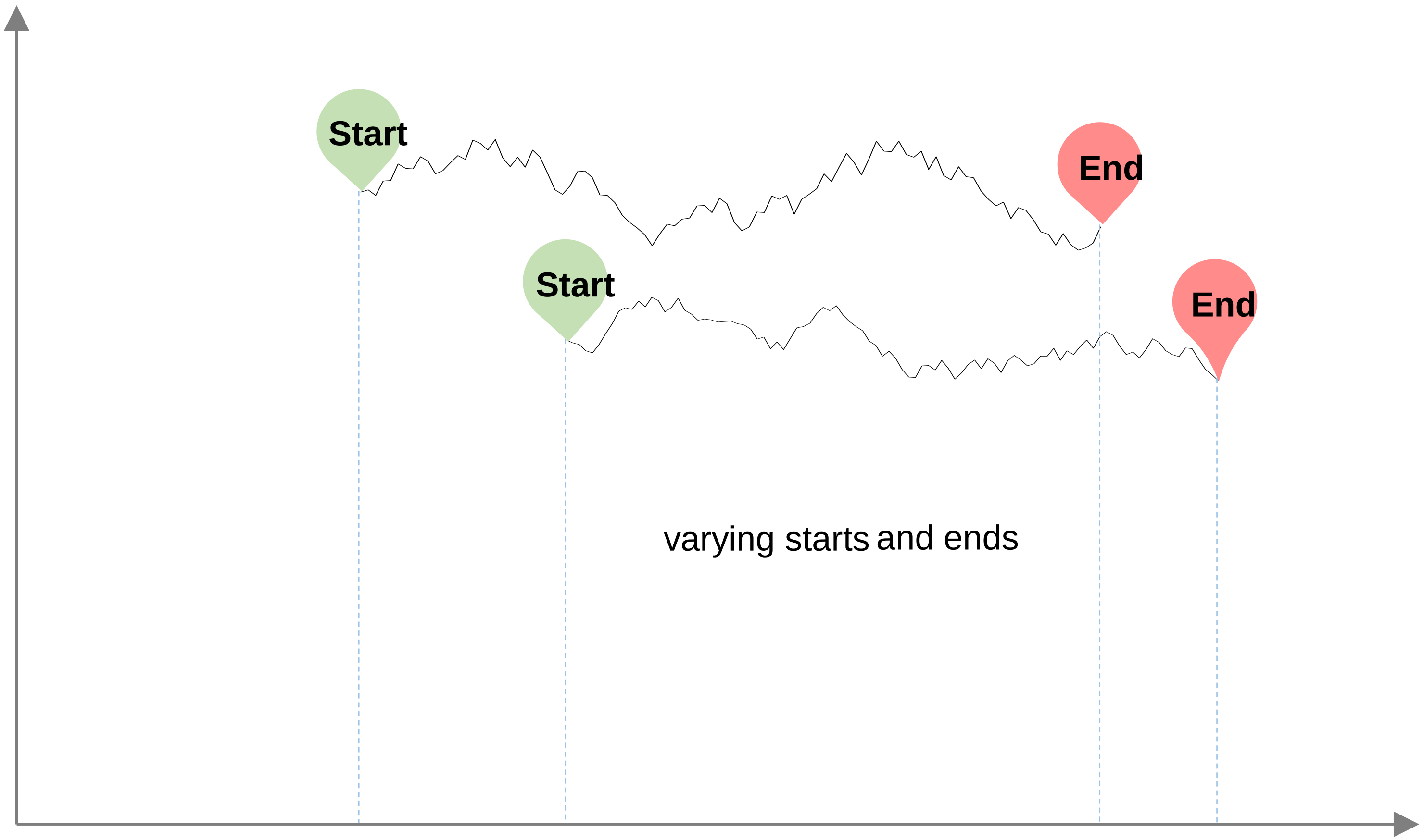
Time-Series Characteristics



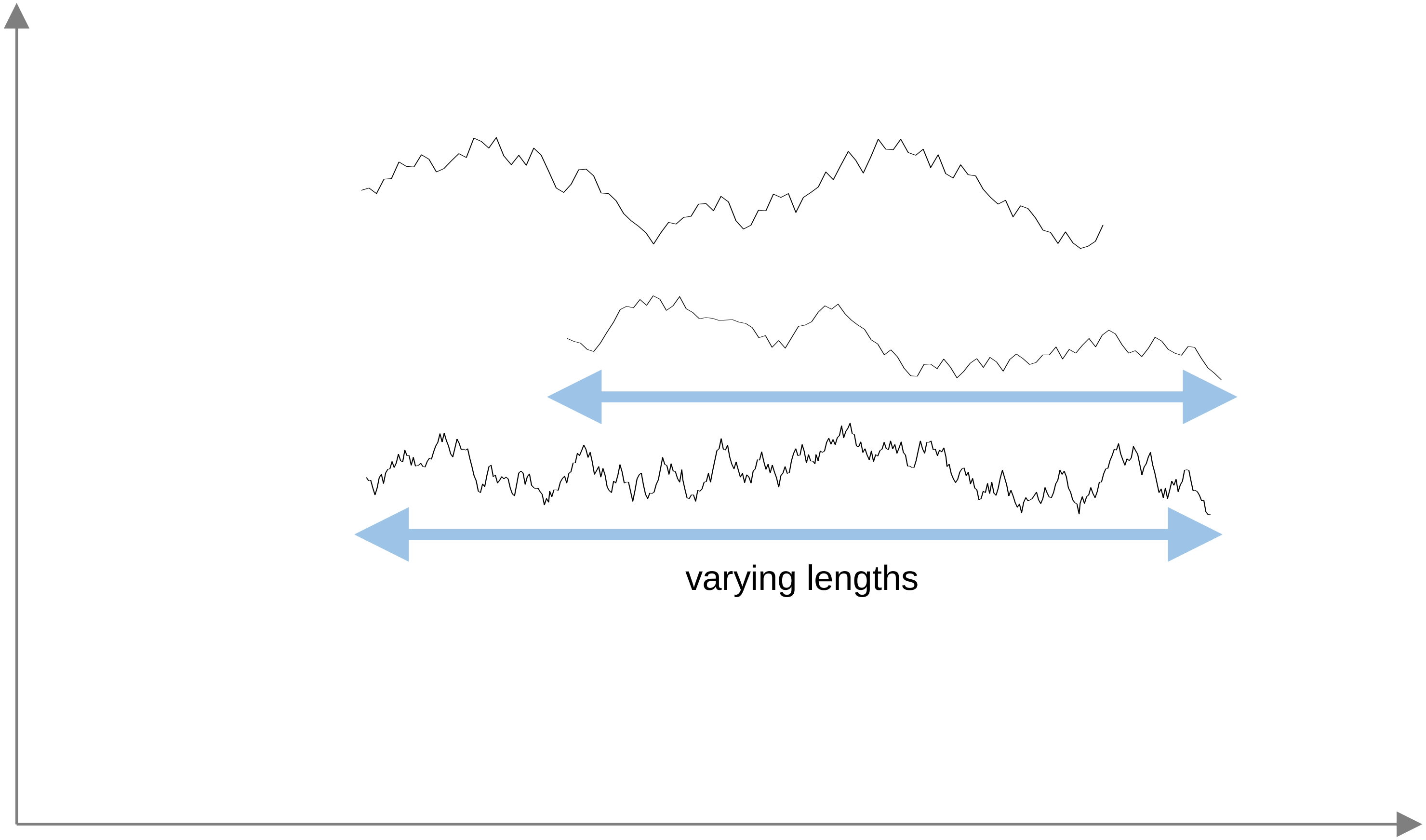
The Data-Type Characteristics

What are the properties of data series?

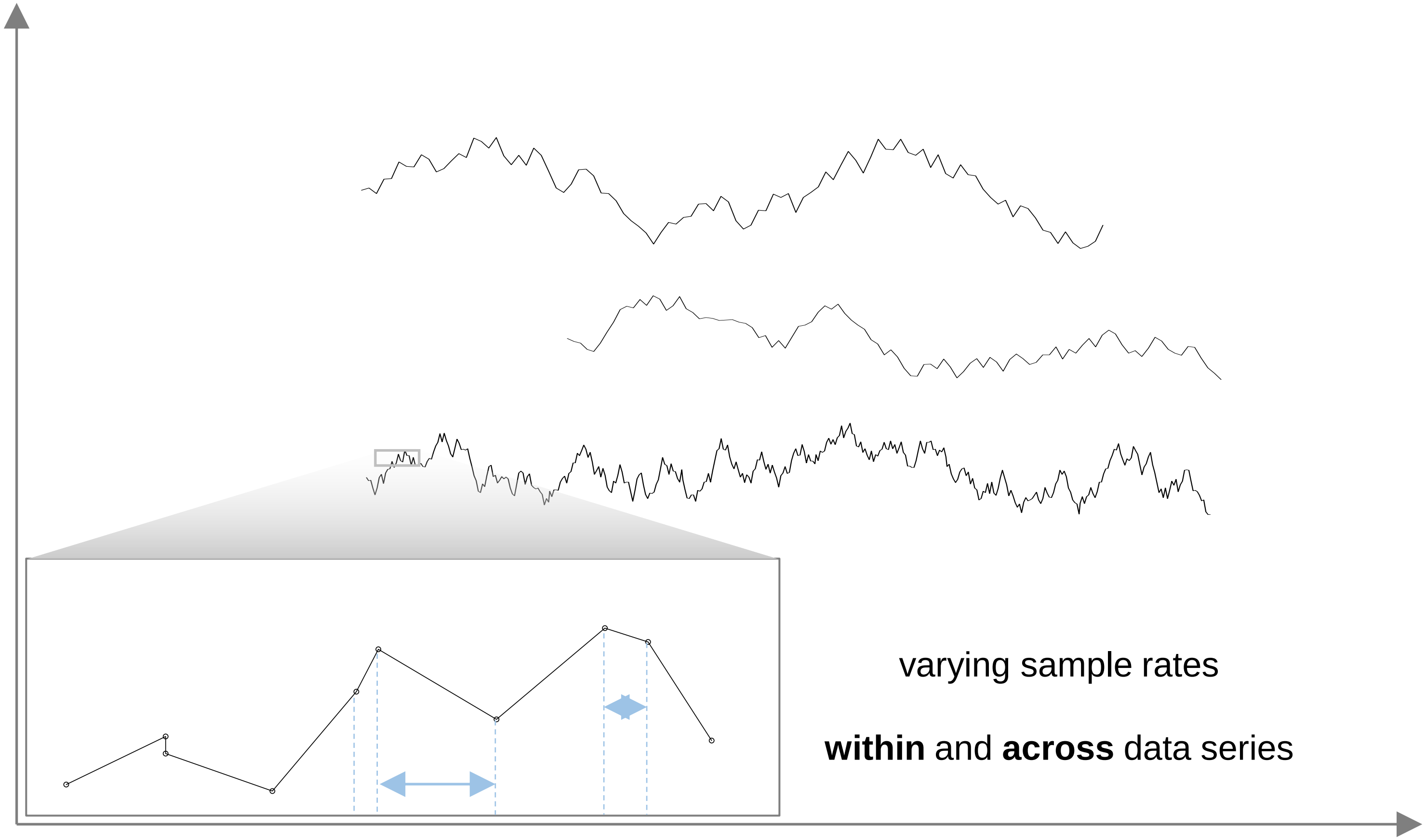
Time-Series Characteristics



Time-Series Characteristics



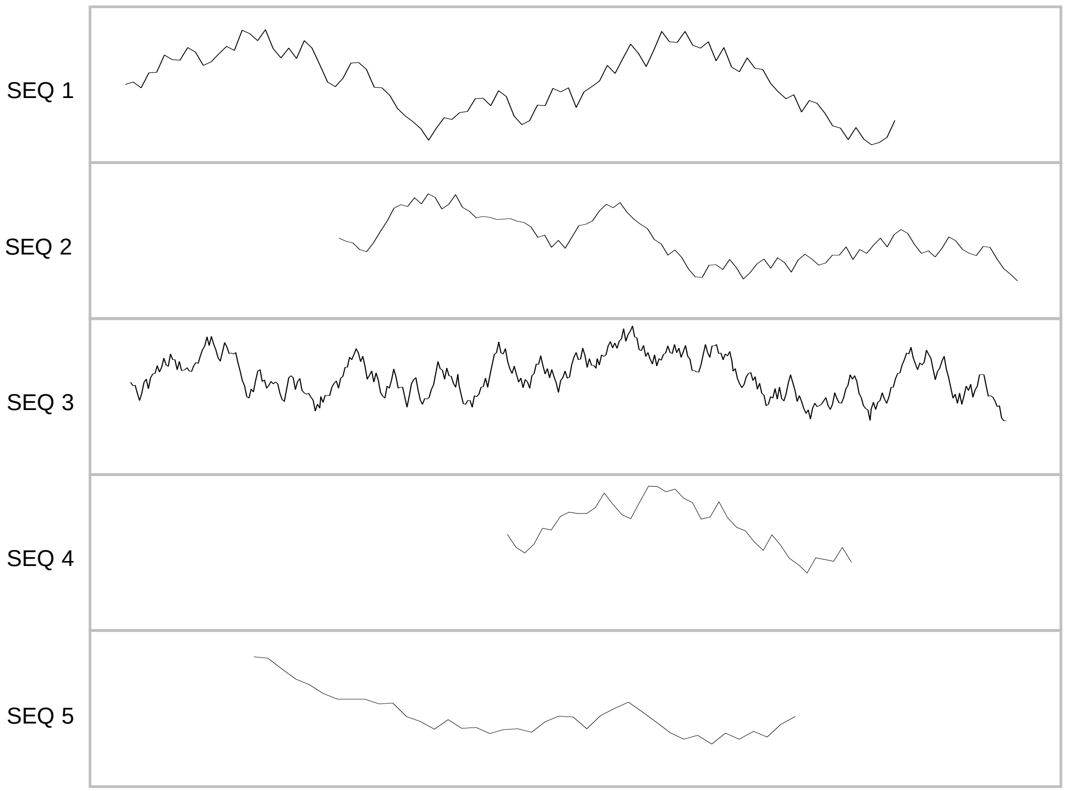
Time-Series Characteristics



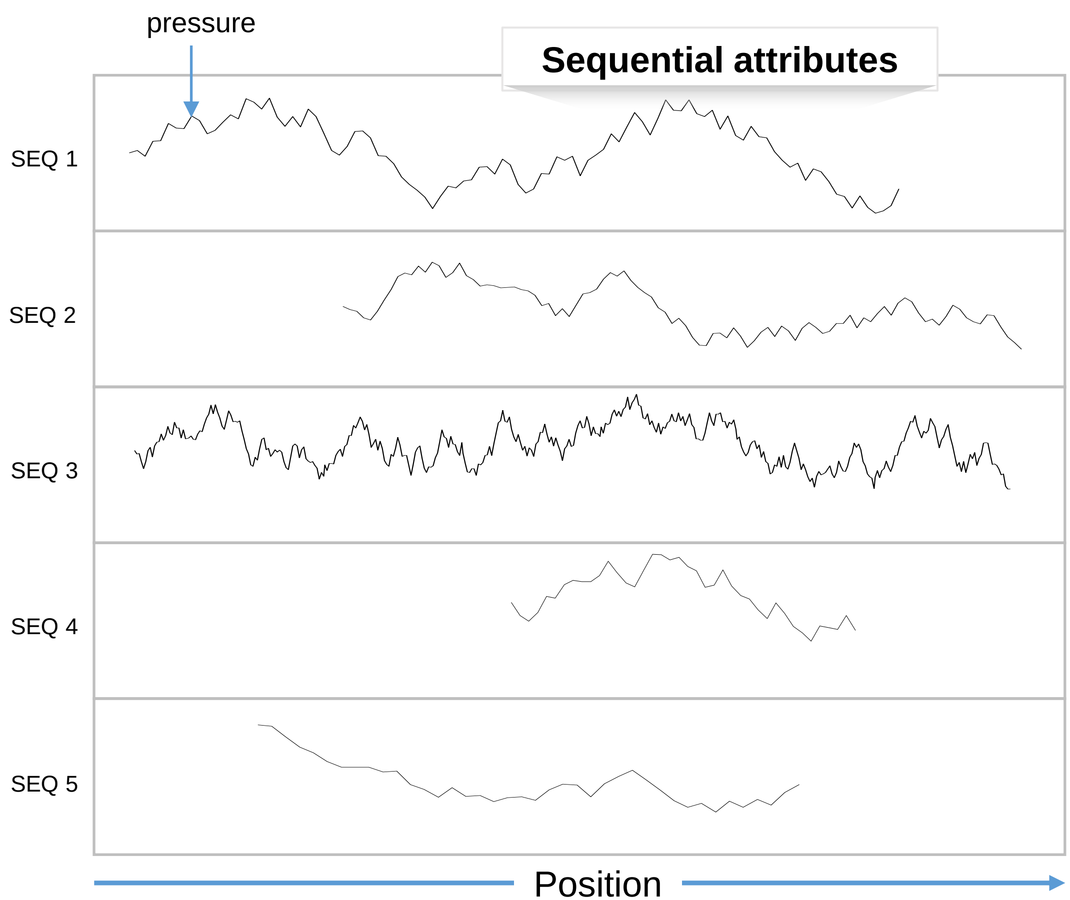
varying sample rates

within and **across** data series

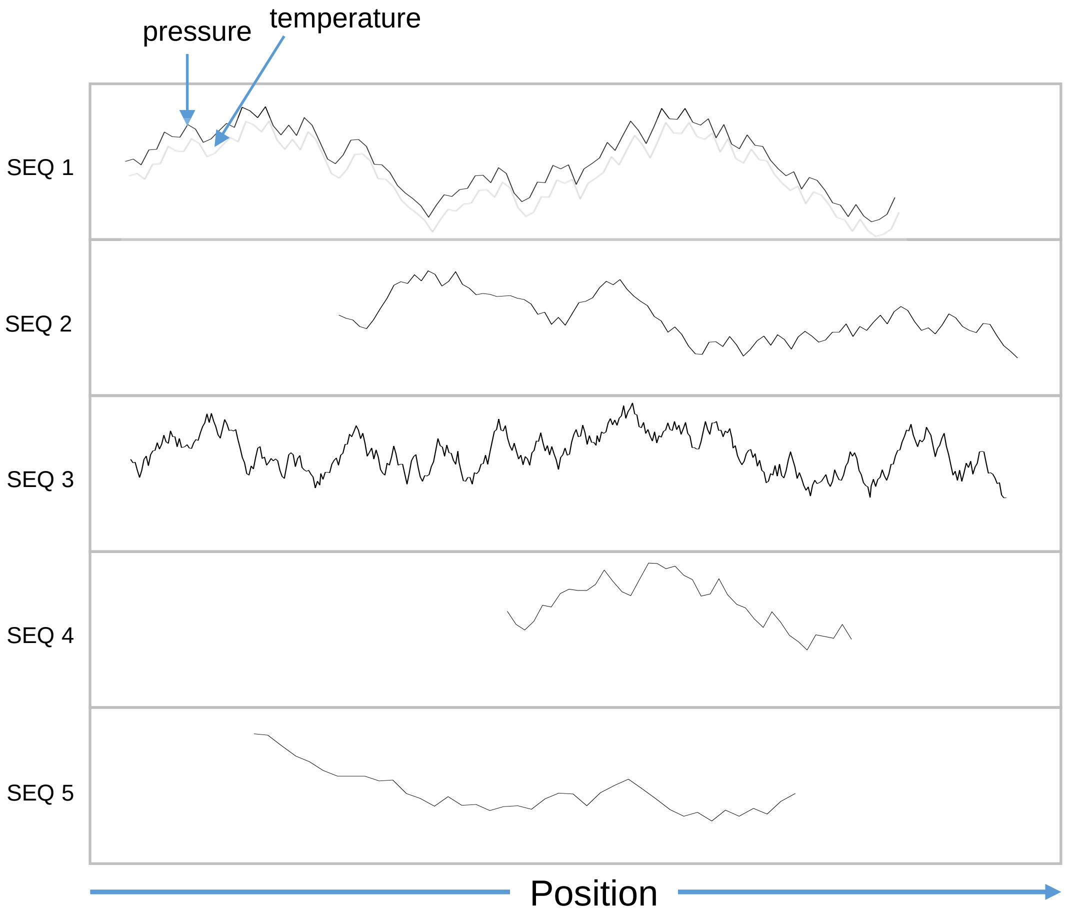
Time-Series Characteristics



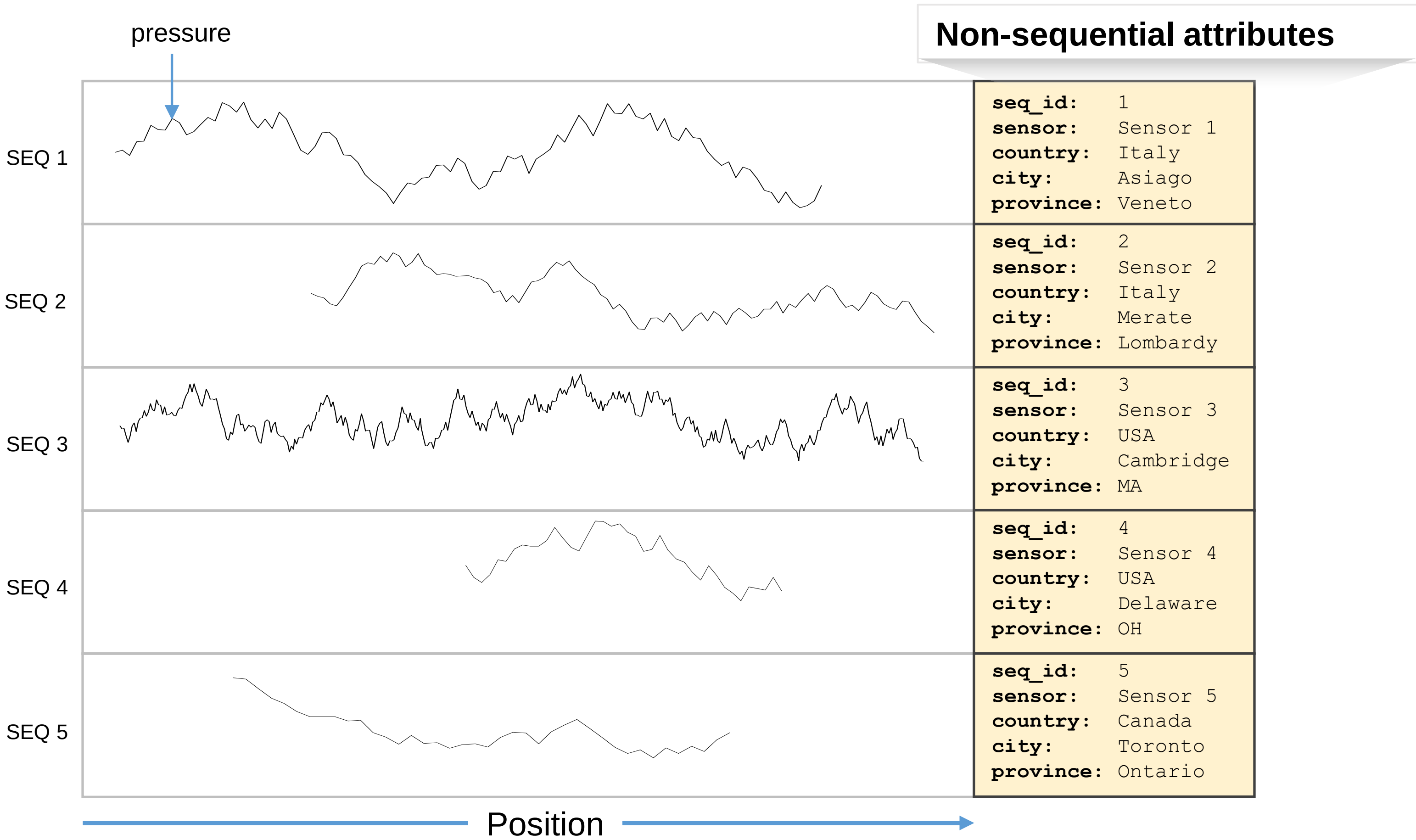
Time-Series Characteristics



Time-Series Characteristics



Time-Series Characteristics



Query Types

The Types of Queries

Simple

Selection-Projection-Transformation

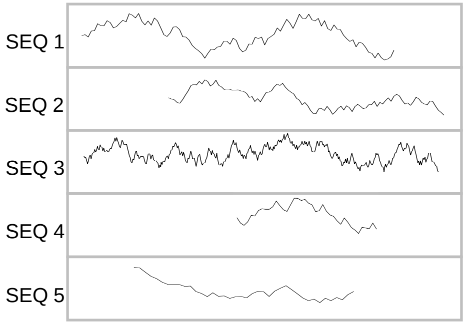
Complex

Analytical/Mining Queries

Query Types

Simple

Selection-Projection-Transformation



Query Types

Simple

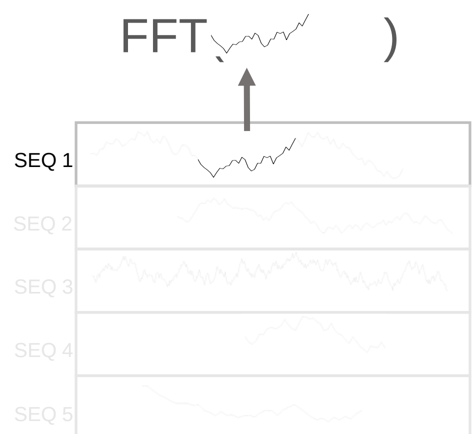
Selection-Projection-Transformation

Select some series → **Project** some points → **Apply** a **function** on them

Query Type 1: Find all points of a subset of data series
*e.g., Bring me the **whole history** of "pressure" for "Sensor 1"*

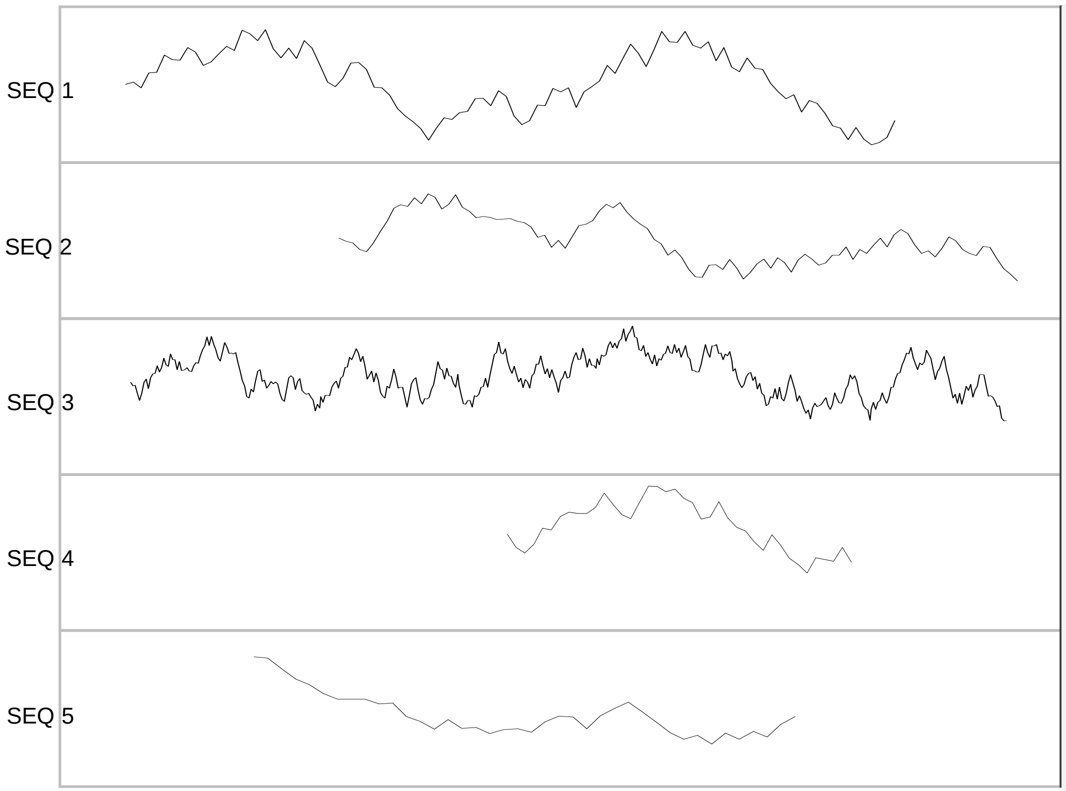
Query Type 2: Look at the points at a subset of the positions
*e.g., Compute the **average** pressure for all sensors for the range of positions that cover the 2nd to the 12th of March.*

Query Type 3: Look at a subset of points **based on a value**
*e.g., Bring me **all pressure** values above a **threshold***



Storage

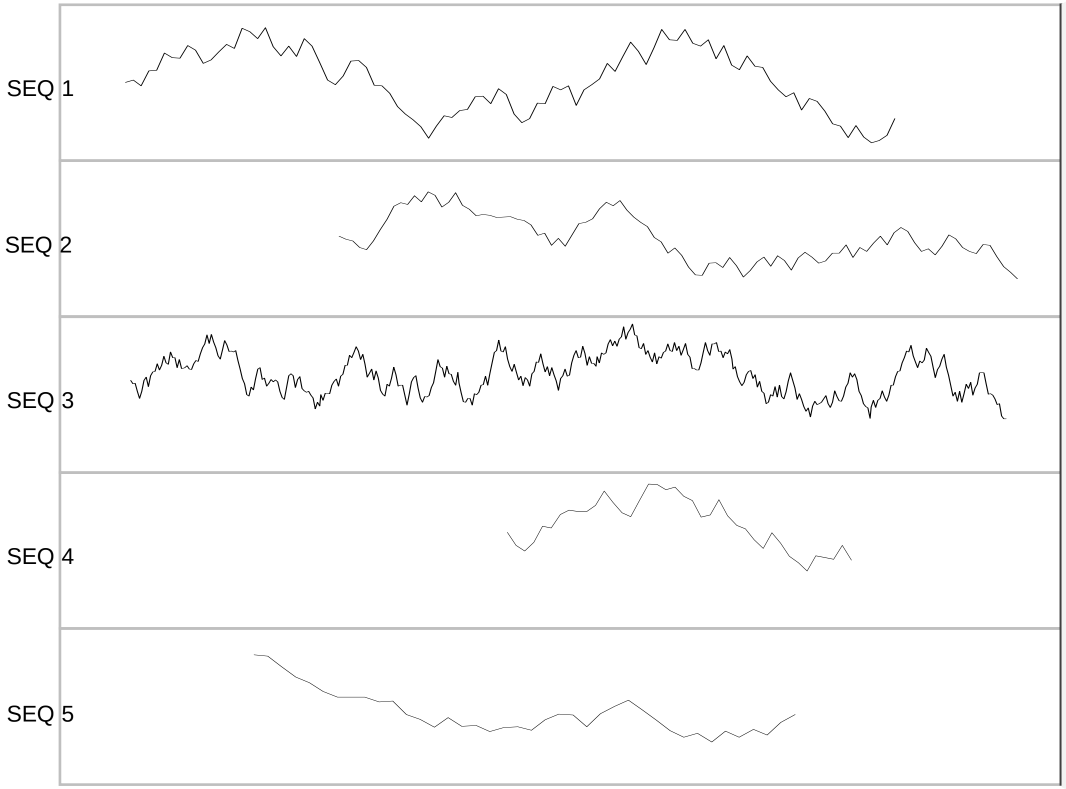
Storing meta-data



Meta-Data tuples

SEQID	COUNTRY	CITY	PROVINCE
1	Italy	Asiago	Veneto
2	Italy	Merate	Lombardy
3	USA	Cambridge	MA
n			

Storage



Meta-Data tuples

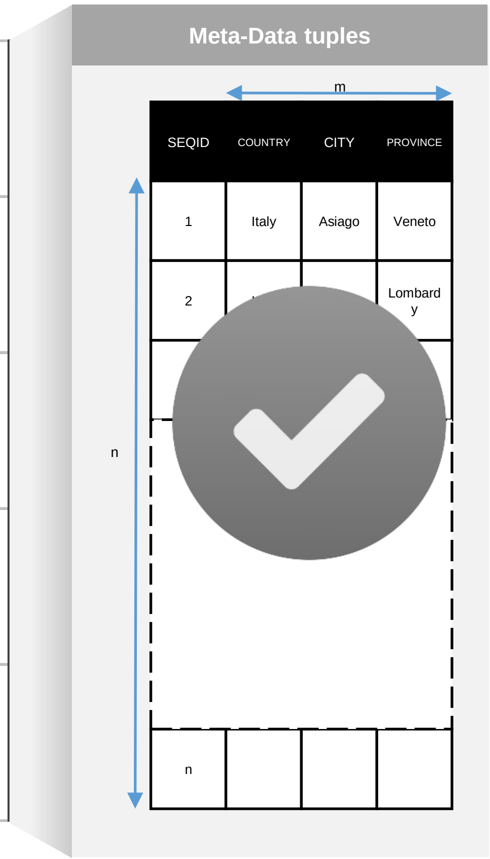
m

n

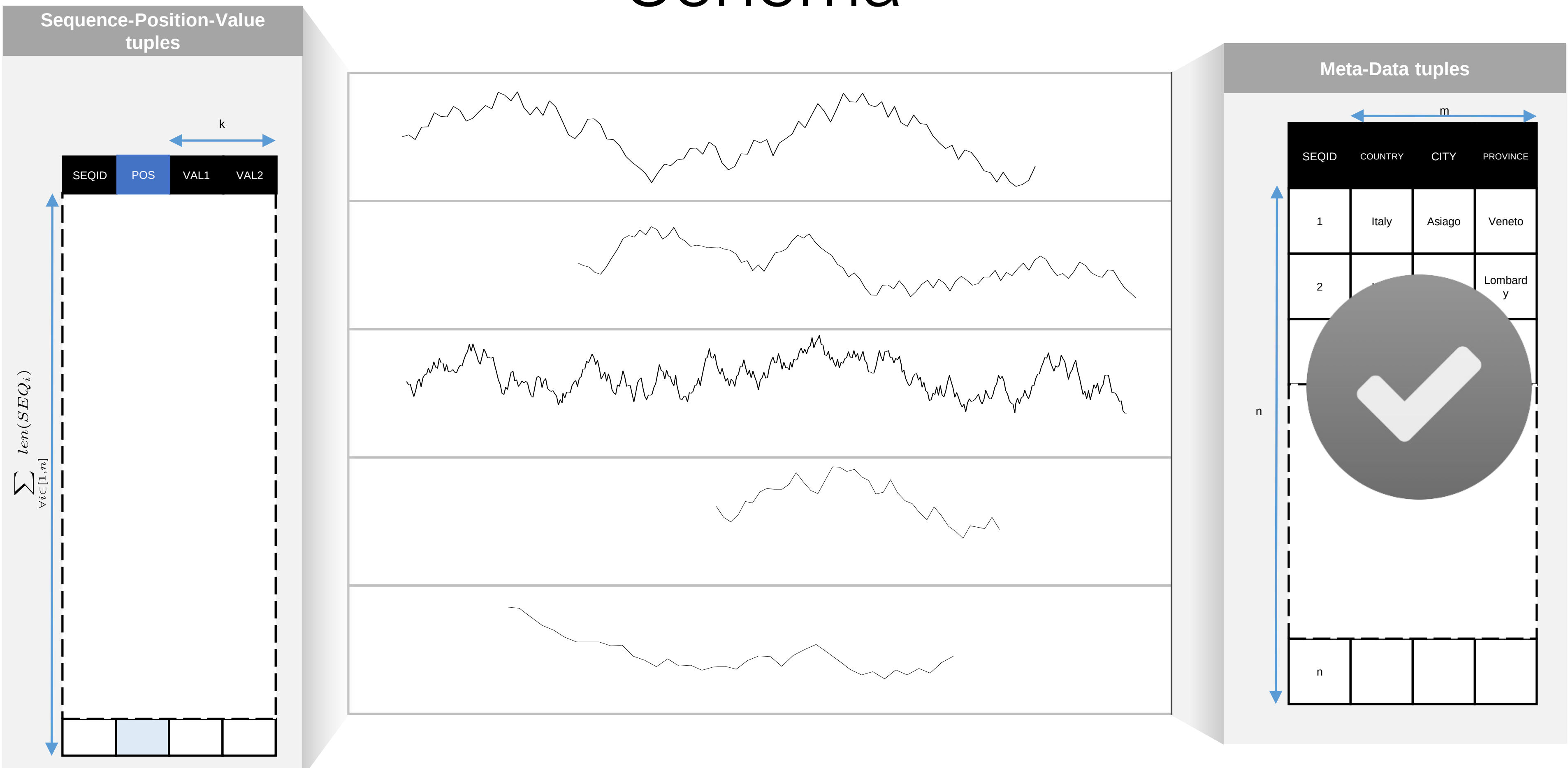
SEQID	COUNTRY	CITY	PROVINCE
1	Italy	Asiago	Veneto
2	Italy	Merate	Lombard y
3	USA	Cambridg e	MA
n			

Storage

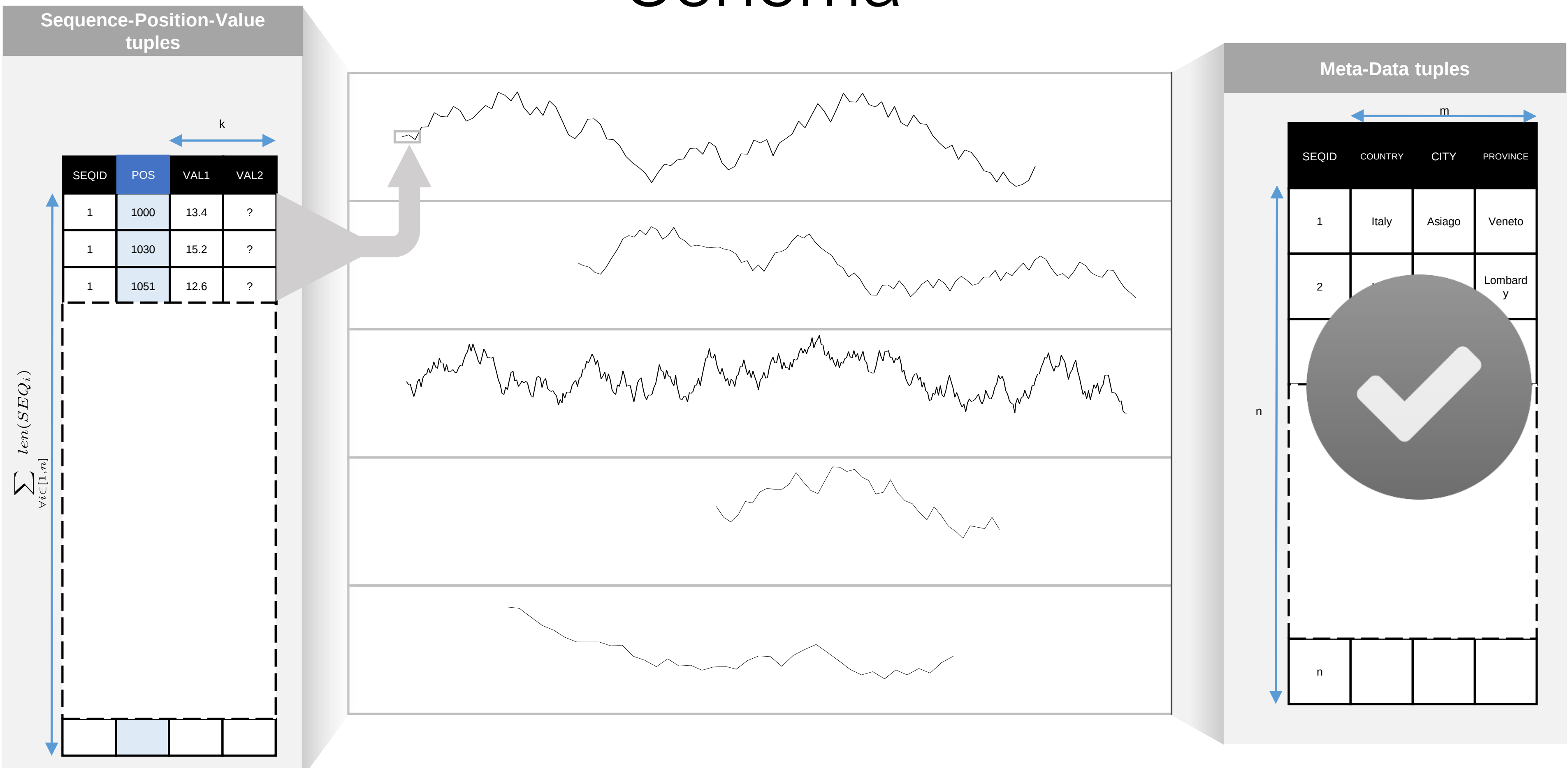
Storing meta-data



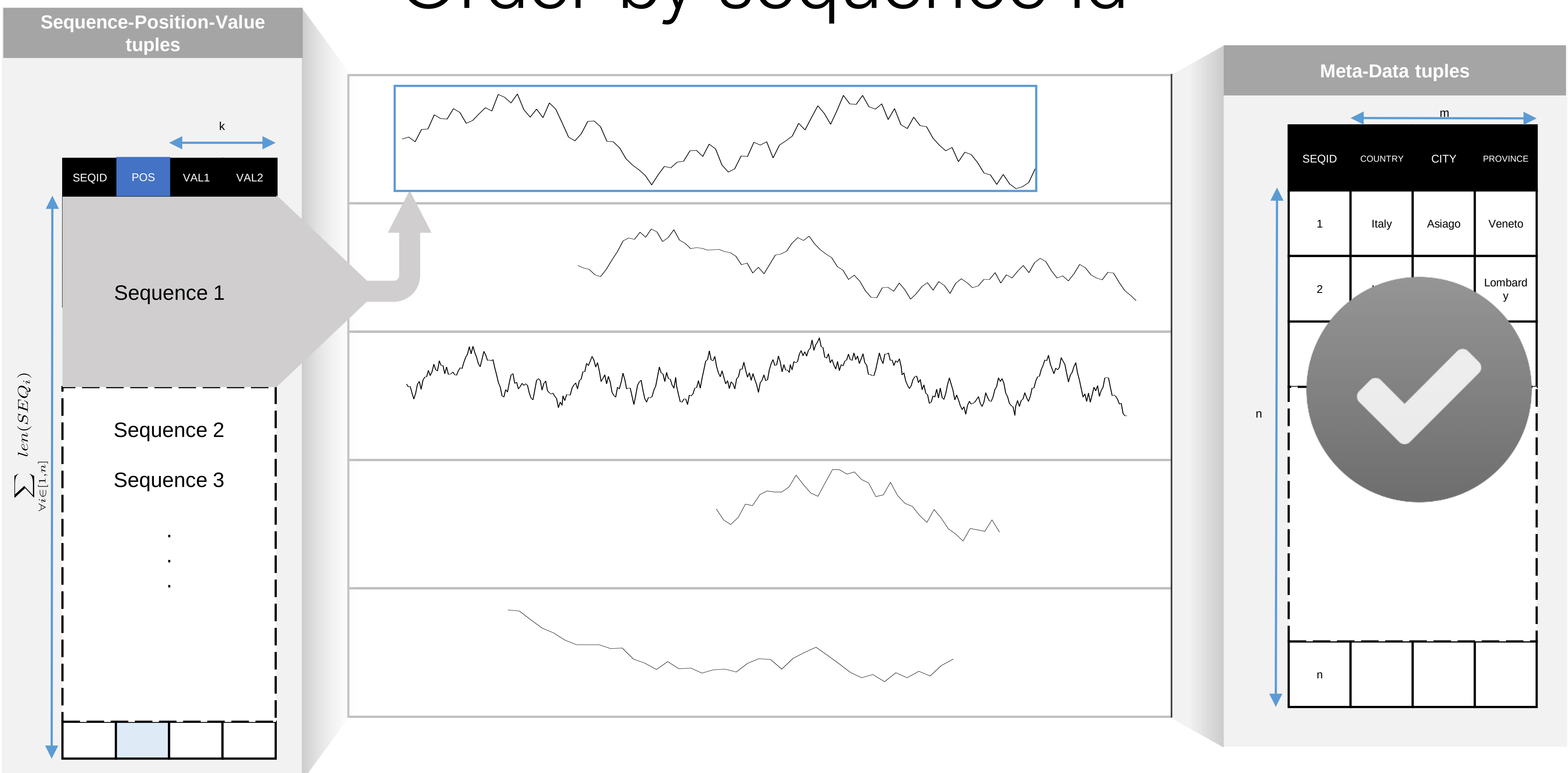
Schema


Sequence-Position-Value tuples
k
SEQID
POS
VAL1
VAL2
$\sum_{\forall i \in [1,n]} len(SEQ_i)$
Meta-Data tuples
m
SEQID
COUNTRY
CITY
PROVINCE
1
Italy
Asiago
Veneto
2
Lombard
y
n
n

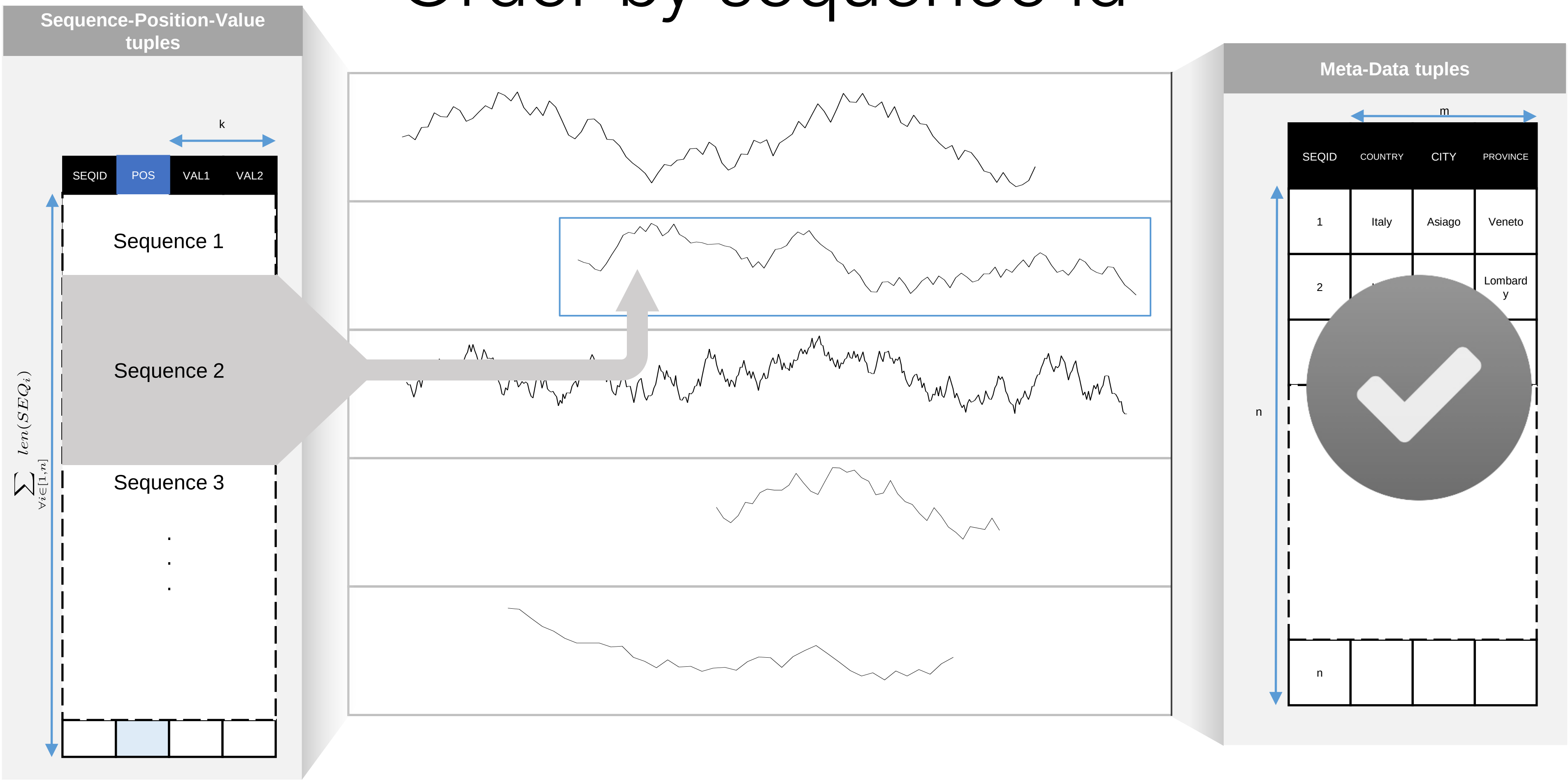
Schema


Sequence-Position-Value tuples
k
SEQID
POS
VAL1
VAL2
1
1000
13.4
?
1
1030
15.2
?
1
1051
12.6
?
$\sum_{\forall i \in [1,n]} len(SEQ_i)$
Meta-Data tuples
m
SEQID
COUNTRY
CITY
PROVINCE
1
Italy
Asiago
Veneto
2
Lombard
y
n
n

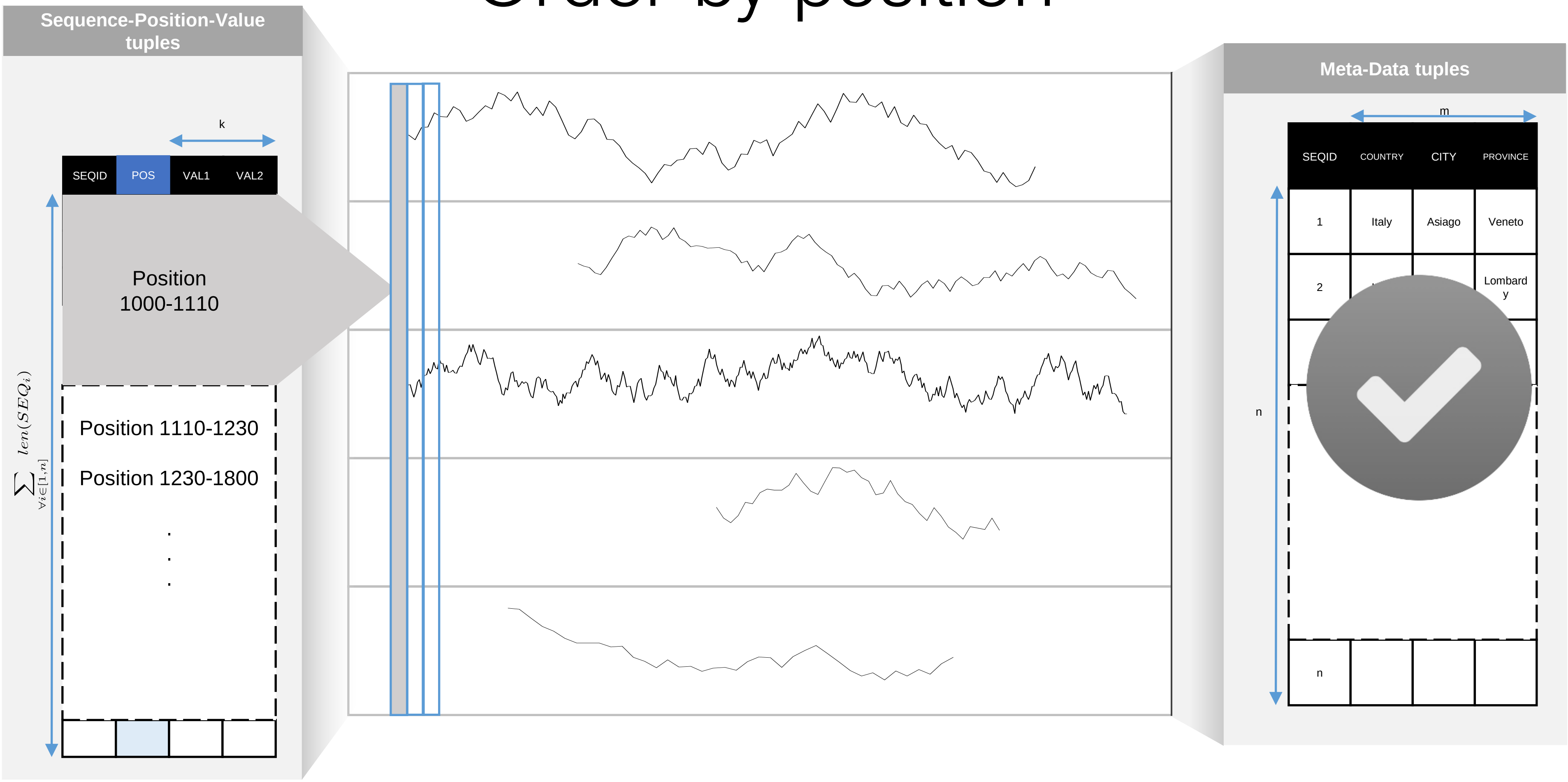
Order by sequence id



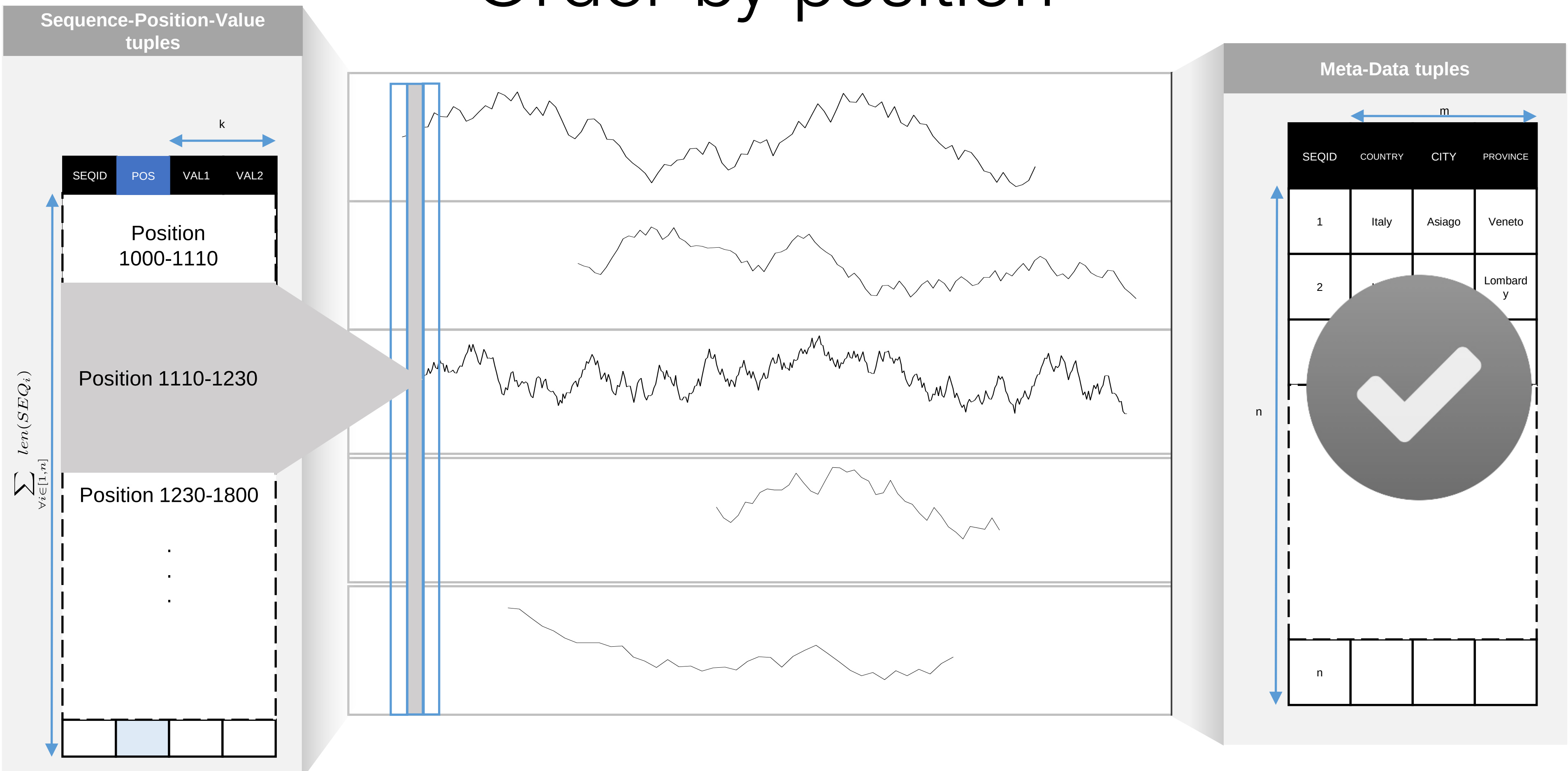
Order by sequence id



Order by position



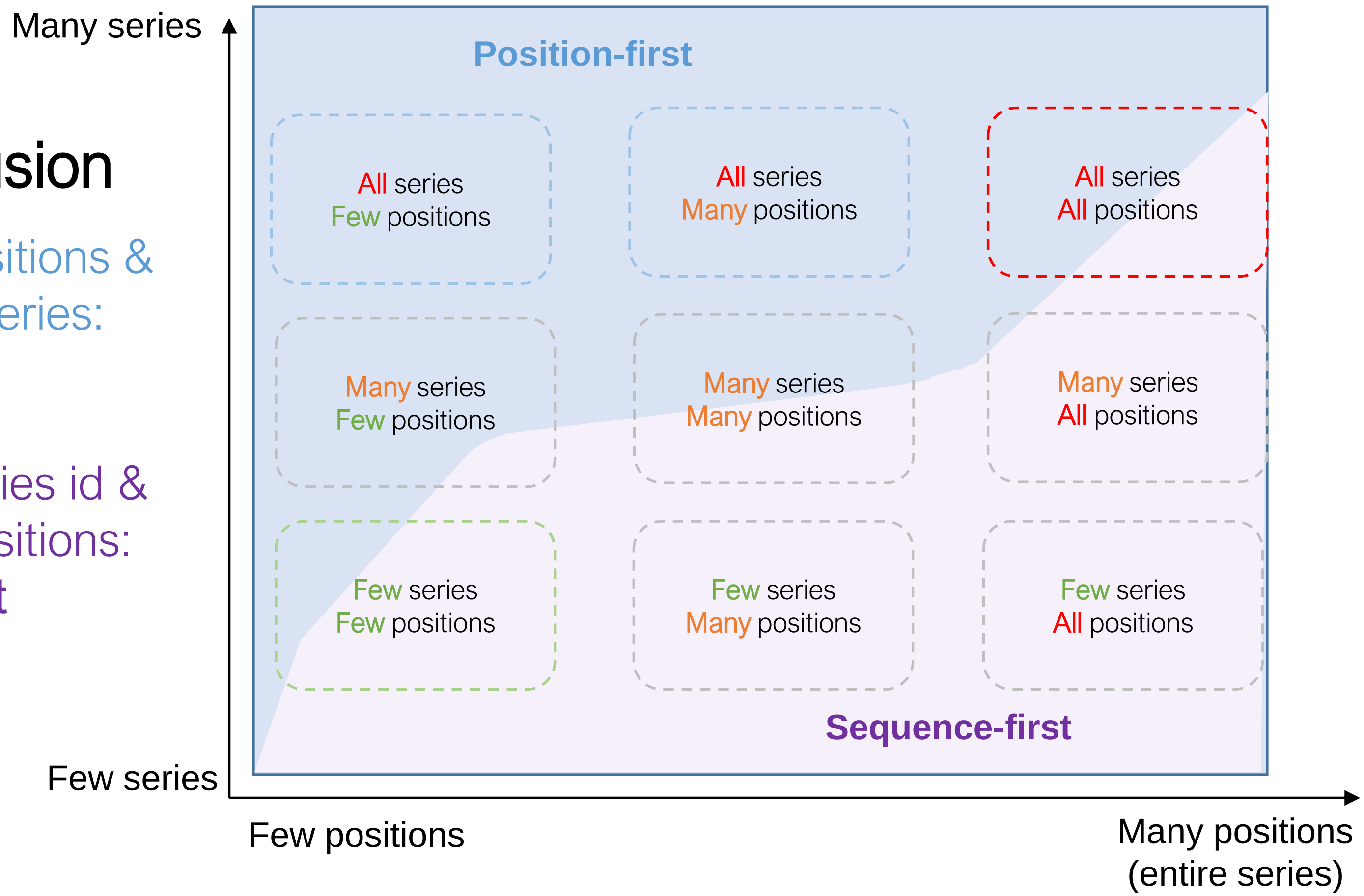
Order by position



Simple Conclusion

Heavy filtering on positions & Accessing lots of series: **position-first**

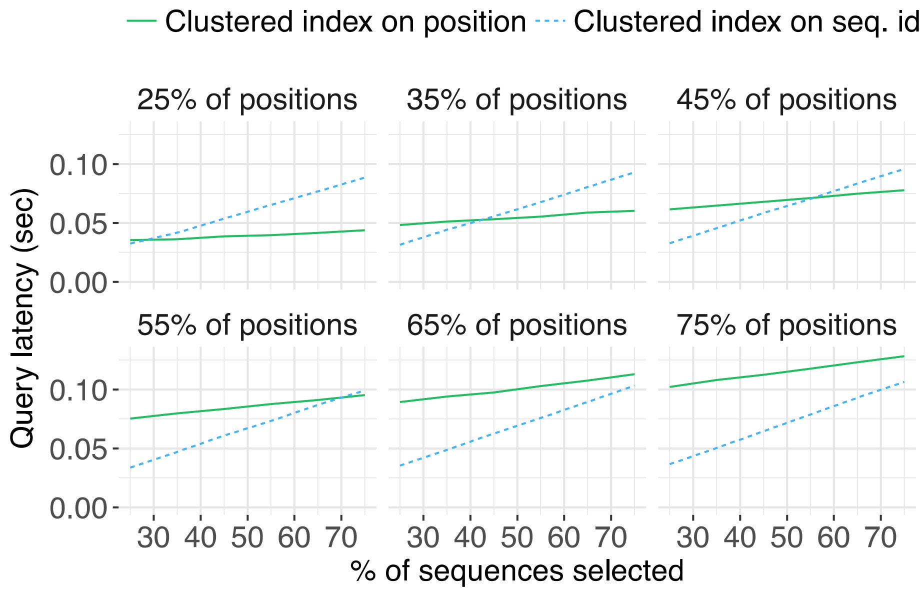
Heavy filtering on series id & accessing lots of positions: **sequence-first**



Simple Conclusion

Heavy filtering on positions &
Accessing lots of series:
position-first

Heavy filtering on series id &
accessing lots of positions:
sequence-first

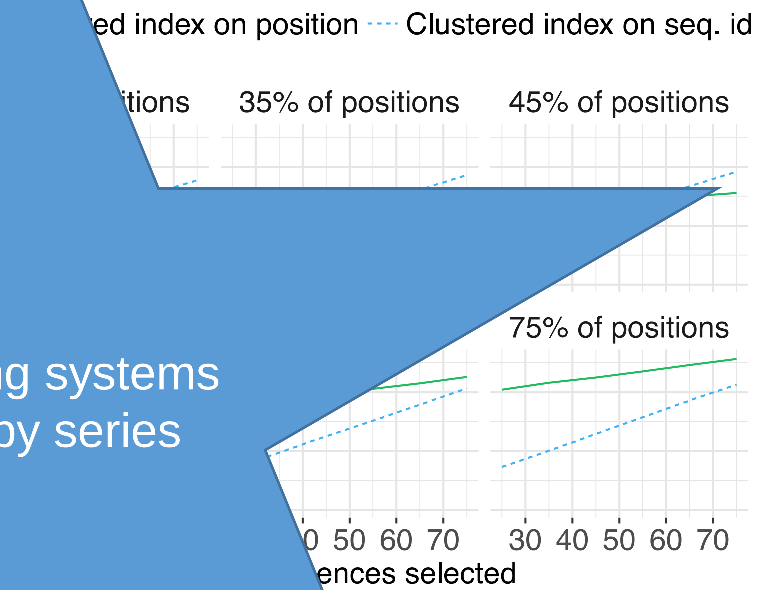


*DBMS-X

Simple Conclusion

Heavy filtering on positions &
Accessing
position

Heavy filtering on series id
accessing lots of positions:
sequence-first



Most existing systems sort data by series

*DBMS-X

Query Types

Simple

Selection-Projection-Transformation

Select some series → **Project** some points → **Apply** a **function** on them

Query Type 1: Find all points of a subset of data series
*e.g., Bring me the **whole history** of "pressure" for "Sensor 1"*

Query Type 2: Look at the points at a subset of the positions
*e.g., Compute the **average** pressure for all sensors for the range of positions that cover the 2^{nd} to the 12^{th} of March.*

Query Type 3: Look at a subset of points **based on a value**
*e.g., Bring me **all pressure** values above a **threshold***

Query Types

Classic **1/n-dimensional indexes & layouts** for **point** and **range queries:**

Point get: Get seq id = 1
Range: Get positions 10 - 100

Simple

Selection-Projection-Transformation

Select some series → **Project** some points → **Apply** a **function** on them

Query Type 1: Find all points of a subset of data series
*e.g., Bring me the **whole history** of "pressure" for "Sensor 1"*

Query Type 2: Look at the points at a subset of the positions
*e.g., Compute the **average** pressure for all sensors for the range of positions that cover the 2^{nd} to the 12^{th} of March.*

Query Type 3: Look at a subset of points **based on a value**
*e.g., Bring me **all pressure** values above a **threshold***

Query Types

Simple

Selection-Projection-Transformation

Select some series → **Project** some points → **Apply** a **function** on them

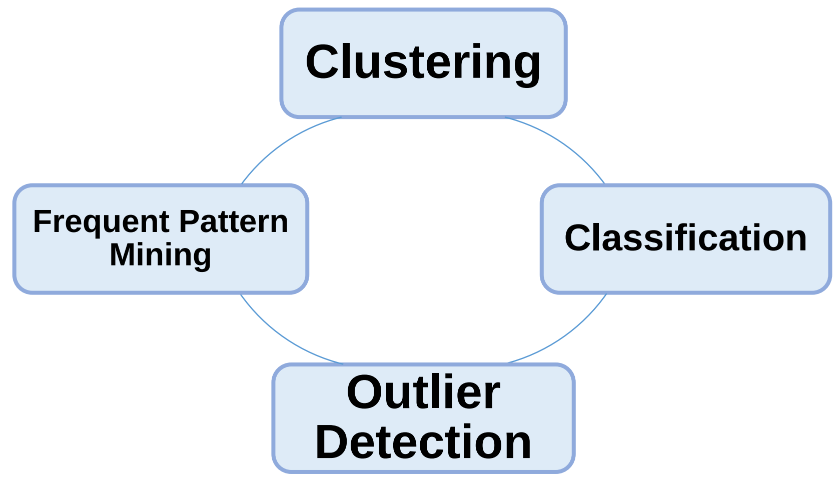
Query Type 1: Find all points of a subset of data series
e.g., Bring me the ***whole history*** *of "pressure" for "Sensor 1"*

Query Type 2: Look at the points at a subset of the positions
e.g., Compute the ***average*** *pressure for all sensors for the range of positions that cover the 2nd to the 12th of March.*

Query Type 3: Look at a subset of points **based on a value**
e.g., Bring me ***all pressure*** *values above a* ***threshold***

Complex

Analytical/Mining Queries



Query Types

Simple

Selection-Projection-Transformation

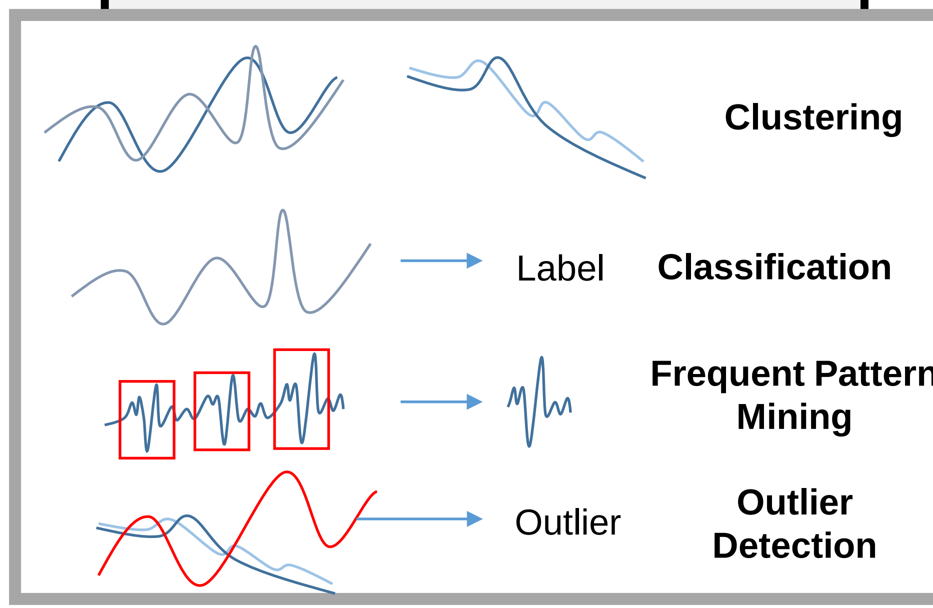
Select some series	**Project** some points	**Apply** a **function** on them

Query Type 1: Find all points of a subset of data series
*e.g., Bring me the **whole history** of "pressure" for "Sensor 1"*

Query Type 2: Look at the points at a subset of the positions
*e.g., Compute the **average** pressure for all sensors for the range of positions that cover the 2nd to the 12th of March.*

Query Type 3: Look at a subset of points **based on a value**
*e.g., Bring me **all pressure** values above a **threshold***

Complex



Query Types

Simple

Selection-Projection-Transformation

Select some series → **Project** some points → **Apply** a **function** on them

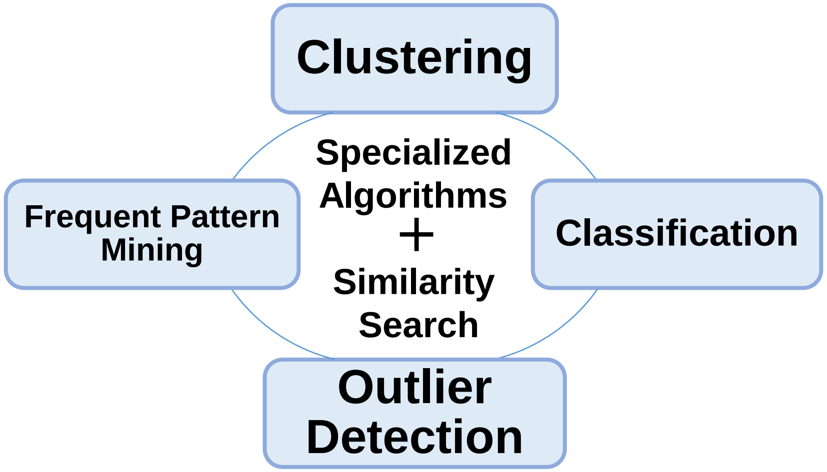
Query Type 1: Find all points of a subset of data series
e.g., Bring me the ***whole history*** *of "pressure" for "Sensor 1"*

Query Type 2: Look at the points at a subset of the positions
e.g., Compute the ***average*** *pressure for all sensors for the range of positions that cover the 2nd to the 12th of March.*

Query Type 3: Look at a subset of points **based on a value**
e.g., Bring me ***all pressure*** *values above a* ***threshold***

Complex

Analytical/Mining Queries



Query Types

Simple

Selection-Projection-Transformation

Select some series → **Project** some points → **Apply** a **function** on them

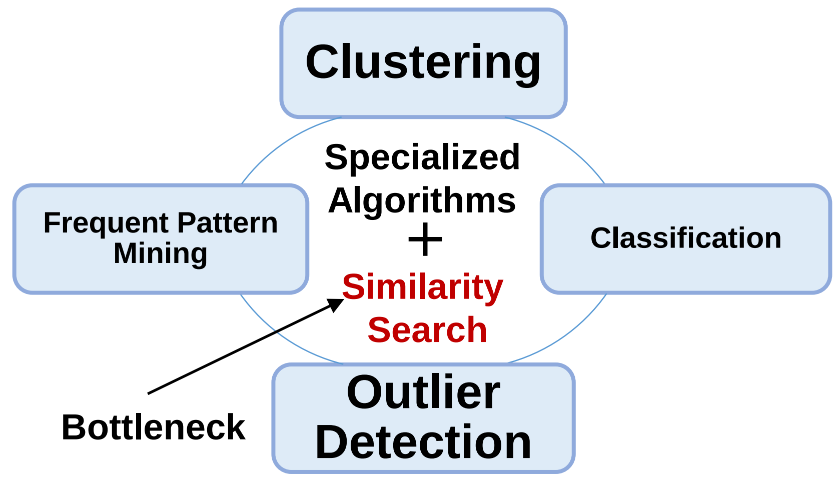
Query Type 1: Find all points of a subset of data series
e.g., Bring me the ***whole history*** *of "pressure" for "Sensor 1"*

Query Type 2: Look at the points at a subset of the positions
e.g., Compute the ***average*** *pressure for all sensors for the range of positions that cover the* 2^{nd} *to the* 12^{th} *of March.*

Query Type 3: Look at a subset of points based on a value
e.g., Bring me ***all pressure*** *values above a* ***threshold***

Complex

Analytical/Mining Queries



Time-Series Management Systems

a few more details on the popular systems:

- InfluxDB
- TimescaleDB

InfluxDB

- Storage Engine:
  - **Log Structured Merge Tree: LSM-Tree** variant that expects data to arrive ordered by time and partitions them by distinct sequence. It then stores each series contiguously.

- Schema:
  - Tags and fields. Tags are used to describe meta-data and fields are used to store quantities that change over time.

- Queries
  - It supports group by (only on tags), join (on timestamps and fields), selections, projections, and aggregations.
  - It also supports continuous queries

TimescaleDB

- **Storage:** Uses PostgreSQL as the backend.
  - It partitions time-series into multiple tables, forming a single virtual entity called a **hypertable**.
  - It allows for the **compression** of data, something that Postgres does not do by default.

- **Schema:** Tables are **normal Postgres tables**, where one has to specify a time column in order to create a hypertable.

- **Queries: Full SQL support**, with the addition of custom time-series functions.
  - **Custom time-series operators:** first, last, histogram, interpolation, time bucketing, gap filling, etc.
  - It also supports **continuous queries**

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 series applications

Massive Data Series Collections

Publications

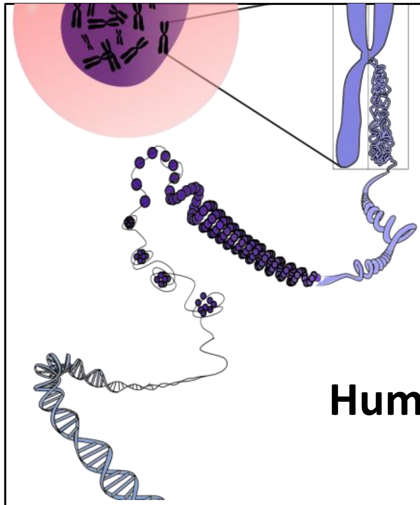
Palpanas-SIGREC'19

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

Outline

- sequence management system
- benchmarking
- general high-dimensional vectors
- deep learning

Publications

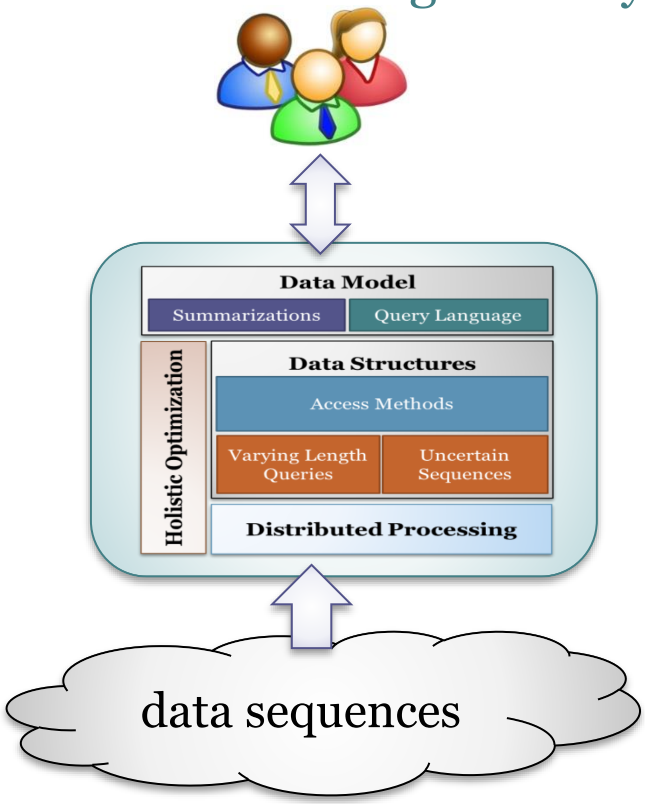
Zoumbatianos ICDE’18

Palpanas-HPCS’17

Palpanas-SIGREC’15

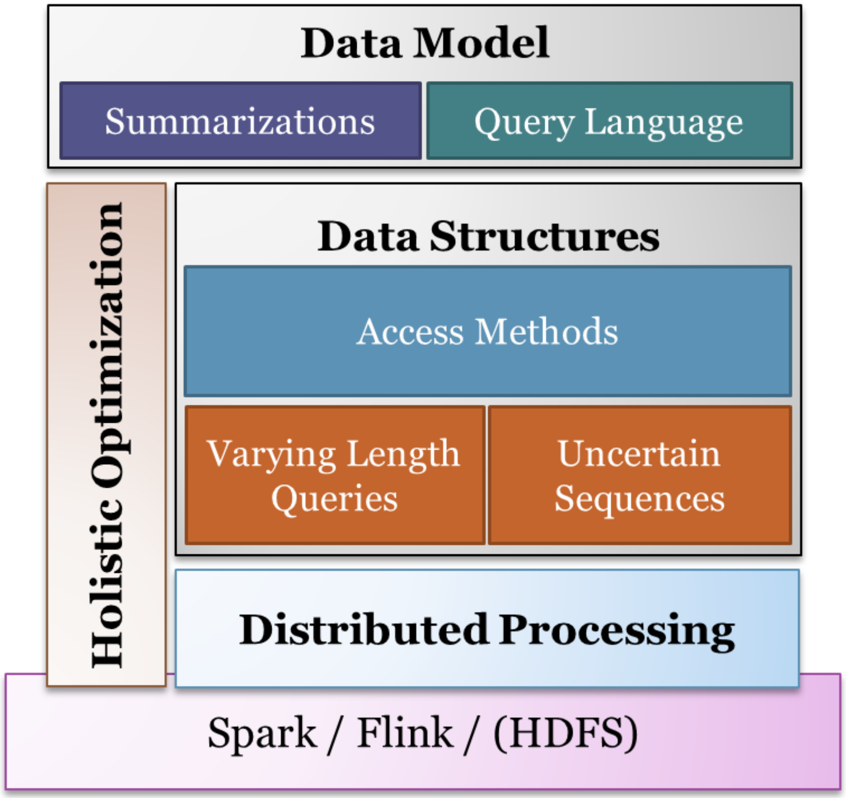
Management System

- Big Sequence Management System
  - general purpose data series management system



Management System

- Big Sequence Management System



Publications

- Zoumbatianos ICDE'18
- Palpanas-HPCS'17
- Palpanas-SIGREC'15

Management System

- Big Sequence Management System

	Dataset	Idx	Exact 100	Idx+ Exact 100	Idx+ Exact 10K	Exact Easy-20	Exact Hard-20
		Scenarios					
HDD	Small	A	D	S	D	D	D
HDD	Large	A	D	S	D	D	D
HDD	Astro	A	U	U	V	V	U
HDD	Deep1B	A	U	U	U	D	U
HDD	SALD	A	D	I	D	D	D
HDD	Seismic	A	D	S	D	D	U
SSD	Small	S	D	I	D	I	D
SSD	Large	S	D	I	D	I	D
SSD	Astro	I	V	V	V	V	V
SSD	Deep1B	S	I	I	V	I	U
SSD	SALD	S	I	I	I	I	V
SSD	Seismic	A	V	V	V	D	V

A: ADS, D: DSTree, I: iSAX2+
S: SFA, U: UCR-Suite, V: VA+file

Publications

- Zoumbatianos ICDE'18
- Palpanas-HPCS'17
- Palpanas-SIGREC'15
- Echihabi-PVLDB'18

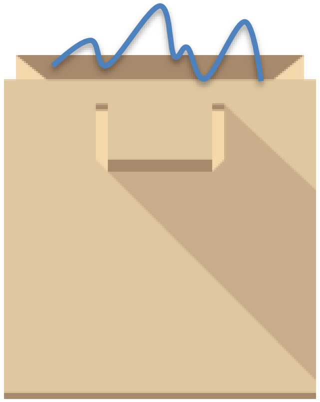
Outline

- sequence management system
- benchmarking
- general high-dimensional vectors
- deep learning

Previous Studies

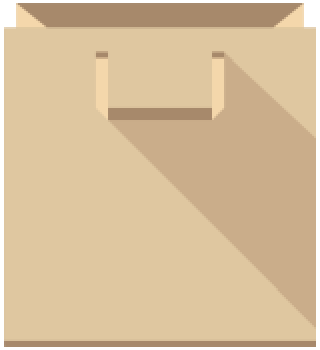
evaluate **performance** of **indexing methods** using **random queries**

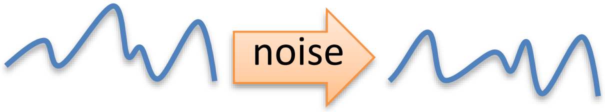
- chosen from the data (with/without noise)



Previous Studies

With or without noise





Problem with Random Queries

No control on their ***characteristics***

We **cannot properly evaluate** summarizations and indexes

We need queries that cover the entire range from easy to hard

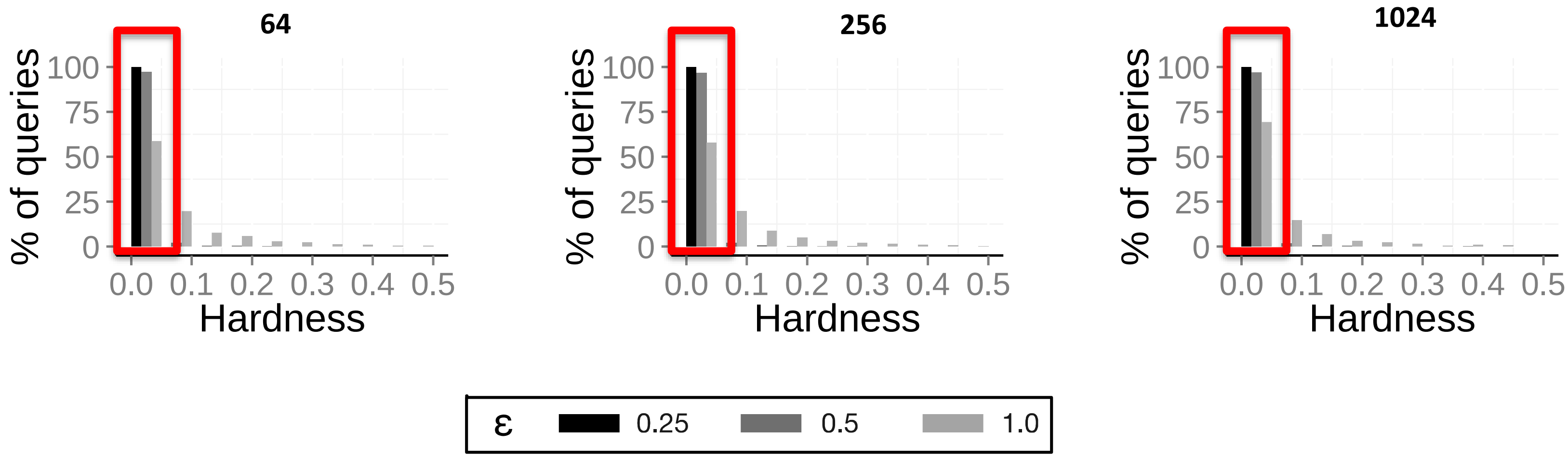
Zoumbatianos KDD '15

Zoumbatianos TKDE '18

Previous Workloads

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

DNA



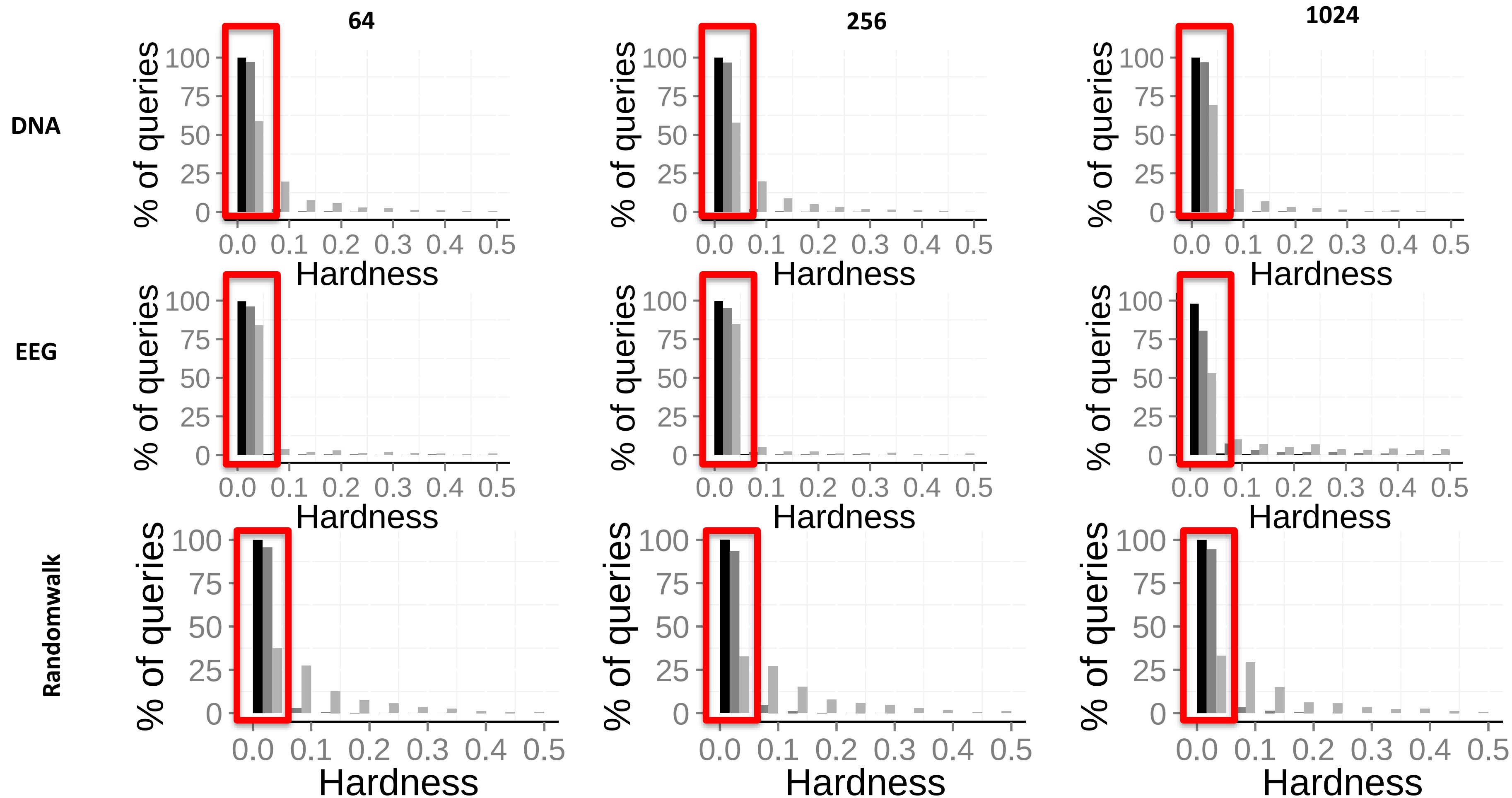
Publications

Zoumbatianos KDD '15

Zoumbatianos TKDE '18

Previous Workloads

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



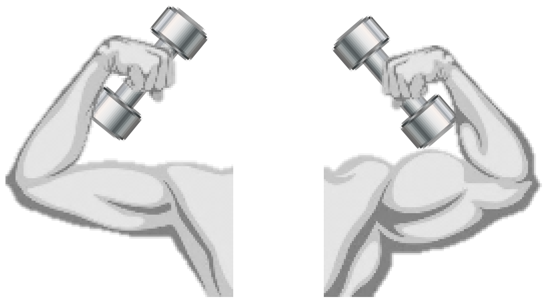
Publications

Zoumbatianos KDD ‘15

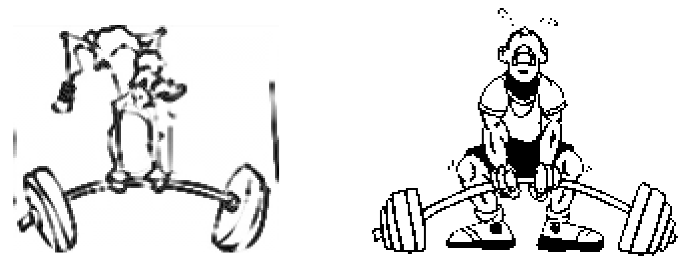
Zoumbatianos TKDE ‘18

Benchmark Workloads

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

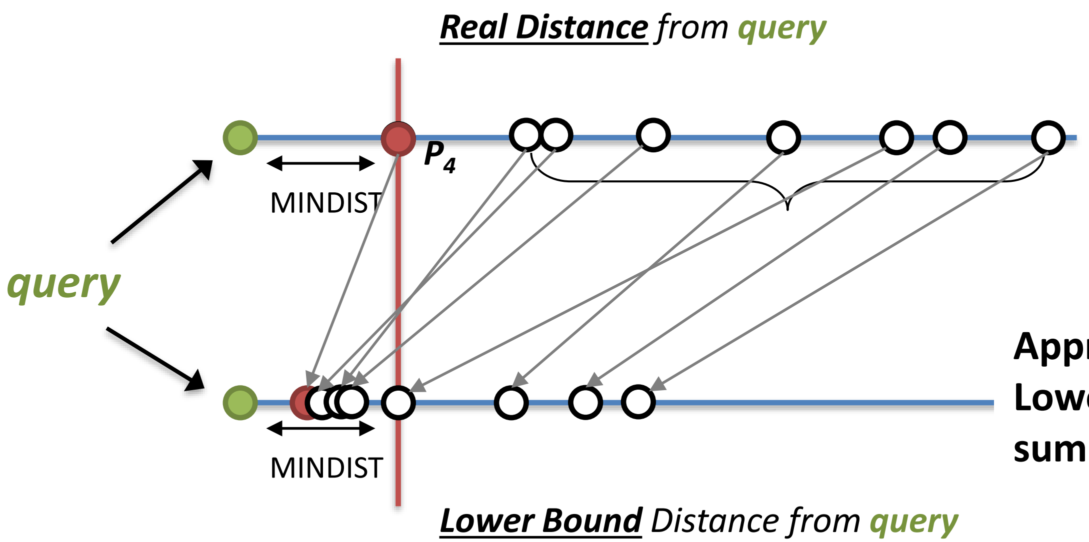


If all queries are **hard**
all indexes look **bad**



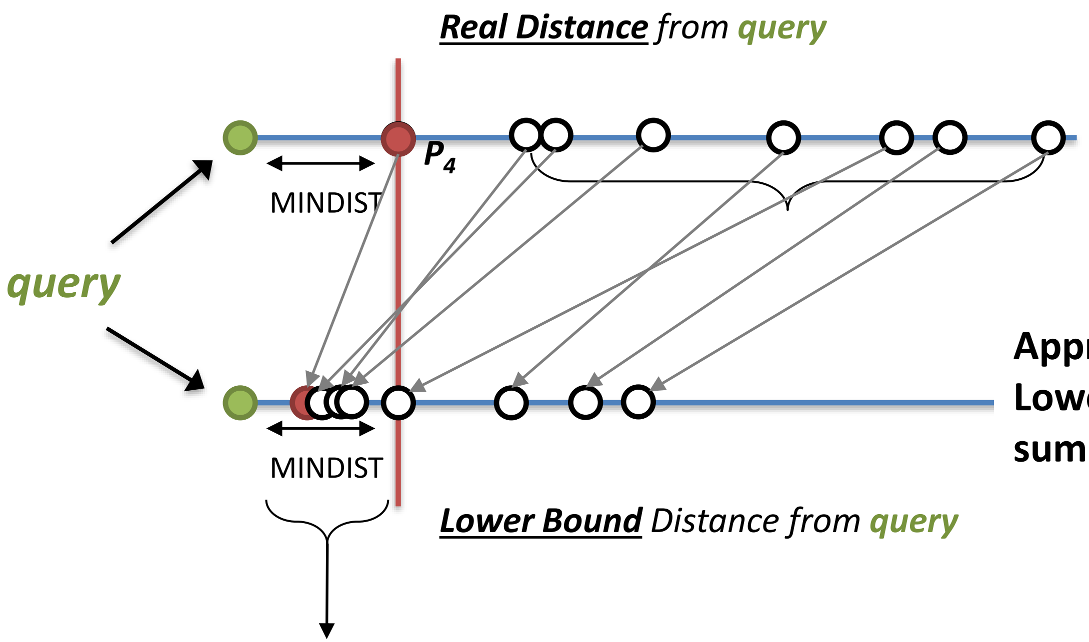
need **methods** for **generating** queries of **varying hardness**

Characterizing Queries



Approximating distances using **Lower Bounding functions** on **summarizations**.

Characterizing Queries



Approximating distances using **Lower Bounding functions** on **summarizations.**

Points with **lower bounds** below **MINDIST cannot be pruned**

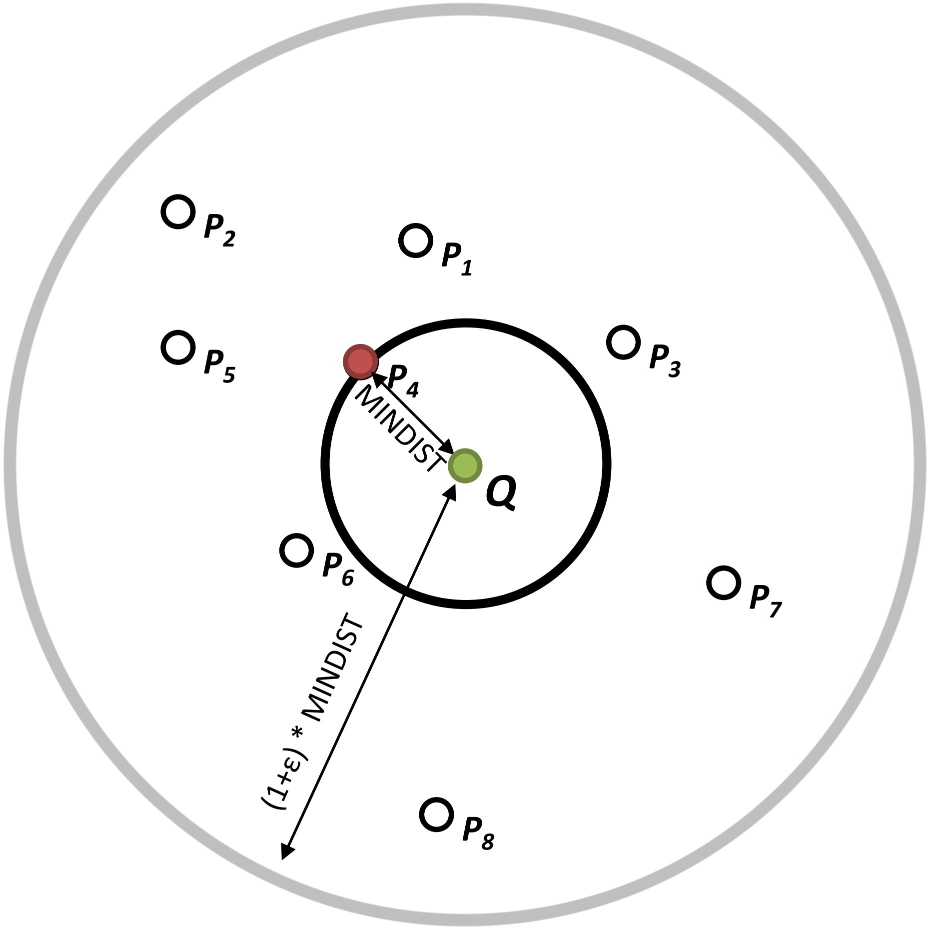
Must be **read from disk** in order **to dismiss false positives**

Publications

Zoumbatianos KDD '15

Zoumbatianos TKDE '18

Hardness

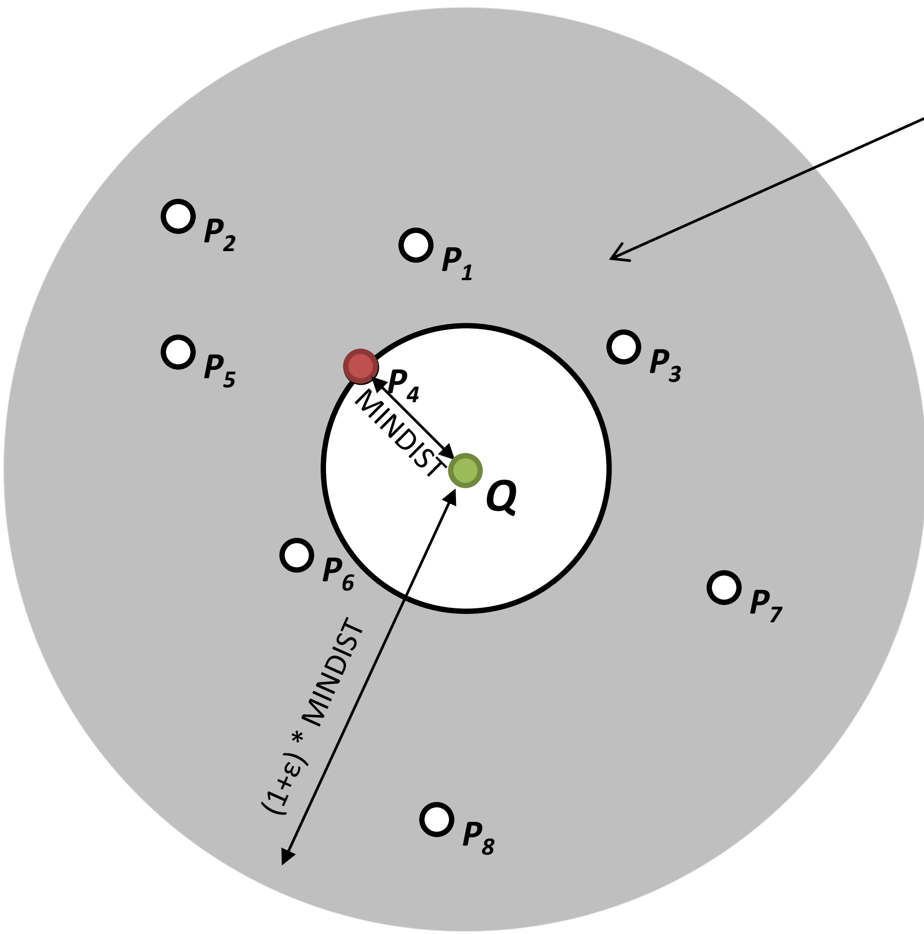


Publications

Zoumbatianos KDD ‘15

Zoumbatianos TKDE ‘18

Hardness



We define an ε-area

*(1+*ε) * MINDIST

Hardness

$$\frac{\text{\# of data –series in ε-area}}{\text{\# all data series}}$$

Publications

Zoumbatianos KDD ‘15

Zoumbatianos TKDE ‘18

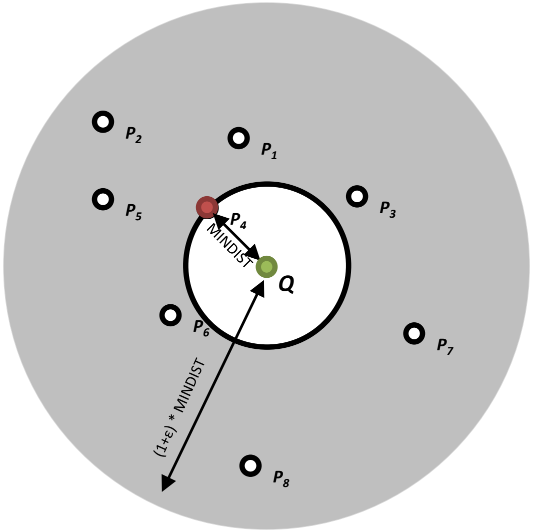
Hardness

Significance

Queries with **larger hardness** tend to have a **larger minimum effort**

data series **close to the answer**

higher chance that their **lower bounding distance** will be **less than MINDIST**



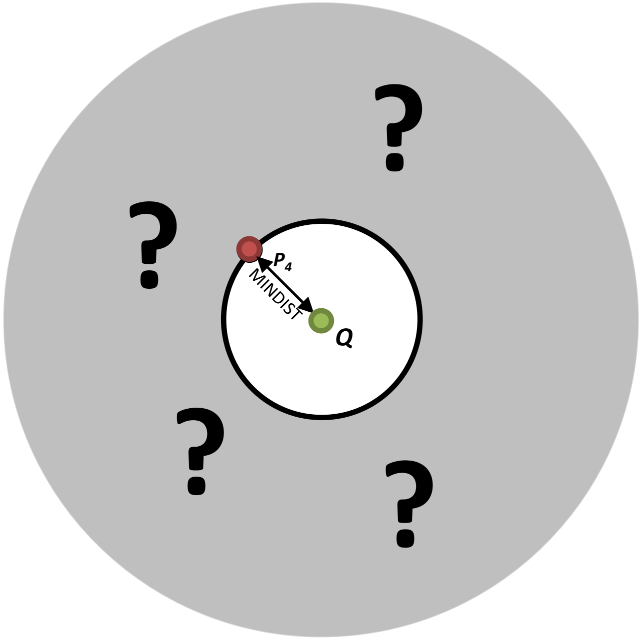
Publications

Zoumbatianos KDD '15

Zoumbatianos TKDE '18

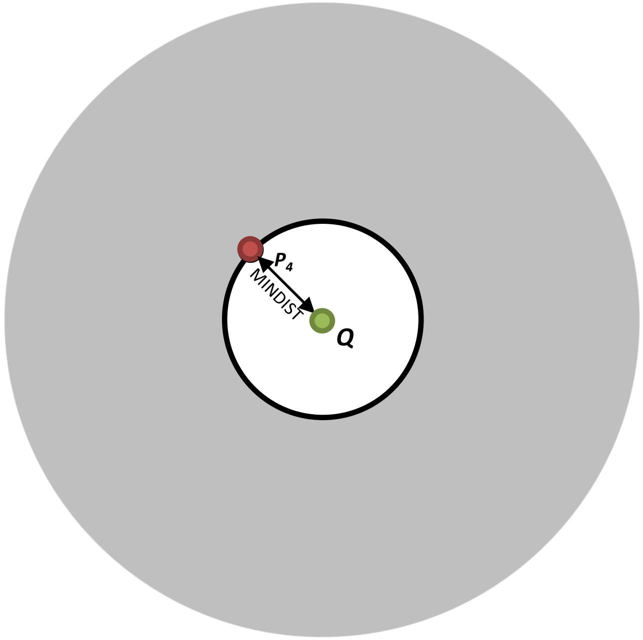
Workload Generation

Random queries have **random hardness**



Workload Generation

Can we **generate** queries of **controlled hardness**?



Publications

Zoumbatianos KDD ‘15

Zoumbatianos TKDE ‘18

3 Step Process

Sample

Random queries from a given dataset

Filter

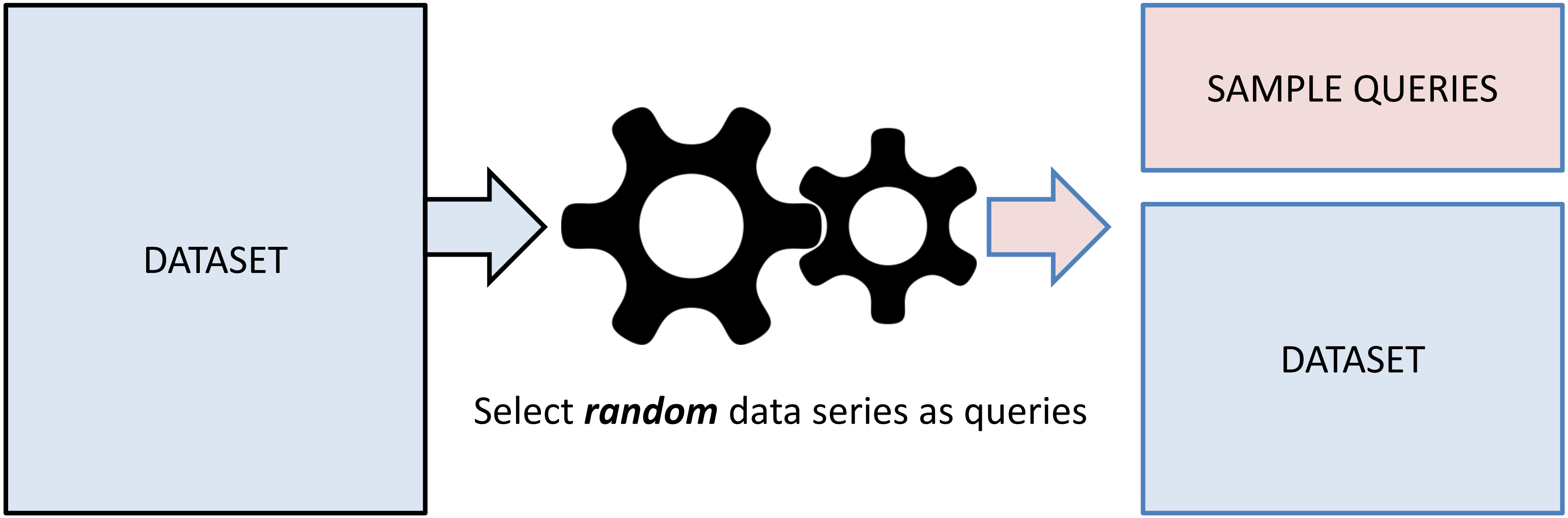
Subset of queries that have “independent” ε-areas

“Densify”

ε-areas to reach given hardness

Publications
Zoumbatianos KDD '15
Zoumbatianos TKDE '18

Step 1: Sampling



Select ***random*** data series as queries

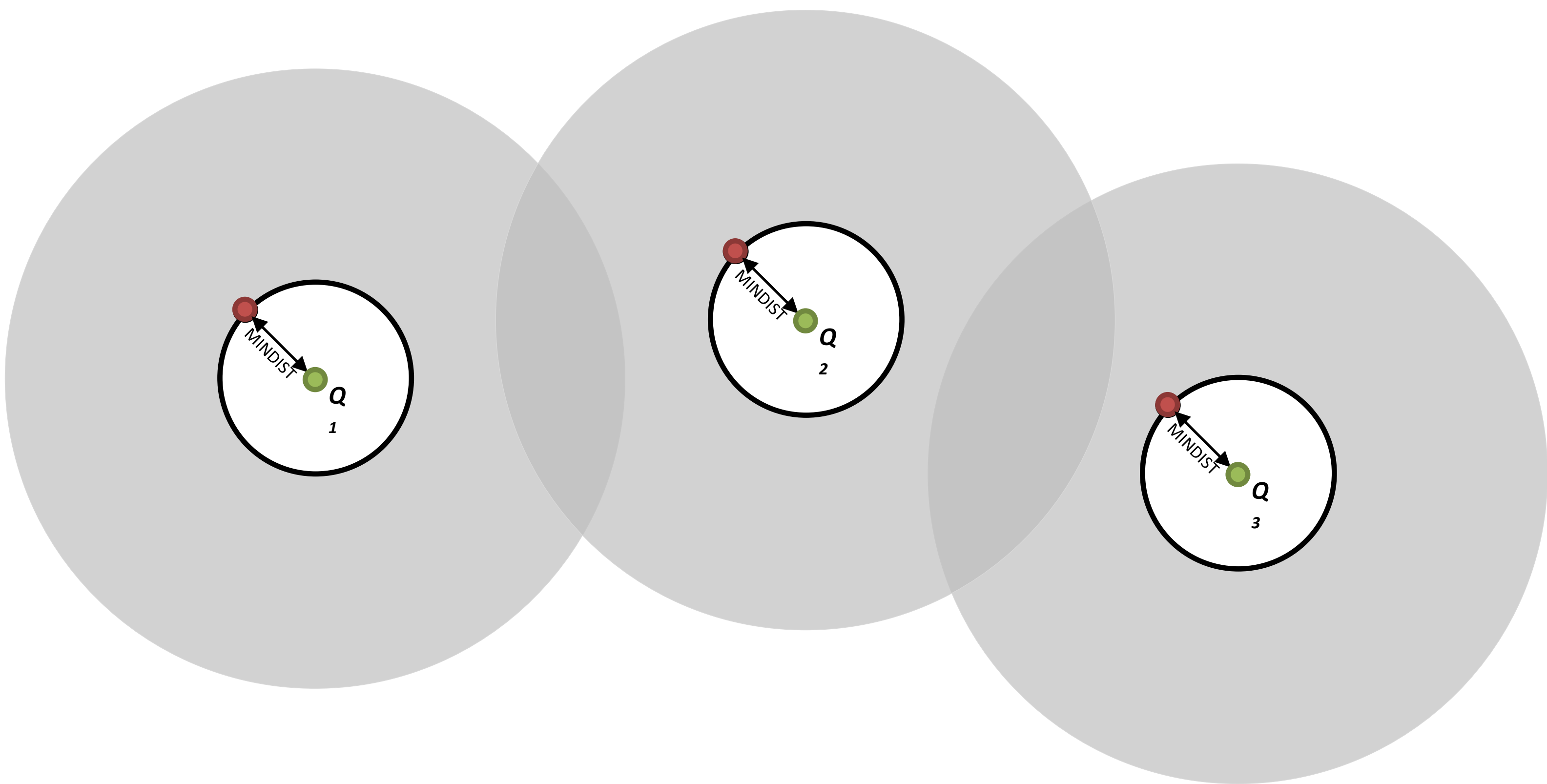
Step 2: Filtering-out "intersecting" queries

Publications

Zoumbatianos KDD '15

Zoumbatianos TKDE '18

We need to **independently** control the ε-areas



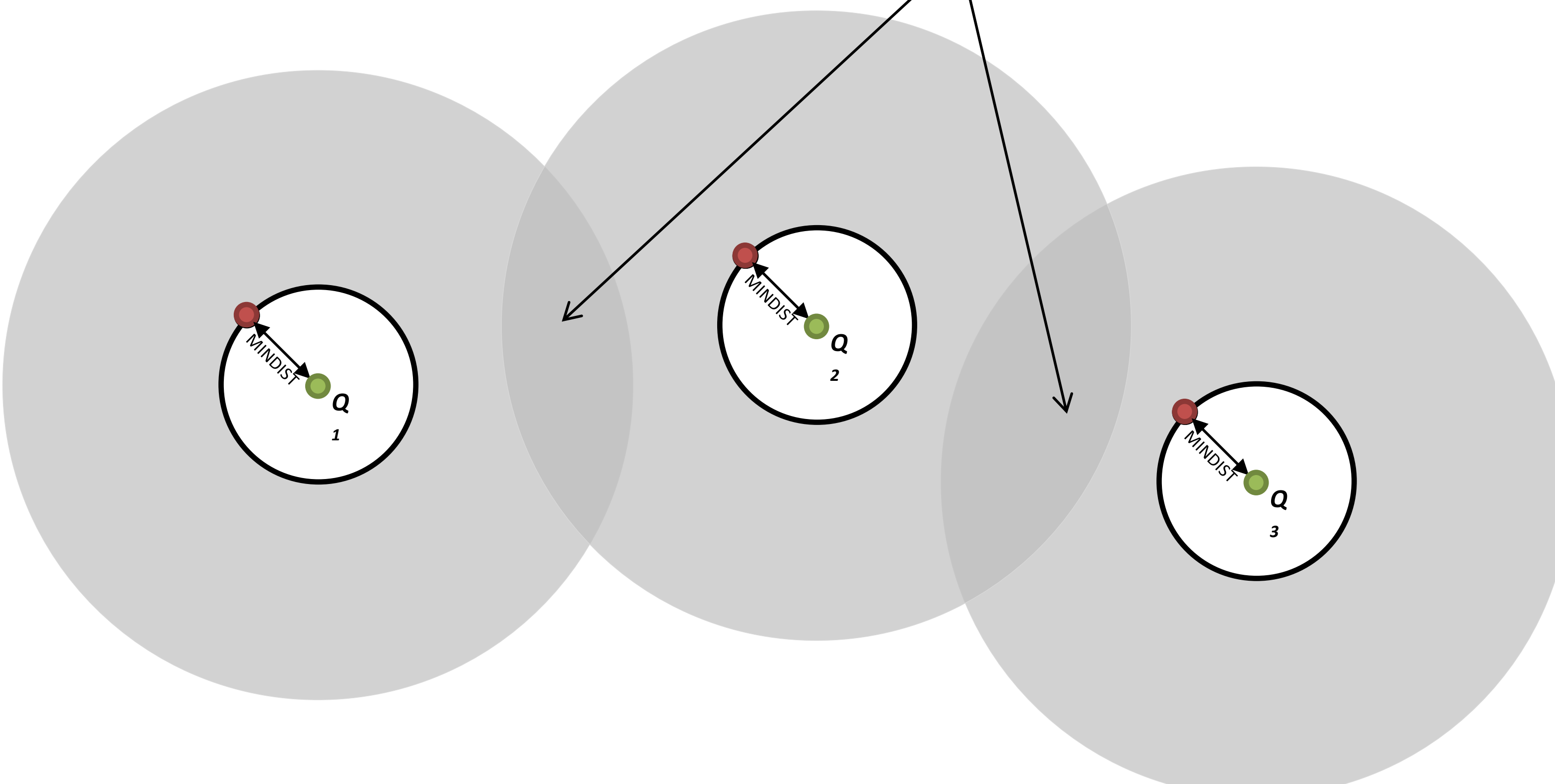
Step 2: Filtering-out "intersecting" queries

Publications

Zoumbatianos KDD '15

Zoumbatianos TKDE '18

The ε-areas of (Q_1, Q_2) and (Q_2, Q_3) cannot be **independently controlled** because they **intersect**



Publications

Zoumbatianos KDD ‘15

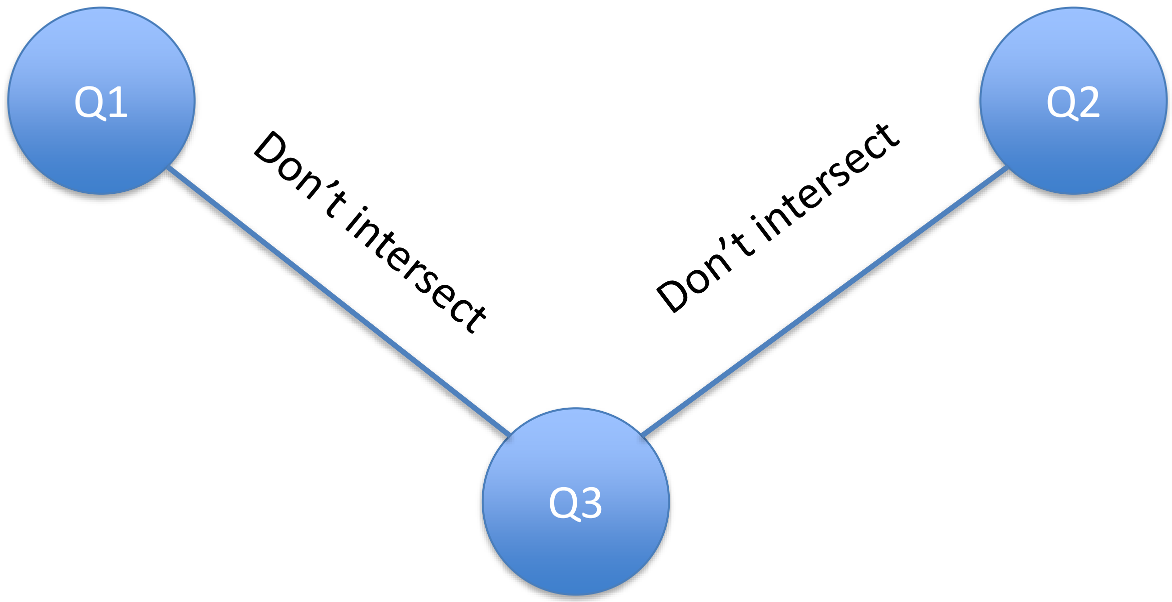
Zoumbatianos TKDE ‘18

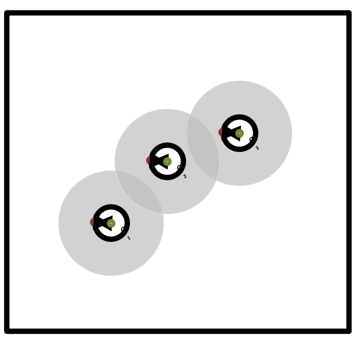
Step 2: Filtering-out “intersecting” queries

Can be formulated as a **graph problem**

1 node per query

1 edge for each pair that doesn’t intersect





Step 2: Filtering-out "intersecting" queries

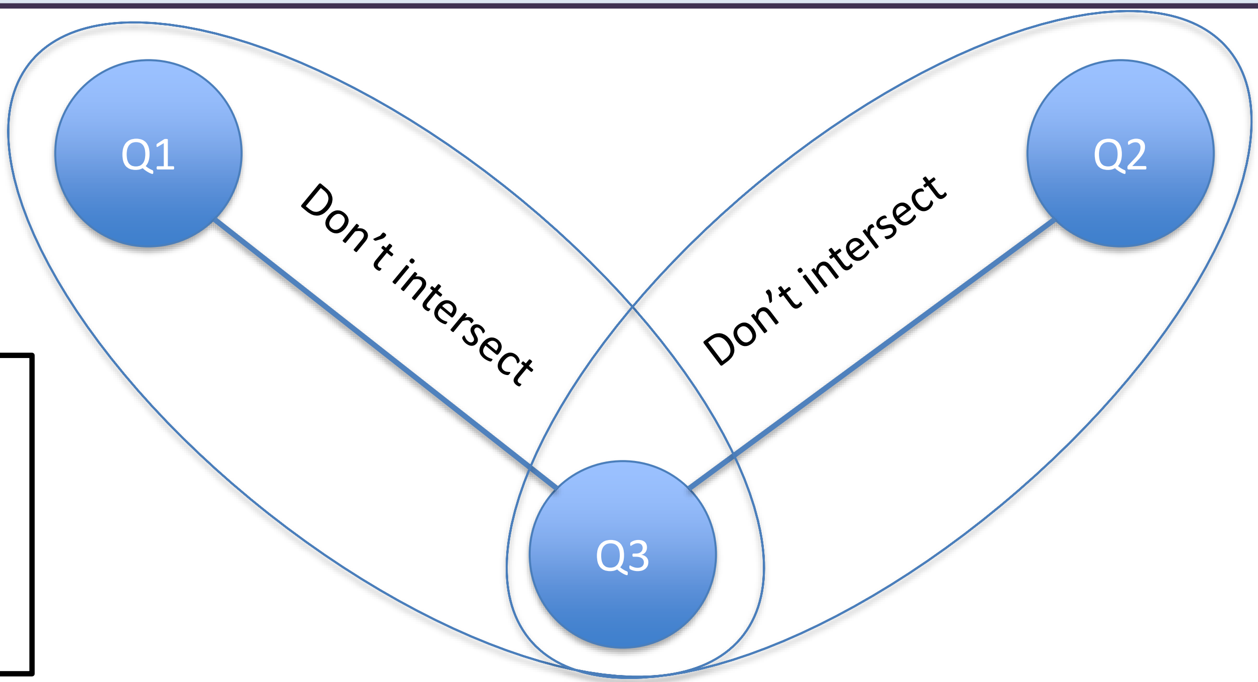
Publications

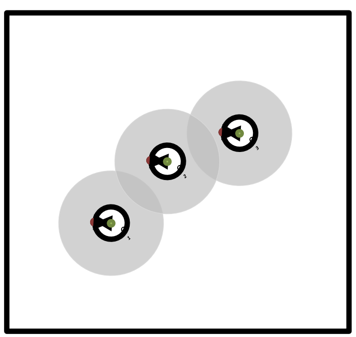
Zoumbatianos KDD '15

Zoumbatianos TKDE '18

Solution

We need to find the **maximum** clique in the graph
(NP-Complete: we find a large enough clique using a heuristic)





Publications

Zoumbatianos KDD '15

Zoumbatianos TKDE '18

Step 3: Densifying Number of data series to add

1. Given a set of hardnesses as input

2. We decide the number of data series to add for each query by solving a linear system of equations:

$$a_i = \frac{N_i + x_i}{N + \sum_{j=1}^{n} x_j}$$

   - α_i : hardness,
   - X_i : number of data series to add
   - N_i : number of data series already in e-area
   - N : Total number of data series

Publications

Zoumbatianos KDD '15

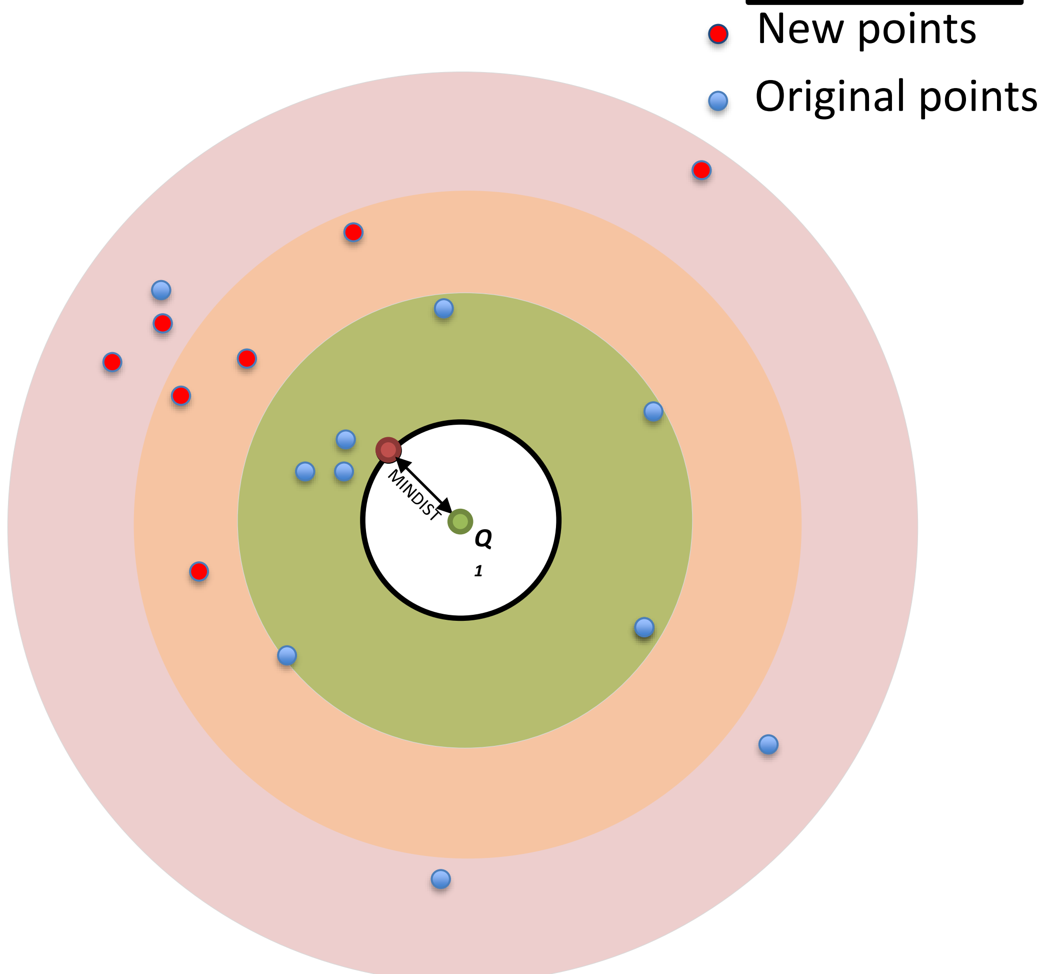
Zoumbatianos TKDE '18

Densification Method:

Equi-densification

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

Equal number of points in every "zone"



Experiments
Densification Methods

Publications

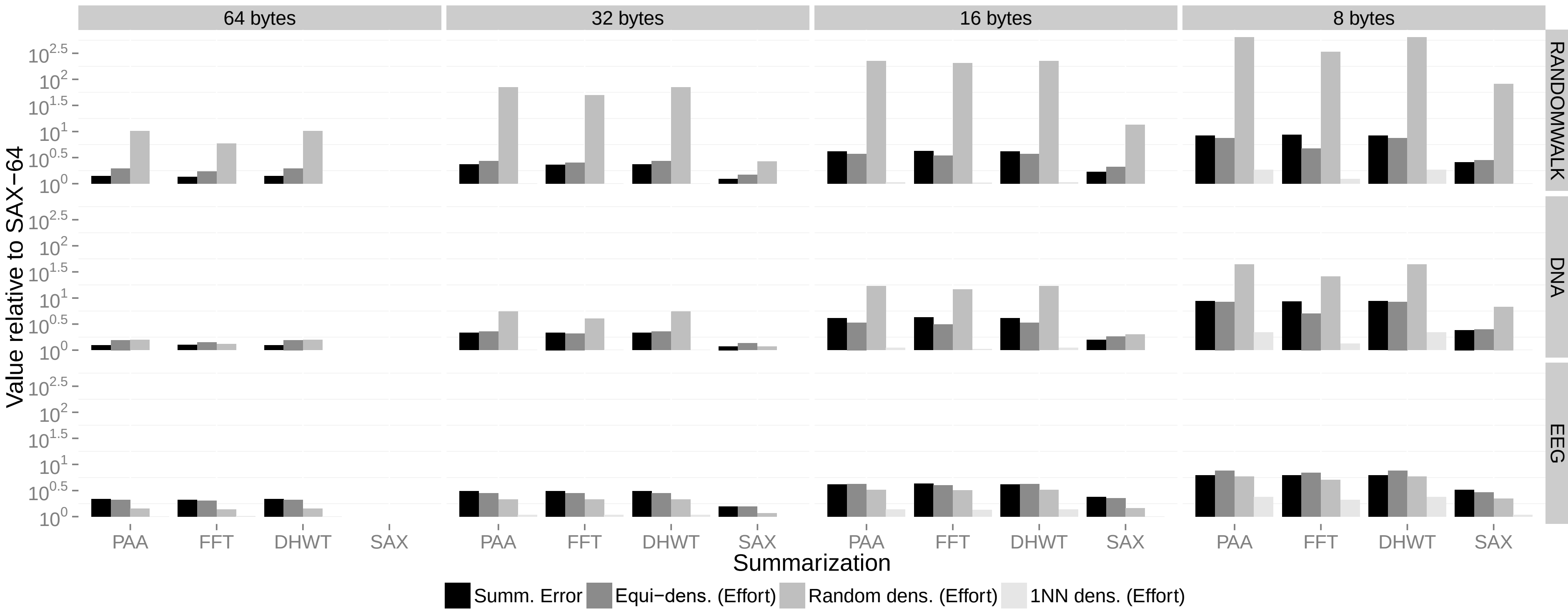
Zoumbatianos KDD '15

Zoumbatianos TKDE '18

Using all datasets of size 256 (100 queries for each dens. method), we measured the:

- ***1-TLB: Summarization Error** (0: perfect bound, 1: worst possible bound)*
- ***Minimum Effort** for a set of summarizations using 8 – 64 bytes.*

Normalized over SAX-64



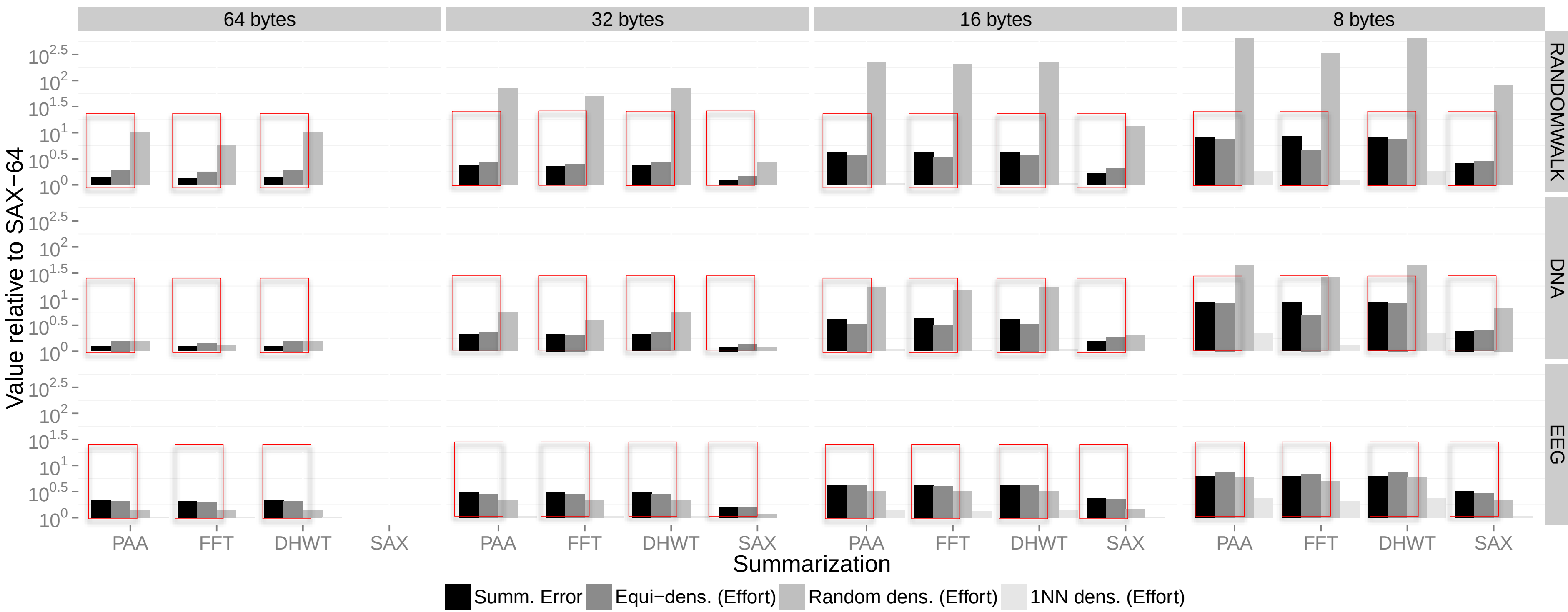
Experiments
Densification Methods

Publications
- Zoumbatianos KDD '15
- Zoumbatianos TKDE '18

For **equi-densification**

normalized Effort is closer to the **normalized Summarization Error**

The worse a summarization the bigger effort it does



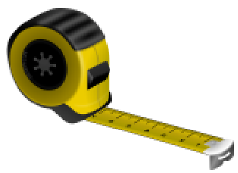
Publications

Zoumbatianos KDD '15

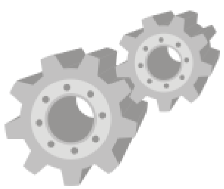
Zoumbatianos TKDE '18

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

Outline

- sequence management system
- benchmarking
- general high-dimensional vectors
- deep learning

Publications

Echihabi-WIMS'20

Data Series vs. high-d Vectors

- two sides of the same(?) coin
  - data series as multidimensional points
  - for a specific ordering of the dimensions
- several techniques for similarity search in high-d vectors
  - using LSH (SRS), space quantization (IMI), k-NN graphs (HNSW)
- how do these high-d vector techniques compare to data series techniques?
  - conducted extensive experimental comparison

Publications

Echihabi-PVLDB'19

Data Series vs. high-d Vectors

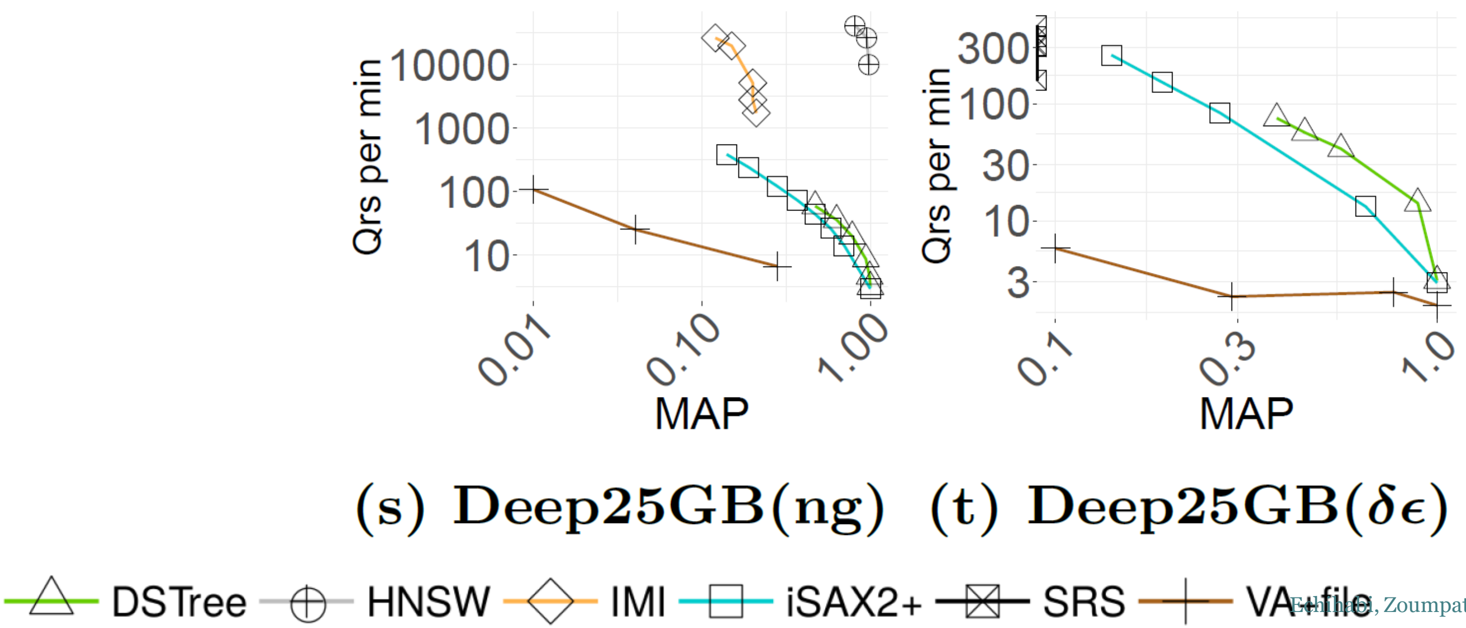
- **data series techniques** are the **overall winners**, even on **general high-d vector** data

Publications

Echihabi-PVLDB'19

Data Series vs. high-d Vectors

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



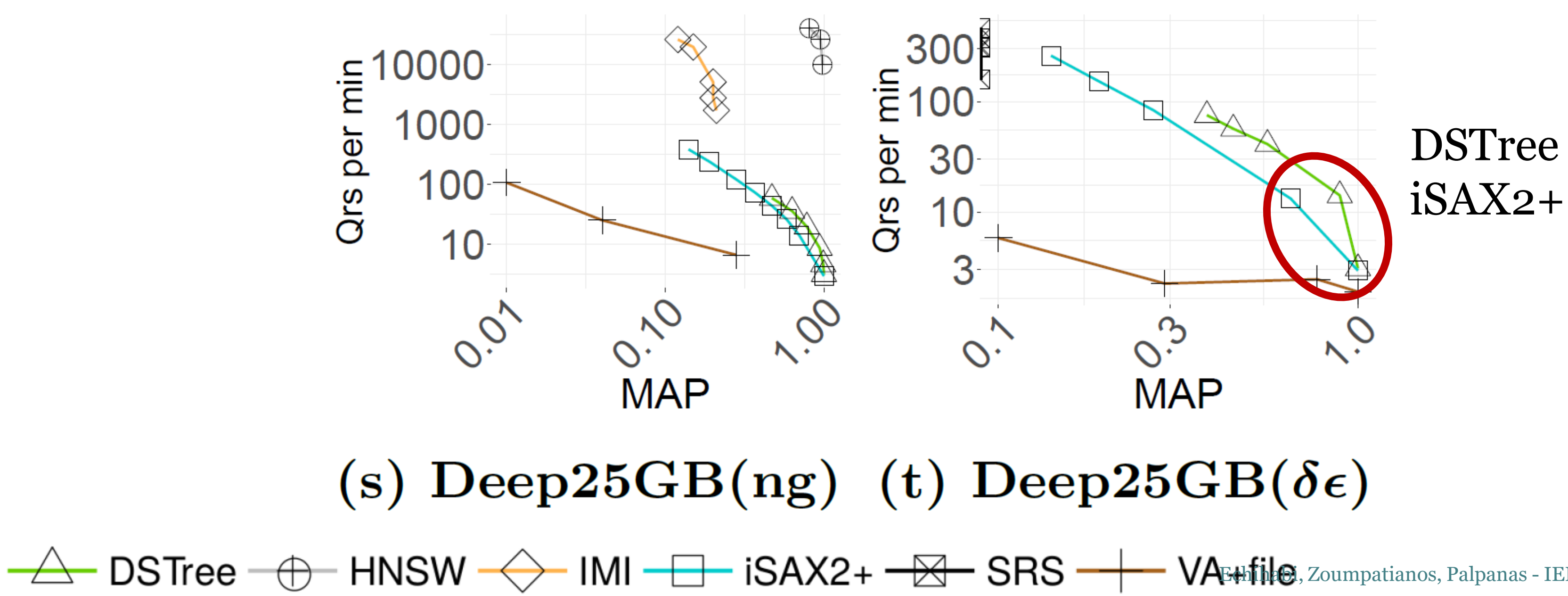
(s) Deep25GB(ng) (t) Deep25GB($\delta\epsilon$)

Publications

Echihabi-PVLDB'19

Data Series vs. high-d Vectors

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



(s) Deep25GB(ng) (t) Deep25GB(δϵ)

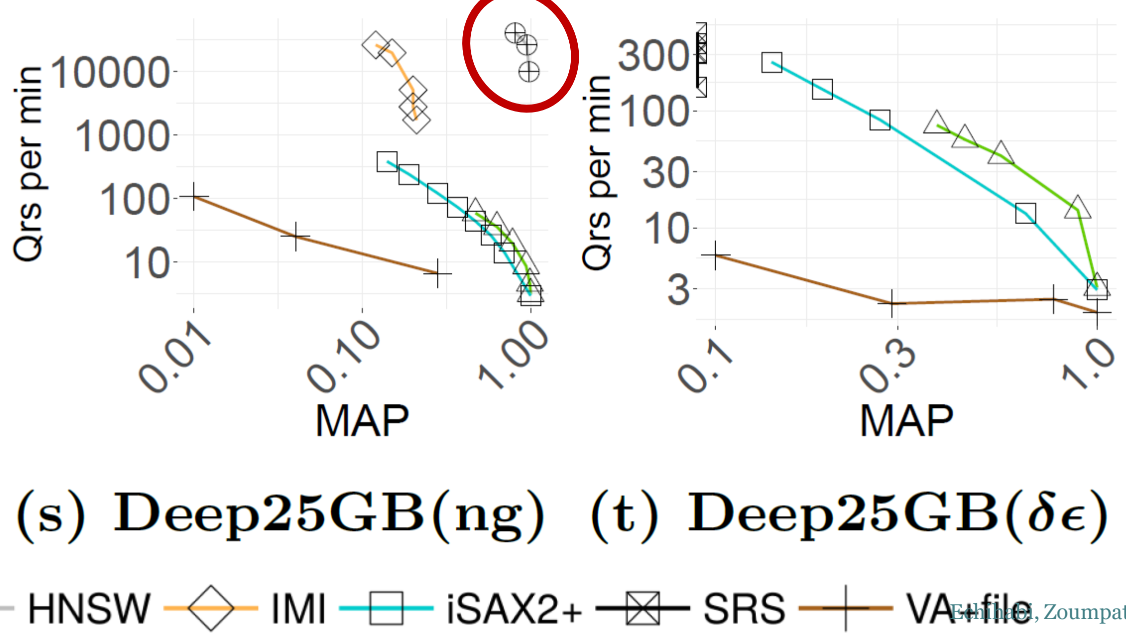
Data Series vs. high-d Vectors

Publications

Echihabi-PVLDB'19

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

HNSW (only in-memory, ng-approximate)



(s) Deep25GB(ng) (t) Deep25GB(δϵ)

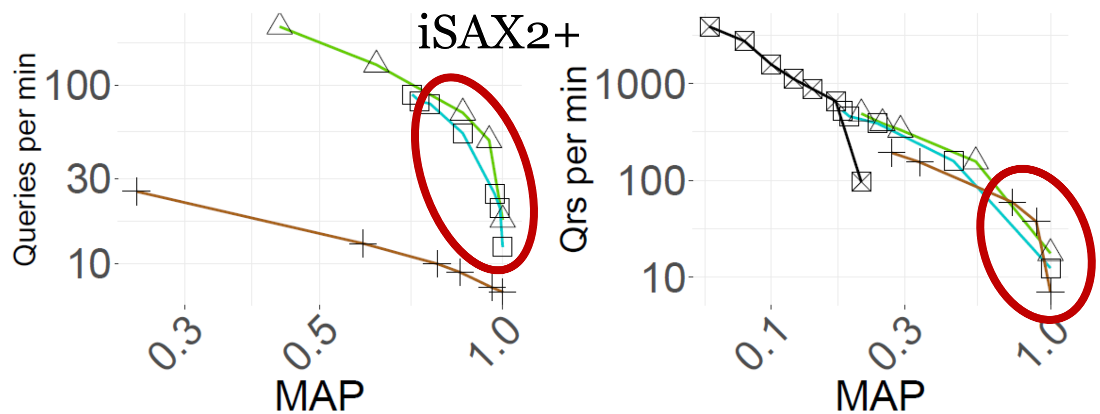
Publications

Echihabi-PVLDB'19

Data Series vs. high-d Vectors

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

DSTree
iSAX2+



DSTree
iSAX2+
VA+file

(g) Rand25GB 16384 (ng)

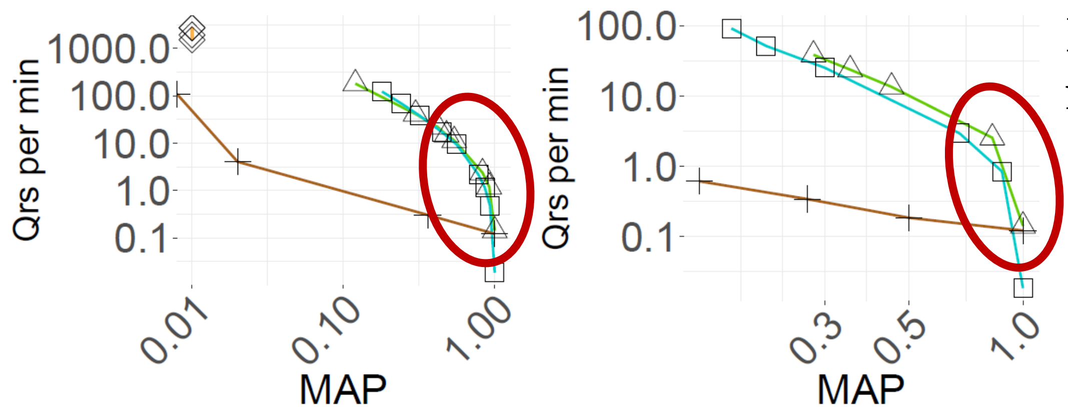
(h) Rand25GB 16384 ($\delta\epsilon$)

Publications

Echihabi-PVLDB'19

Data Series vs. high-d Vectors

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



DSTree
iSAX2+

(m) Deep250GB(ng)

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

DSTree HNSW IMI iSAX2+ SRS VA+file

Publications

Echihabi-PVLDB'19

Data Series vs. high-d Vectors

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

- several new applications (and challenges) for data series similarity search techniques!

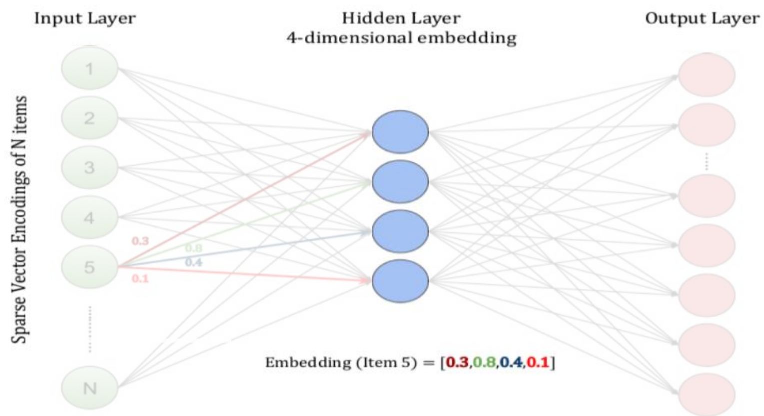
Outline

- sequence management system
- benchmarking
- general high-dimensional vectors
- deep learning

Connections to Deep Learning

- data series indexing for deep embeddings

sequences
text
images
video
graphs
...



deep embeddings
high-d vectors learned using a DNN

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 data series
  - 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

Conclusions

- data series is a very common data type
  - across several different domains and applications
- complex data series analytics are challenging
  - have very high complexity
  - efficiency comes from data series management/indexing techniques
- need for Sequence Management System
  - optimize operations based on data/hardware characteristics
  - transparent to user
- several exciting research opportunities

thank you!

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

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

References

- 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.
- Pavlidis, T. (1976). Waveform segmentation through functional approximation. *IEEE Transactions on Computers*.
- 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

- 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.
- 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.

References

- 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.
- 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)
- 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.
- 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)

References

- 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
- 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)
- P. Schafer and M. Hogvist. Sfa: A symbolic fourier approximation and index for similarity search in high dimensional datasets. EDBT 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.
- 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
- 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

References

- 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.
- 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
- 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
- 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

- Katsiaryna Mirylenka, Michele Dallachiesa, Themis Palpanas: Correlation-Aware Distance Measures for Data Series. EDBT 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
- 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)
- 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

References

- 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)
- 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)
- C. Fu, C. Xiang, C. Wang, and D. Cai. Fast Approximate Nearest Neighbor Search with the Navigating Spreading-out Graph. PVLDB, 12(5):461–474, 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)
- 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

References

- 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)
- 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)
- Botao Peng, Panagiota Fatourou, Themis Palpanas. Paris+: Data series indexing on multi-core architectures. TKDE, 2020
- 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
- Botao Peng, Panagiota Fatourou, Themis Palpanas. SING: Sequence Indexing Using GPUs. ICDE, 2021

References

- InfluxDB: https://www.influxdata.com/
- Timescale: https://www.timescale.com
- Beringei: https://github.com/facebookarchive/beringei
- Druid: https://druid.apache.org
- Prometheus: https://Prometheus.io
- CrateDB: https://crate.io
- IoTDb: https://iotdb.apache.org
- OpenTSDB: http://opentsdb.net/
- QuasarDB: https://www.quasardb.net/
- Timestream: https://aws.amazon.com/timestream/
- Apache IoTDB: https://iotdb.apache.org/