Compressing Big OLAP Data Cubes in Big Data Analytics Systems: New Paradigms and Future Research Perspectives

Alfredo Cuzzocrea
University of Calabria, Italy

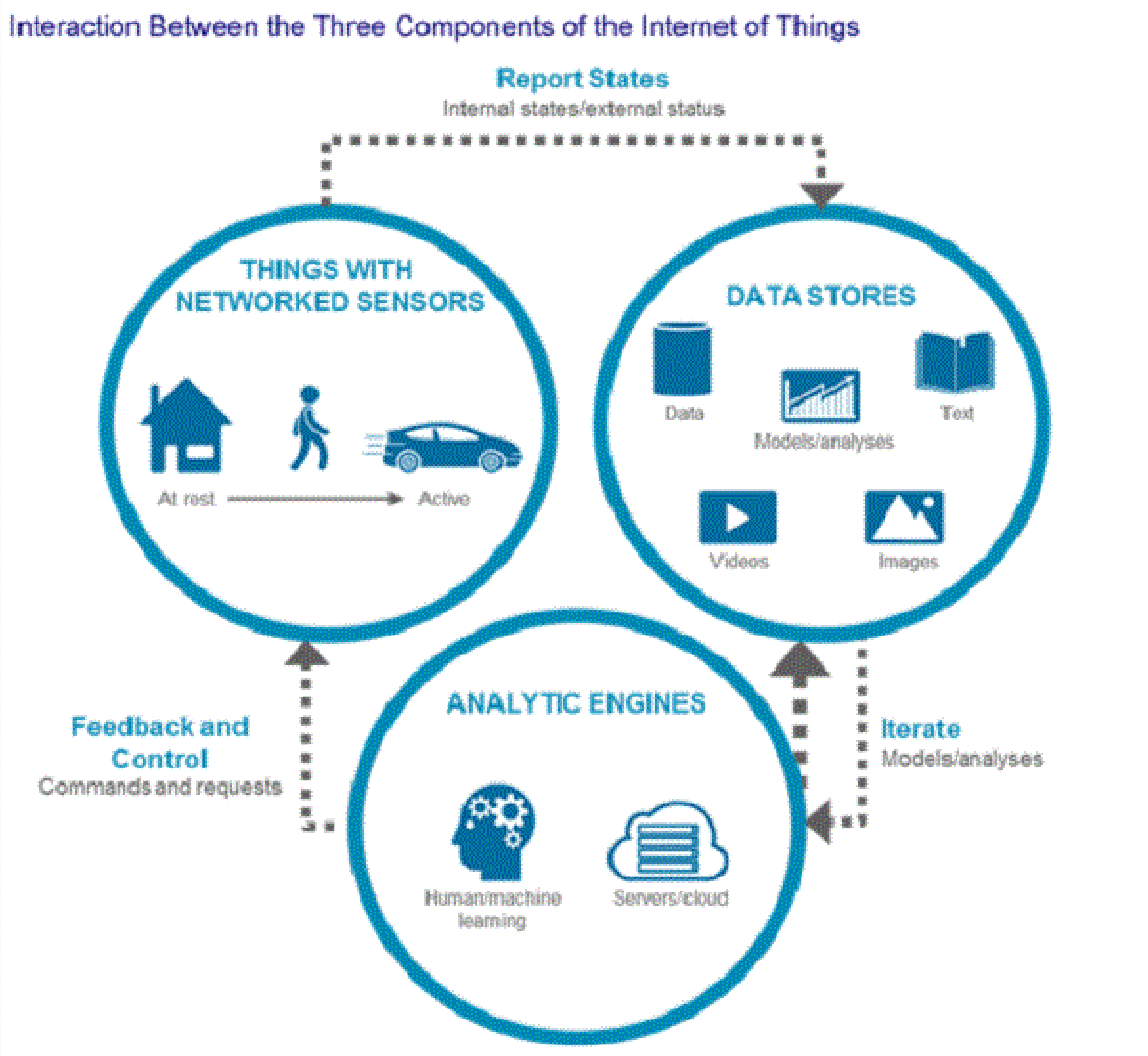
keynote talk @ ICSBT 2022

Lisbon, Portugal - July 16, 2022

Big Data Principles

- Massive amounts of heterogeneous data
  - Relational Data (Tables/Transactions/Legacy Data)
  - Text Data (Web)
  - Semi-structured Data (XML)
  - Graph Data (Social Networks, Semantic Web)
- Large-scale data (distributed repositories, clouds)
- Scalability issues: running on very-large, growing data sets
- Elastic metaphors – Cloud Computing paradigms
- Database As A Service (DaaS)
- Easy and Interpretable Analytics
- Privacy-Preserving and Secure Data Management

A Big Data IoT Framework Reference Architecture



Main Issues in Big Data IoT Frameworks

- *Performance*
- *Support for heterogeneous data formats and types*
- *Transparency vs autonomy*
- *Data security, privacy and confidentiality*
- *Analytics support*
- *Communication protocols*
- [...]

Big Data Compression/1

- Among these research challenges, performance of big data management plays a critical role
- Indeed, it is easy to understand how the complexity of big data management tasks heavily influence all the other activities
- One solution to this relevant problem is represented by so-called big **data compression paradigms**
- Basically, the idea behind big data compression initiatives consists in reducing the size of (big) data as to gain into querying and management efficiency

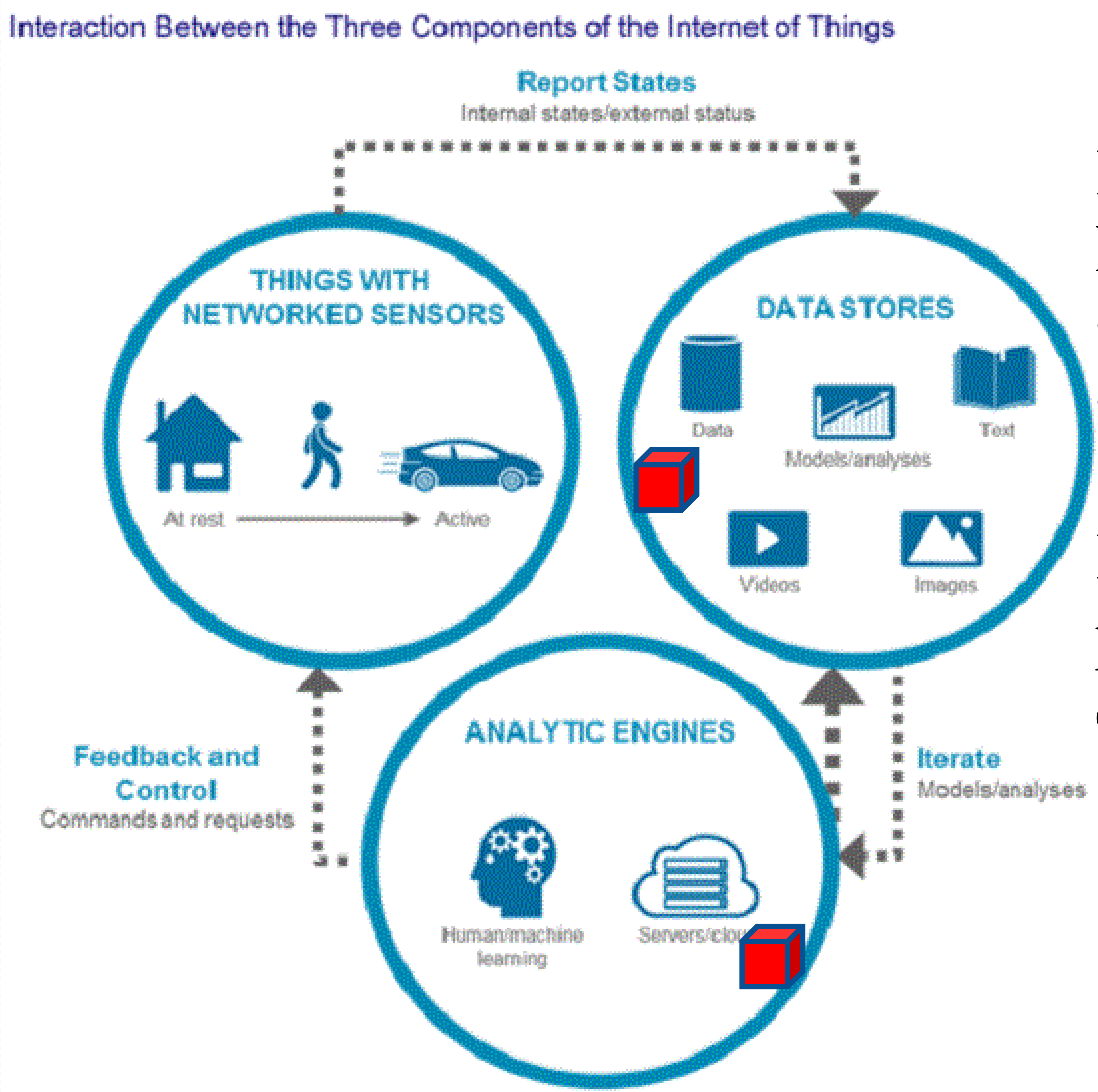
Big Data Compression/2

- Big data compression methods:
  - *Lossless approaches*
  - *Statistical approaches*
  - *Error-metrics approaches*
- Many solutions in classical contexts (e.g., OLAP data cube compression)
- Real-life implementation (e.g., MS SQL Server platform)
- Effective and practical systems

Big Data Compression Main Challenges

- *Big Data Compression as a Paradigm for Big Data Understanding*
- *Accuracy-Aware Big Data Compression Techniques*
- *Feature Correlation Analysis for Enhanced Big Data Compression*
- *Indexing Data Structures for Compressed Big Data*
- *Query Optimization Issues*
- *Scalability Issues*
- *Cloud-based Big Data Compression*
- *Secure and Privacy-Preserving Big Data Compression*
- *Compressed Big Data Visualization Tools*

A Special Case: Compressing Data Cube in Big Data IoT Frameworks



Data cubes arise in several data layers of the reference framework:

- in the proper **data storage layer**;
- in the extended **analytical layer**.

Hence, compressing data cubes can provide a relevant reduction in the overall data processing flow

Outline

- Problem Statement
- Approximate Query Answering Techniques in OLAP
- Synopsis Data Structures: Overview
- Limitations of AQA
- The Δ-Syn Approach

Outline

- **Problem Statement**
- Approximate Query Answering Techniques in OLAP
- Synopsis Data Structures: Overview
- Limitations of AQA
- The Δ-Syn Approach

Problem Statement/1

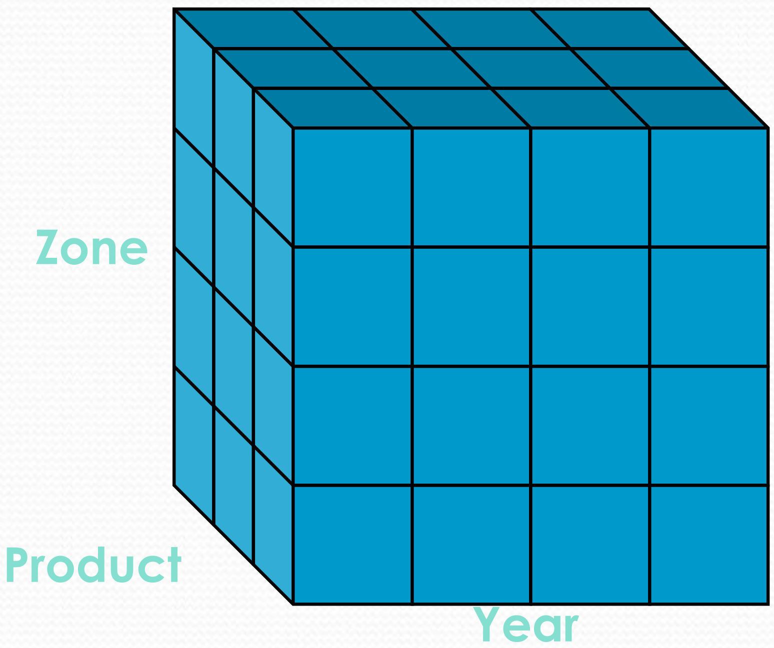
- OLAP: performing fast aggregations on huge amounts of data to support decision making processes.

Dimensions Measure

Product	Year	Zone	Sale

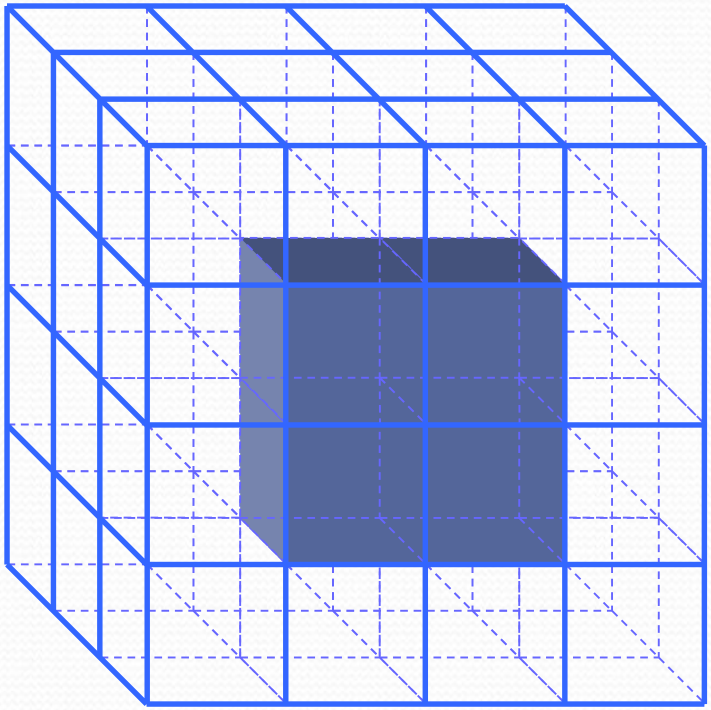
multidimensional representation

data cube



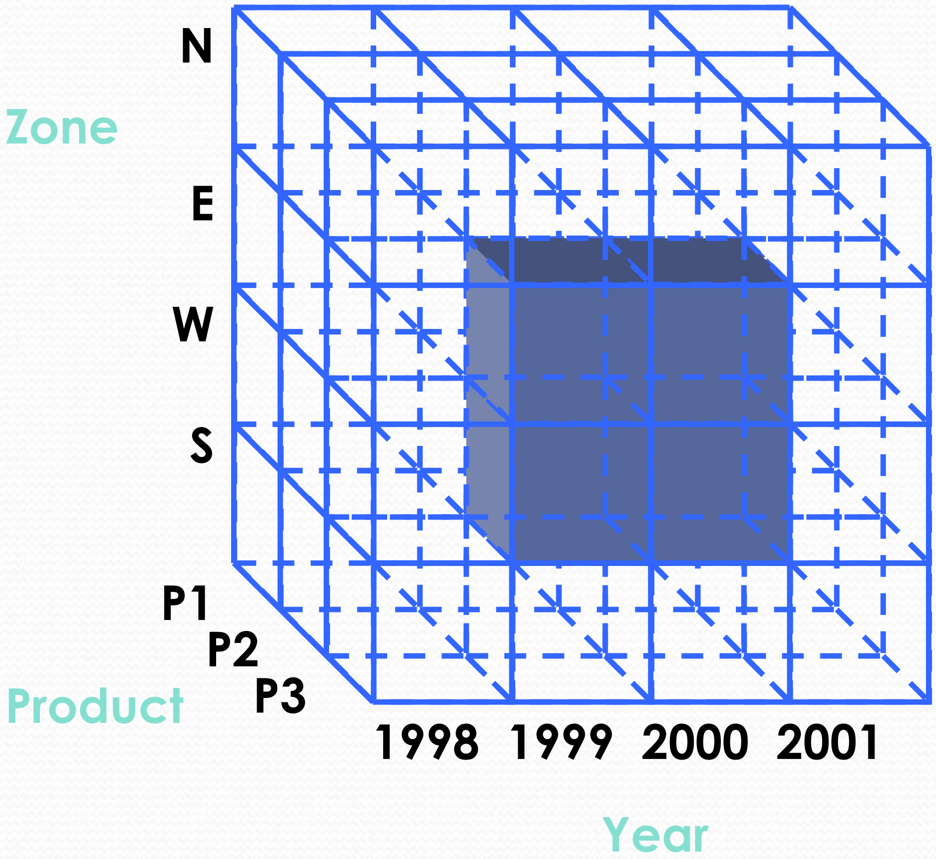
Problem Statement/2

- A range query over a data cube is defined as the application of a SQL aggregation operator (such as SUM, COUNT, AVG etc) on the subset of data which belong to a given range.



Problem Statement/3

- Example: total amount of sales of the product *P3* between *1999* and *2000* in zones *East* and *West*.



Product	Year	Zone	Sale
…	…	…	…
P3	1999	W	
…	…	…	…
P3	2000	E	
P3	2000	W	
…	…	…	…

Outline

- Problem Statement
- **Approximate Query Answering Techniques in OLAP**
- Synopsis Data Structures: Overview
- Limitations of AQA
- The Δ-Syn Approach

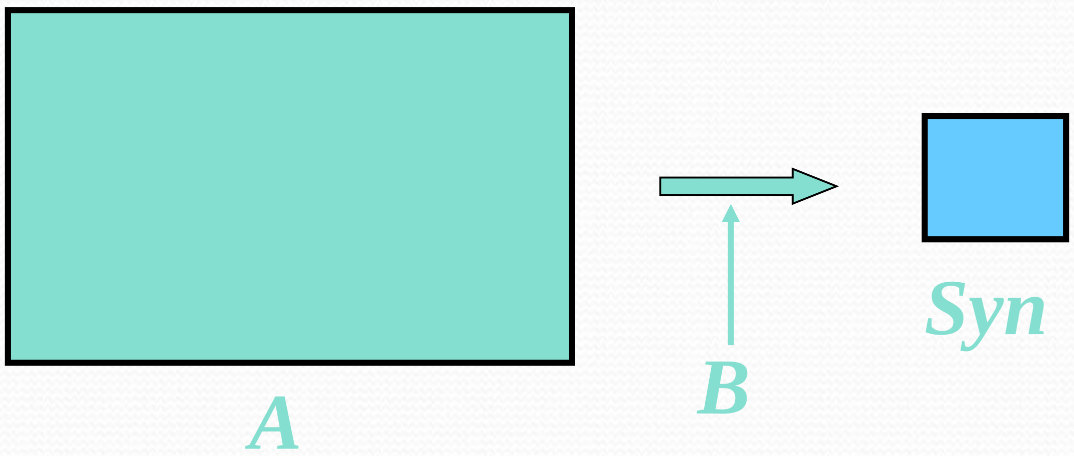
Approximate Query Answering

- Approximate query answering (AQA) is a widely investigated issue in OLAP research.
- Fast approximate answers generally suffice to support decision making processes as decimal precision is not needed in OLAP.
- Some traditional approaches:
  - *Histograms* (Poosala, Ioannidis, Haas, Shekita);
  - *Wavelets* (Vitter, Wang, Iyer);
  - *Random Sampling* (Gibbons, Matias).

Outline

- Problem Statement
- Approximate Query Answering Techniques in OLAP
- **Synopsis Data Structures: Overview**
- Limitations of AQA
- The Δ-Syn Approach

Synopsis Data Structures: Overview



- Requirements:
  - size(*Syn*) << size(A);
  - for each query Q, the approximate answer $A(Q)$ evaluated on *Syn* must be "very close" by the exact answer $E(Q)$ evaluated on A, i.e. $A(Q) \cong E(Q)$.

Outline

- Problem Statement
- Approximate Query Answering Techniques in OLAP
- Synopsis Data Structures: Overview
- **Limitations of AQA**
- The Δ-Syn Approach

Limitations of AQA Techniques in OLAP/1

- Conventional approximate OLAP query answering techniques suffer from two main limitations:
  - *Lack of Accuracy Control*: they do not provide any mechanism for controlling accuracy of retrieved answers approximate answers;
  - *Scalability Issues*: their quality in both representing the input data cube and evaluating OLAP queries over the compressed data cube decreases when data cubes grow in dimension number and size.

Limitations of AQA Techniques in OLAP/2

- Across 10 years, we proposed a set of AQA techniques for OLAP data cubes aimed at overcoming previous limitations:
  - Δ-Syn, an analytical synopsis data structure that introduces a polynomial approximation technique for OLAP data cubes;
  - K_{LSA} (or accuracy-aware Δ-Syn), which further extends the Δ-Syn proposal in order to provide accuracy control over compressed OLAP data cubes.

Outline

- Problem Statement
- Approximate Query Answering Techniques in OLAP
- Synopsis Data Structures: Overview
- Limitations of AQA
- The Δ-Syn Approach

Δ-Syn Outline (Sub-Section)

- The Δ-Syn Synopsis Data Structure: Overview
- The LSA Method and its Adaptation to AQA
- Improving the Δ-Syn Technique
- Building the Δ-Syn
- Δ-Syn Physical Representation
- The Accuracy-Aware LSA Method
- The Accuracy-Aware Δ-Syn
- Δ-Syn Optimizations
- Δ-Syn Query Model

Δ-Syn Outline (Sub-Section)

- **The Δ-Syn Synopsis Data Structure: Overview**
- The LSA Method and its Adaptation to AQA
- Improving the Δ-Syn Technique
- Building the Δ-Syn
- Δ-Syn Physical Representation
- The Accuracy-Aware LSA Method
- The Accuracy-Aware Δ-Syn
- Δ-Syn Optimizations
- Δ-Syn Query Model

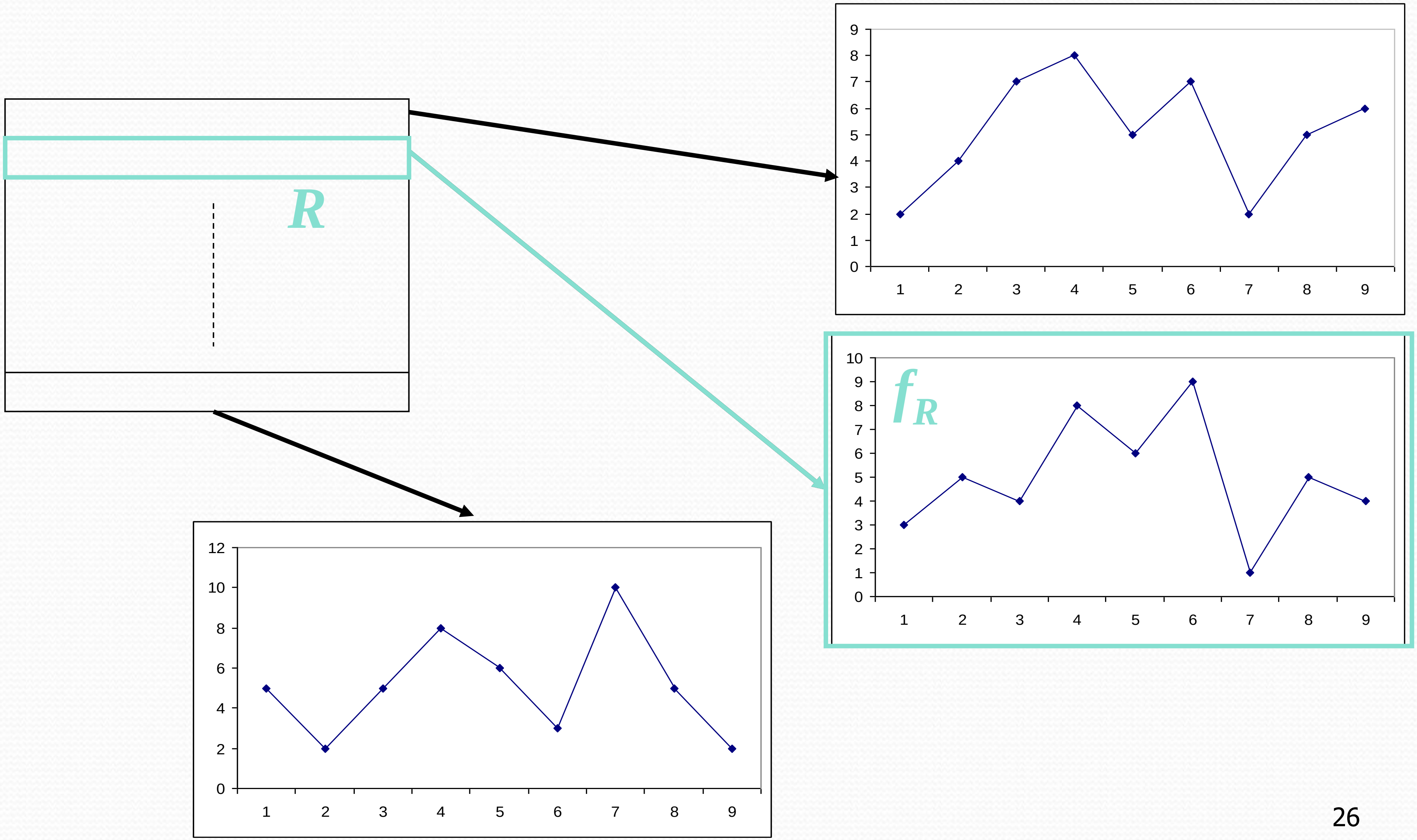
Analytical Interpretation of Multidimensional Data Cubes/1

- Our approach starts from an **analytical interpretation of multidimensional data cubes**: a data cube is treated as a collection of data rows, such that a representing discrete data distribution is associated to each row.
- Each data distribution is then approximated via the well-known **Least Square Approximation (LSA)** method and the resulting set of polynomial coefficients are stored instead of the original data, thus obtaining a synopsis data structure called Δ-Syn.
- Queries are issued on the compressed representation, thus reducing the number of disk accesses needed to evaluate the answers.

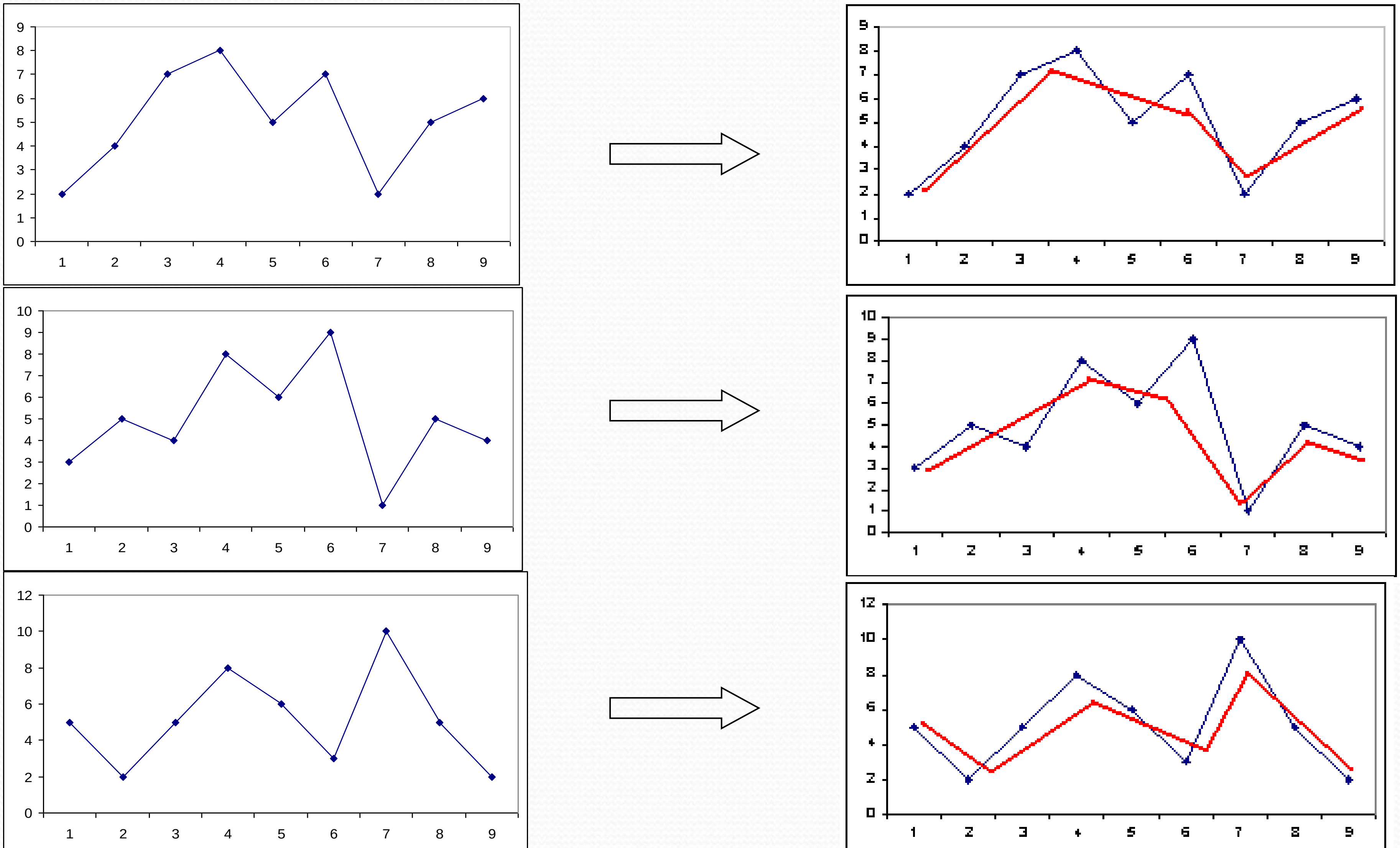
Analytical Interpretation of Multidimensional Data Cubes/2

- Without any loss of generality, we refer to data cubes stored according to the MOLAP data organization.
- A MOLAP data cube A is a multidimensional array from which we can select the i-th row according to a certain access strategy $A[i]$.
- In other words, from a logical point of view **a MOLAP data cube is a set of rows**.
- This realizes our analytical interpretation of multidimensional data cubes.

Example: 2D Data Cube



Data Distribution Approximation via LSA



Δ-Syn Building Steps

- INPUT A multidimensional data cube A, the available storage space B.
- OUTPUT Δ-Syn.
- STEPS
  1. Allocate the available storage space B.
  2. For each row R belonging to A, extract the data distribution f_R.
  3. For each f_R, build the approx function g_R via applying the LSA method.
  4. For each g_R, store the approximating coefficients $\{c_R\}$.

Δ-Syn Outline (Sub-Section)

- The Δ-Syn Synopsis Data Structure: Overview
- **The LSA Method and its Adaptation to AQA**
- Improving the Δ-Syn Technique
- Building the Δ-Syn
- Δ-Syn Physical Representation
- The Accuracy-Aware LSA Method
- The Accuracy-Aware Δ-Syn
- Δ-Syn Optimizations
- Δ-Syn Query Model

The LSA Method

- Given a discrete function f with n samples, LSA finds the **"best" polynomial function** g approximating f via minimizing the sum of the squares of distances between points of f and g.
- g is defined as the linear combination of T basis functions Φ_k belonging to Φ, such that Φ is the set of basis functions of the g functional space, and T coefficients as follows:

$$g = \sum_{k=0}^{T-1} c_k \cdot \Phi_k \qquad c_k = \frac{\Phi_k \times f}{\left\|\Phi_k\right\|^2}$$

Adapting the LSA Method to Approximate Query Answering

- T is also the polynomial degree of g.
- In the original LSA method, T is an input parameter and, intuitively enough, the greater is T the greater is the degree of accuracy (i.e., the "quality") of g. In our algorithm, T **depends on the storage space** B available for representing Δ-Syn.
- It follows that a critical component of the Δ-Syn proposal is the **allocation scheme**, which is presented next.

Δ-Syn Outline (Sub-Section)

- The Δ-Syn Synopsis Data Structure: Overview
- The LSA Method and its Adaptation to AQA
- **Improving the Δ-Syn Technique**
- Building the Δ-Syn
- Δ-Syn Physical Representation
- The Accuracy-Aware LSA Method
- The Accuracy-Aware Δ-Syn
- Δ-Syn Optimizations
- Δ-Syn Query Model

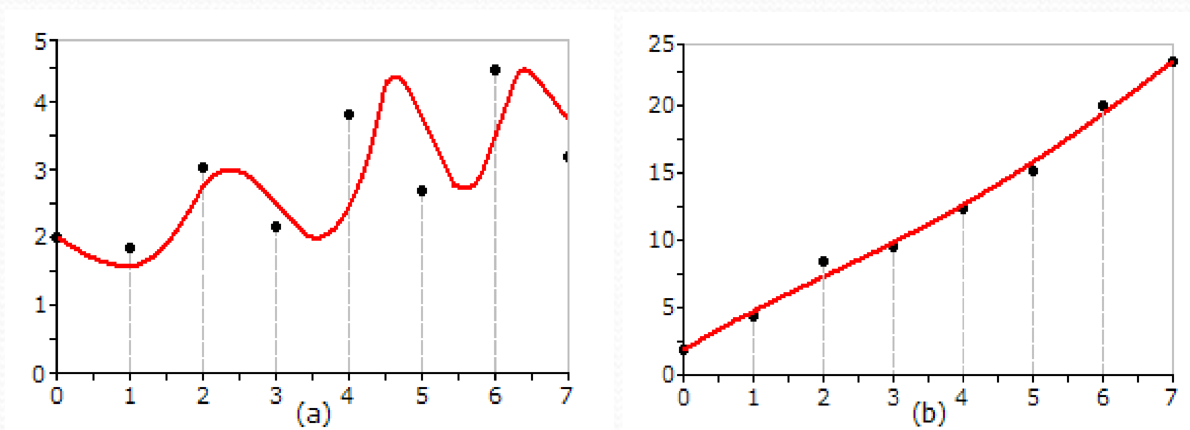
Improving the Δ-Syn Technique

- In order to improve the quality of our technique, for each row R of A, instead of approximating f_R directly, we build and approximate the **cumulative distribution** of f_R, denoted by f_R^+, and defined as follows:

$$f_R^+(x) = \begin{cases} f_R(x) & x = 0 \\ f_R^+(x-1) + f_R(x) & x \geq 1 \end{cases}$$

Benefits due to f_R^+ /1

- f_R^+ is an always-increasing function (*b*) and, as a consequence, it can be approximated with a polynomial function having a polynomial degree smaller than the one needed to approximate a skewed function (*a*):



Benefits due to /2 f_R^+

- Improved approximate query evaluation as a lower number of disk accesses is needed.
- 2D Range-SUM Query: $Q(\langle l_0, u_0 \rangle, \langle l_1, u_1 \rangle)$

Therefore, we apply the LSA method on functions f_R^+ instead that functions f_R.

- Approximate answer with f_R:

$$A_f(Q) = \sum_{i_0=l_0}^{u_0} \sum_{i_1=l_1}^{u_1} \Delta - Syn_f[i_0, i_1] = \sum_{i_0=l_0}^{u_0} \sum_{i_1=l_1}^{u_1} g_{i_0}(i_1)$$

- Approximate answer with f_R^+:

$$A_{f^+}(Q) = \sum_{i_0=l_0}^{u_0} \sum_{i_1=l_1}^{u_1} \Delta - Syn_{f^+}[i_0, i_1] = \sum_{i_0=l_0}^{u_0} \left[g_{i_0}^+(u_1) - g_{i_0}^+(l_1 - 1) \right]$$

Δ-Syn Building Steps – Revised

- INPUT A multidimensional data cube A, the available storage space B.
- OUTPUT Δ-Syn.
- STEPS

1. Allocate the available storage space B.
2. For each row R belonging to A, extract the data distribution f_R.
3. For each f_R f_R^+ build the approx function g_R g_R^+ via applying the LSA method.
4. For each g_R g_R^+ store the approximating coefficients $\{c_R\}$.

For each f_R, build the cumulative distribution f_R^+.

Δ-Syn Outline (Sub-Section)

- The Δ-Syn Synopsis Data Structure: Overview
- The LSA Method and its Adaptation to AQA
- Improving the Δ-Syn Technique
- **Building the Δ-Syn**
- Δ-Syn Physical Representation
- The Accuracy-Aware LSA Method
- The Accuracy-Aware Δ-Syn
- Δ-Syn Optimizations
- Δ-Syn Query Model

Building Δ-Syn/1

- A basic issue in our work is how to allocate the storage space B available for housing Δ-Syn.
- We propose a **proportional storage space allocation scheme** based on statistical properties of data distributions.
- Similarly to the other components of the technique, the allocation scheme is also oriented to rows.
- We basically use two parameters of row data distributions: the **skewness** and its **standard deviation**, and drive the space allocation accordingly.

Building Δ-Syn/2

- Let R be a row of A and f_R be its representing function.
- The skewness value $\gamma_1(R)$ of f_R is defined as follows:

$$\gamma_1(R) = \frac{\left(\mu_3(R)\right)^2}{\left(\mu_2(R)\right)^3}$$

where $\mu_r(R)$ is the r^{th} central moment of f_R, which is defined as follows:

$$\mu_r(R) = \sum_{k=0}^{n-1} \left(k - \mu\right)^r \cdot f_R\left(k\right)$$

Building Δ-Syn/3

- The standard deviation of the skewness $\sigma_\gamma(R)$ can be computed as follows [Stuart&Ord98]:

$$\sigma_\gamma(R) = \sigma\left(\gamma_1(R)\right) = \sqrt{\frac{6}{n}}$$

- A well-known result of theoretical statistics [Stuart&Ord98] claims that the skewness value of a data distribution is "significant" if **it is greater than its standard deviation by a factor of 2.6**.
- In this condition, it can be assumed that data are not distributed according to a normal distribution, so that the distribution is skewed.

Building Δ-Syn/4

- We introduce the function $\Gamma(R)$ for detecting whether the skewness value of a given row is significant:

$$\Gamma(R) = \begin{cases} 1 & \dfrac{\gamma_1(R)}{\sigma_\gamma(R)} > 2.6 \\ 0 & otherwise \end{cases}$$

- We denote the factor $\dfrac{\gamma_1(R)}{\sigma_\gamma(R)} - 2.6$ by $\lambda(R)$.

Building Δ-Syn/5

- We introduce the function $m(R)$ that captures the statistical properties of the distribution f_R of a given row R, as follows:

$$m(R) = \sigma^2(R) + \frac{abs[\gamma_1(R)]}{|R|}$$

"global" effect

"local" effect

such that $\sigma^2(R)$ is the variance of f_R, which is defined as follows:

$$\sigma^2(R) = \sum_{k=0}^{n-1} (k - \mu)^2 \cdot f_R(k)$$

Building Δ-Syn/6

- In conclusion, given a row R, the storage space $B(R)$ allocated to R as a portion of the whole available storage space B is given by the following formula:

$$B(R) = \left\lfloor \frac{m(R) + \Gamma(R) \cdot \lambda(R)}{\sum_{k=0}^{RN(A)-1} m(k) + \sum_{k=0}^{RN(A)-1} \Gamma(k) \cdot \lambda(k)} \cdot B \right\rfloor$$

- This also determines the number of coefficients and basis functions T used to approximate f_R.

Building Δ-Syn/7

- Therefore, for all rows of the input data cube A, the proportional allocation scheme is described by the following system:

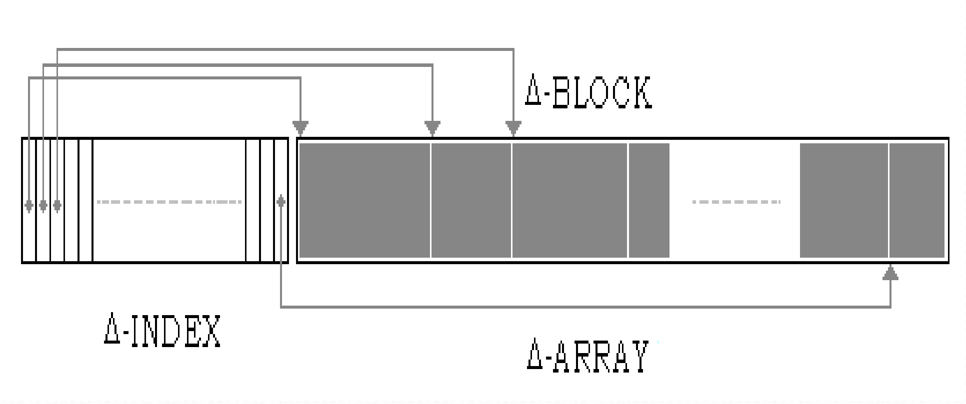
$$
\begin{cases}
B(R_0) = \left\lfloor \dfrac{m(R_0) + \Gamma(R_0) \cdot \lambda(R_0)}{\sum\limits_{k=0}^{RN(A)-1} m(k) + \sum\limits_{k=0}^{RN(A)-1} \Gamma(k) \cdot \lambda(k)} \cdot B \right\rfloor \\
\ldots \\
B(R_{RN(A)-1}) = \left\lfloor \dfrac{m(R_{RN(A)-1}) + \Gamma(R_{RN(A)-1}) \cdot \lambda(R_{RN(A)-1})}{\sum\limits_{k=0}^{RN(A)-1} m(k) + \sum\limits_{k=0}^{RN(A)-1} \Gamma(k) \cdot \lambda(k)} \cdot B \right\rfloor \\
\sum\limits_{k=0}^{RN(A)-1} B(R_k) \leq B
\end{cases}
$$

Δ-Syn Outline (Sub-Section)

- The Δ-Syn Synopsis Data Structure: Overview
- The LSA Method and its Adaptation to AQA
- Improving the Δ-Syn Technique
- Building the Δ-Syn
- **Δ-Syn Physical Representation**
- The Accuracy-Aware LSA Method
- The Accuracy-Aware Δ-Syn
- Δ-Syn Optimizations
- Δ-Syn Query Model

Δ-Syn Physical Representation

- Δ-Syn physical representation consists, for each R belonging to the input data cube A, of the set of coefficients representing the approximating function g_R^+.



Δ-Syn Outline (Sub-Section)

- The Δ-Syn Synopsis Data Structure: Overview
- The LSA Method and its Adaptation to AQA
- Improving the Δ-Syn Technique
- Building the Δ-Syn
- Δ-Syn Physical Representation
- **The Accuracy-Aware LSA Method**
- The Accuracy-Aware Δ-Syn
- Δ-Syn Optimizations
- Δ-Syn Query Model

How to Control the Accuracy of the Compression Process?

- We need a formal **theoretical framework to model and handle accuracy**.
- In our proposal, theoretical foundations are provided by the LSA method.
- We achieve the definition of the so-called **accuracy-aware LSA method**, which allows us to control the degree of approximation of the overall compression process.

The Accuracy-Aware LSA Method/1

- Given a discrete data distribution f and a degree of accuracy δ, from the theoretical foundations of the LSA method, it follows that the constraint to be satisfied to obtain a T_δ–degree approximating function g_δ for f with degree of approximation equal to δ is:

$$\| f - g_\delta \|_2 \leq \delta$$

where $\|\bullet\|_2$ is the **norm operator** modeling the "distance" between f and g_δ.

The Accuracy-Aware LSA Method/2

- The goal is to determine the value of the parameter T_δ to be set **as input** for the execution of the LSA method generating g_δ.
- In our research, we found that such value is the one for which the corresponding approximating function g_δ satisfies the following constraint:

$$g_\delta \cdot (2 \cdot f + g_\delta) \geq f^2 - \delta^2$$

thus, we can control the process generating g_δ and, as a consequence, the overall compression process of the input data cube.

The Accuracy-Aware LSA Method/3

- In order to determine T_δ, we adopt a routine that, starting from an empirical parameter T_δ^*, **iteratively computes** the corresponding approximating function g_δ and checks the main constraint.
- If it is true, then the desired value of T_δ is determined, otherwise we increment T_δ^* and iterate the previous step.
- It is trivial to demonstrate that, for any input distribution f, an upper bound for the parameter T_δ^* exists.
- This routine also gives us the allocation for the current row.

Δ-Syn Outline (Sub-Section)

- The Δ-Syn Synopsis Data Structure: Overview
- The LSA Method and its Adaptation to AQA
- Improving the Δ-Syn Technique
- Building the Δ-Syn
- Δ-Syn Physical Representation
- The Accuracy-Aware LSA Method
- **The Accuracy-Aware Δ-Syn**
- Δ-Syn Optimizations
- Δ-Syn Query Model

Δ-Syn Building Steps – Accuracy Control

- INPUT A MD data cube A, the degree of accuracy δ, the available storage space B.
- OUTPUT Δ-Syn.
- STEPS

1. Allocate the available storage space B.
2. For each row R belonging to A, extract the data distributions f_R.
3. For each f_R, build the cumulative distribution f_R^+
4. For each f_R^+ build the approx function g_R^+ via applying the accuracy-aware LSA method.
5. For each g_R^+ store the approximating coefficients $\{c_R\}$.

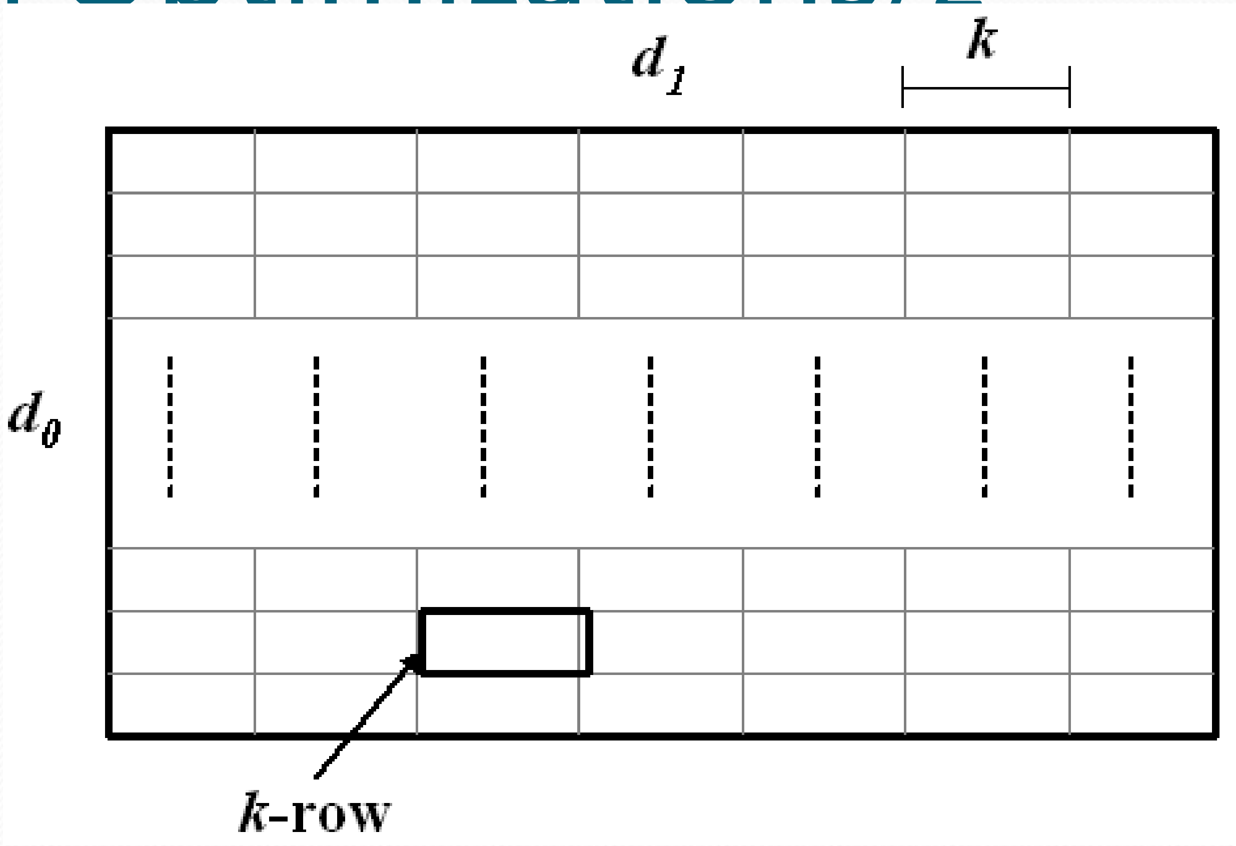
Δ-Syn Outline (Sub-Section)

- The Δ-Syn Synopsis Data Structure: Overview
- The LSA Method and its Adaptation to AQA
- Improving the Δ-Syn Technique
- Building the Δ-Syn
- Δ-Syn Physical Representation
- The Accuracy-Aware LSA Method
- The Accuracy-Aware Δ-Syn
- **Δ-Syn Optimizations**
- Δ-Syn Query Model

Δ-Syn Optimizations/1

- To further improve the capabilities of Δ-Syn (i.e., achieving higher compression ratios), two optimizations are proposed.
- The first one consists in a **partitioning strategy for data rows**, i.e. we apply the accuracy-aware LSA method to parts of rows instead that to the entire rows.
- The second one consists in an **approximation-driven similarity metrics** for the partitioned representation (provided by the first optimization).

Δ-Syn Optimizations/2



The second optimization consists in pruning all the k-row for which the LSA-based "distance" from other rows is less than 10 %.

Δ-Syn Outline (Sub-Section)

- The Δ-Syn Synopsis Data Structure: Overview
- The LSA Method and its Adaptation to AQA
- Improving the Δ-Syn Technique
- Building the Δ-Syn
- Δ-Syn Physical Representation
- The Accuracy-Aware LSA Method
- The Accuracy-Aware Δ-Syn
- Δ-Syn Optimizations
- **Δ-Syn Query Model**

Δ-Syn Query Model/1

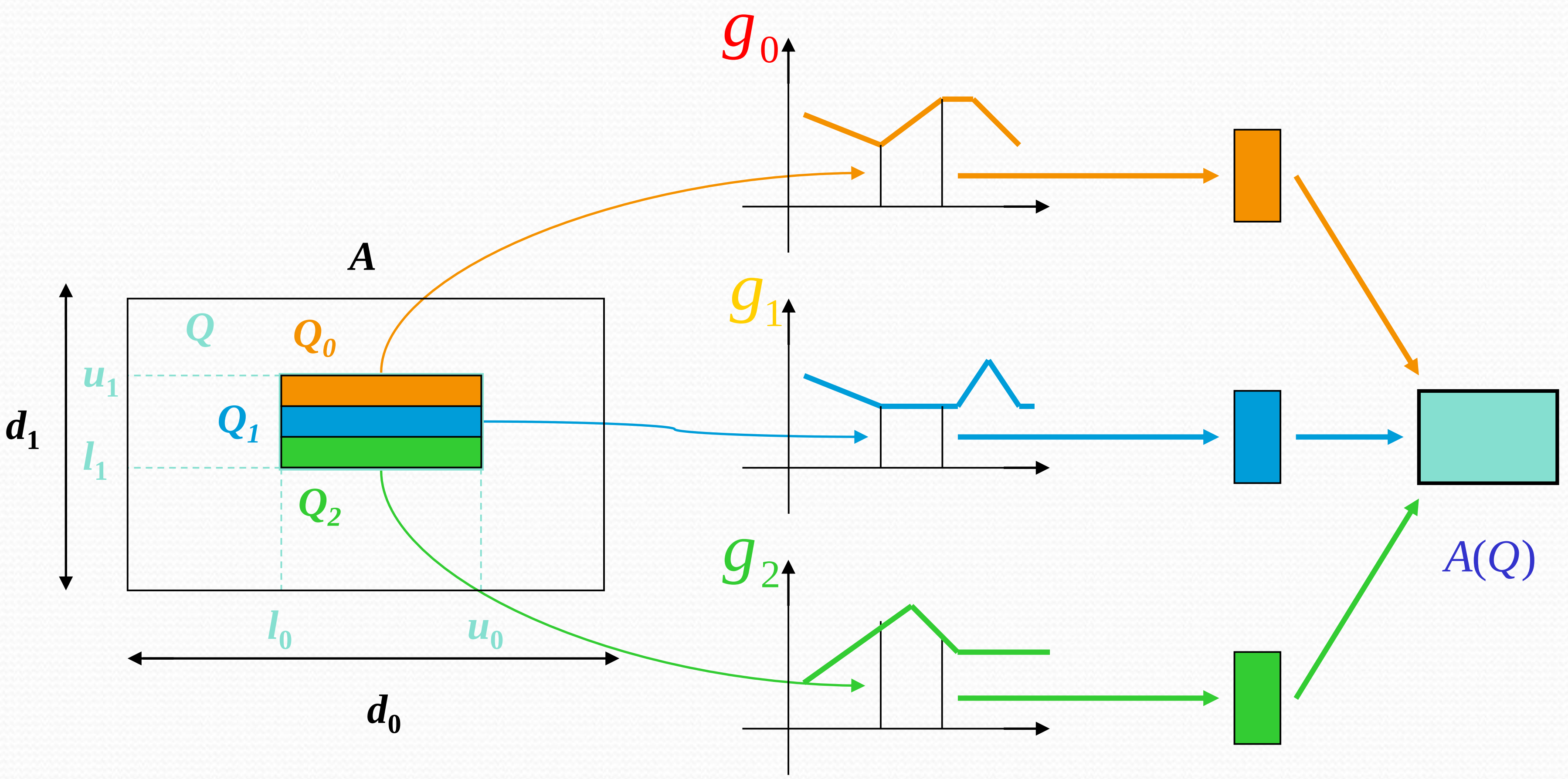
- Let $Q(\langle l_0, u_0 \rangle, \langle l_1, u_1 \rangle)$ be a two-dimensional query on a two-dimensional data cube A.
- Q is decomposed in the following set of queries, each one evaluated on Δ-Syn:

$$Q_0(\langle l_0, u_0 \rangle, \langle l_1, l_1 + 1 \rangle)$$
$$Q_1(\langle l_0, u_0 \rangle, \langle l_1 + 2, l_1 + 3 \rangle)$$
$$\vdots$$
$$Q_{u_1 - l_1}(\langle l_0, u_0 \rangle, \langle u_1 - 1, u_1 \rangle)$$

- Then, the final approximate answer is obtained as:

$$A(Q) = \sum_{k=0}^{u_1 - l_1} A(Q_k)$$

Δ-Syn Query Model/2



Conclusions

- We provided paradigms for improving the performance of big-data-based IoT frameworks
- Analysis and trade-offs have been discussed as well
- We focused on the top-quality solution represented by data cube compression paradigms
- Many other compressiong paradigms to explore and to adapt to IoT frameworks

Compressing Big OLAP Data Cubes in Big Data Analytics Systems: New Paradigms and Future Research Perspectives

Alfredo Cuzzocrea
University of Calabria, Italy

Thanks for Your Attention!