The Data-as-a-Service Framework for Cyber-Physical-Social Big Data

Laurence T. Yang (杨天若)

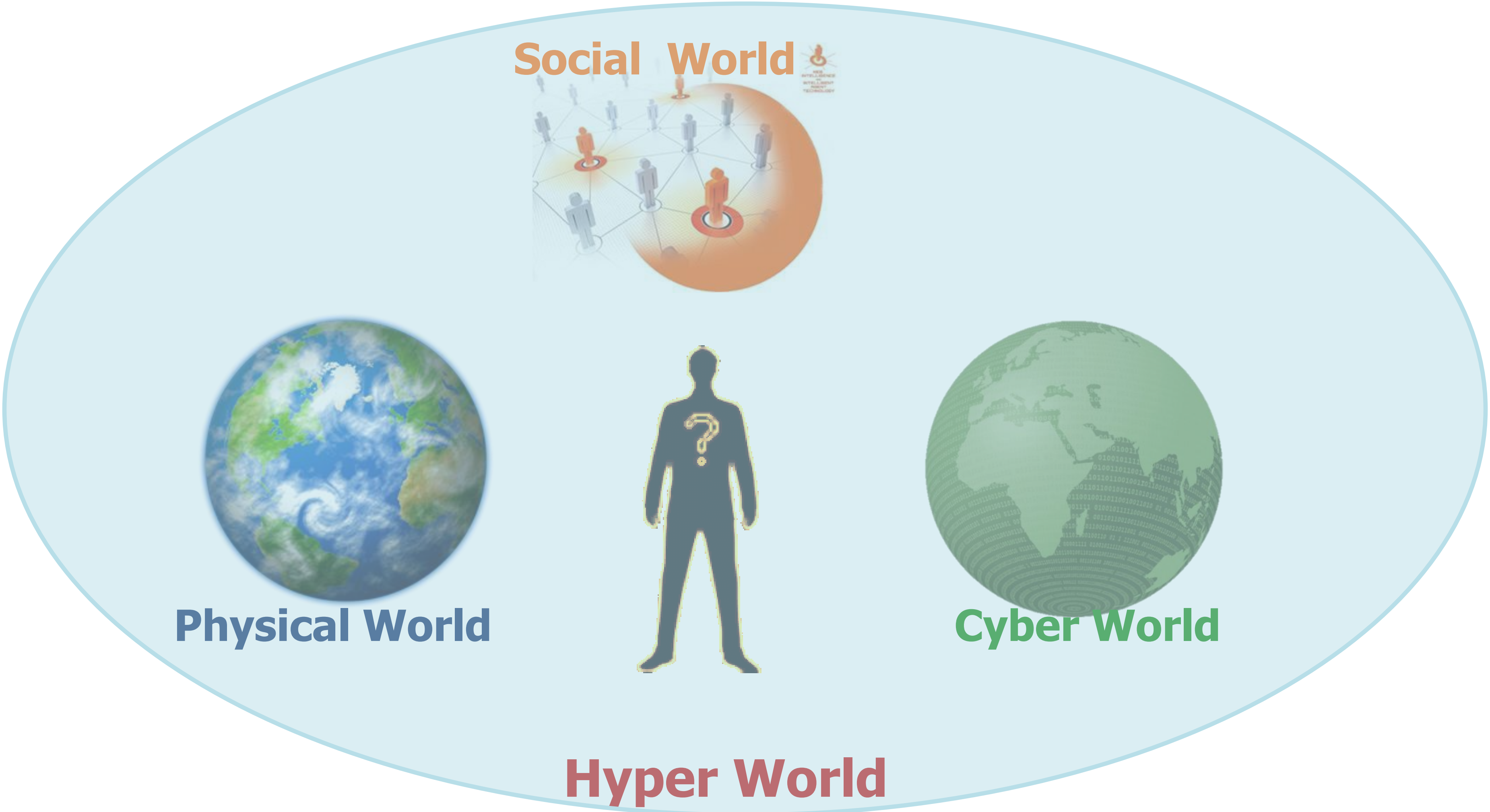
Huazhong University of Science and Technology, China

St Francis Xavier University, Canada

Outline

1. Hyper (Cyber-Physical-Social)World
2. Big Data and Challenges
3. A Tensor-Based Approach
4. Illustrations and Examples
5. Conclusion

Hyper (Cyber-Physical-Social) World

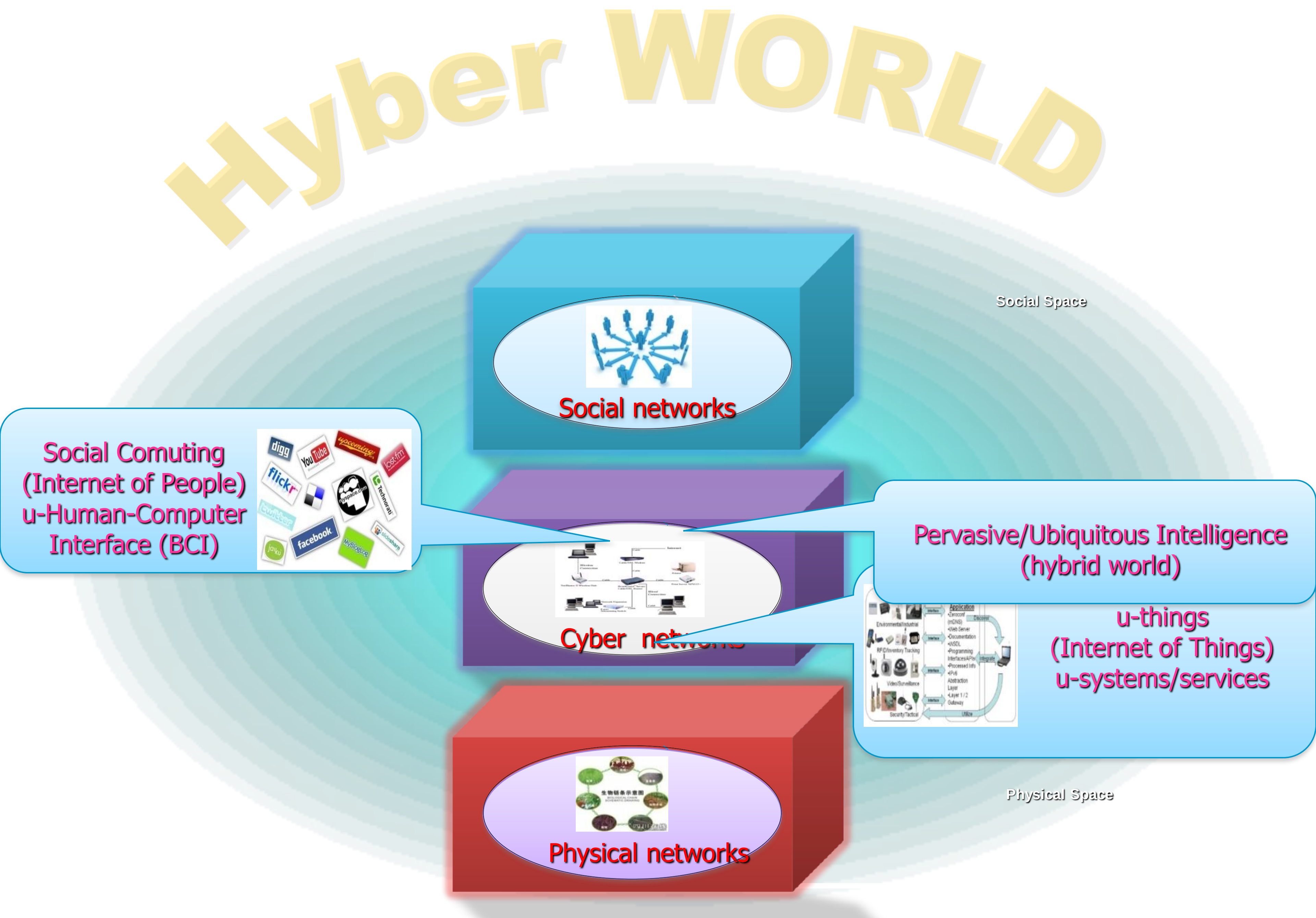


What we are and what we can do in the Hyper World?

Hyperspace

- Cyber – computation, communication, and control that are discrete, logical, and switched
- Physical – natural and human-made systems governed by the laws of physics and operating in continuous time
- Social – socially produced space, serves as a tool of thoughts, action, mind, sense, interaction, communications, etc.

Hyperspace = Cyber Space + Physical Space + Social Space


Hyber WORLD
Social Space
Social networks
Social Comuting
(Internet of People)
u-Human-Computer
Interface (BCI)
Pervasive/Ubiquitous Intelligence
(hybrid world)
Cyber networks
u-things
(Internet of Things)
u-systems/services
Physical Space
Physical networks

Characteristics of Hyperworld

"Its basic characteristic is direct mapping between virtual and real worlds via active devices including sensors, actuators, micro-machines, robots, etc."

-- *Kunii, Ma & Huang, Keynote "Hyperworld Modeling" in VIS'96*

- <u>Data/Info Explosion</u>! Last 5 years' web data > 5,000 years' whole data
- <u>Connection Explosion</u>! C2C → D2D → H2H → M2M → CPS → T2T (IoT)
- <u>Service/App Explosion</u>! WbS → SaaS → PaaS → IaaS → EaaS → Cloud
- <u>Intelligence Explosion</u>! Ubi-Sensing → CA → TTT → AmI → Smart W/P

Intelligent Computing Waves

- <u>1st: AI (Logic/KL-based, cyber-based)</u>
  - *Machine learning*
  - *NLP & Comp-Vision*
  - *Robot & game theory*
  - *Expert system*
  - *Knowledge/Reasoning*
  - *DAI (Distributed AI)*

- <u>2nd: Physical-based</u>
  - *Probabilistic computing*
  - *Fuzzy logic*
  - *Neural network*
  - *GA/Evolutionary computing*
  - *Chaotic/Swarm computing*
  - *Biologic computing*

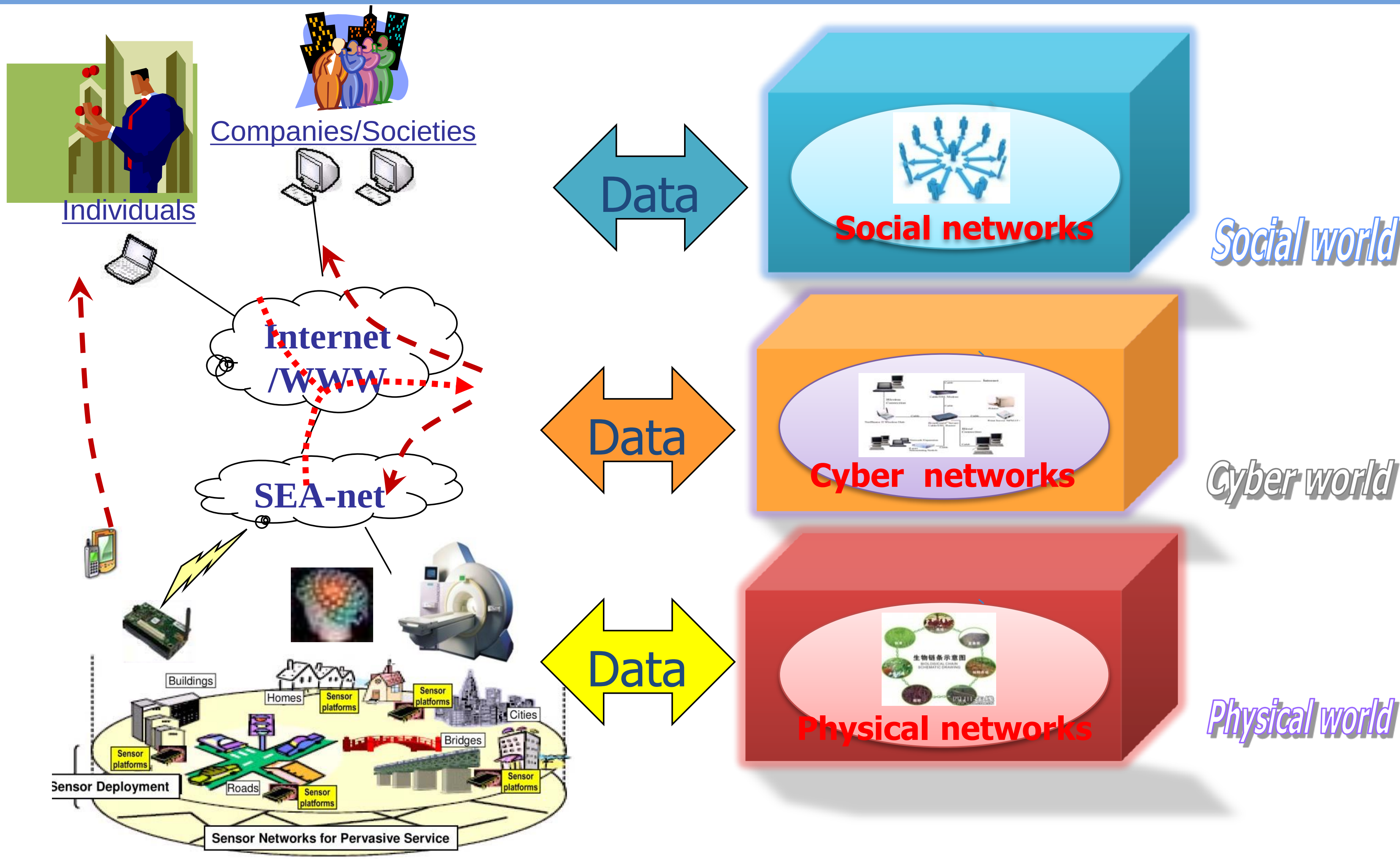
- <u>3rd: Social-based</u>
  - *Autonomous software*
  - *Multi/Massive agents*
  - *Agent language*
  - *Agent negotiation & cooperation*
  - *Personal/social behavior*
  - *Web intelligence/semantics*

- <u>4th: Cyber-Physical-Social</u>
  - *Physical/everyday things' intelligence*
  - *Atop of the above 3 intelligent comp*
  - *Scale, dynamic, heterogeneous, spontaneous*
  - *Predictable, controllable, adaptable, manageable, ethic, …*
  - *Others-aware & self-aware → mind/spirit?*

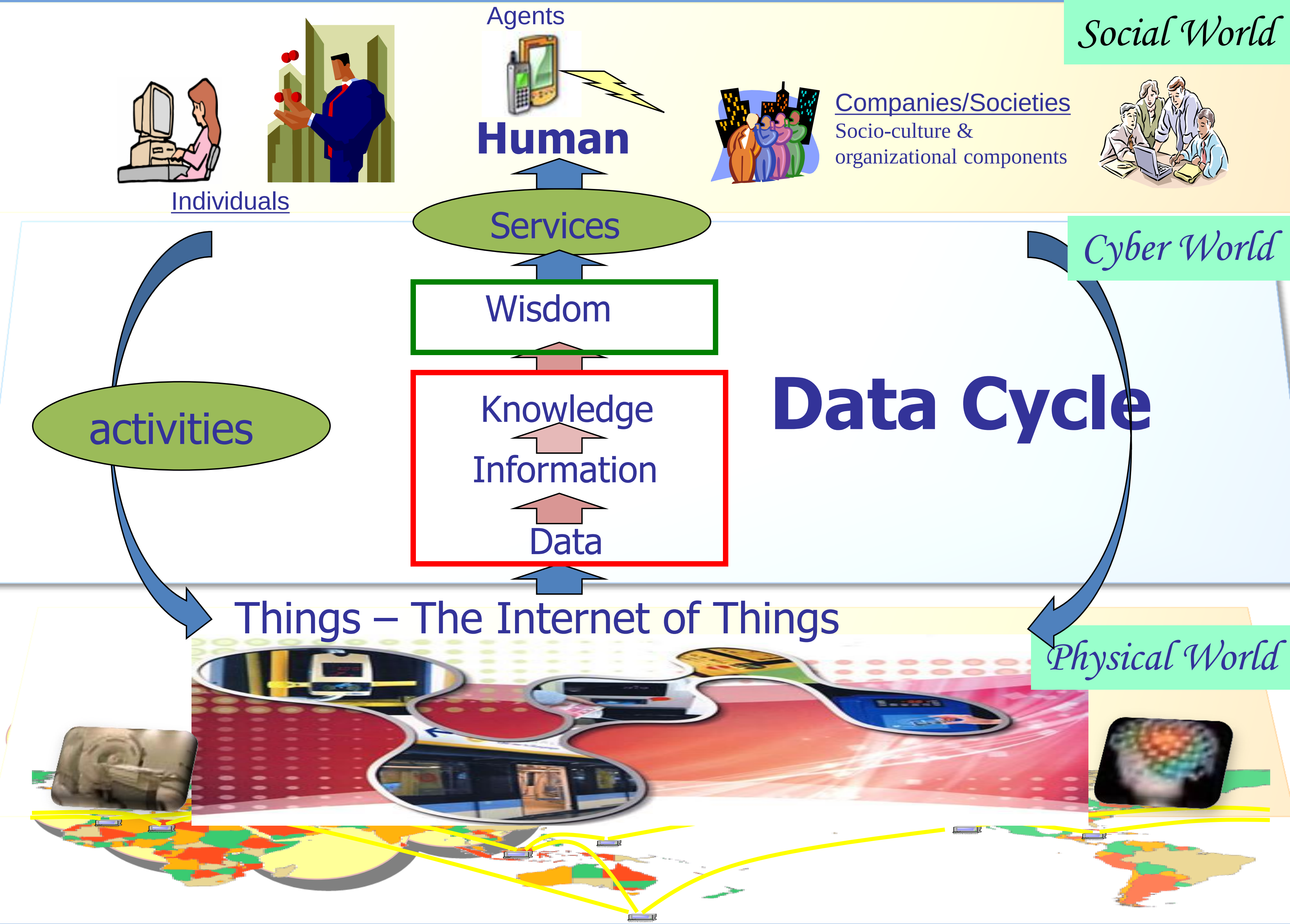
Four Fundamental Issues of HyperWorld

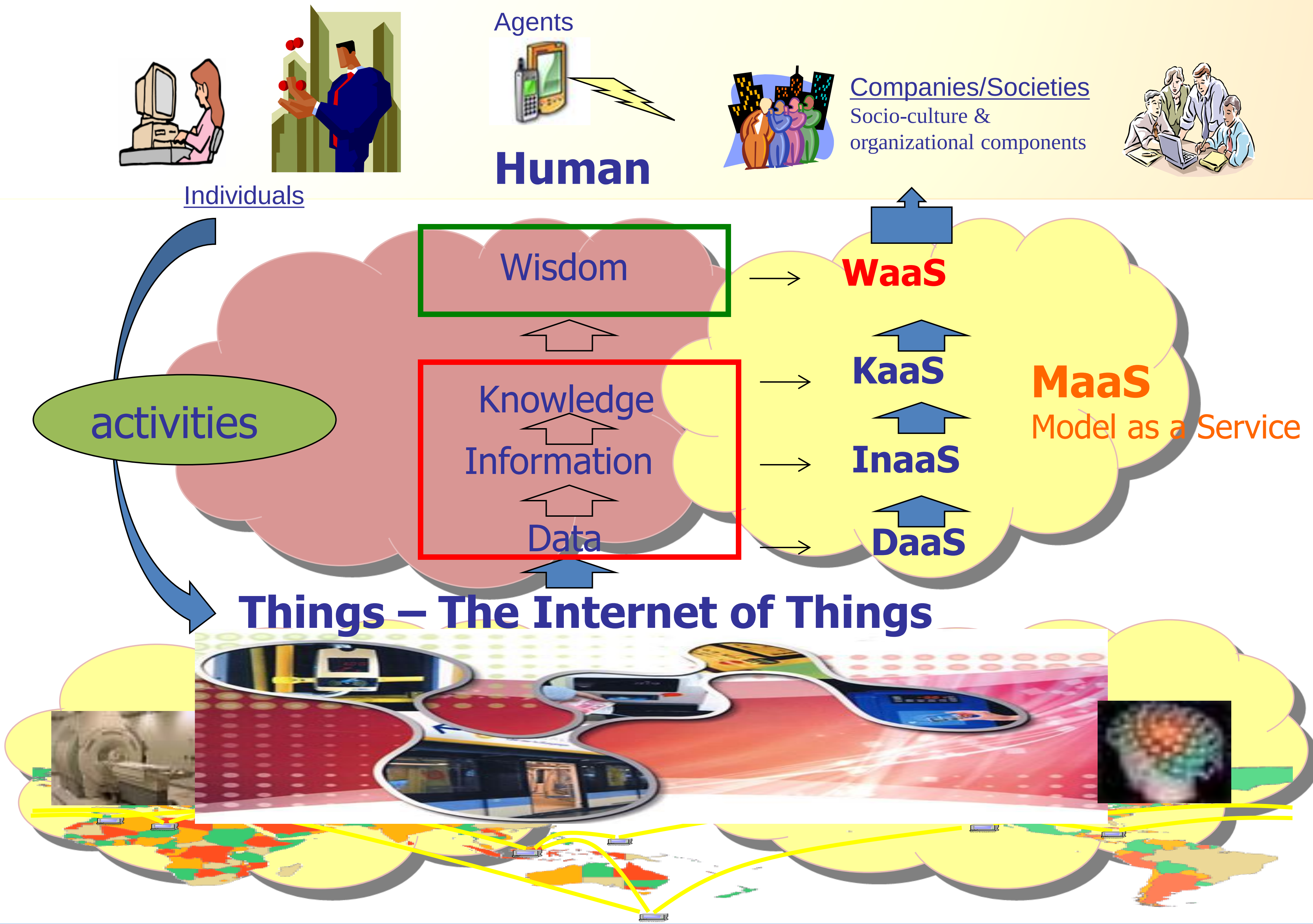
- How to realize the harmonious symbiosis of humans, information and things in the hyper world?
- How to implement the data-cycle system as a practical way to realize the harmonious symbiosis of humans, information and things in the hyper world?
- How to investigate intelligence from a holistic way in the hyper world?
- How to do unifying studies of humans, networks and information granularity in the hyper world?

HyperWorld



Companies/Societies
Individuals
Internet
/WWW
SEA-net
Buildings
Homes
Sensor platforms
Cities
Bridges
Roads
Sensor Deployment
Sensor Networks for Pervasive Service
Data
Data
Data
Social networks
Cyber networks
Physical networks
Social world
Cyber world
Physical world


Social World
Agents
Human
Individuals
Companies/Societies
Socio-culture &
organizational components
Services
Cyber World
Wisdom
Knowledge
Information
Data
Data Cycle
activities
Things – The Internet of Things
Physical World



Agents
Human
Individuals
Companies/Societies
Socio-culture &
organizational components
Wisdom
WaaS
Knowledge
KaaS
Information
InaaS
Data
DaaS
MaaS
Model as a Service
activities
Things – The Internet of Things

Outline

Big Data

What’s the difference between large data and big data?

Just Size?

Features of Big Data

4Vs Features

■**Volume**

Terabytes, Petabytes,…

■**Variety**

Structured
Semi-Structured
Un-Structured

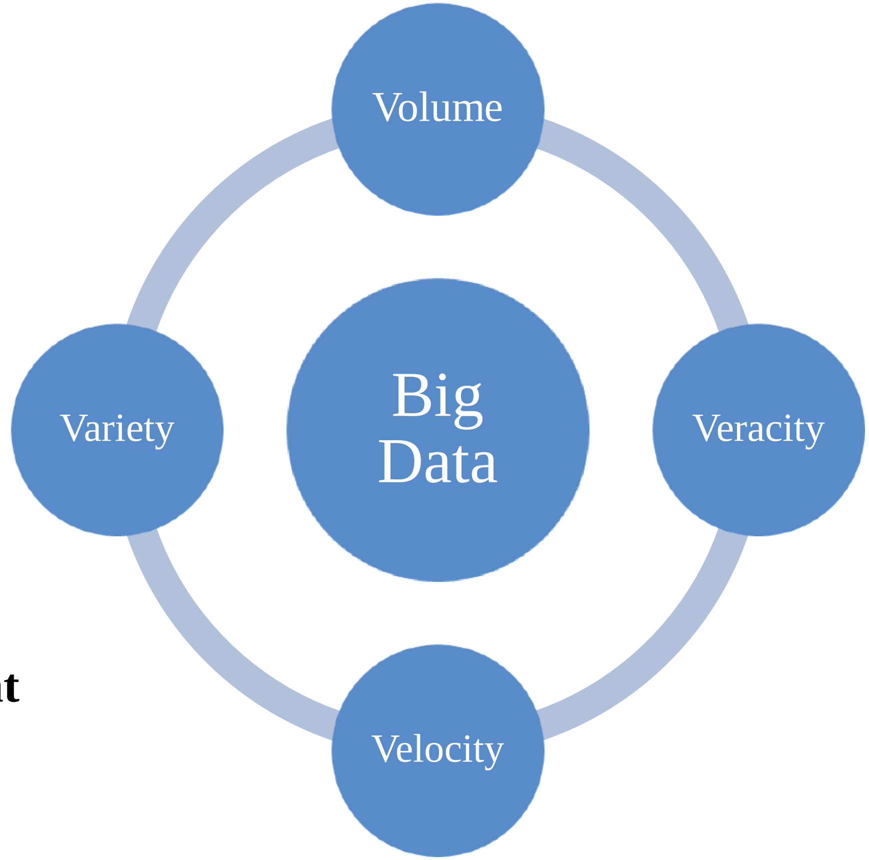
■**Velocity**

Streaming Data

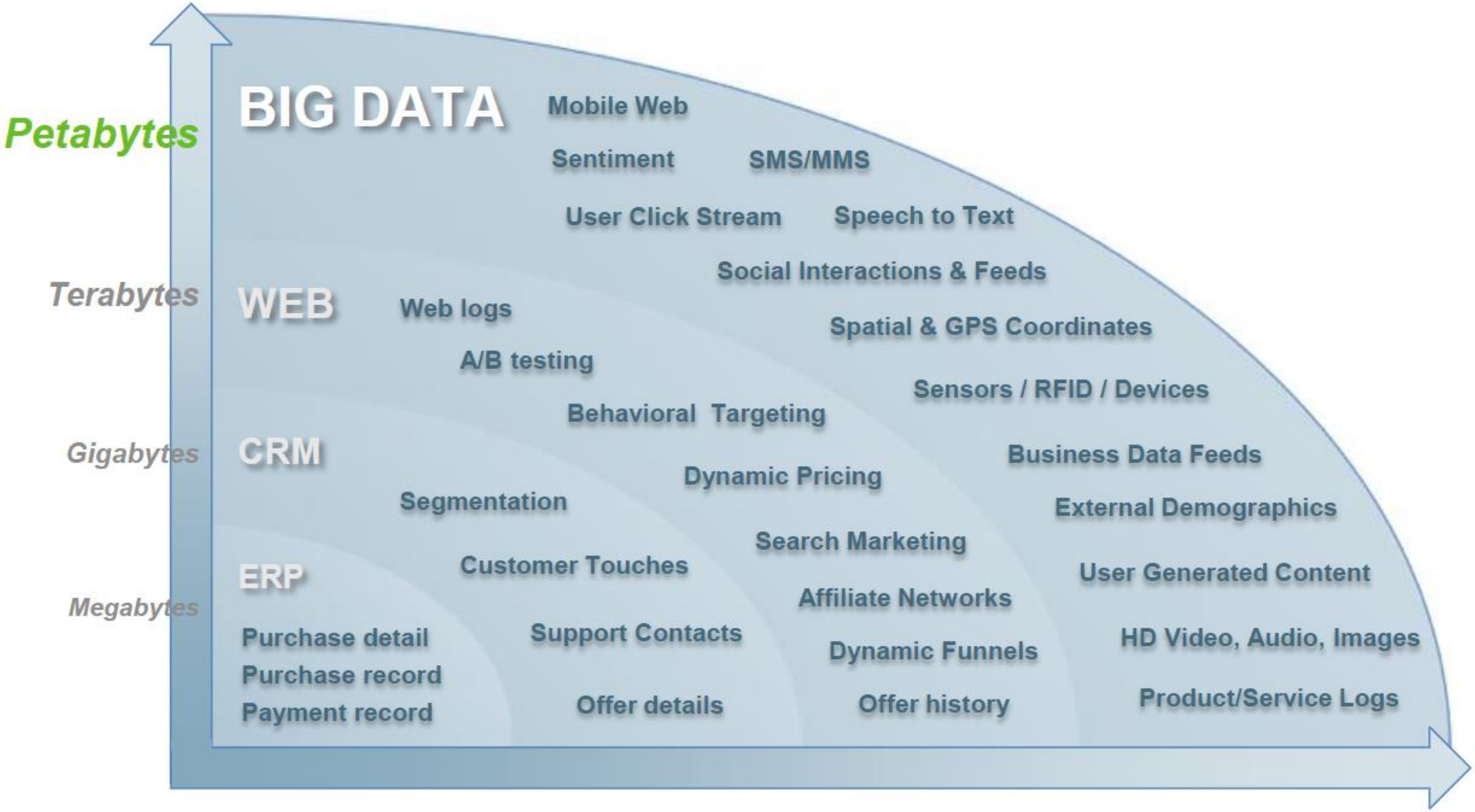
Processing time requirement

■**Veracity**

Quality of Data



Cyber-Physical-Social Data



Challenges Caused by Big Data

Volume	• Large Scale Data , Limited Compute Resources
Variety	• Diversified Format, Unified Representation
Velocity	• Requirement for quick processing
Veracity	• Inconsistent, Incomplete, Redundant Data

How to tackle the problems caused by the 4Vs of Big Data?

Outline

1. Smart Word and Hyper World
2. Big Data and Challenges
3. A Tensor-Based Framework
4. Illustrations and Examples
5. Conclusion

A Tensor-Based Approach

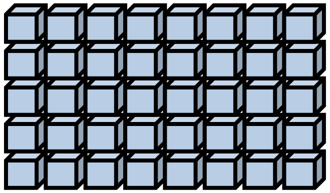
3.1	• Tensor and Tensor Decomposition
3.2	• ‘4Vs Problem’ with Tensor
3.3	• A Framework for Big Data Analysis and Mining

Tensor Definition

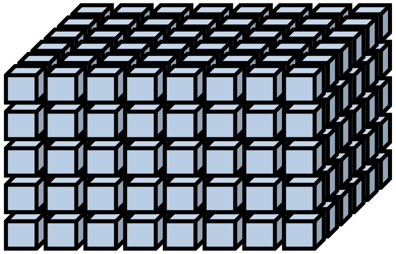
Mathematics: Tensor

Computer Science: Multiple Dimensional Arrays

One Order Tensor: **Vector**

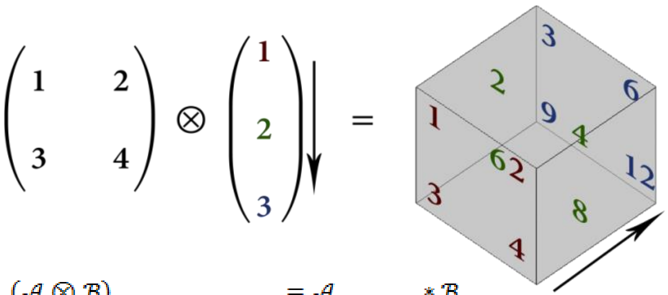


Two Order Tensor: **Matrix**



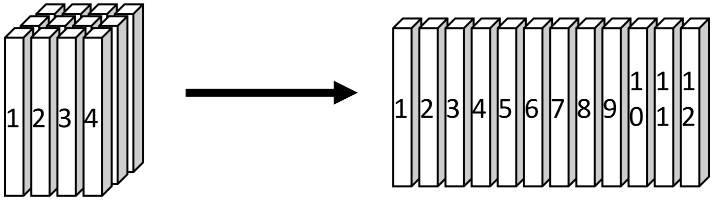
Three Order **Tensor**

Tensor Computations and Algebra



$(\mathcal{A}\otimes\mathcal{B})_{i_1,i_2,\dots,i_m,j_1,j_2,\dots,j_n} = \mathcal{A}_{i_1,i_2,\dots,i_m} * \mathcal{B}_{j_1,j_2,\dots,j_n}$

- Kronecker product.
- Tensor-tensor product
- Inverse
- Identity tensor
- Transpose
- Tensor-matrix mode product
- Stefan Ragnarsson, Ph.D thesis, 2012, Cornell

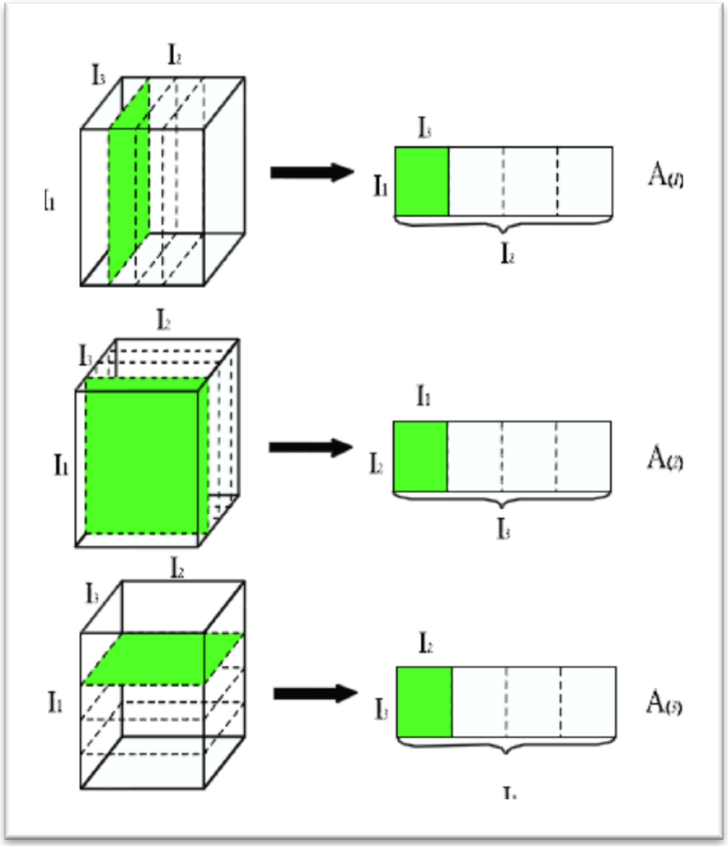


Unfolding on mode 1 (take mode 1 fibers and linearize).

$$\mathcal{A}\times_i \mathcal{B} = fold_i(\mathbf{B} * unfold_i(\mathcal{A}))$$

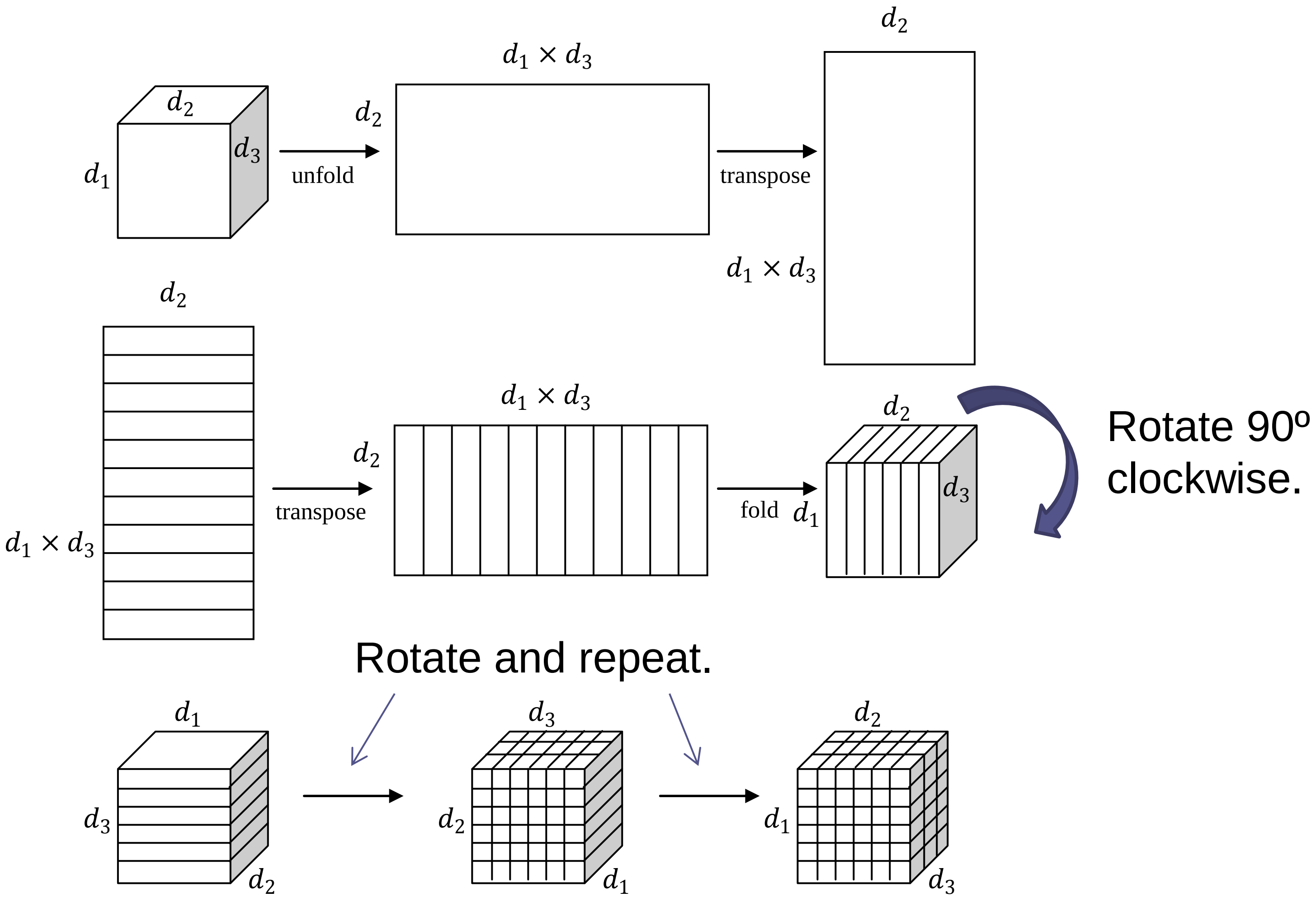
Equivalently: $(\mathcal{A}\times_i \mathbf{B})_{j_{1:i-1},k,j_{i+1:r}} = \sum_{\ell=1}^{d_i} \mathcal{A}_{j_{1:i-1},\ell,j_{i+1:r}} \mathbf{B}_{k,\ell}$

Illustration



Model-N Unfolding

Illustration

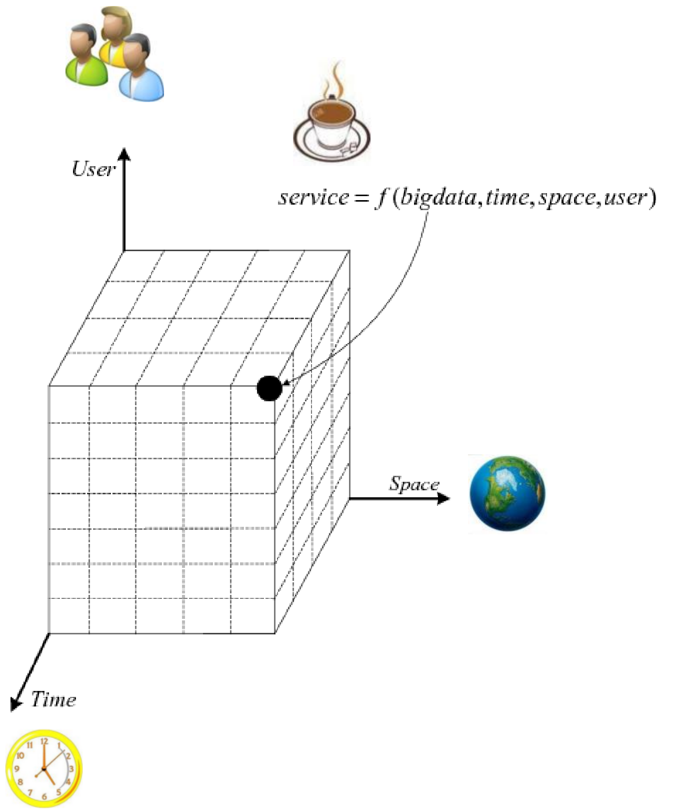


Formal Description

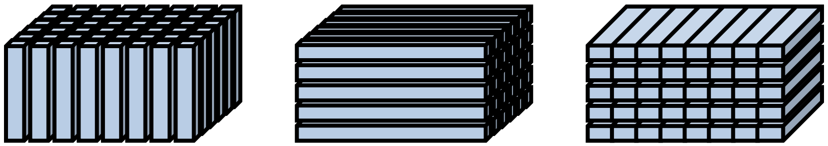
$$A \in R^{I_t \times I_x \times I_y \times I_z \times I_c \times I_u}$$

$$a_{ijklmn}, i \in I_t, j \in I_x, k \in I_y, l \in I_z, m \in I_c, n \in I_u$$

where $\Re$ is defined on the real number domain; $I_t, I_x, I_y, I_z, I_c, I_u$ refer to the dimensions on time, space coordinates (I_x, I_y, I_z) and cyber resources. $I_t \times I_x \times I_y \times I_z \times I_c \times I_u$ denotes the Cartesisan product operation of $I_t, I_x, I_y, I_z, I_c, I_u$. The value of each

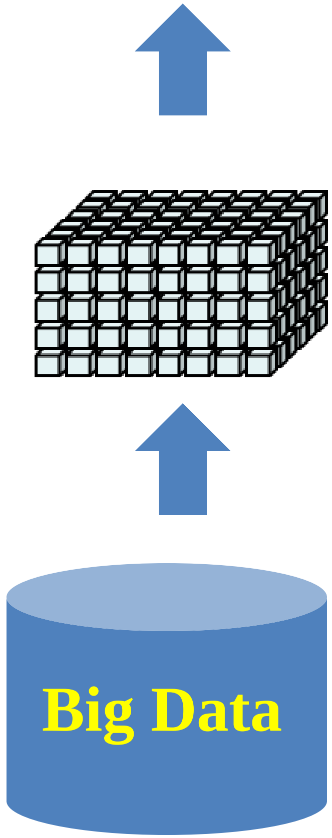


Tensor Decomposition



Tensor decomposition is an important data processing tool.

Big data are modeled as tensors



Decomposition Process

Data | SVD | Matrix data | SVD | Approximation

Data

$A_{(1)}$

$A_{(2)}$

…

$A_{(k)}$

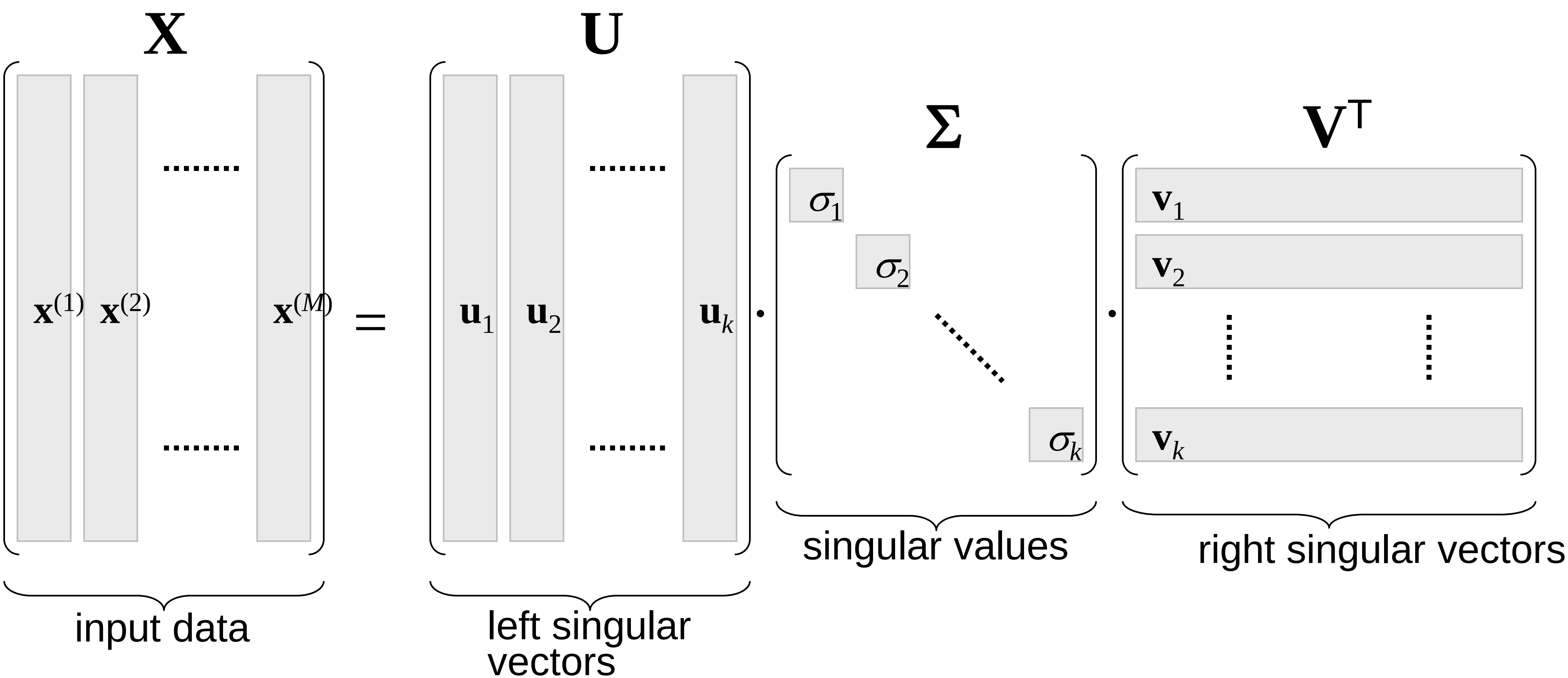
$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

$$A = U\Sigma V^{\mathrm{T}}$$

Coresets

Singular Value Decomposition (SVD)

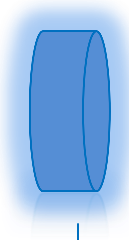
$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathsf{T}}$$



@Carnegie Mellon Databases

Incremental SVD

$$A_0 = U \Sigma V^{\mathrm{T}}$$

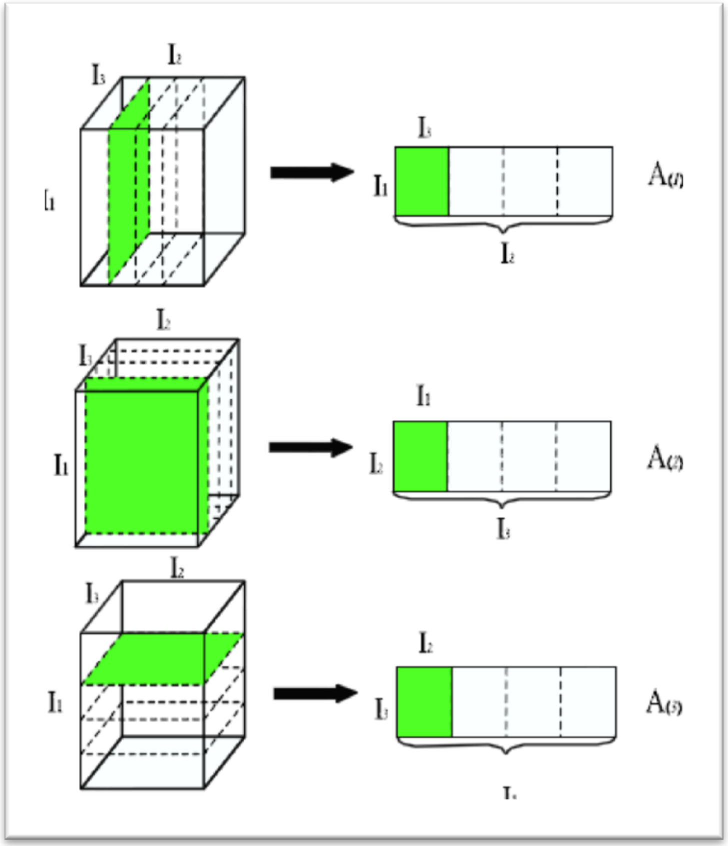


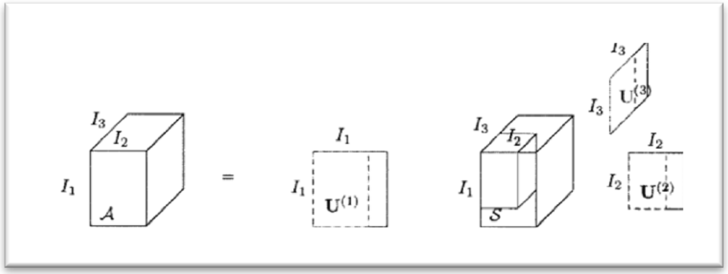
Update the singular vectors and singular values by using the incremental decomposition results of matrix A1

Incremental Matrix

$$\begin{bmatrix} A_0 & A_1 \end{bmatrix} = \underbrace{\begin{bmatrix} U & J \end{bmatrix} U'}_{\text{Updated U}} \underbrace{\Sigma'}_{\text{Updated } \Sigma} \underbrace{V'^{\mathrm{T}} \begin{pmatrix} V & 0 \\ 0 & I \end{pmatrix}^{\mathrm{T}}}_{\text{Updated V}}$$

HO-SVD Tensor Decomposition





Approximation Assemble

Model-N Unfolding

Approximate Tensor Calculation

Matrix Unfolding

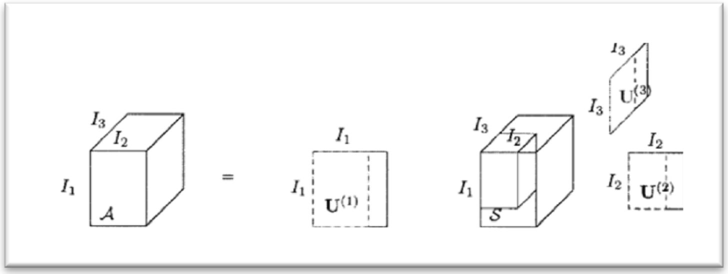
$$A_1 \in R^{I_1 \times I_2 I_3}$$

$$A_2 \in R^{I_2 \times I_1 I_3}$$

$$A_3 \in R^{I_1 I_2 \times I_3}$$

SVD for Every Unfolding Matrix

$$A_i = U_i \Sigma V_i^{\mathrm{T}}$$



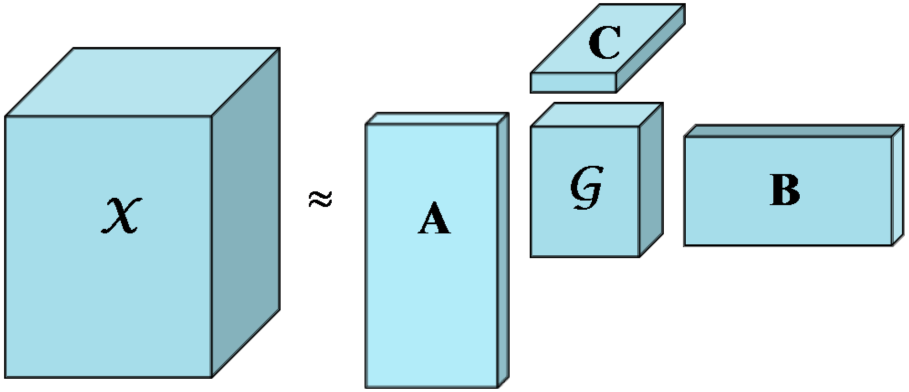
Core Tensor

$$S = A \times_1 (U^{(t)}_{c_1})^T \times_2 (U^{(x)}_{c_2})^T \times_3 (U^{(y)}_{c_3})^T \times_4 (U^{(z)}_{c_4})^T \times_5 (U^{(c)}_{c_5})^T \times_6 (U^{(u)}_{c_6})^T$$

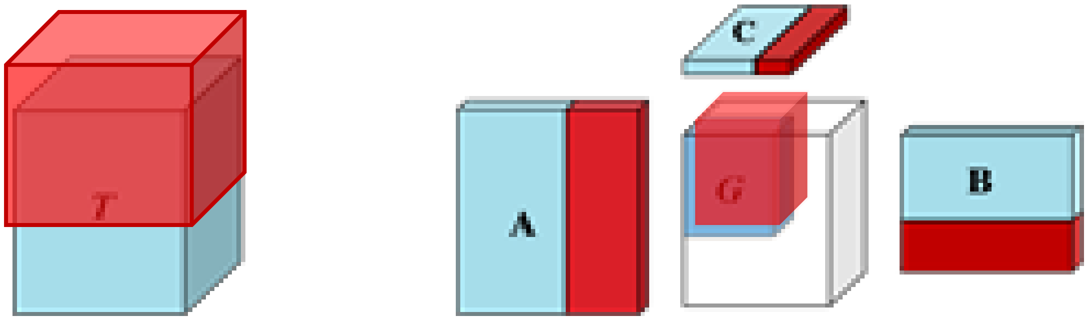
Approximate Tensor

$$\tilde{A} = S \times_1 U^{(t)}_{c_1} \times_2 U^{(x)}_{c_2} \times_3 U^{(y)}_{c_3} \times_4 U^{(z)}_{c_4} \times_5 U^{(c)}_{c_5} \times_6 U^{(u)}_{c_6}$$

Incremental HOSVD



Original SVD

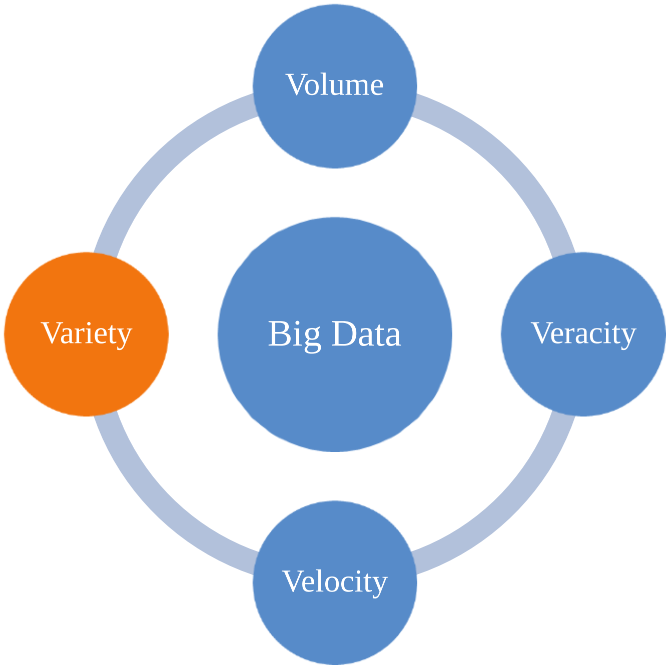


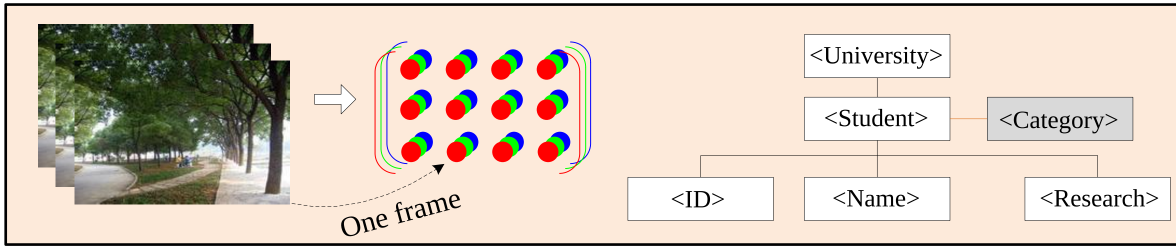
Incremental SVD

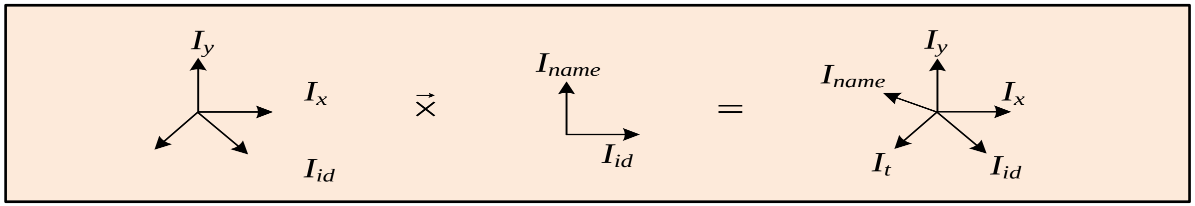
A Tensor-Based Approach

3.1	• Tensor and Tensor Decomposition
3.2	• ‘4Vs Problem’ with Tensor
3.3	• A Framework for Big Data Analysis and Mining

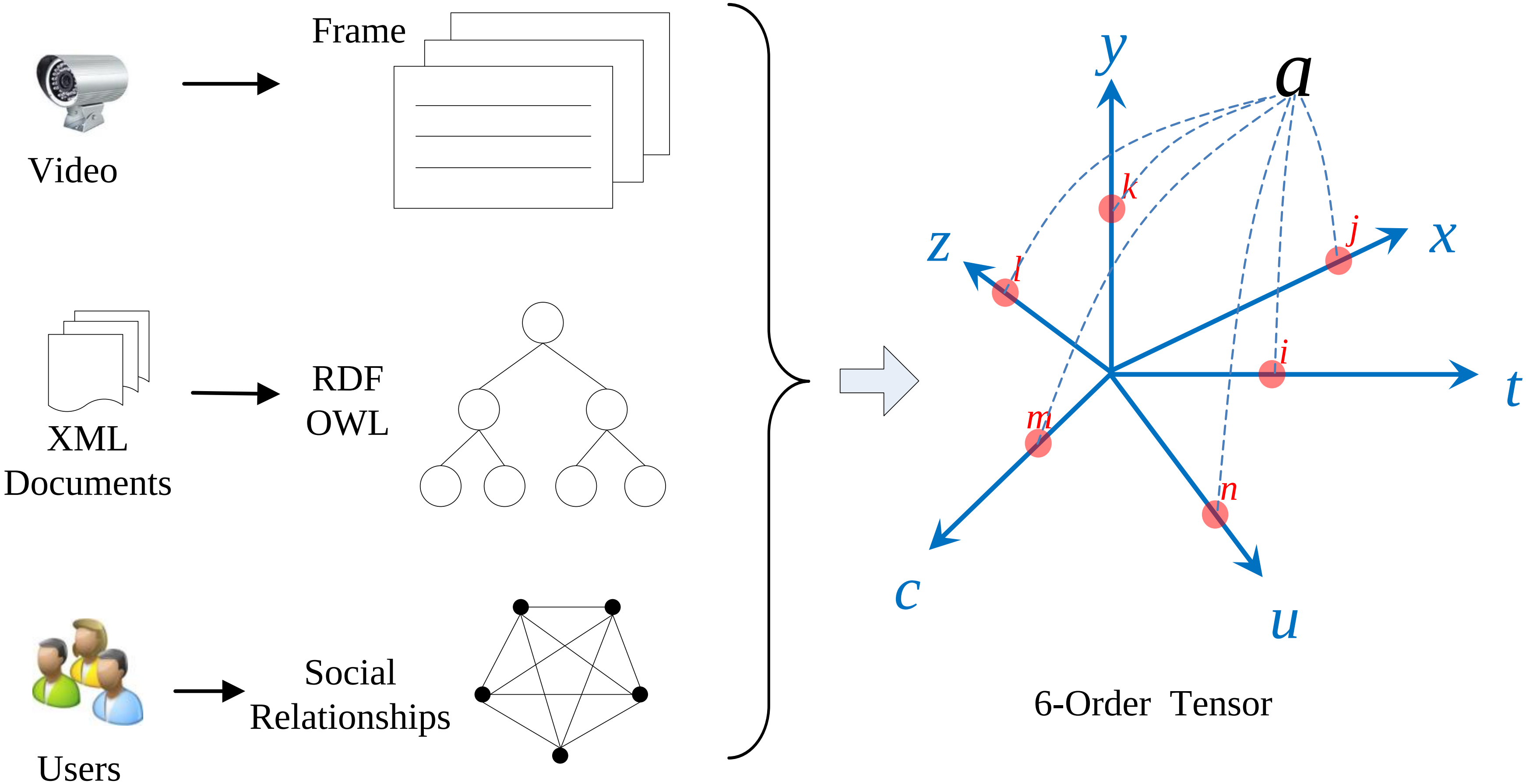
Variety





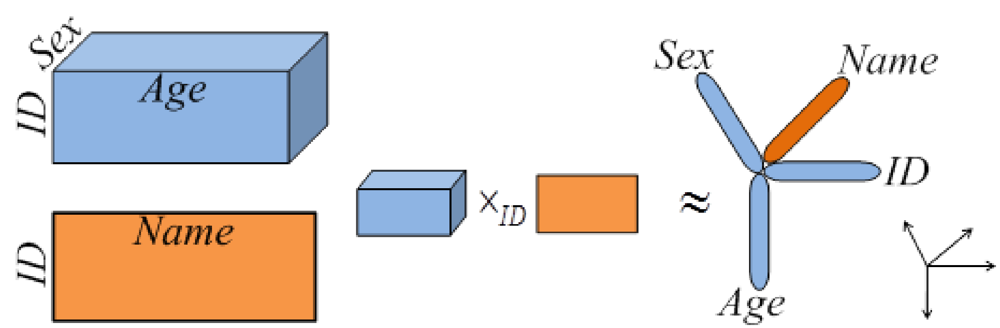


$$f : (d_u \cup d_{semi} \cup d_s) \rightarrow \underbrace{T_u \cup T_{semi} \cup T_s}_{T}$$

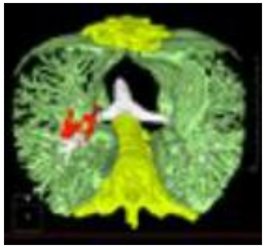


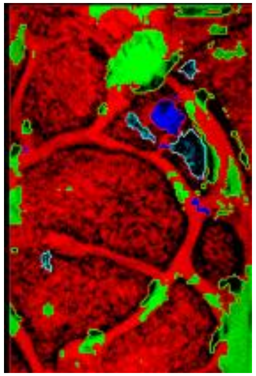
ID	Age	Sex
1	31	M
2	29	F
3	6	M

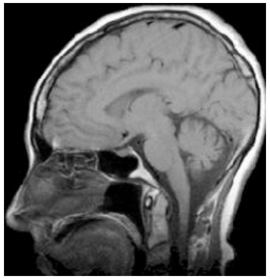
ID	Name
1	Charles
2	Alice
3	Bob

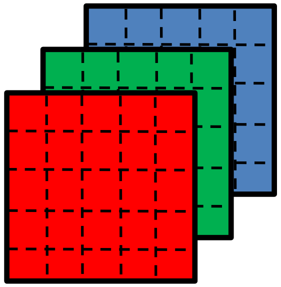


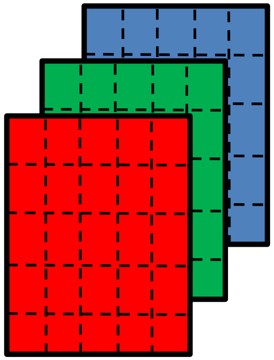
Variety

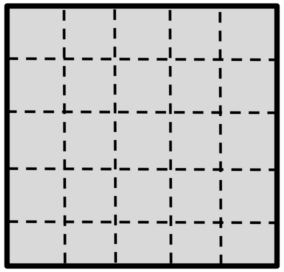


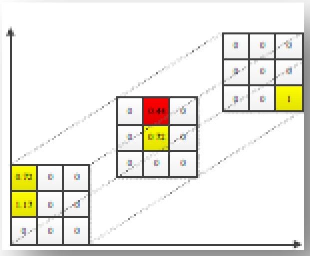


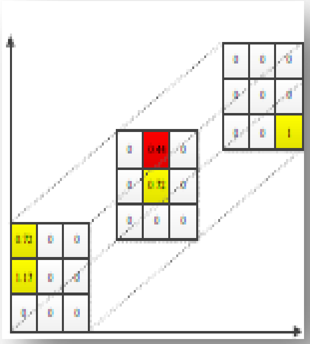












$$\begin{pmatrix} 201 & 76 & 44 \\ 234 & 123 & 67 \\ 9 & 134 & 84 \end{pmatrix}$$

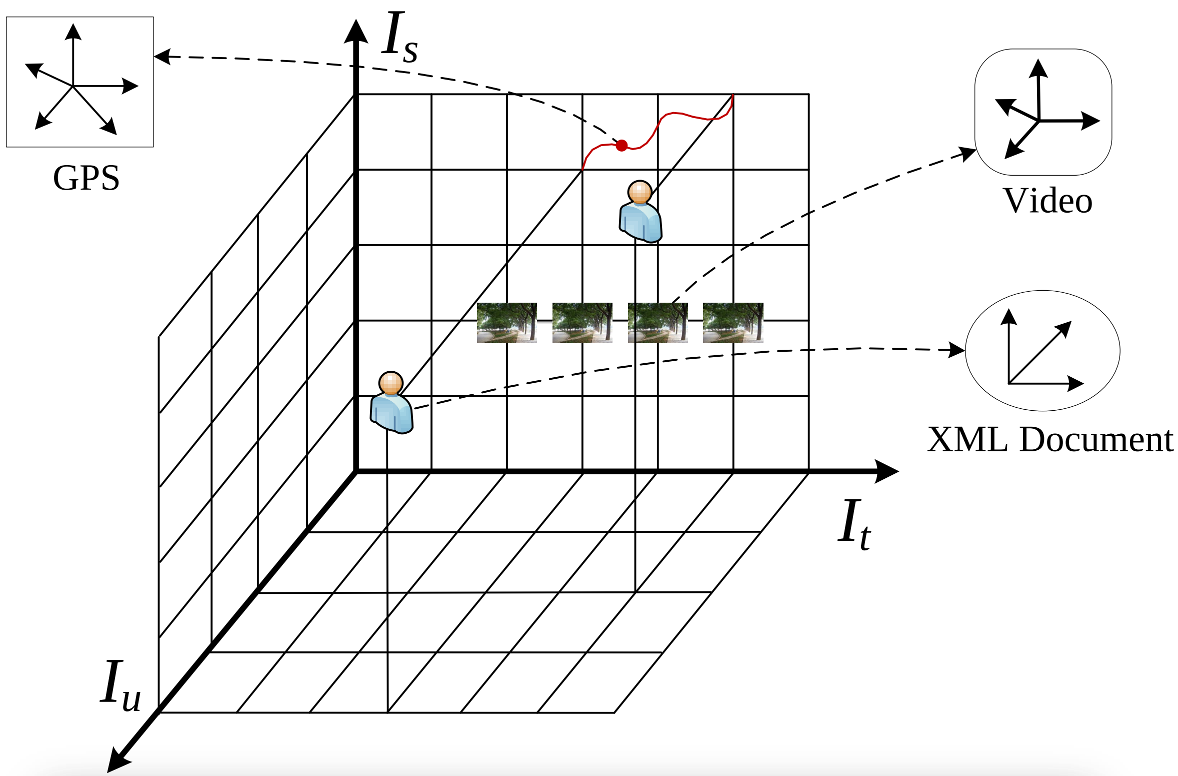
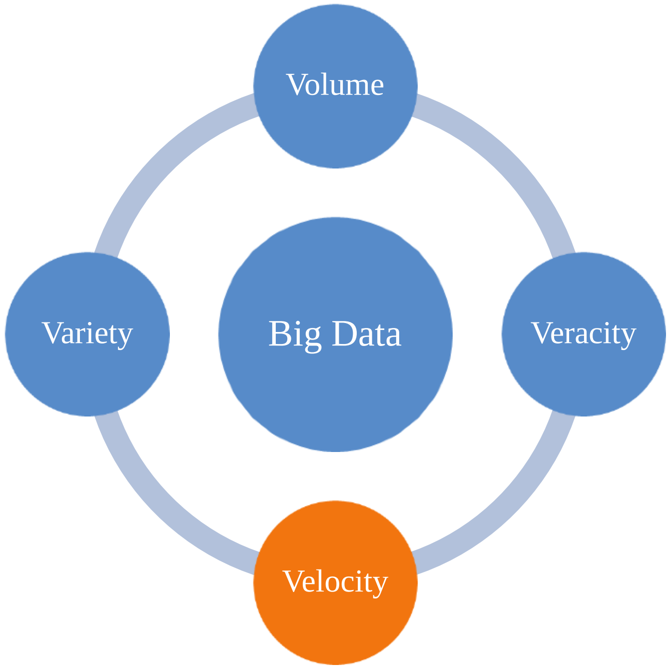
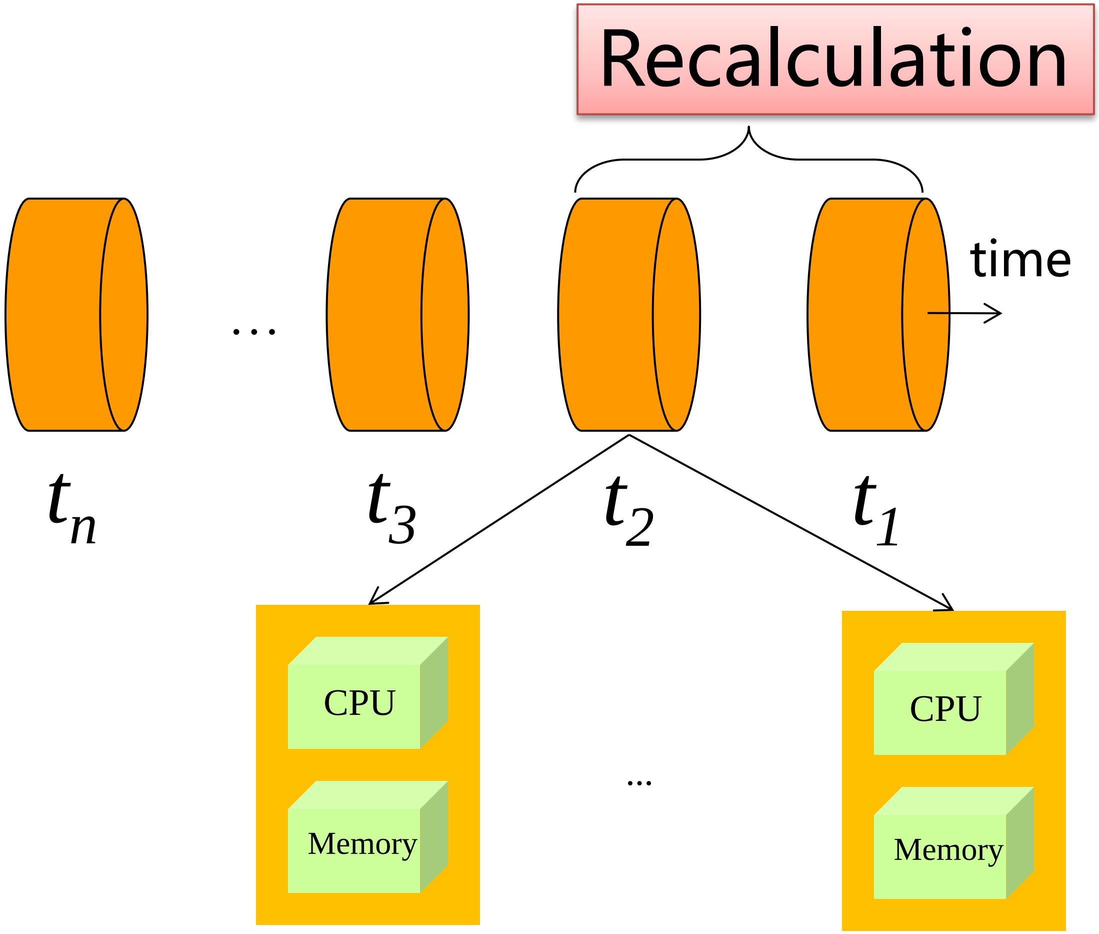


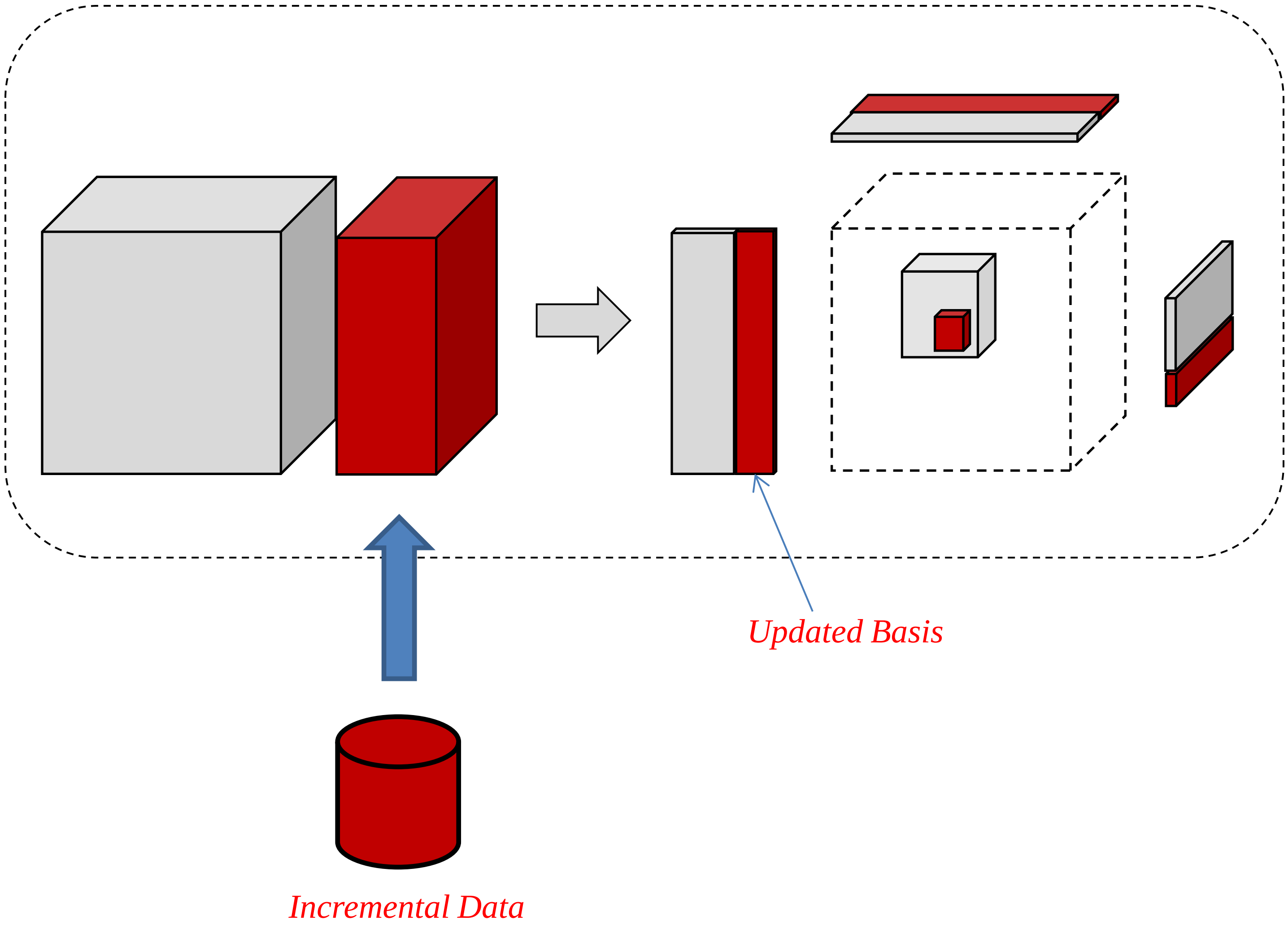
Fig. 8. Using the tensor extension operator, the sub-tensors are integrated together to a higher unified tensor.

Unified Representation for Big Data

Velocity


Volume
Variety
Big Data
Veracity
Velocity


Recalculation
time
…
t_n
t_3
t_2
t_1
CPU
Memory
…
CPU
Memory



Updated Basis
Incremental Data

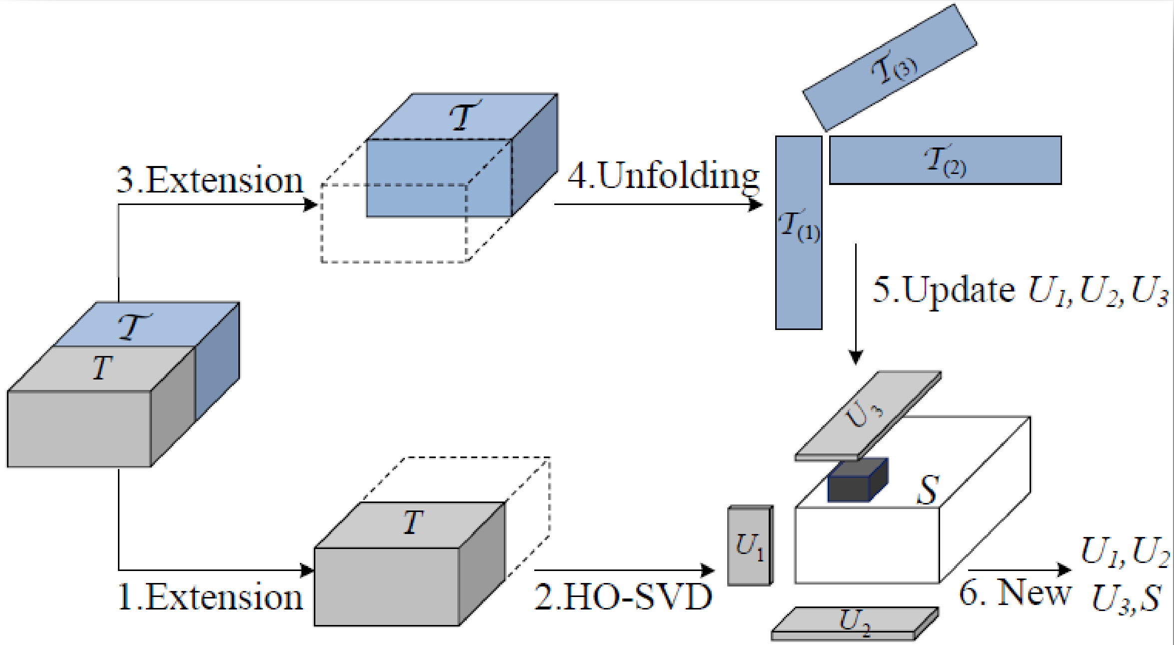
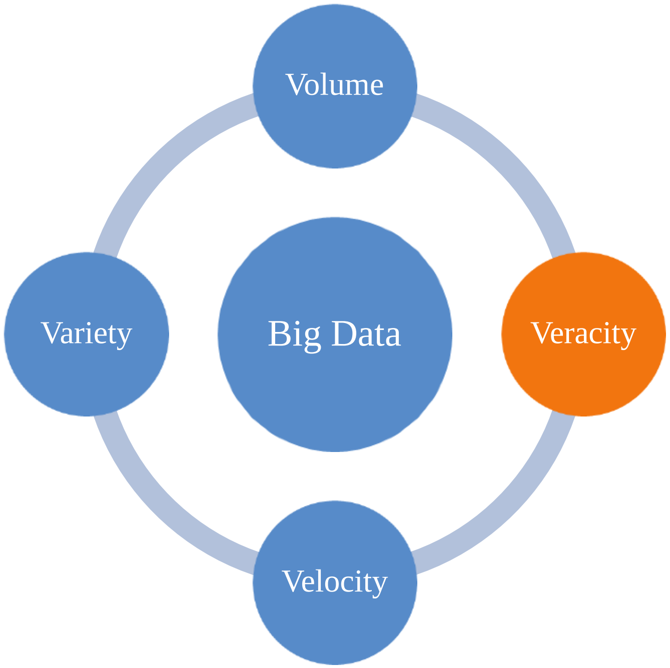
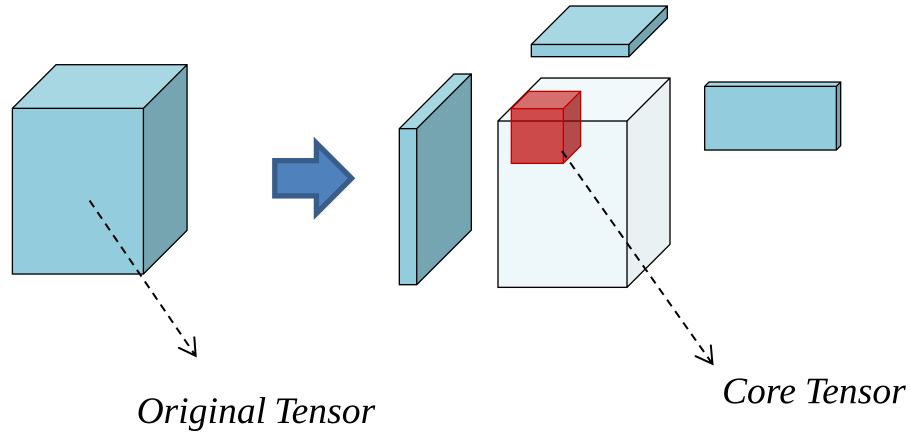


Fig. 10. Example of incremental tensor decomposition, truncated orthogonal bases U_1, U_2, U_3 of new tensor $\mathcal{T}$ are updated incrementally.

Veracity





Extract High Quality Core Tensor from PB(TB) Level Medial Tensor

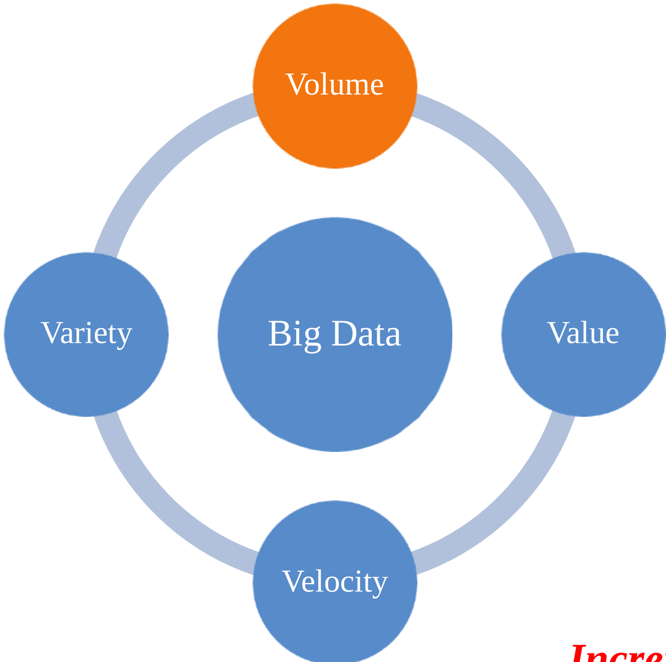
Error Constraints

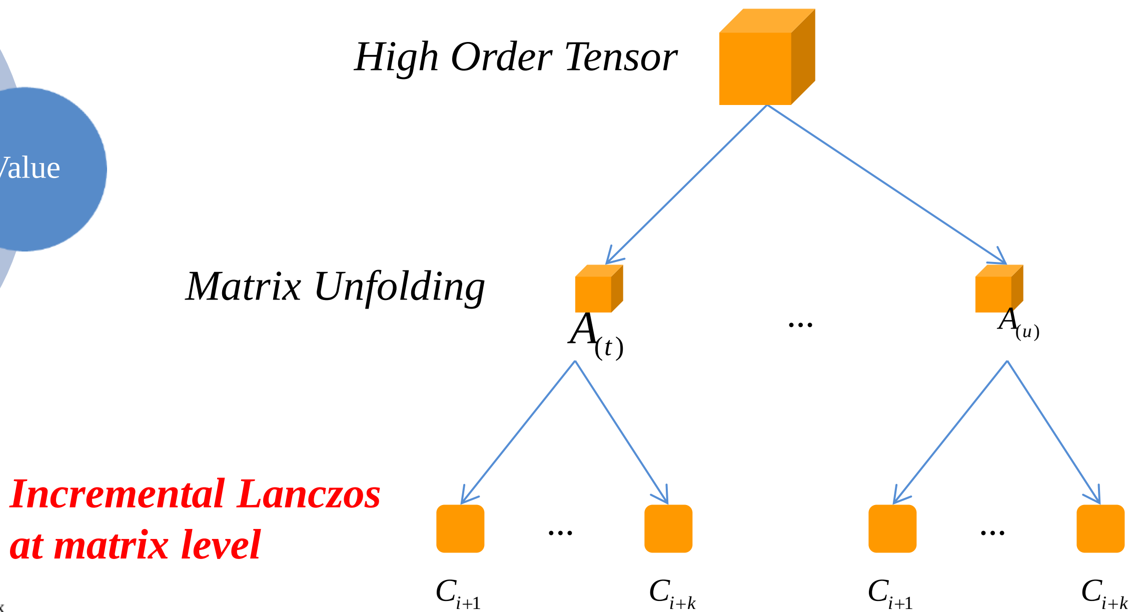
$$\min_{\chi\in\lambda(A)} |\theta_i-\chi| \le |\beta_k|\cdot|s_{ki}|, i=1:k$$

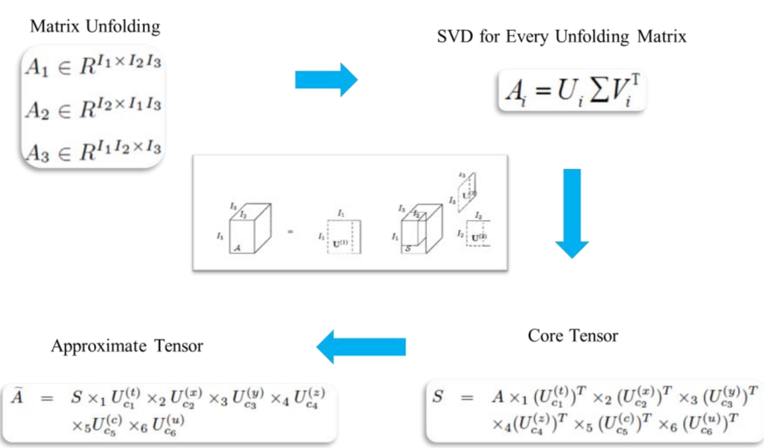
$$\|T\|_F = \Big(\sum_{ijklmn} a_{ijklmn}^2\Big)^{\frac{1}{2}}$$

$$\max_{U_i} \left\|T\times_t U_t^{\mathrm{T}}\times_x U_x^{\mathrm{T}}\times_y U_y^{\mathrm{T}}\times_z U_z^{\mathrm{T}}\times_r U_r^{\mathrm{T}}\times_u U_u^{\mathrm{T}}\right\|_F^2$$

Volume





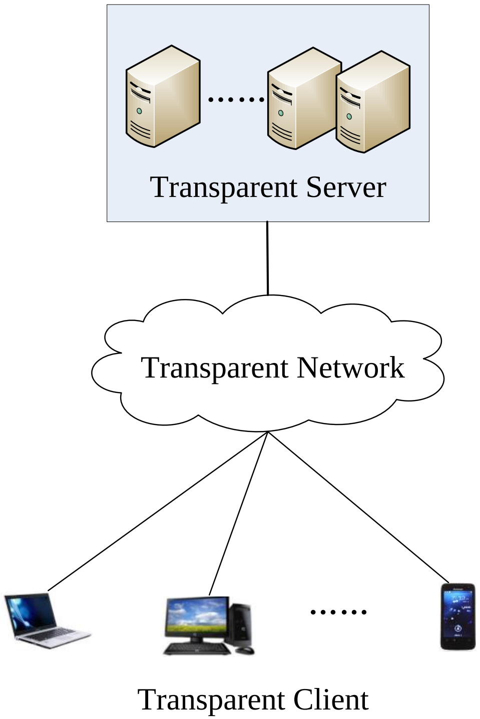


Two layers incremental SVD parallel strategies on servers

Transparent Computing vs. Cloud Computing

Computation in sever:

- Synthetic SVD Results
- Compute Core Tensor

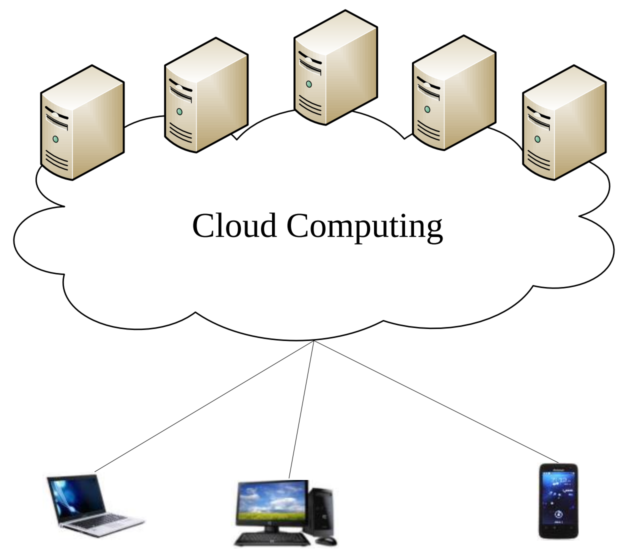


Computation in client:

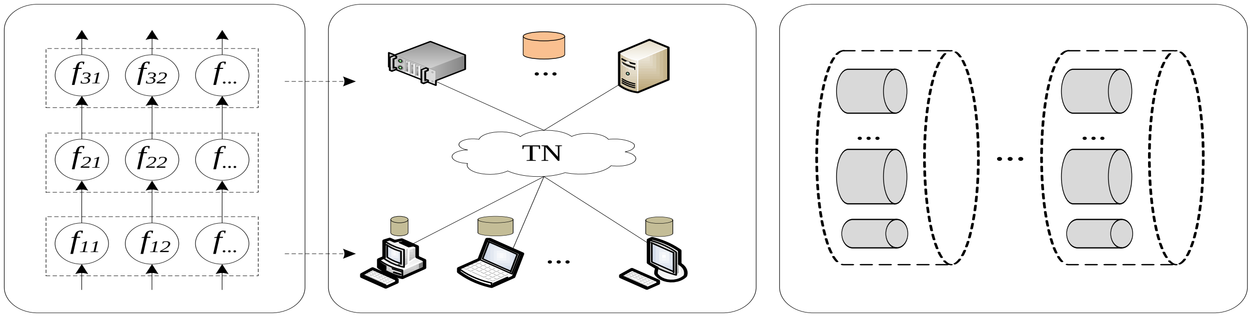
- SVD
- Incremental SVD

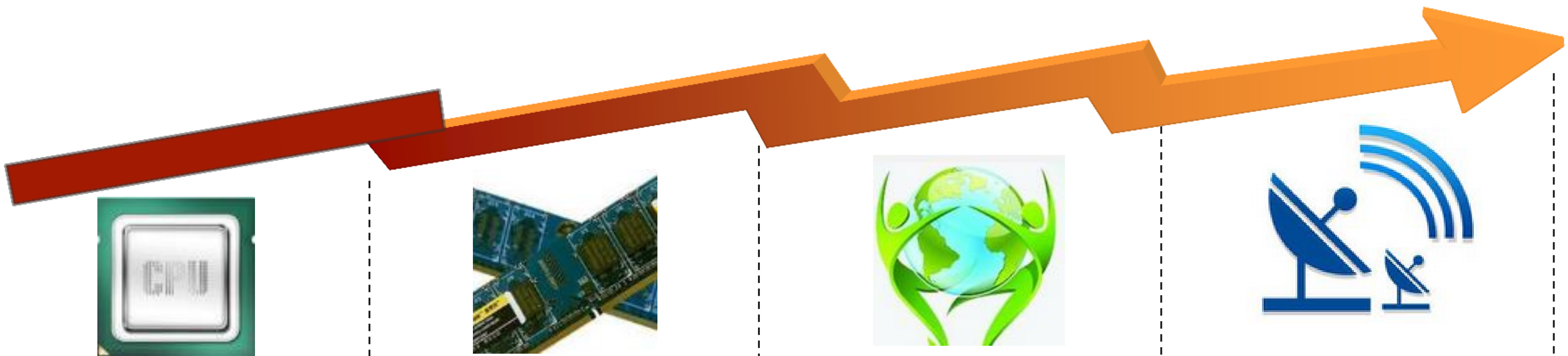
All computation are performed in sever

- SVD
- Incremental SVD
- Compute Core Tensor



Distributed Computing with Multi-Objective Optimization





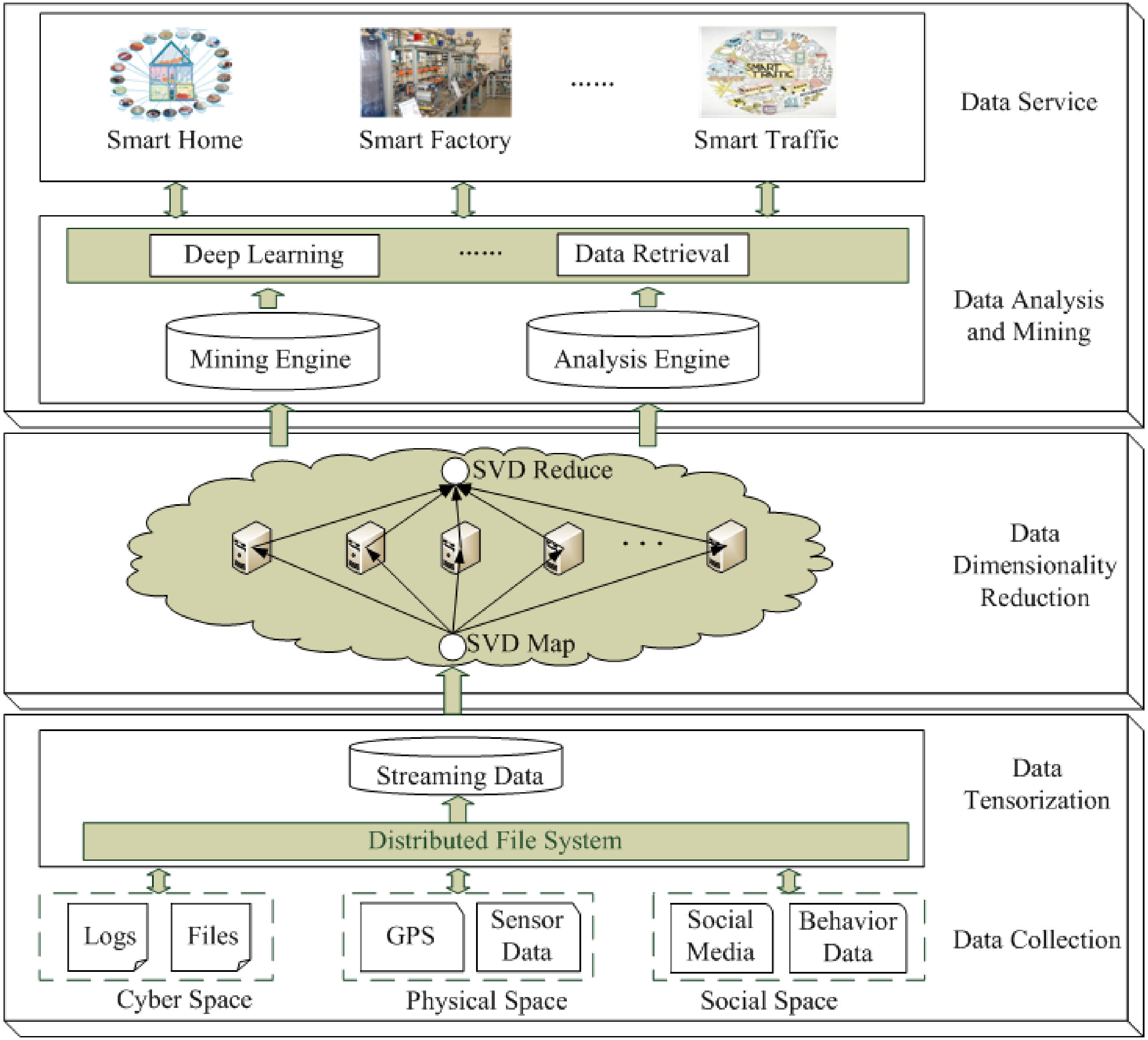
Computation Time | **Memory Usage** | **Energy Consumption** | **Communication Traffic**

$$OPT: minZ = \beta_1 Tim + \beta_2 Mem + \beta_3 Egy + \beta_4 Tra$$

A Tensor-Based Approach

3.1	• Tensor and Tensor Decomposition
3.2	• ‘4Vs Problem’ with Tensor
3.3	• A Framework for Big Data Analysis and Mining

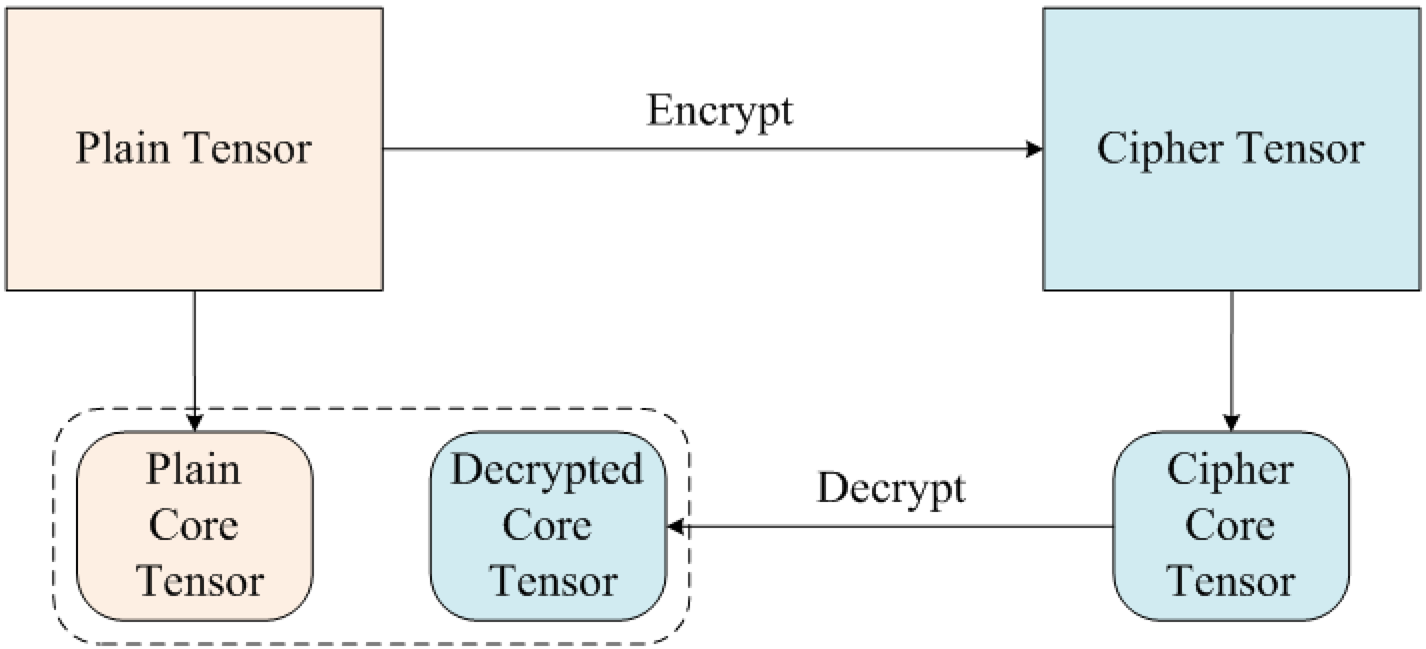
Big Data Framework



The Framework

1. Secure Dimensionality Reduction
2. A Deep Computation Model
3. Big Data Ranking and Retrieval
4. A Promising Technique: Fuzzy Tensor

1. Secure Dimensionality Reduction



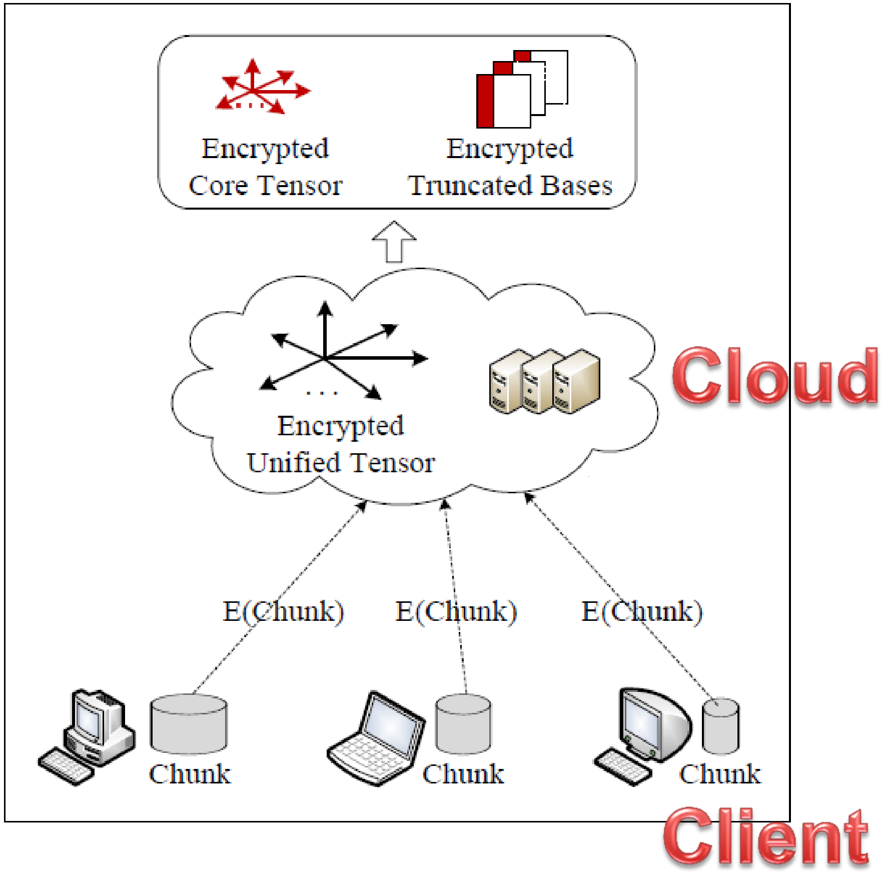
Paillier Encryption. Paillier encryption is a partially homomorphic encryption scheme which is more efficient than the fully homomorphic schemes. The encryption and decryption procedure are defined as follows:

$$c = g^m \cdot r^n \bmod n^2,$$
$$m = \frac{L(c^\lambda \bmod n^2)}{L(g^\lambda \bmod n^2)} \bmod n.$$

The public key cryptosystem is based on the composite degree residuosity classes. It preserves homomorphic addition and multiplication that can be described as

$$E(x) + E(y) = E(x+y),$$
$$E(xy) = (E(x))^y \bmod n^2.$$

No.	Operation	Homomorphic
1	$+$	yes
2	$-$	yes
3	$\times$	yes
4	$\div$	no
5	$\sqrt{x}$	no
6	mod	no



Algorithm 1 Secure Lanczos Iteration on Cloud.

Input:
A symmetric matrix $M \in R^{n \times n}$.
Output:
The Eigen Values and Eigen Vectors.

Select two vectors $w_1 \in R^n, v_1 \in R^n$ where $w_1^T v_1 = 1$.
Compute $v = M v_1, w = M^T w_1$;
for j from 1 to k **do**
  Compute $\alpha_j = w_j^T v$, $v = v - \alpha_j v_j$, $w = w - \alpha_j w_j$.
  if $\|v\|_2 = 0$ or $\|w\|_2 = 0$ **then**
    Exit.
  else
    Compute $w_j = v^T w$.
  end if
  if $w_j = 0$ **then**
    Exit.
  else
    Send w_j, v, w to Client.
    Receive β_j, γ_j, v_{j+1}, w_{j+1} from Client.
  end if
  $v = A v_{j+1} - \gamma_j v_j$; $w = A^T w_{j+1} - \beta_j w_j$.
end for
Obtain the Tridiagonal Matrix
Send the Tridiagonal Matrix to Client.

Algorithm 2 Secure Lanczos Iteration on Client.

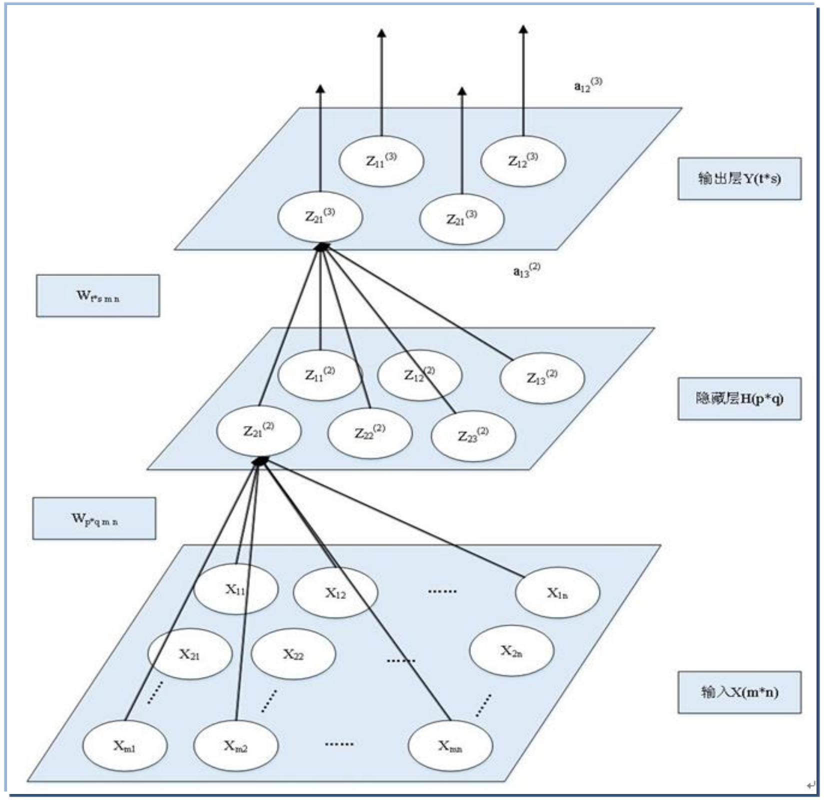
Input:
Middle results from Cloud.
Output:
Processed Middle Results.

Receive w_j, v, w from Cloud.
$\beta_j = \sqrt{|w_j|}$; $\gamma_j = w_j / \beta_j$.
$v_{j+1} = v / \beta_j$; $w_{j+1} = w / \gamma_j$.
Send β_j, γ_j, v_{j+1}, w_{j+1} to Cloud.
Receive the Tridiagonal Matrix from Cloud.
Compute the Core Tensor and Truncated Bases.

2. Big Data + Tensor +Deep Learning = Deep Computation

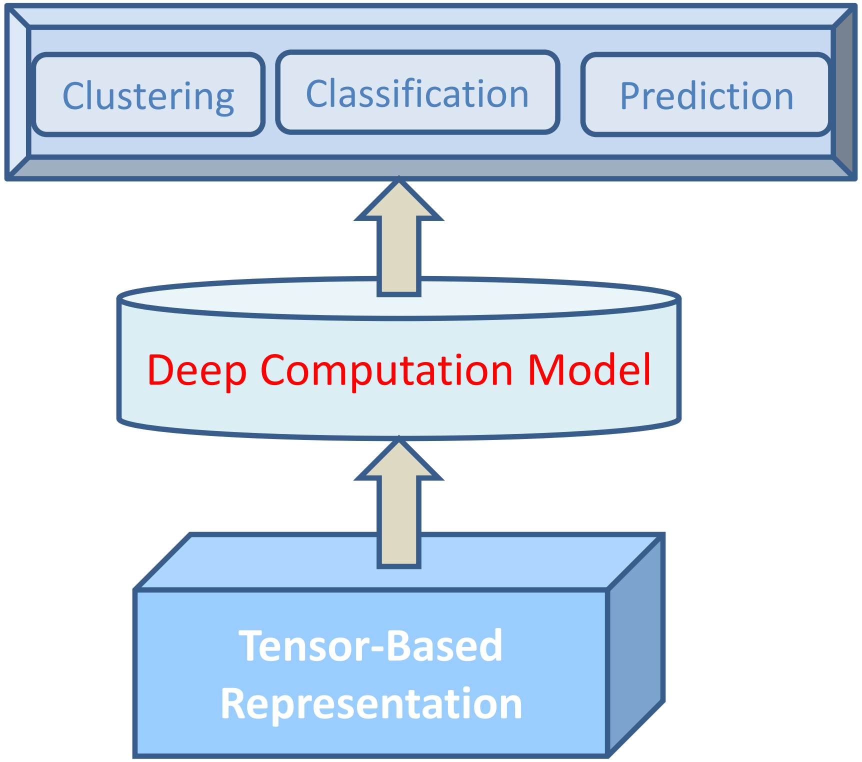
Deep Computation Model:

- Input/Output: Tensor-Based Data
- Hidden: Tensor-Based Features
- Parameter: Tensor-Based Parameters



Deep Computation Model:

- A General Deep Learning Model for Big Data
- Unsupervised Feature Learning
- Bridge Between Representation and Mining



2. Big Data + Tensor+ Deep Learning = Deep Computation

Deep Computation Model:

- Input/Output: Tensor-Based Data
- Hidden: Tensor-Based Features
- Parameter: Tensor-Based Parameters

（1）input layer→hidden layer:

$$H = f_\theta(W^{(1)} \circ X + b^{(1)})$$

$$\theta = \{W^{(1)}, b^{(1)}\}, W^{(1)} \in R^{\alpha \times I_1 \times I_2 \times \cdots \times I_n}, b^{(1)} \in R^{J_1 \times J_2 \times \cdots \times J_n}$$

$$\alpha = J_1 \times J_2 \times \cdots \times J_n$$

（2）hidden layer→output layer:

$$h_{W,b}(X) = f_{\theta'}(W^{(2)} \circ H + b^{(2)})$$

$$\theta' = \{W^{(2)}, b^{(2)}\}, W^{(2)} \in R^{\beta \times J_1 \times J_2 \times \cdots \times J_n}, b^{(2)} \in R^{K_1 \times K_2 \times \cdots \times K_n}$$

$$\beta = K_1 \times K_2 \times \cdots \times K_n$$

Deep Computation Model

- Tensor-Based Computing
- Tensor Distance → Data Distribution
- Cost Function → Tensor Distance

$$J(W,b) = [\frac{1}{m}\sum_{i=1}^{m} J(W,b;\mathrm{X}^{(i)},\mathrm{Y}^{(i)})] + \frac{\lambda}{2}(\sum_{j=1}^{\alpha}\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_n}^{I_n}(\mathrm{W}^{(1)}_{i_1 i_2 \ldots i_n})^2$$

$$+\sum_{k=1}^{\gamma}\sum_{j_1=1}^{J_1}\sum_{j_2=1}^{J_2}\cdots\sum_{j_n}^{J_n}(\mathrm{W}^{(2)}_{j_1 j_2 \ldots j_n})^2) + \beta\sum_{j_1=1}^{J_1}\sum_{j_2=1}^{J_2}\cdots\sum_{j_n}^{J_n}(\rho \,||\, \hat{\rho}_{j_1 j_2 \ldots j_n})$$

$$J(W,b;\mathrm{X}^{(i)},\mathrm{Y}^{(i)}) = \frac{1}{2} d^2_{\mathrm{TD}(\mathrm{X}^{(i)},\mathrm{Y}^{(i)})}$$

$$\alpha = J_1 \times J_2 \times \cdots \times J_n; \gamma = K_1 \times K_2 \times \cdots \times K_n$$

$$X \in R^{I_1 \times I_2 \times \cdots \times I_N}; x_l = X_{i_1 i_2 \ldots i_N} (1 \le i_j \le I_j, 1 \le j \le N)$$

$$l = i_1 + \sum_{j=2}^{N}(i_j - 1)\prod_{0=1}^{j-1} I_0$$

$$d_{TD} = \sqrt{\sum_{l,m=1}^{I_1 \times I_2 \times \cdots \times I_N} g_{lm}(x_l - y_l)(\mathrm{x}_m - y_m)} = \sqrt{(x-y)^T G (x-y)}$$

$$g_{lm} = \frac{1}{2\pi\sigma^2}\exp\{-\frac{||\mathrm{p}_l - p_m||_2^2}{2\sigma^2}\}$$

$$||p_l - p_m||_2 = \sqrt{(i_1 - i_1')^2 + (i_2 - i_2')^2 + \cdots + (i_N - i_N')^2}$$

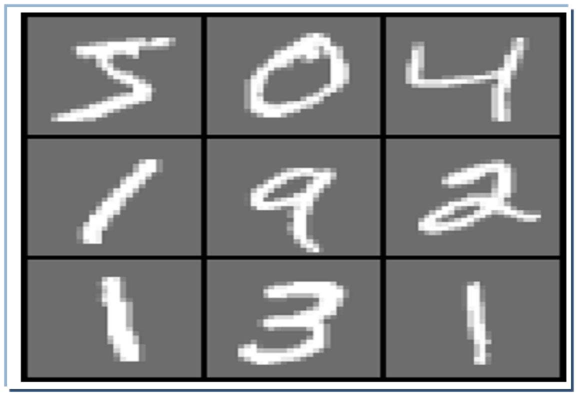
2. Big Data+Tensor+Deep Learning=Deep Computation Model

Deep Computation Model:

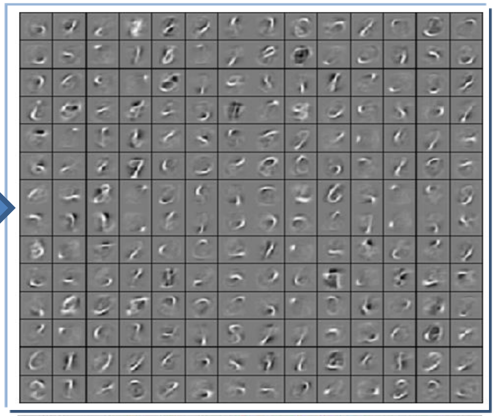
- Input/Output: Tensor-Based Data
- Hidden: Tensor-Based Features

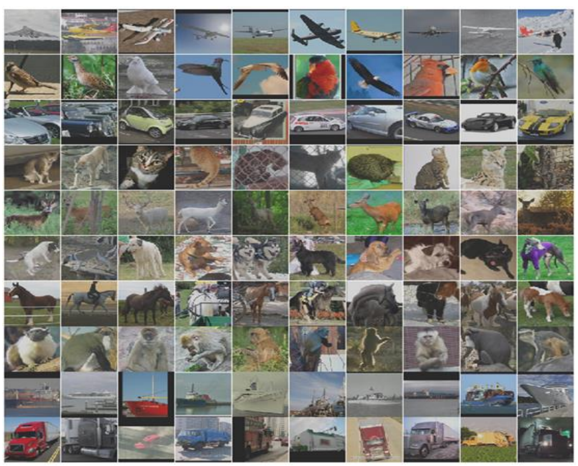
Deep Computation Model

- Unsupervised Feature Learning
- Bridge Between Representation and Mining

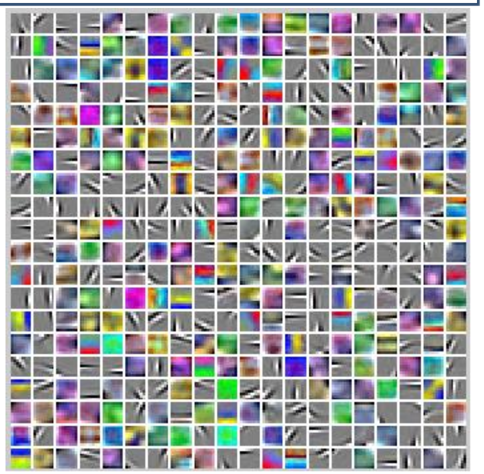


Edge Feature

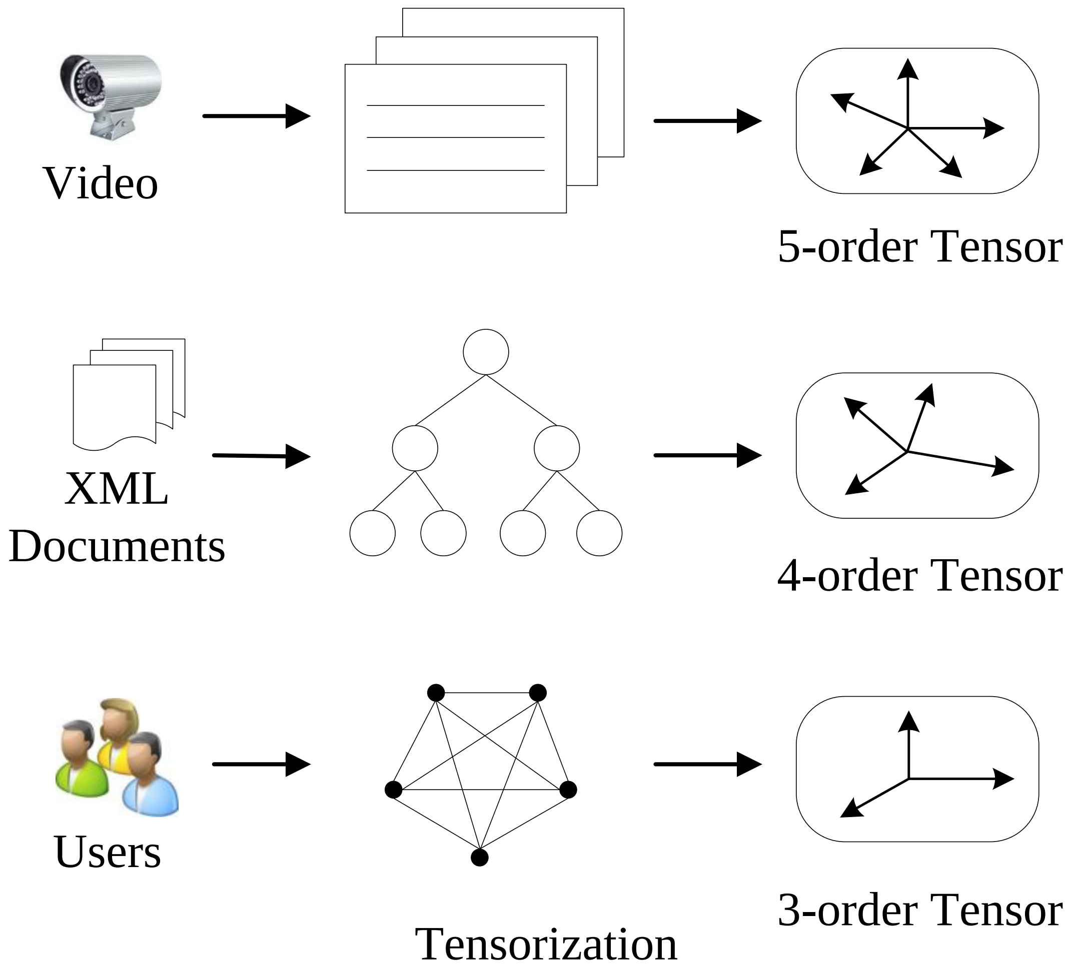




Color Feature

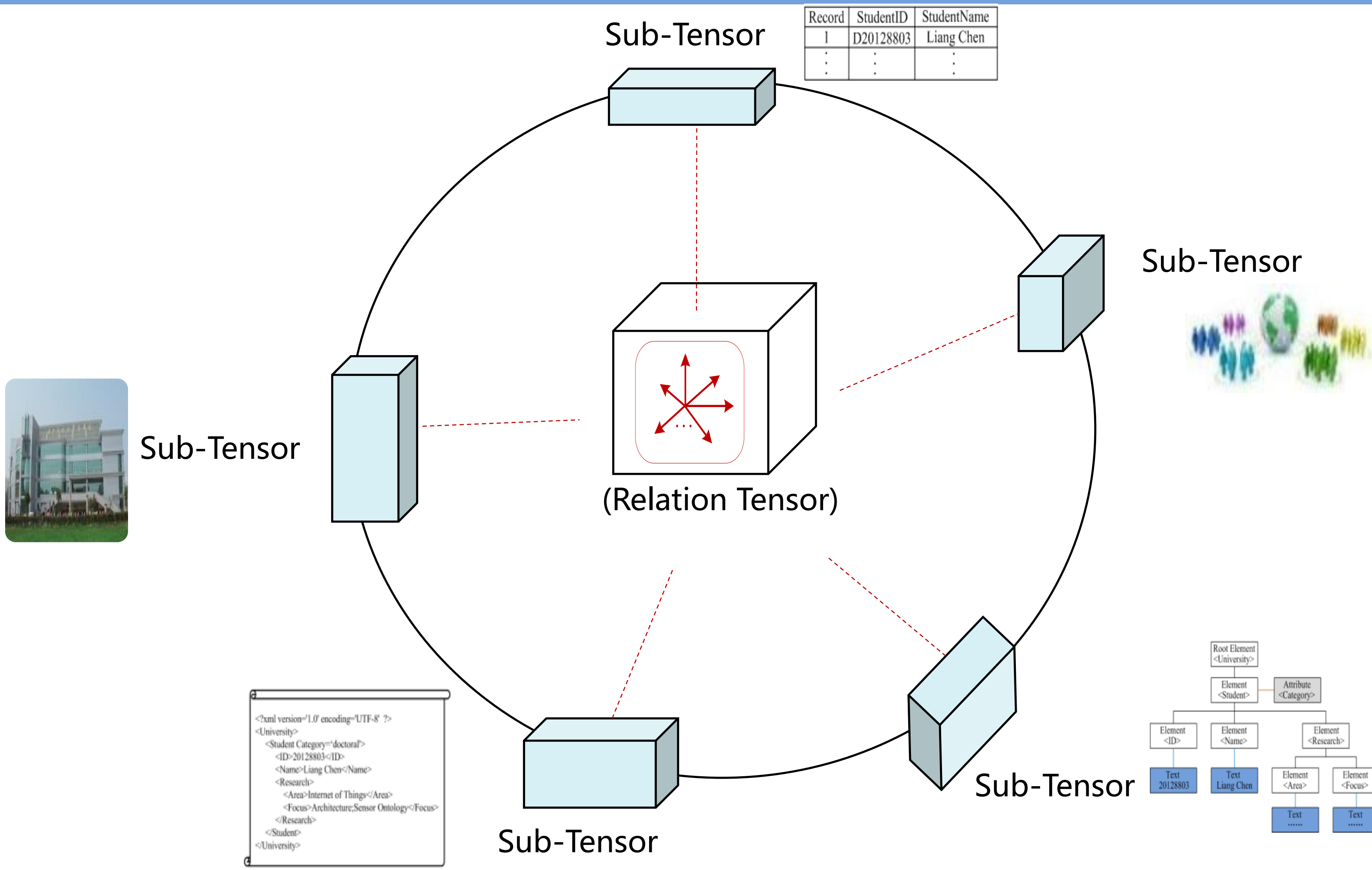


3. Big Data Ranking and Retrieval



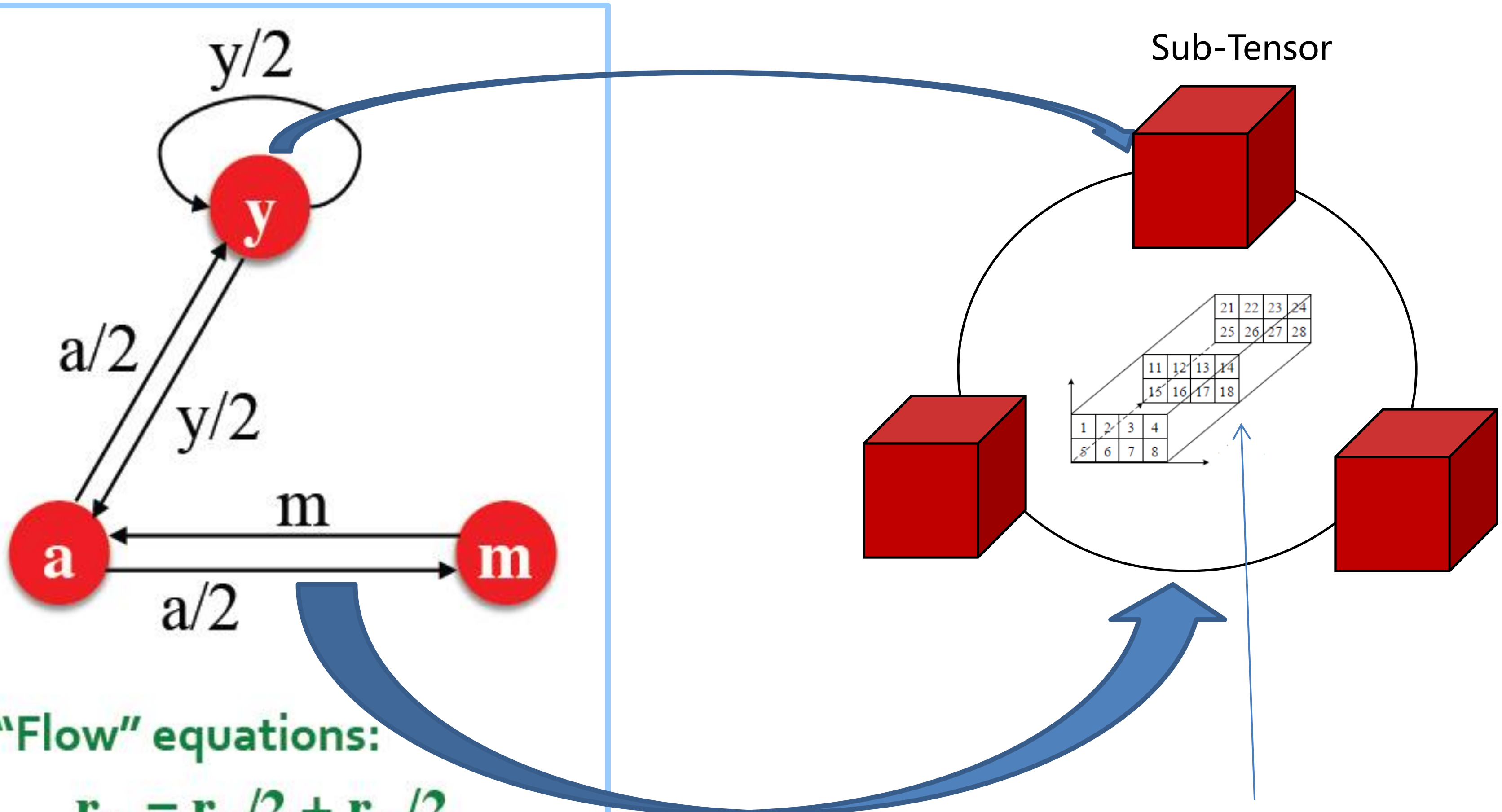
Data Space: Represent heterogeneous data as sub tensors.

3. Big Data Ranking and Retrieval



Metric Space: Map all the sub tensors to a Hamming Space, and obtain the similarity between all sub tensors. The similarities are stored in the relation tensor.

3. Big Data Ranking and Retrieval

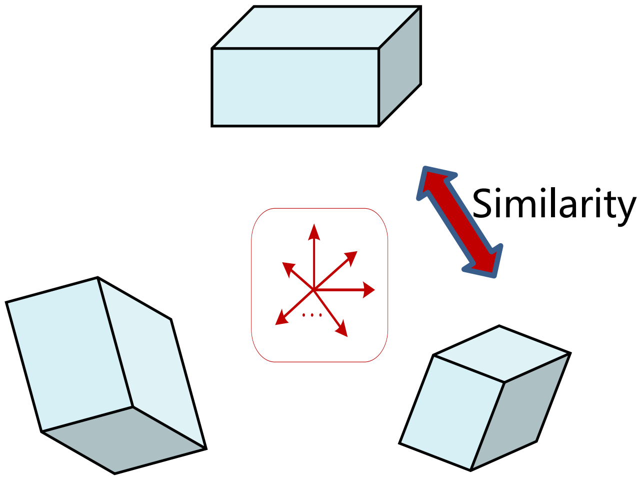


"Flow" equations:

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$

$$\boldsymbol{r} = \boldsymbol{T} \times \boldsymbol{r}$$

Vector ***r*** denotes the rank of the data object



In the metric space, the similarity is computed with tensor distance.

Tensor Distance

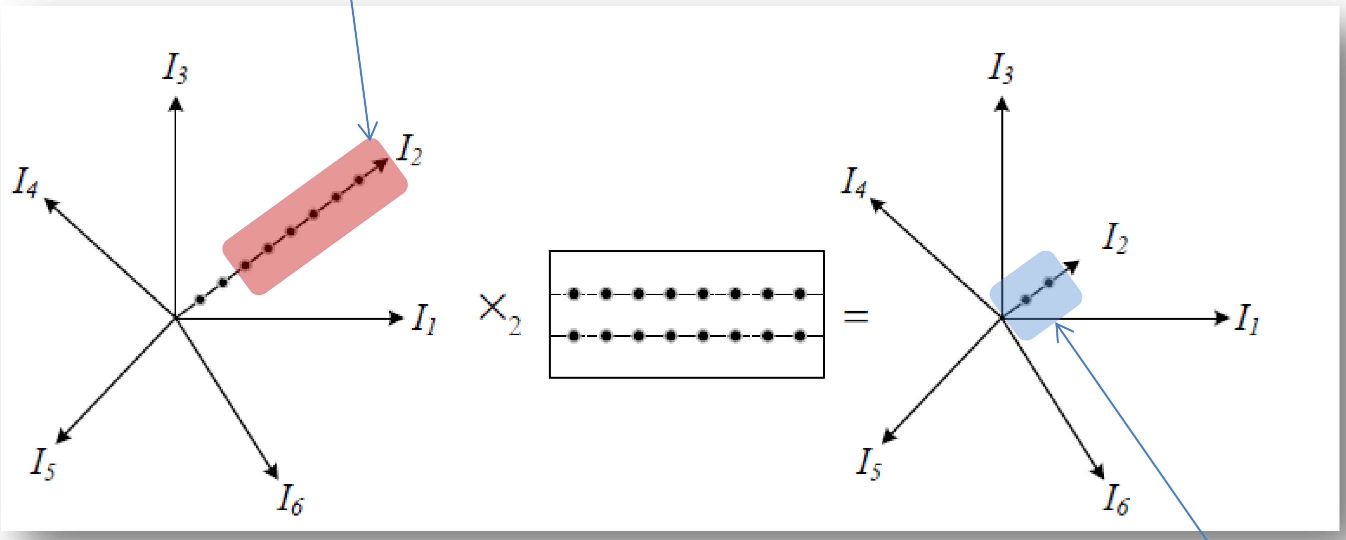
$$d_{TD} = \sqrt{\sum_{l,m}^{I_1 \times I_2 \times \ldots \times I_N} g_{lm}(x_l - y_l)(x_m - y_m)}$$

$$= \sqrt{(x-y)^T G(x-y)}$$

$$g_{lm} = \frac{1}{2\pi\sigma_1^2} \exp\left\{ \frac{-\|P_l - P_m\|_2^2}{2\sigma_1^2} \right\}$$

Veracity

Redundancy, Uncertainty, Inconsistency, Incompleteness



Extract the distinguish dimensions that can best capture the characteristics of big data

4. Fuzzy Tensor

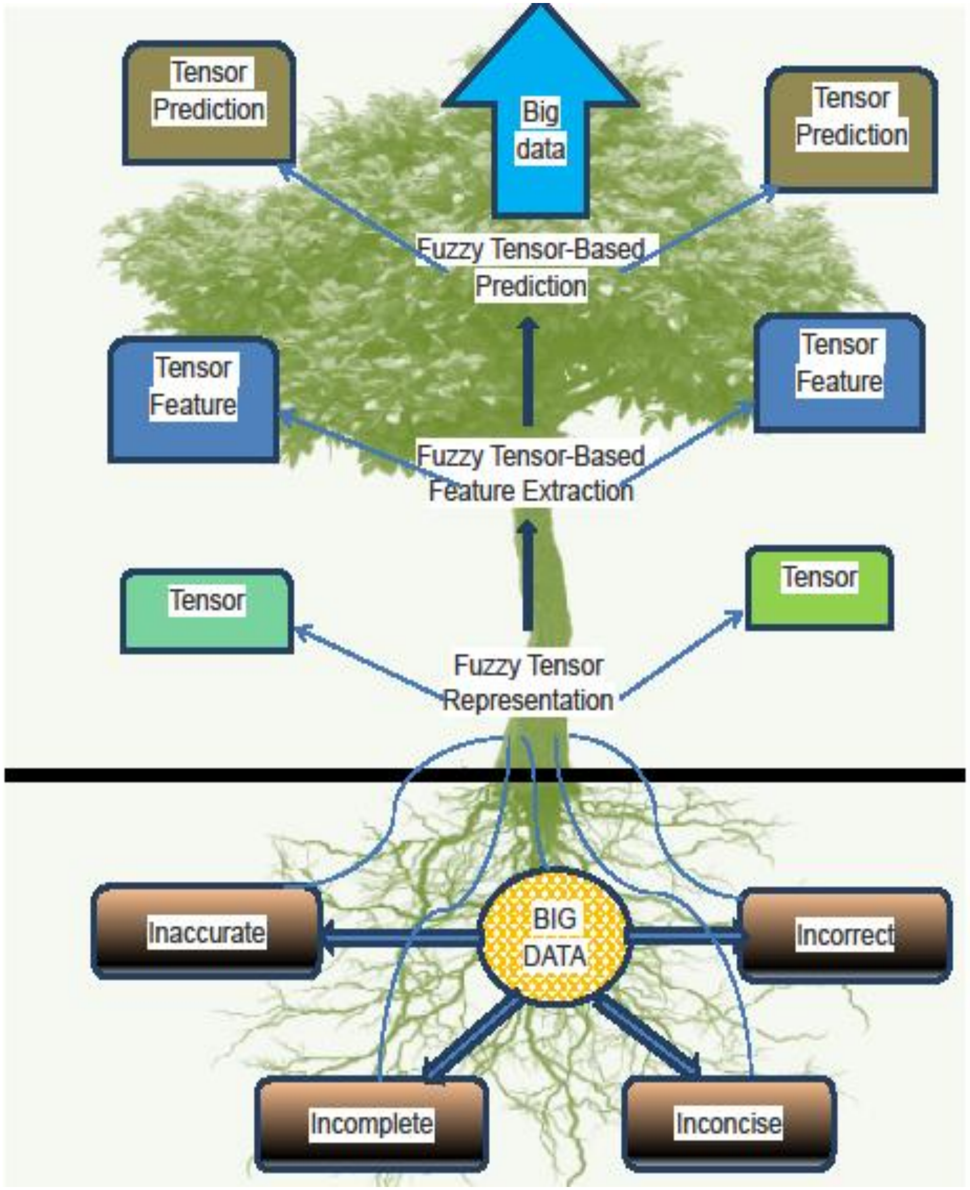
Data Structure: 4V

Data Feature: 4I

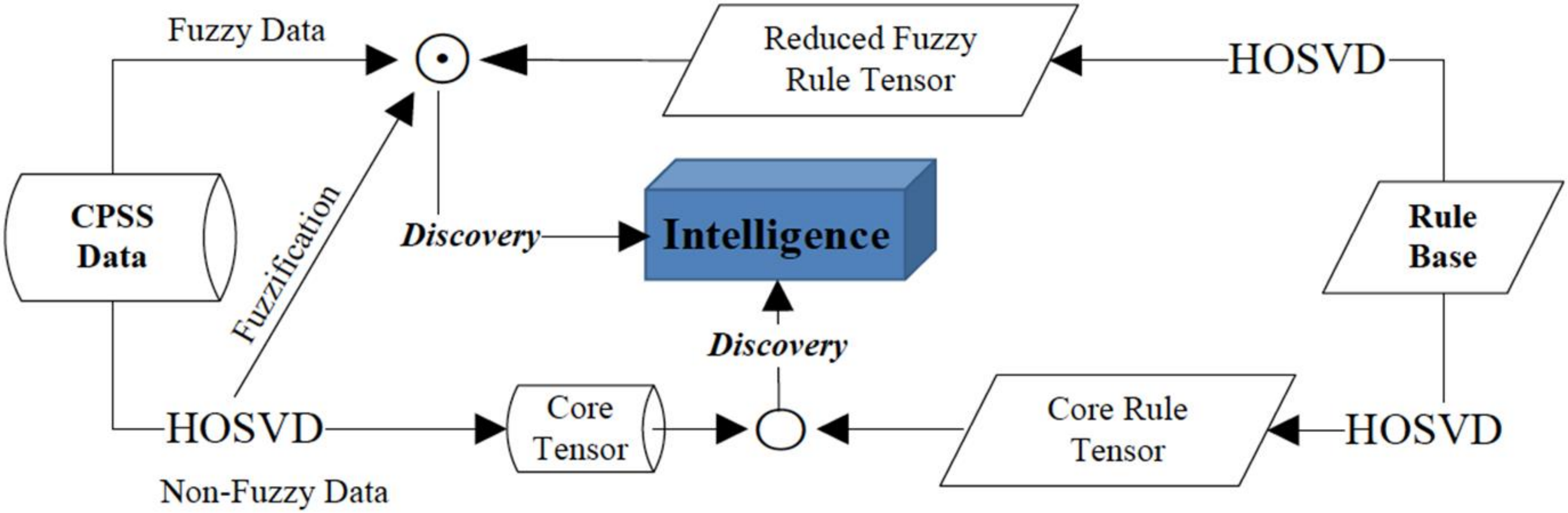
Tensor Generalization:

Tensor → Fuzzy Tensor

What/How



4. Fuzzy Tensor



Outline

1. Smart Word and Hyper World

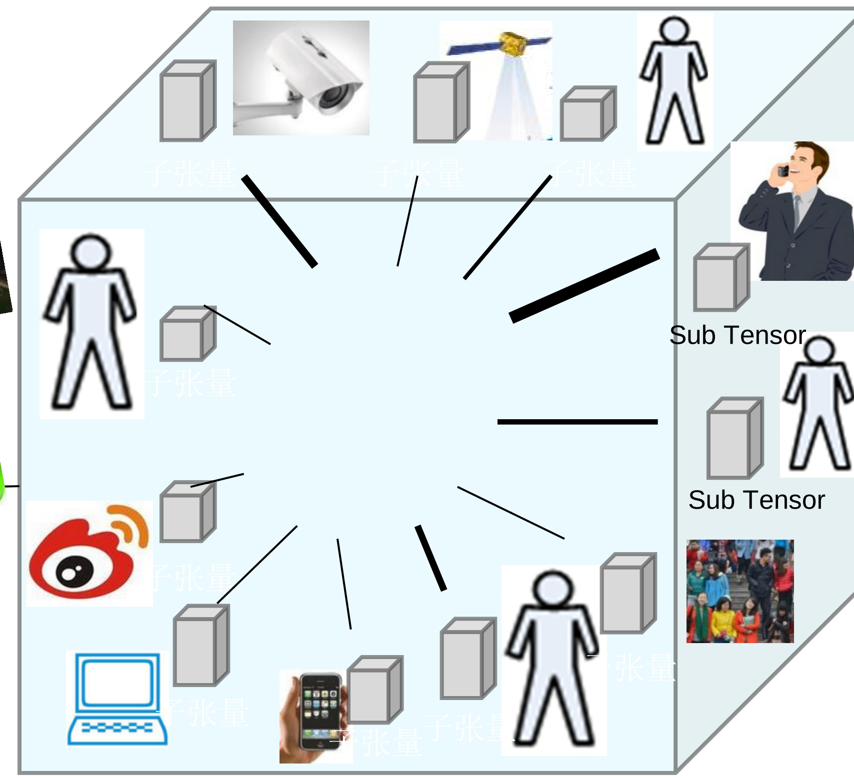
2. Big Data and Challenges

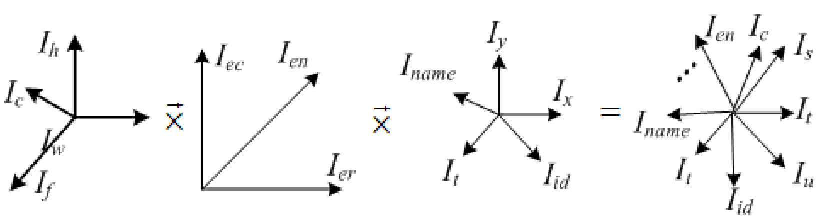
3. A Tensor-Based Approach

4. Illustration and Example

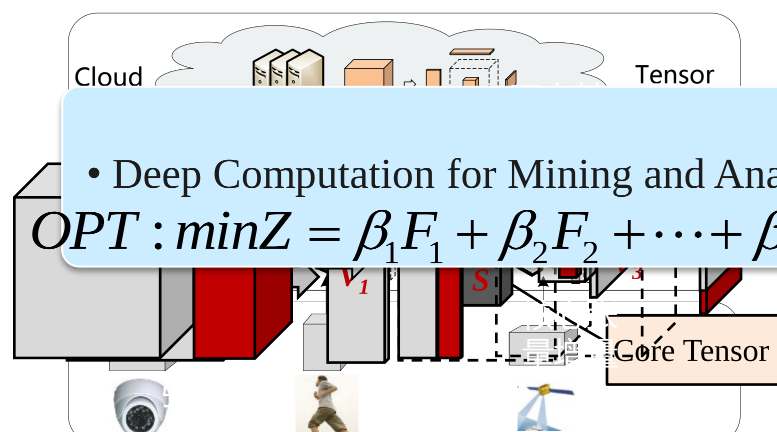
5. Conclusion

Smart City





$$f : (d_u \cup d_{semi} \cup d_s) \rightarrow \underbrace{T_u \cup T_{semi} \cup T_s}_{T}$$



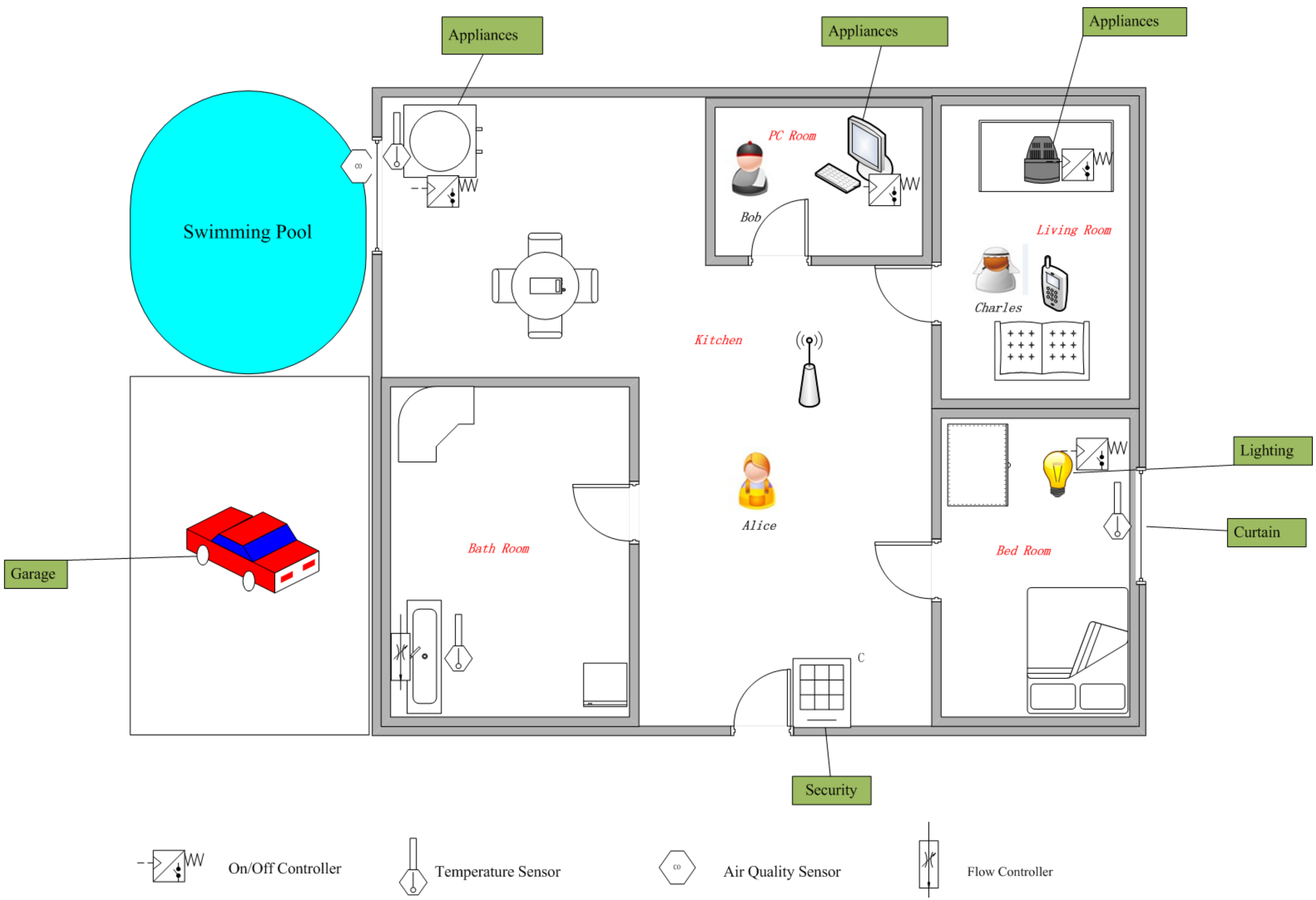
- Deep Computation for Mining and Analysis

$$OPT : minZ = \beta_1 F_1 + \beta_2 F_2 + \cdots + \beta_n F_n$$

Illustration 2:

Smart Home

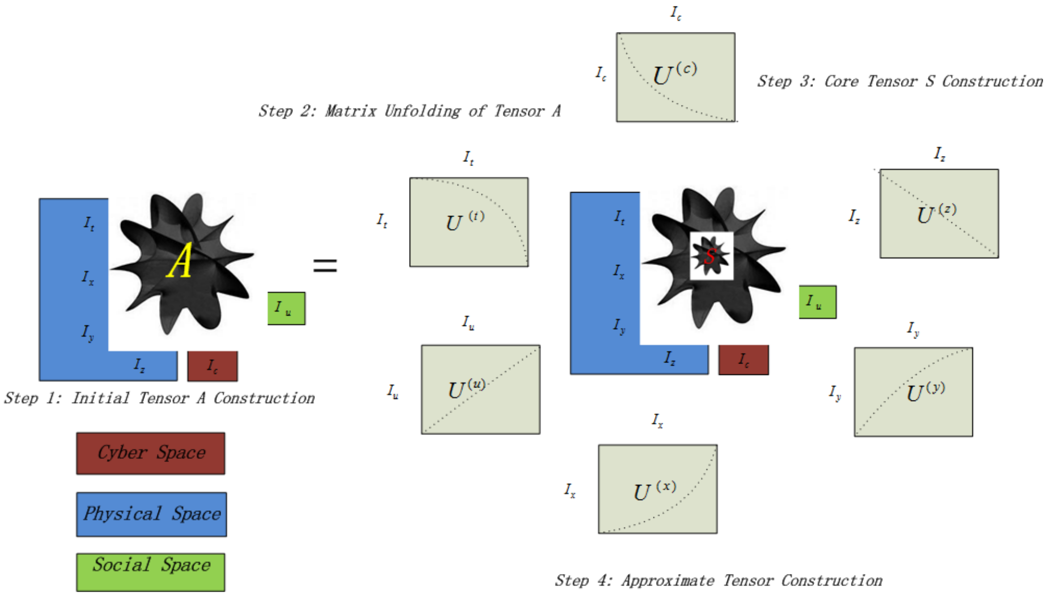
Heterogeneous Data from Intelligent Device



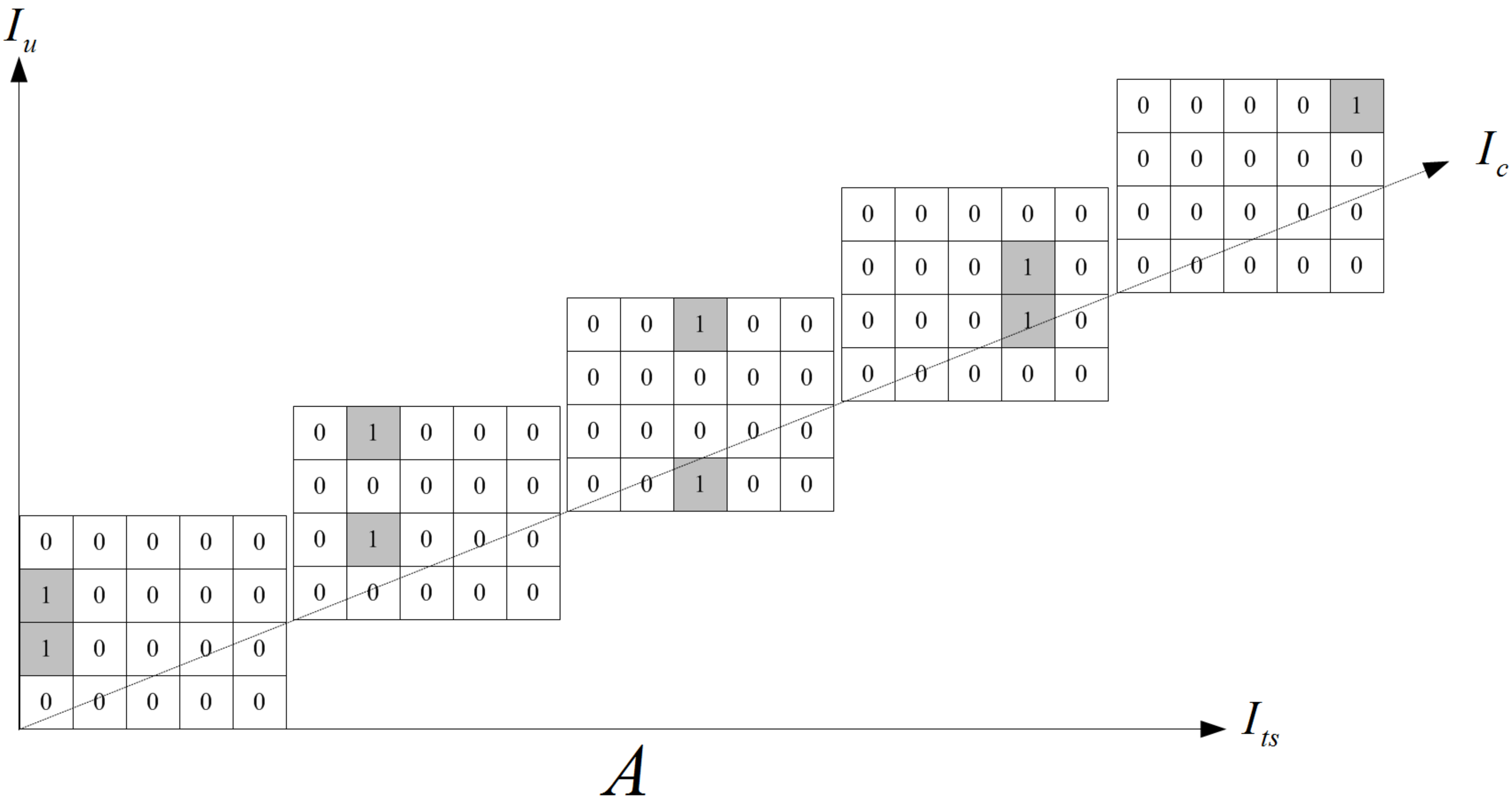
Four Step to Construct Six-Order Tensor

Smart Home

- *Social Space: Father, Mother, Children…*
- *Physical Space: Bedroom, Dining room…*
- *Cyber Space: Computer, Electronic devices…*



Initial Tensor



Tensor Decomposition

$U^{(c)}$

1	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
0	0	0	0	1

I_c

0	0	0	0	0.436
0	0	0	0	0.795

0	0	0	1.175	0
0	0	0	0.554	0

0	0	0.547	0	0
0	0	1.063	0	0

0	1.240	0	0	0
0	-0.658	0	0	0

I_u

1.240	0	0	0	0
-0.658	0	0	0	0

I_{ts}

S

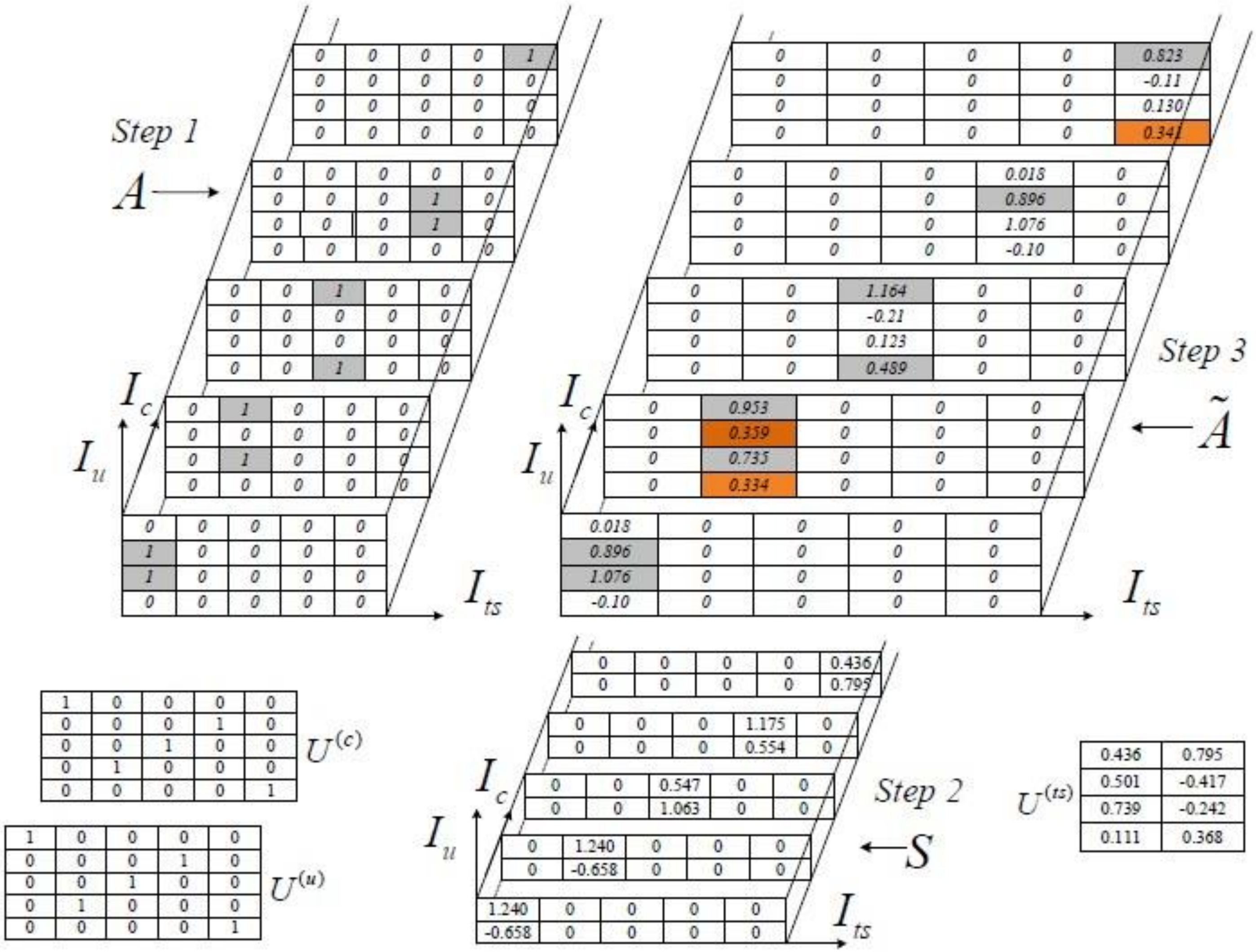
0.436	0.795
0.501	-0.417
0.739	-0.242
0.111	0.368

$U^{(ts)}$

1	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
0	0	0	0	1

$U^{(u)}$

Approximate Tensor Construction

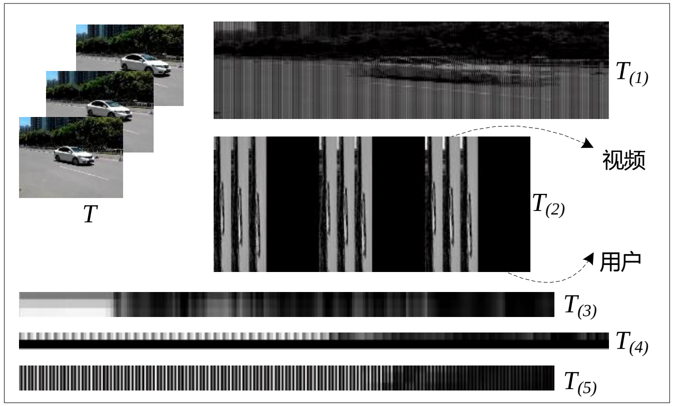


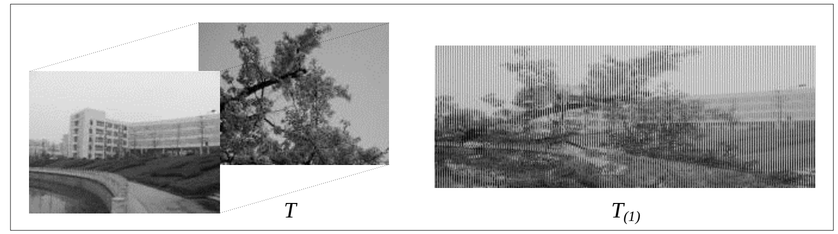
Likelihood Generated by Approximation Tensor

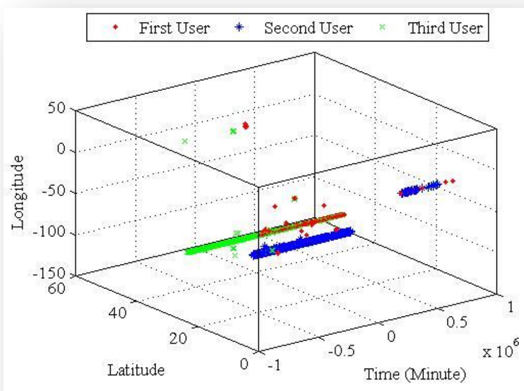
Record	User	Time & Space	Cyber Resource	Weight 1	Weight 2
1	Charles	6:00PM, Living Rm.	$TV_{on}=1$	1	0.896
2	Lucy	6:30PM, Swim.Pool	$Faucet_{on}=1$	1	0.823
3	Alice	7:10PM, Kitchen	$Pot_{on}=1$	1	1.076
4	Lucy	8:00PM, PC Rm.	$PC_{on}=1$	1	1.164
5	Alice	8:00PM, Living Rm.	$TV_{on}=1$	1	1.076
6	Charles	9:00PM, Kitchen	$Pot_{on}=1$	1	0.896
7	Lucy	9:30PM, Bed Rm.	$LE_{off}=1$	1	0.953
8	Alice	10:10PM, Bed Rm.	$LE_{off}=1$	1	0.735
9	Bob	11:30PM, PC Rm.	$PC_{on}=1$	1	0.489
10	Charles	10:10PM, Bed Rm.	$LE_{off}=1$	-	0.359
11	Bob	9:30PM, Bed Rm.	$LE_{off}=1$	-	0.334
12	Bob	6:30PM, Swim.Pool	$Faucet_{on}=1$	-	0.341

Published Research Work

Unified Representation Model for Big Data



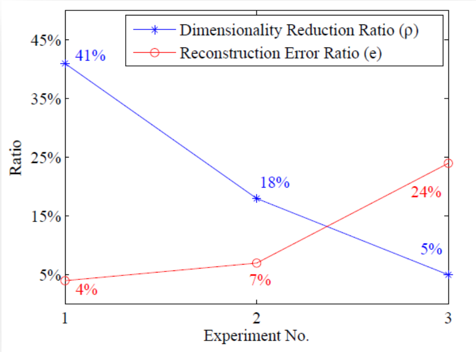


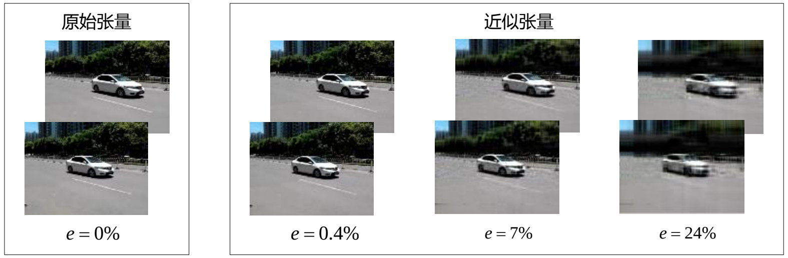


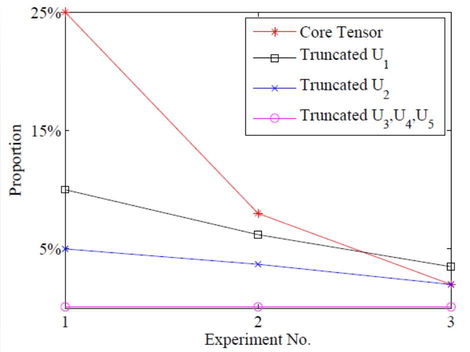
<?xml version='1.0' encoding='UTF-8' ?>
<University>
<Student Category=‘doctoral'>
<ID>20128803</ID>
<Name>Liang Chen</Name>
<Research>
<Area>Internet of Things</Area>
<Focus>Architecture;Sensor Ontology</Focus>
</Research>
</Student>
</University>

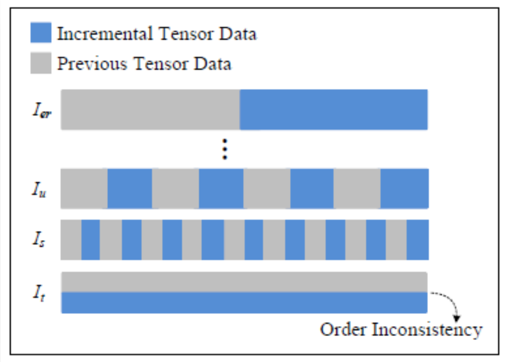
L. Kuang, F. Hao, L.T. Yang, M. Lin, C. Luo and G. Min, “A Tensor-Based Approach for Big Data Representation and Dimensionality Reduction”, IEEE Transactions on Emerging Topics in Computing (TETC), 2014, DOI: 10.1109/TETC.2014.2330516.

Incremental HOSVD



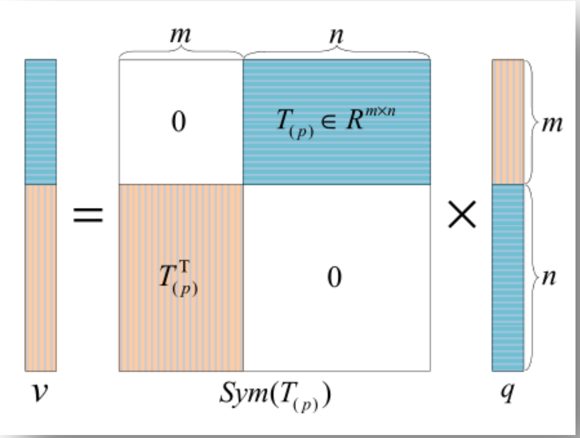


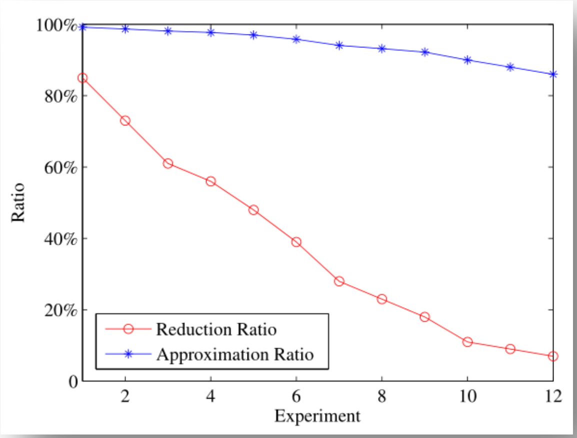


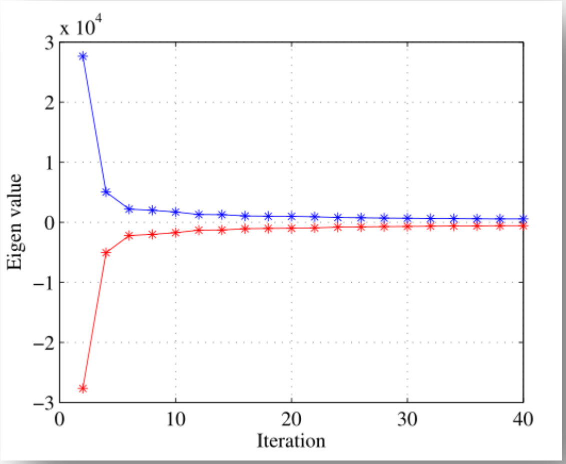


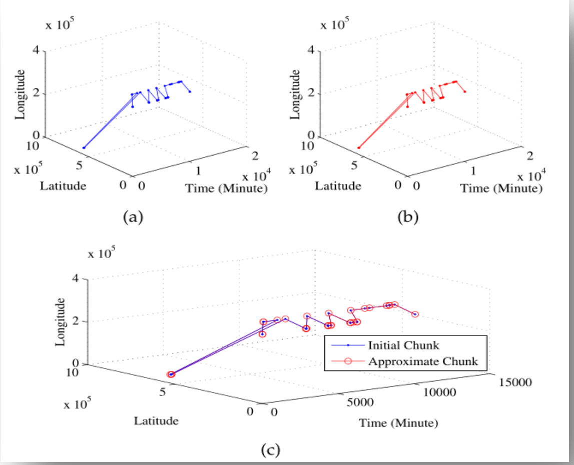
L. Kuang, F. Hao, L.T. Yang, M. Lin, C. Luo and G. Min, "A Tensor-Based Approach for Big Data Representation and Dimensionality Reduction", IEEE Transactions on Emerging Topics in Computing (TETC), 2014, DOI: 10.1109/TETC.2014.2330516.

Distributed Dimensionality Reduction of Big Data









L. Kuang, Y. Zhang, L.T. Yang, J. Chen, F. Hao, and C. Luo, "A Holistic Approach to Distributed Dimensionality Reduction of Big Data", IEEE Transactions on Cloud Computing (TCC), 2014, Accepted.

Outline

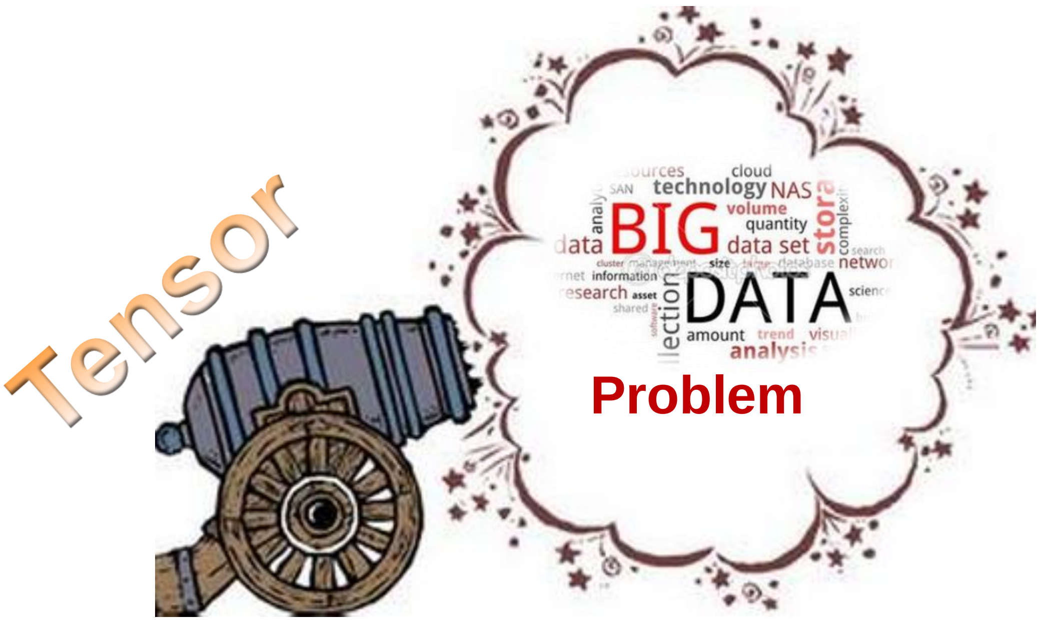
1. Smart Word and Hyper World

2. Big Data and Challenges

3. A Tensor-Based Approach

4. Illustration and Example

5. Conclusion



A tensor-based framework for big data Representations, Relations, Reductions, Retrieval, Reasoning and Recommendations!