Knowledge Engineering with Big Data

(joint work with Nanning Zheng, Huanhuan Chen, Qinghua Zheng, Aoying Zhou, Xingquan Zhu, Gong-Qing Wu, Wei Ding, Kui Yu et al.)

Xindong Wu (吴信东)
Department of Computer Science
University of Vermont, USA;
中国 ·合肥工业大学计算机与信息学院

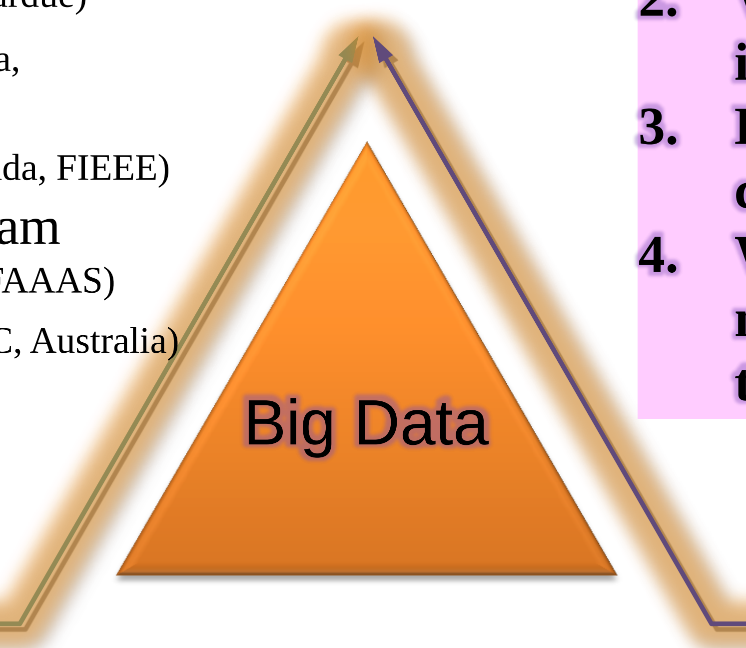
Outline

ICDM '13 Panel: Data Mining with Big Data

Panel Chair: Xindong Wu

Panelists:

- Chris Clifton (NSF & Purdue)
- Vipin Kumar (Minnesota, FIEEE,FACM,FAAAS)
- Jian Pei (TKDE EiC, Canada, FIEEE)
- Bhavani Thuraisingham (UTDallas, Security, FIEEE, FAAAS)
- Geoff Webb (DMKD EiC, Australia)
- Zhi-Hua Zhou (Nanjing,China,FIEEE)



1. **Big Data: a hot topic, but what useful content?**
2. **What new aspects? or is it just data mining?**
3. **How does data mining change with Big Data?**
4. **What should data miners do to cope with these changes?**

Big Data, from 70s to Now, and 2046

- The 1st International Conference on Very Large Data Bases (September 22-24, 1975, Framingham, MA, USA)
  - Very large = big?
  - The first ER model paper, QBE, …
- XLDB – Extremely Large Databases and Data Management, started on October 25, 2007
- ?LDB in 2046?
  - ULDB – Upmost Large Databases ☺
- Cent 01: being big is relative, going big is a deterministic trend
- Data mining: keep evolving

Some comments on big data

David Hand
Imperial College, London

The power law theorem of data set size:

- The number of data sets of size *n* is inversely proportional to *n*
- There are vastly more small data sets than very large ones
- So small data sets are likely to have a much larger impact on the world than big data sets

No-one actually wants data

- What people want are answers
- Which may be extracted from data
- So data are only half the answer
- The other half is statistics, data mining, machine learning, and other data analytic disciplines

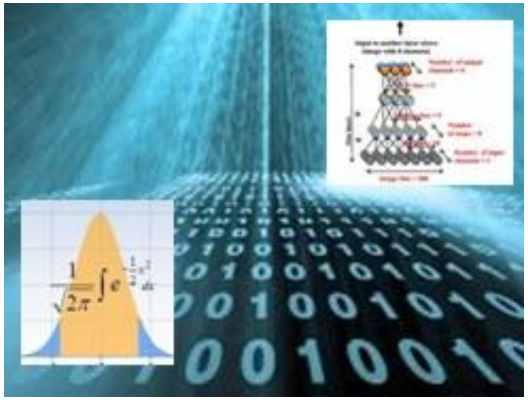
David Hand: Some comments on big data, December 2013

The manure heap theorem of data discoveries

The probability of finding a gold coin in a heap of manure tends towards 1 as the size of the heap tends to infinity.

(***This theorem is false***)

David Hand: Some comments on big data, December 2013



Data Science not just for Big Data

Gregory Piatetsky, @kdnuggets

KDnuggets™

[Analytics, Big Data, Data Mining, and Data Science Resources]

What do we call it?

- Statistics, 1830-
- Data mining, 1980-
- Knowledge Discovery in Data (KDD), 1989-
- Business Analytics, 1997-
- Predictive Analytics, 2002-
- Data Analytics,2011-
- Data Science, 2011-
- Big Data, 2012 -

Same Core Idea:

Finding Useful Patterns in Data

Different Emphasis

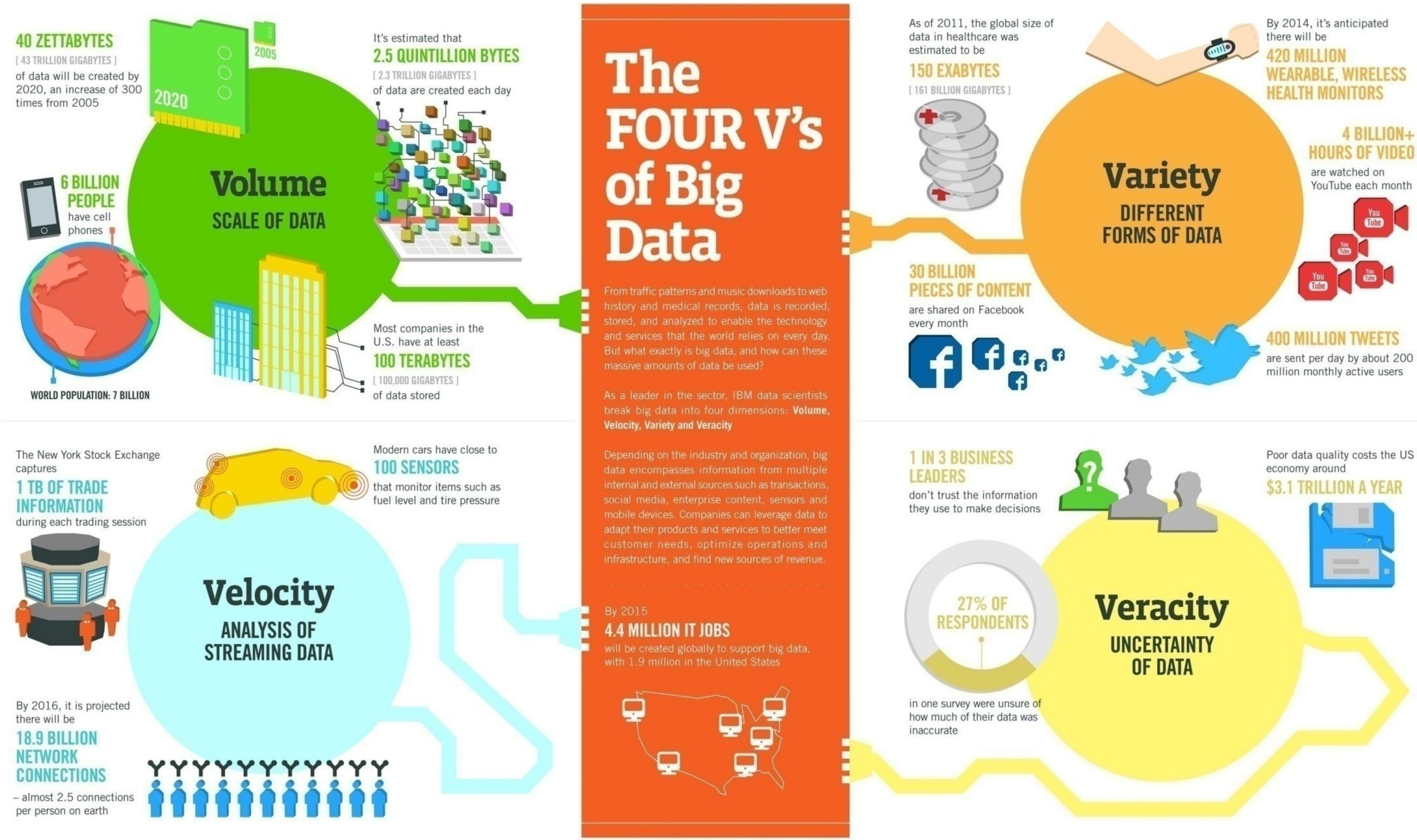
What Changes in Data Science with Big Data?

- Data munging becomes much more complex
- New algorithms, technology needed to deal with Big Data Volume, Velocity, & Variety
- New, effective algorithms that require Big Data: e.g.: deep belief networks, recommendations
- Predictions become (somewhat) more accurate
- **New things become visible**: social networks, recommendations, mobility, knowledge ?

- However, **basic principles remain**

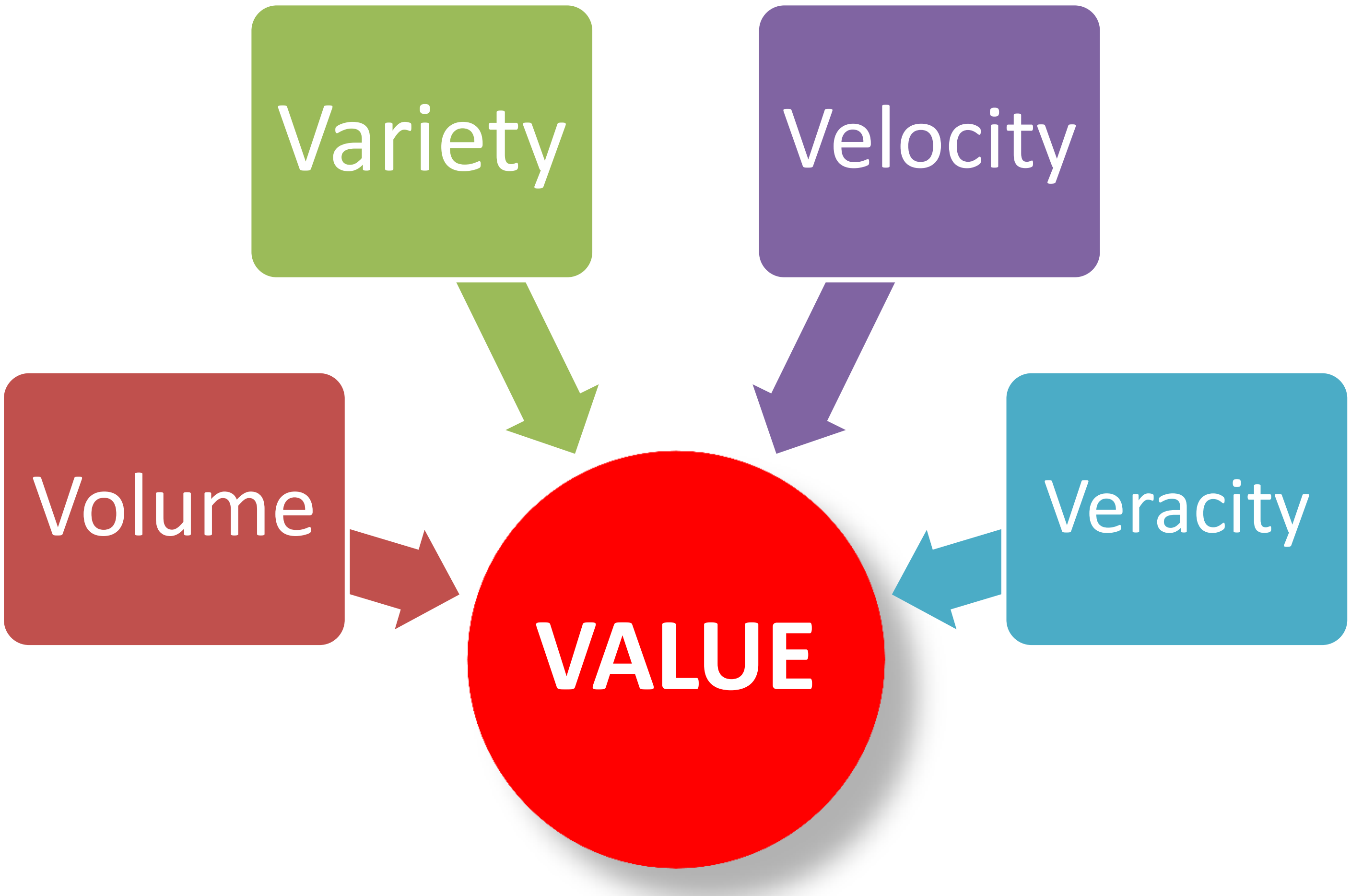
Outline

The IBM 4-V Model

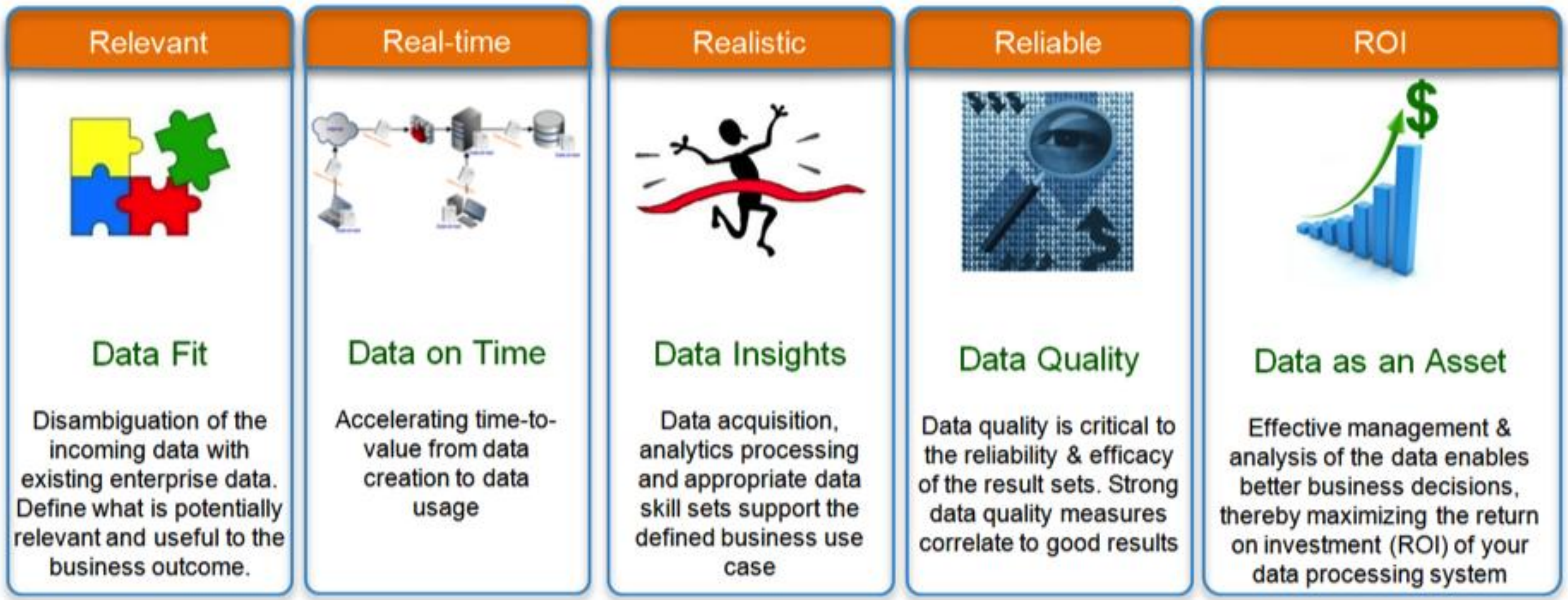


http://www.ibmbigdatahub.com/infographic/four-vs-big-data

Big Data: 5V's



The 5 R's of Big Data



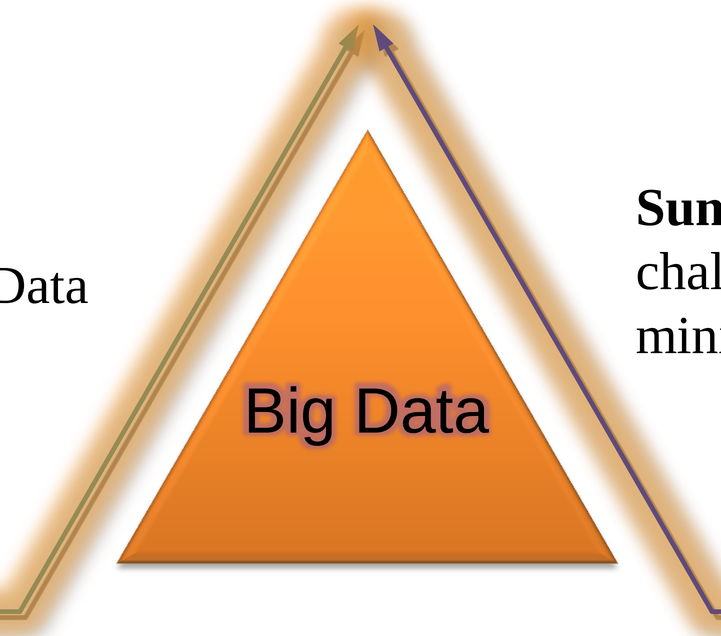
https://www.mapr.com/blog/business-leaders-need-r%E2%80%99s-not-v%E2%80%99s-5-r%E2%80%99s-big-data

Big Data Characteristics: HACE Theorem

Xindong Wu, Xinquan Zhu, Gongqing Wu, Wei Ding. **Data Mining with Big Data.** *IEEE Transactions on Knowledge and Data Engineering (TKDE),* **26**(2014), 1: 97-107.

The most downloaded paper in the IEEE XPLORE Digital Library (among all IEEE Publications (all journals and conferences, in all years) **every month for 18 consective months** (Jan. 2014 ~ Jun. 2015)

HACE Theorem:
a theorem to model Big Data characteristics



Summarizing the key challenges for Big Data mining

Google Scholar Citations So Far: 278

HACE Theorem

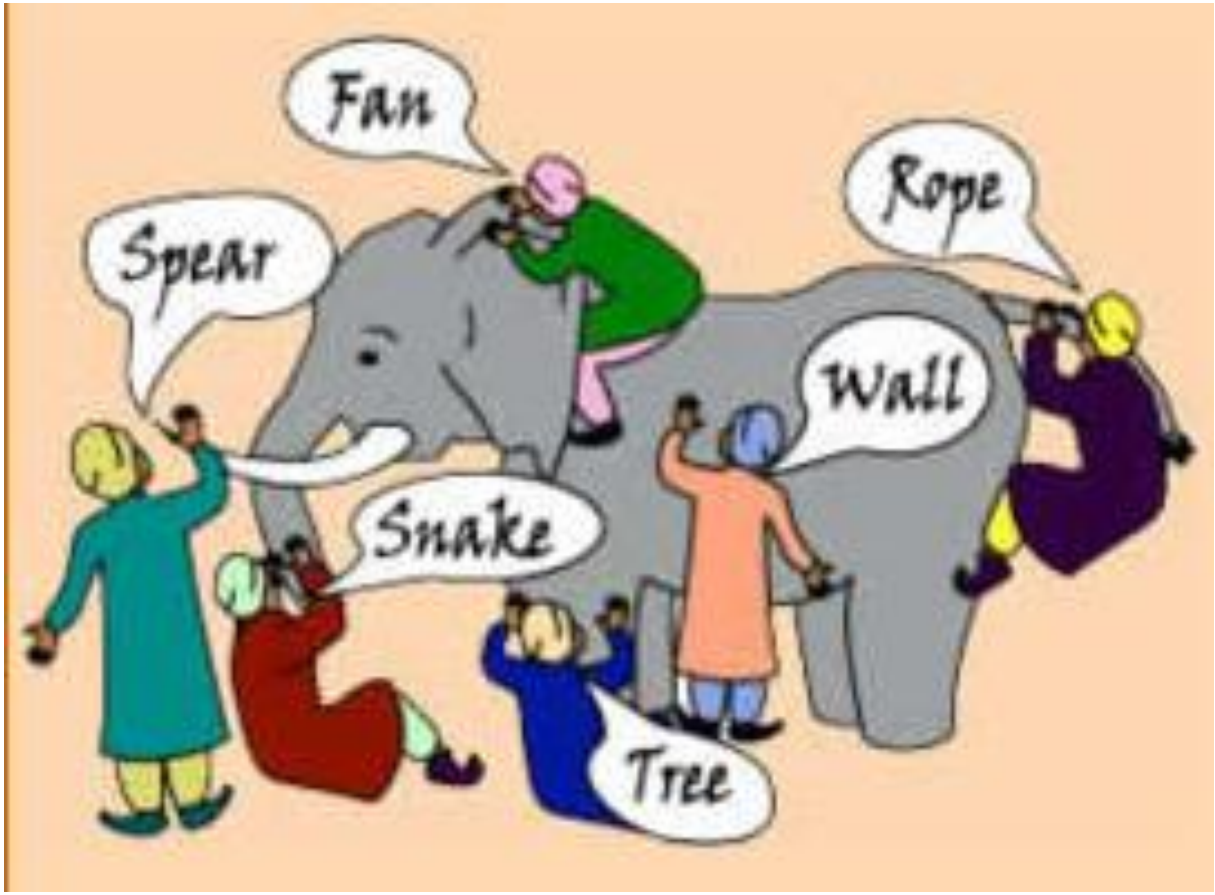
Big Data starts with large-volume,

- **H**eterogeneous,
- **A**utonomous sources with distributed and decentralized control,

•and seeks to explore

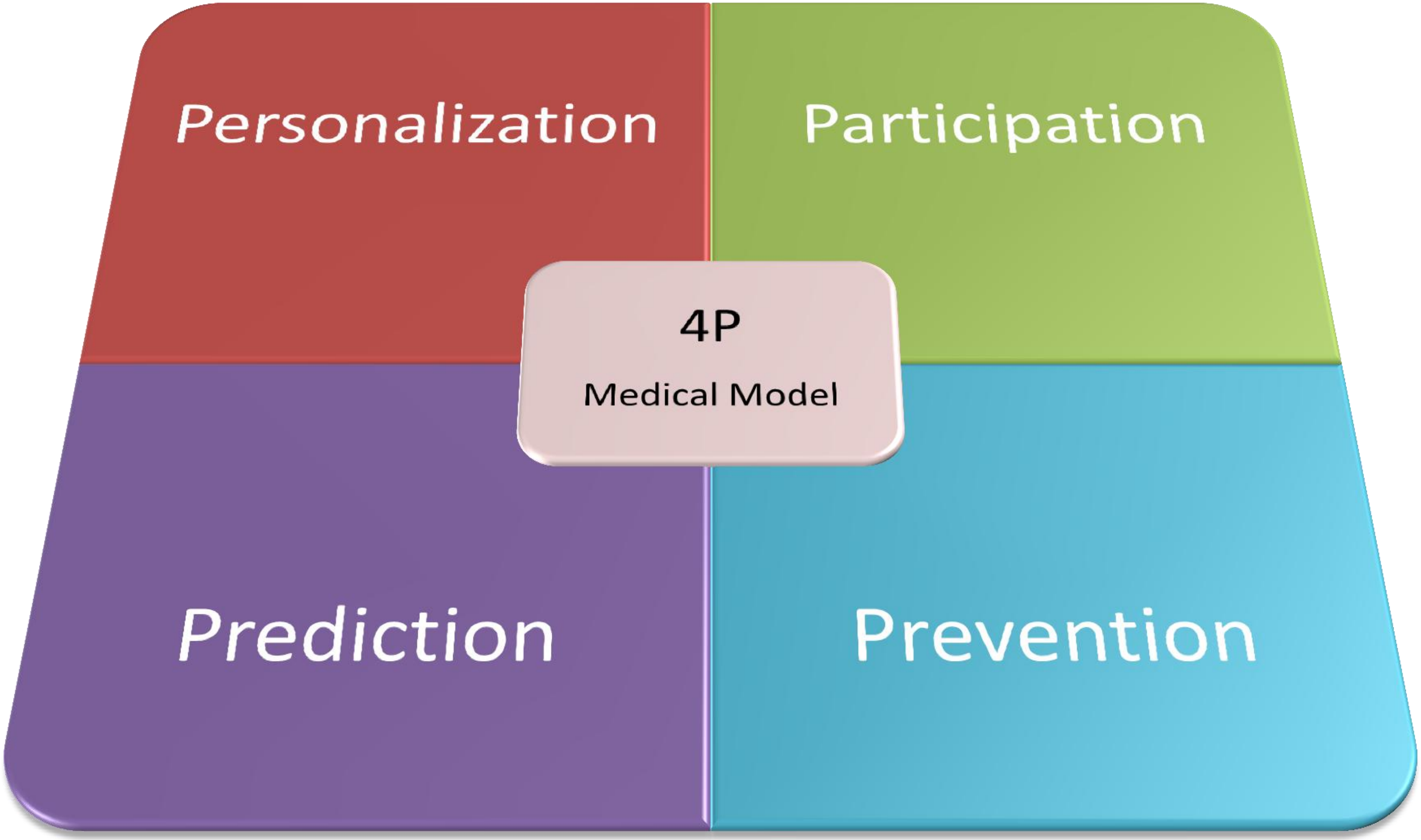
- **C**omplex and
- **E**volving relationships among data.

Small Example for Big Data
(a **moving, growing** elephant with blind men)



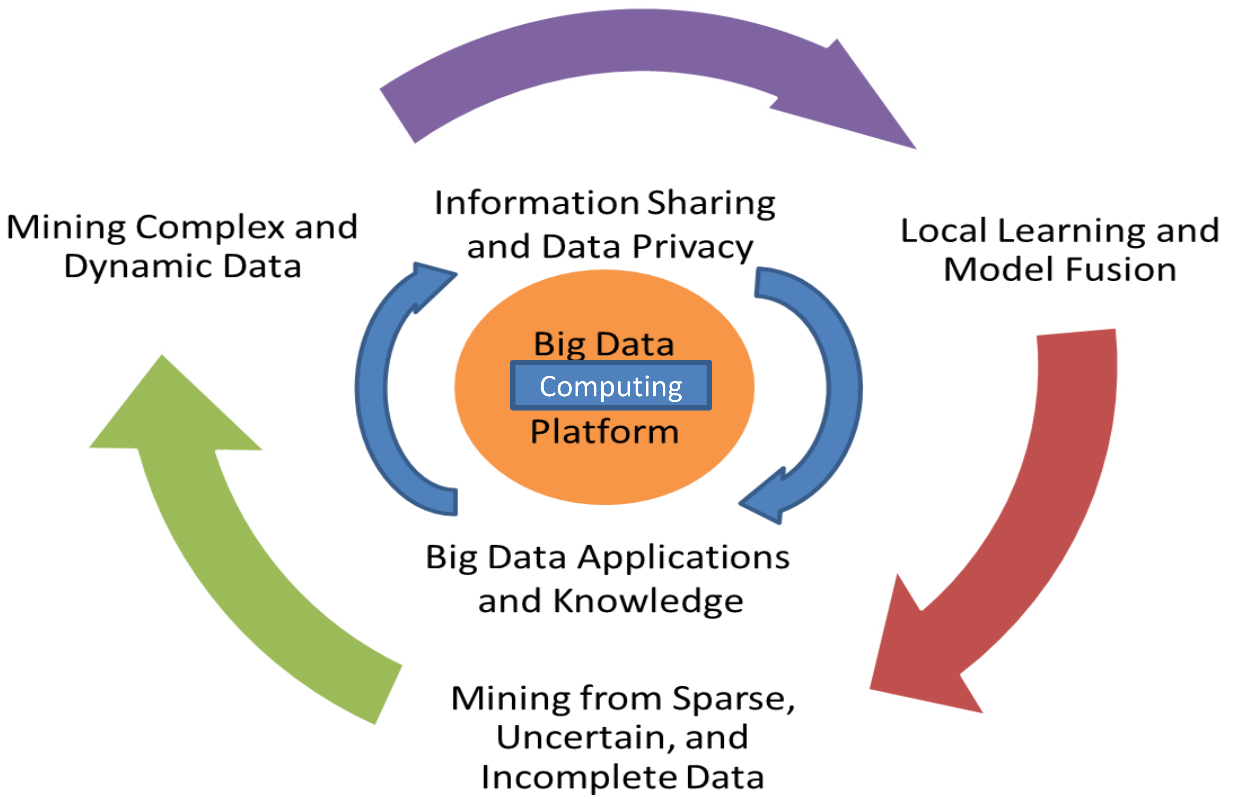
Source: Internet (http://www.nice-portal.com/English/what_i_culture.htm)

The 4P Medical Model



Outline

A Big Data Processing Framework



A Big Data Processing Framework

- **Tier I (databases):** Big Data computing platform, focusing on distributed, decentralized, low-level data accessing and computing.
- **Tier II (knowledge engineering):** Information sharing & privacy, and application domains, with
  - high level semantics,
  - application domain knowledge,
  - user privacy issues.
- **Tier III : Data mining**: Knowledge discovery.

Big Data Mining Challenges (1)

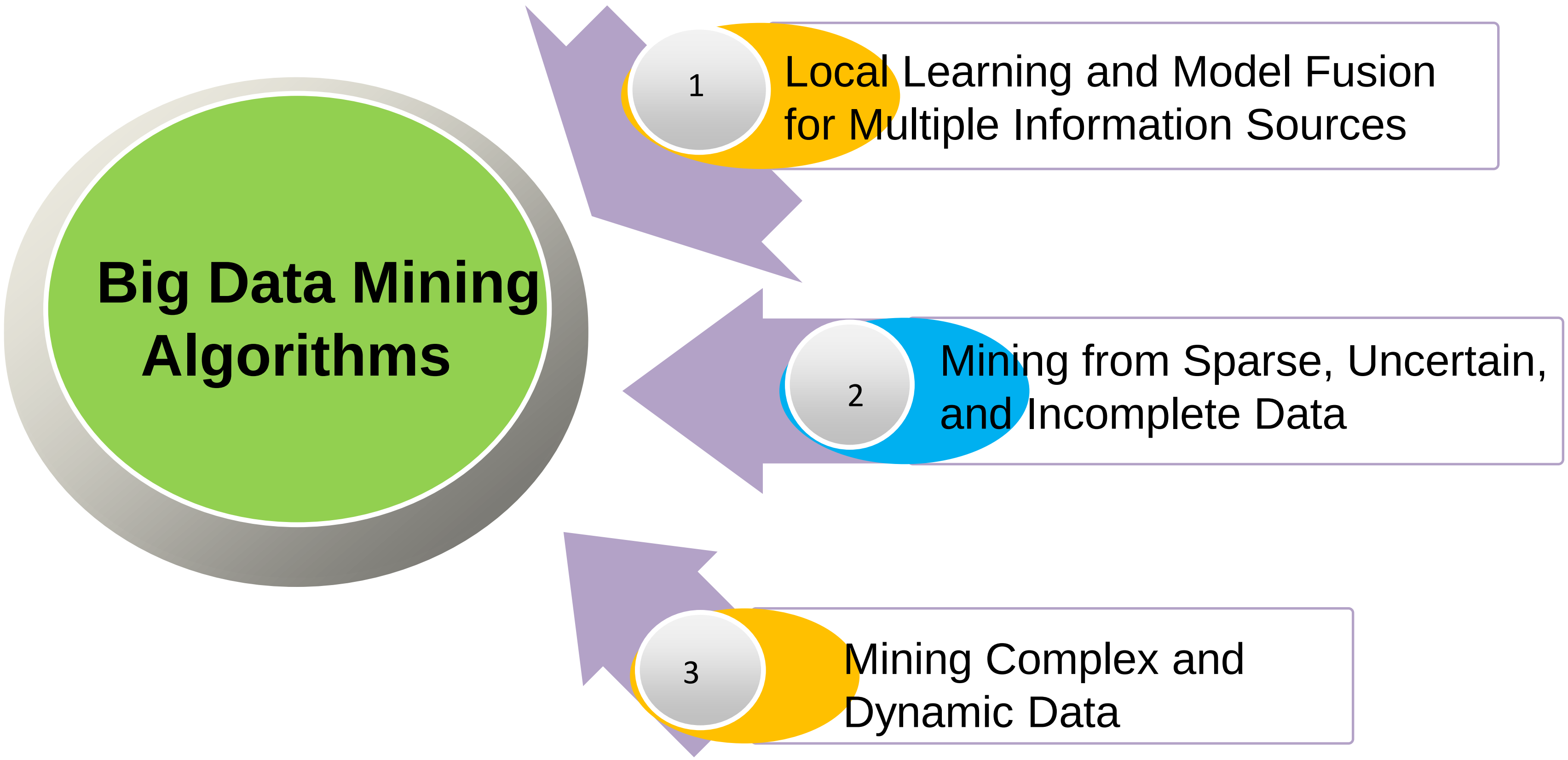
- **Big Data Computing Platform Challenges**
  - **Data accessing**
    - ✓Huge and evolving data volumes
    - ✓Heterogeneous and autonomous sources
    - ✓Diverse representations
    - ✓Unstructured data
  - **Computing processors**

    High performance computing platforms

Big Data Mining Challenges (2)

- **Big Data Semantics and Application Knowledge**
  - **Data sharing and privacy**
    - How data are maintained, accessed, and shared
  - **Domain and application knowledge**
    - What are the underlying applications ?
    - What are the knowledge or patterns users intend to discover from the data ?

Big Data Mining Challenges (3)



Outline

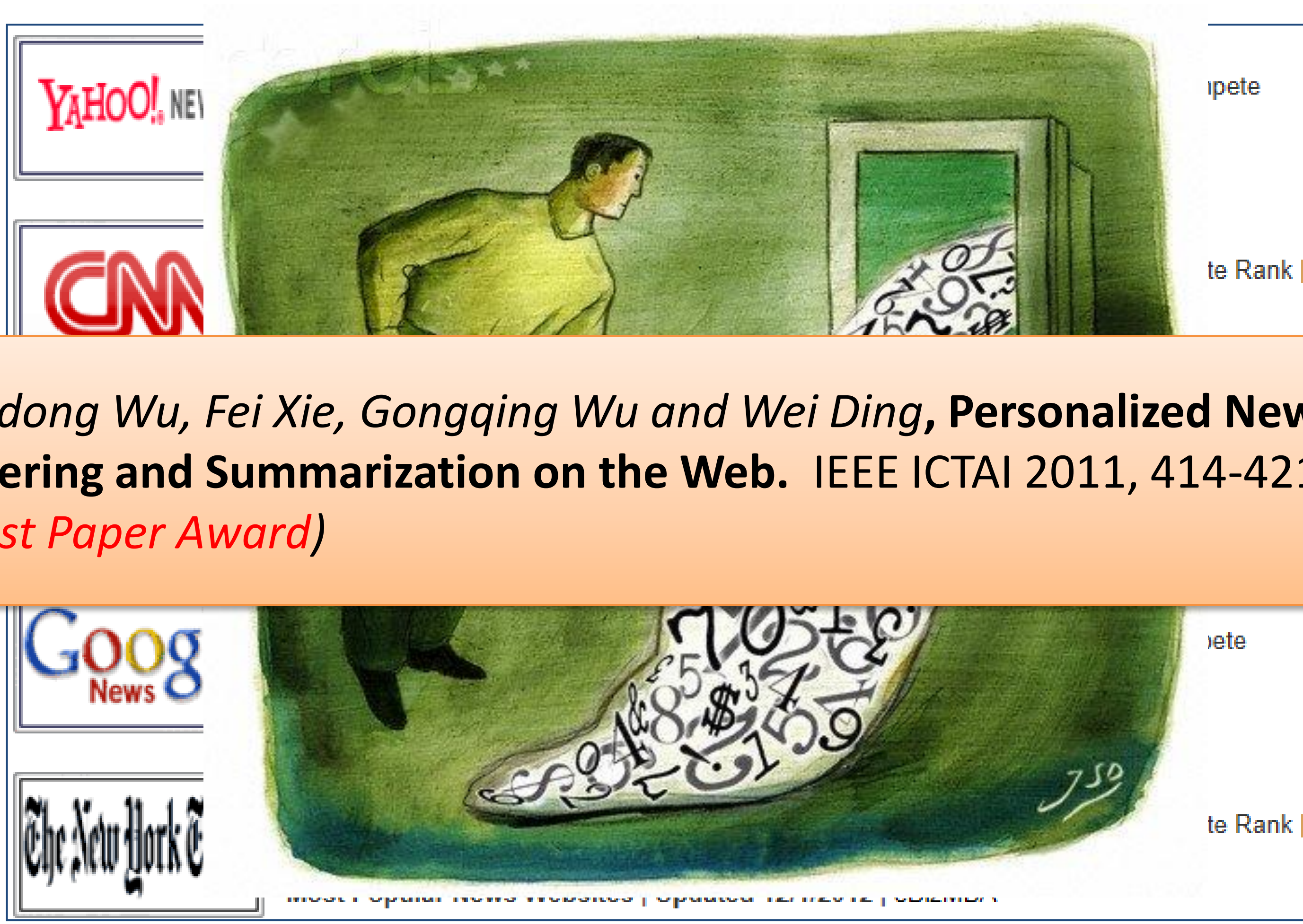
Handling Huge and Evolving Big Data

- **Real-time processing of Big data streams**

- **Real-time processing of Big feature streams**

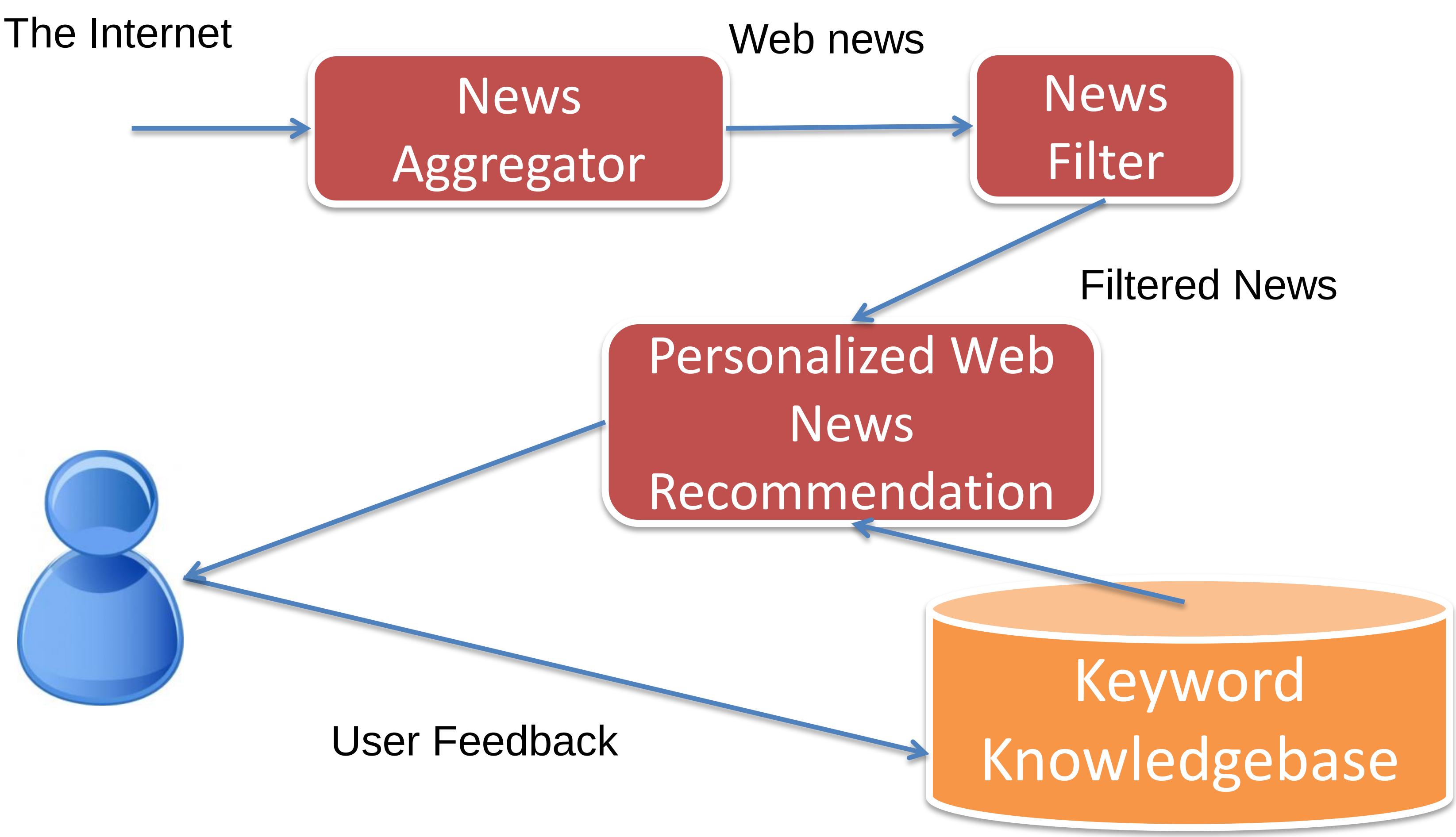
PNRS

- Personalized news recommendation system

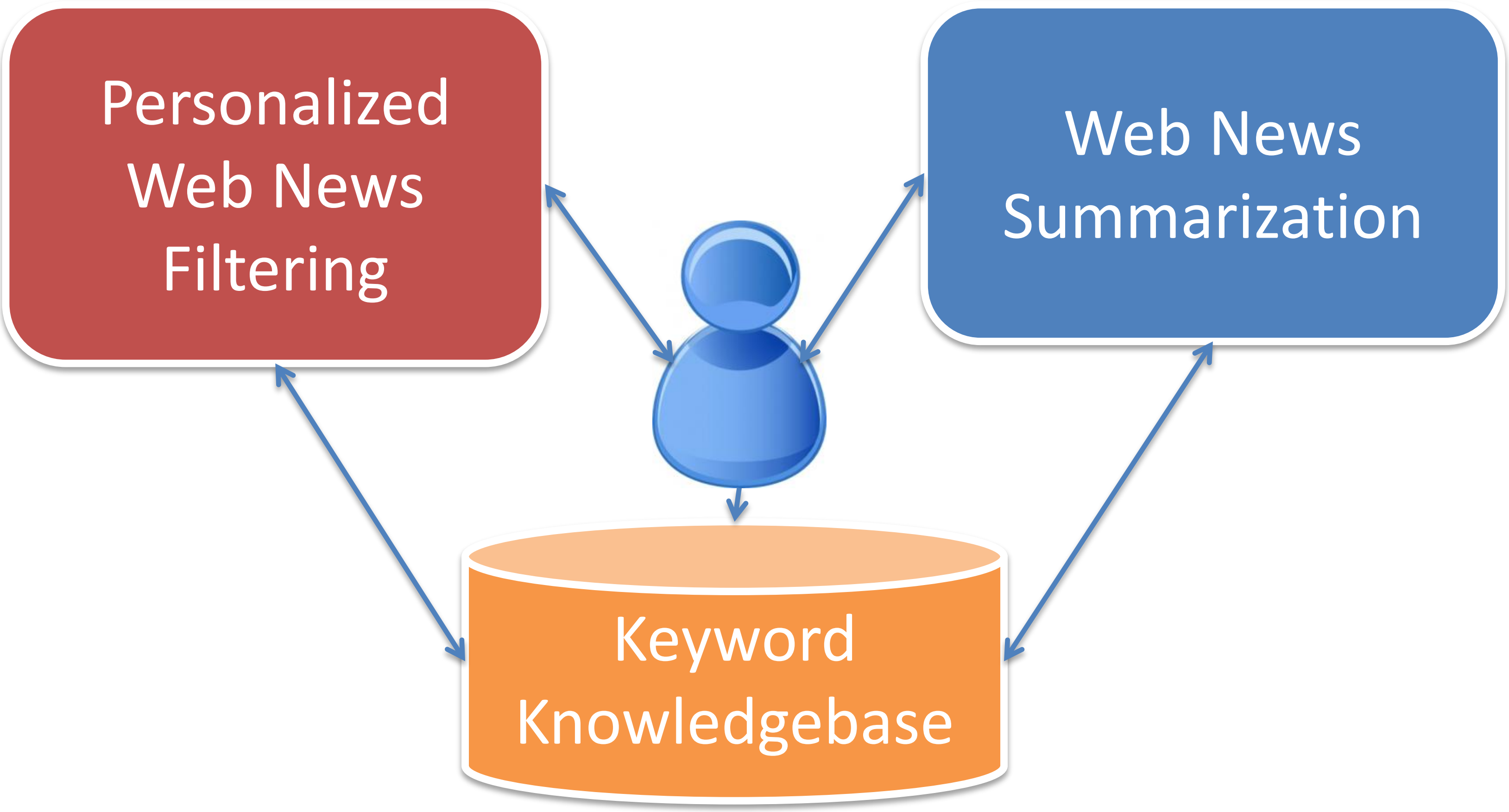


Xindong Wu, Fei Xie, Gongqing Wu and Wei Ding, **Personalized News Filtering and Summarization on the Web.** IEEE ICTAI 2011, 414-421. *(Best Paper Award)*

Personalized Web News Filtering



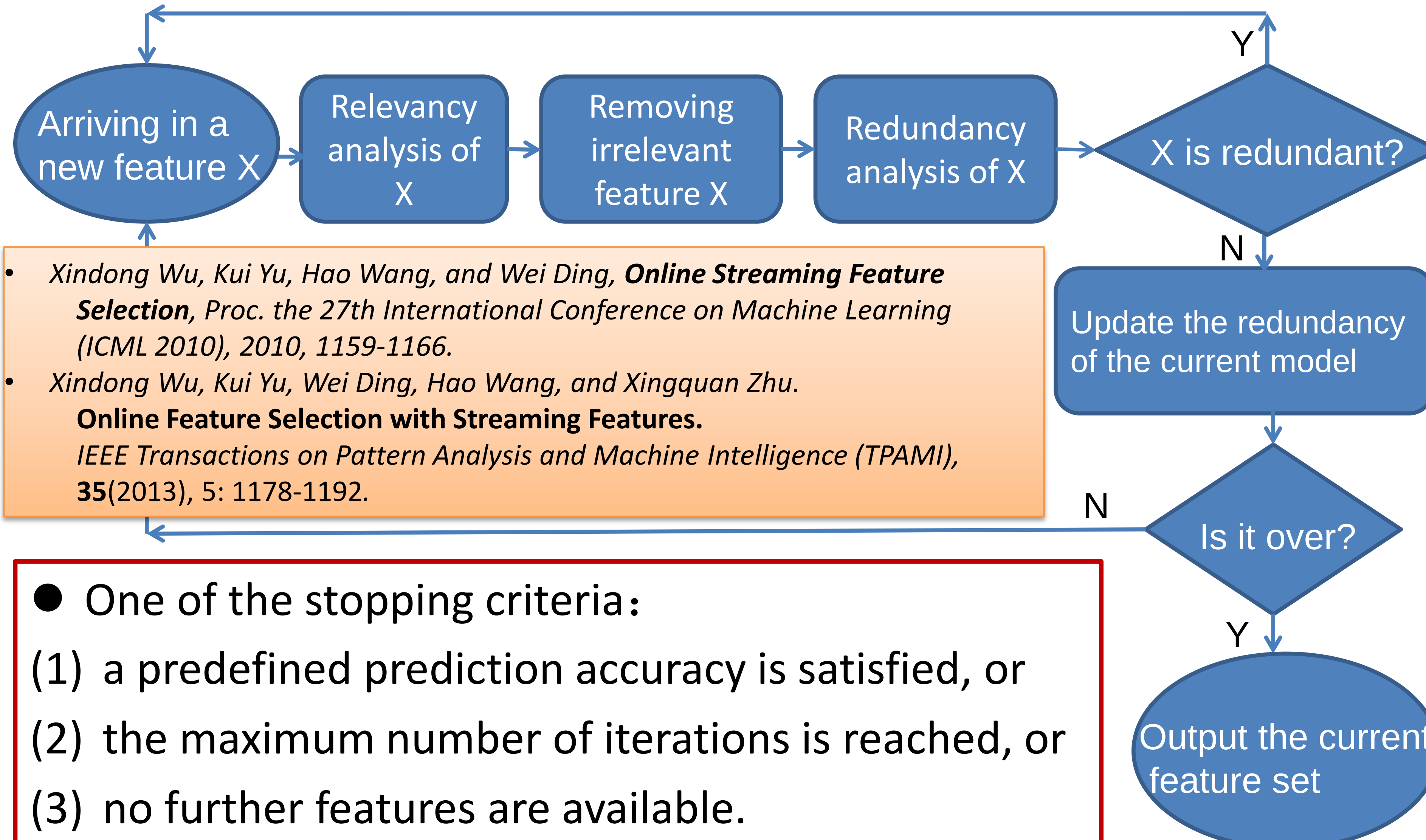
Personalized **N**ews **F**iltering & **S**ummarization (PNFS)



Streaming Features

- **Streaming features:** involve a feature stream that flows in one by one over time while the number of training examples remains fixed.
- **Online feature selection with streaming features**： To maintain an optimal feature subset over time from a feature stream, by online identify a redundant feature or an irrelevant feature upon its arrival.

A Framework for Streaming Feature Selection



- *Xindong Wu, Kui Yu, Hao Wang, and Wei Ding,* ***Online Streaming Feature Selection****, Proc. the 27th International Conference on Machine Learning (ICML 2010), 2010, 1159-1166.*
- *Xindong Wu, Kui Yu, Wei Ding, Hao Wang, and Xingquan Zhu.* **Online Feature Selection with Streaming Features.** *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI),* **35**(2013), 5: 1178-1192.

- One of the stopping criteria：

(1) a predefined prediction accuracy is satisfied, or

(2) the maximum number of iterations is reached, or

(3) no further features are available.

The OSFS algorithm

Online relevancy analysis

The OSFS algorithm

BCF={};
repeat
  added=0;
/* Stream in a new feature*/
  X←get_new_feature()
/*Online relevance analysis*/
  if $Dep(C, X|\emptyset)$
    added=1;
/*Add X to BCF */
    $BCF = BCF \cup X$;
 endif
/*Online redundancy analysis*/
if (added)
for each feature $Y \in BCF$
if $\exists S \subseteq$ BCF-Y s.t. $Ind(C, Y|S)$
/*Remove Y from BCF */
      BCF = BCF-Y;
   endif
 endfor
endif
until a predefined accuracy satisfied
output BCF

Online redundancy analysis

Fast-OSFS (a fast version of OSFS）

The Fast-OSFS algorithm

```
1.  BCF ={};
2.  repeat
3.      added=0;
4.  /*Stream in a new feature*/
5.     X←get_new_feature()
6.  /*online relevance analysis */
7.      if Dep(C,X|∅)
8.             added=1;
9.     endif
10. /*Redundancy analysis 1:*/
11. /* for X */
12. if (added)
13.    if ∃S ⊆ BCF s.t. Ind(C,X|S)
14. /*Discard X */
15.      go to Step 2
16.  endif
17.  /*Add X to BCF */
18. BCF = BCF ∪ X;
19. /*Redundancy analysis 2: */
20. /*for each feature within BCF*/
21. for each feature Y ∈ BCF-X
22. /*Find S ⊆ BCF containing X*/
23. if ∃S ⊆ BCF-Y s.t. Ind(C,Y|S)
24. /*Remove Y from BCF */
25.          BCF = BCF-Y;
26.    endif
27.  endfor
28. endif
29. until a predefined accuracy satisfied
30. output BCF
```

Online relevancy analysis for X

Online redundancy analysis for X

Redundancy analysis for the current feature set

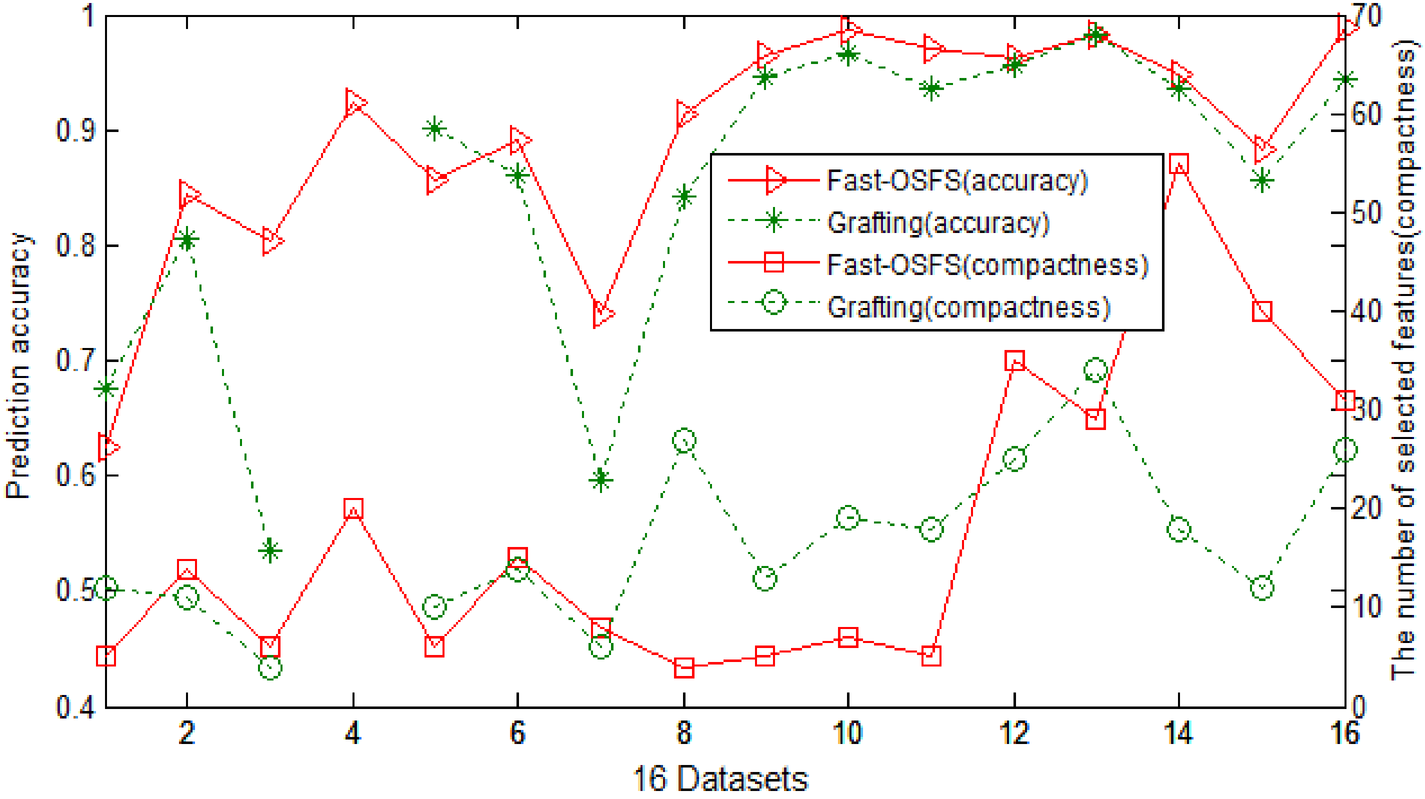
Experimental Results

- **Datasets: 16 high-dimensional datasets**

Dataset	#Features	# instances	Dataset	#Features	# instances
bankruptcy	147	7063	leukemia	7129	72
sylva	216	14374	prostate	6033	102
madelon	500	2000	lung-cancer	12533	181
arcene	10000	100	breast-cancer	17816	286
dexter	20000	300	ovarian-cancer	2190	216
dorothea	100000	800	sido0	4932	12678
lymphoma	7399	227	ohsumed	14373	5000
colon	2000	62	apcj-etiology	28228	15779

- **Competing algorithms： Grafting and Alpha-investing**
- **Evaluation metrics: Prediction accuracy and running time**

Fast-OSFS and Grafting



Prediction accuracy： the y-axis to the left (top two figures) ;
The size of the selected feature subset: the y-axis to the right (bottom two figures).

Running time-OSFS vs.Fast-OSFS

Runtime performance (in seconds) of OSFS and Fast-OSFS (alpha=0.01). (A/B in the second column denotes the runtime of OSFS, i.e., A, vs. the runtime of Fast-OSFS, i.e., B)

Dataset	Runtime		Dataset	Runtime
dexter	4/**1**		lymphoma	0/0
dorothea	64/**34**		breast-cancer	20/**4**
arcene	0/0		ovarian-cancer	1/0
madelon	0/0		sylva	1892/**170**
colon	0/0		bankruptcy	1272/**127**
prostate	0/0		sido0	10085/**410**
lung-cancer	6/**1**		apcj-etiology	11141/**139**
leukemia	0/0		ohsumed	2851/**66**

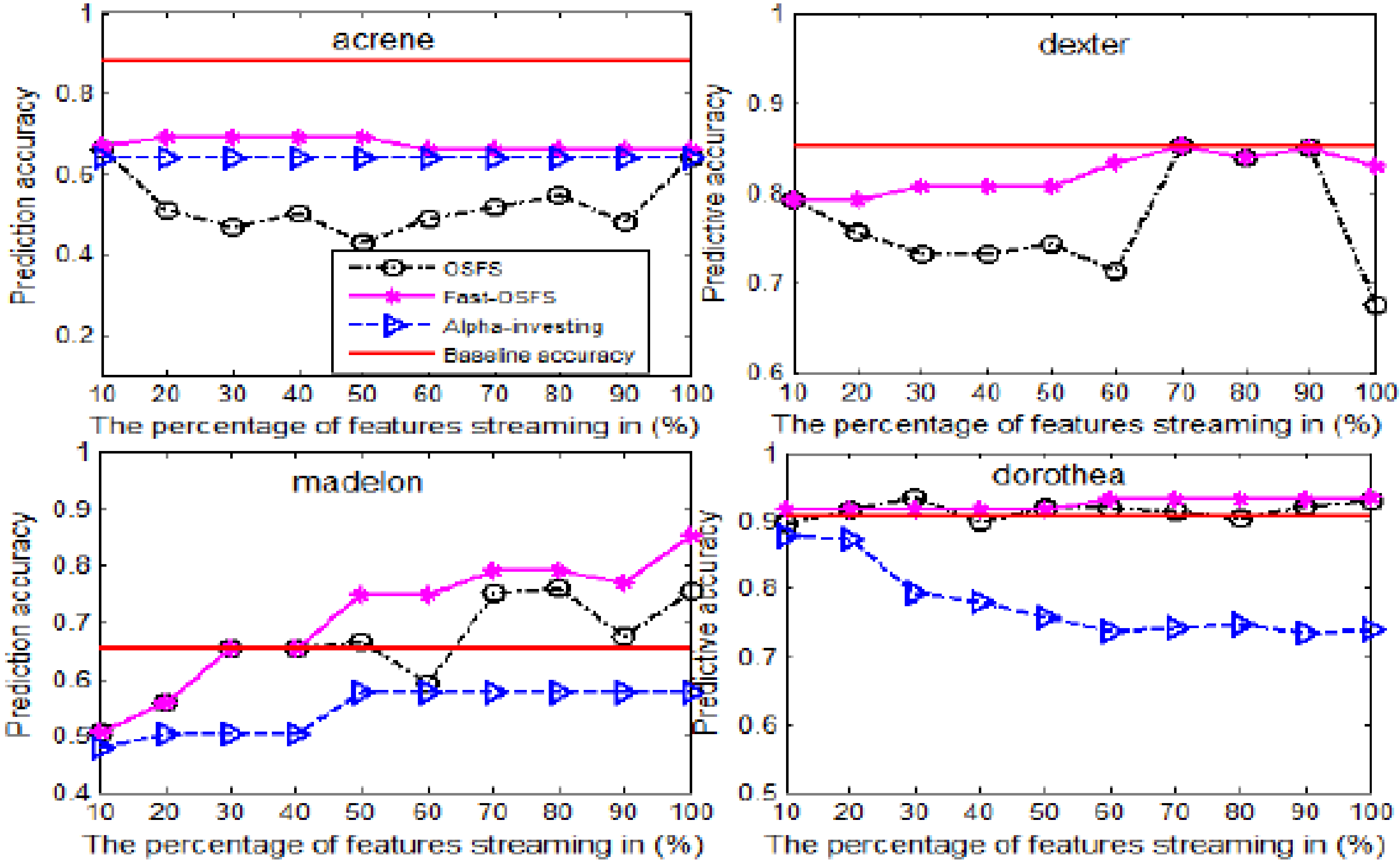
Runtime performance (in seconds) of OSFS and Fast-OSFS (alpha=0.05). (A/B in the second column denotes the runtime of OSFS, i.e., A, vs. the runtime of Fast-OSFS, i.e., B)

Dataset	Runtime		Dataset	Runtime
dexter	38/**2**		lymphoma	2/**1**
dorothea	1988/**78**		breast-cancer	97/**9**
arcene	1/0		ovarian-cancer	4/0
madelon	0/0		sylva	4807/**348**
colon	0/0		bankruptcy	3645/**261**
prostate	1/0		sido0	42789/**2014**
lung-cancer	10/**2**		apcj-etiology	118329/**676**
leukemia	0/0		ohsumed	156271/**1103**

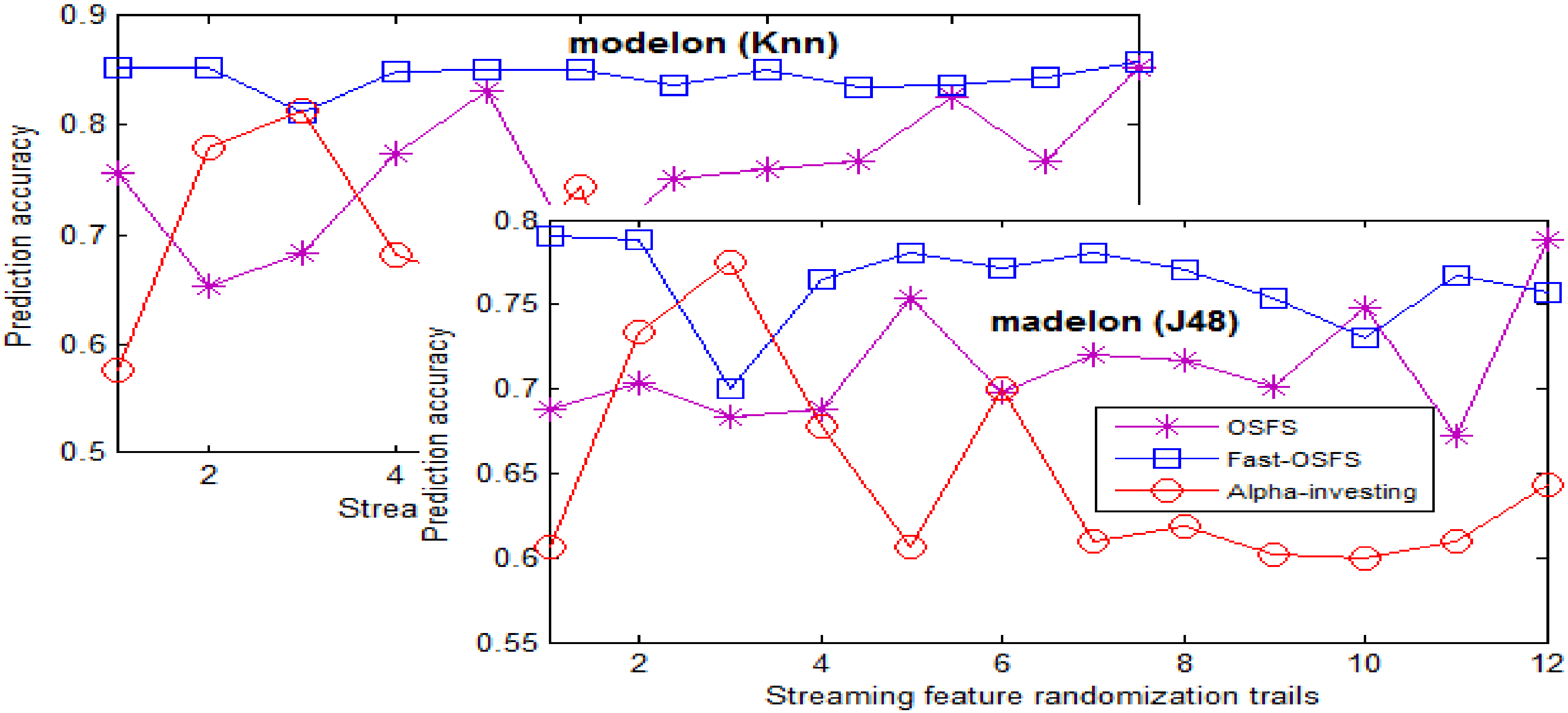
Prediction Accuracy with # Features

- Study the change of the prediction accuracy on *K*nn with respect to the features continuously arriving over time.
- In comparison with the prediction accuracy of the baseline *K*nn classifier trained using all features.
- **Conclusion:** Fast-OSFS achieves better and more stable performance on the models trained from selected streaming features.

Prediction Accuracy with # Features



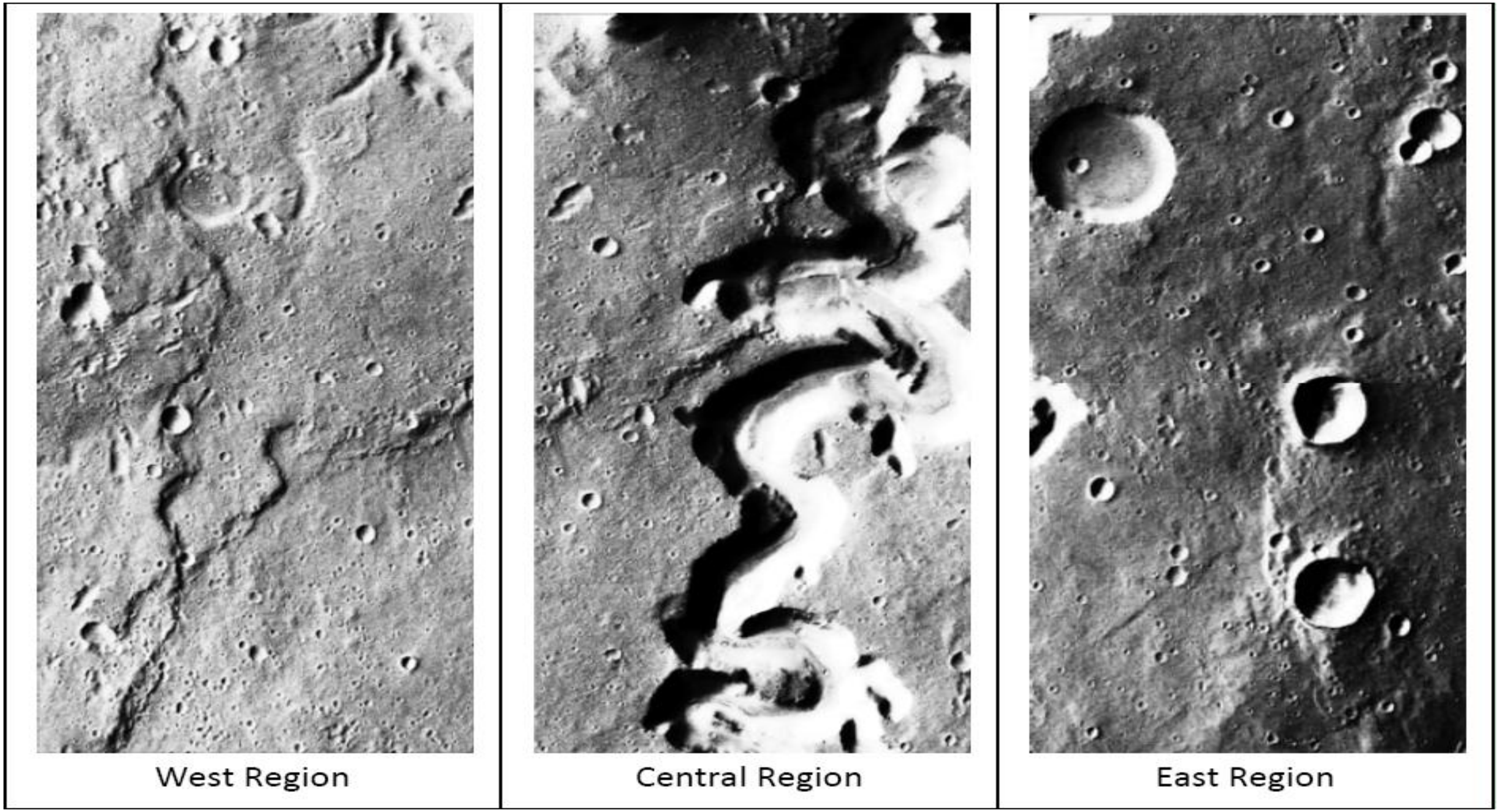
Impact of Ordering of Incoming Features



Observations:

1. Varying the order of the incoming features does impact on the final outcomes.
2. The results demonstrate that Fast-OSFS is the most stable method and Alpha-investing appears to be highly unstable.

A Case Study: Automatic Impact Crater Detection



Impact craters in a 37,500×56,250 m^2 test image from Mars

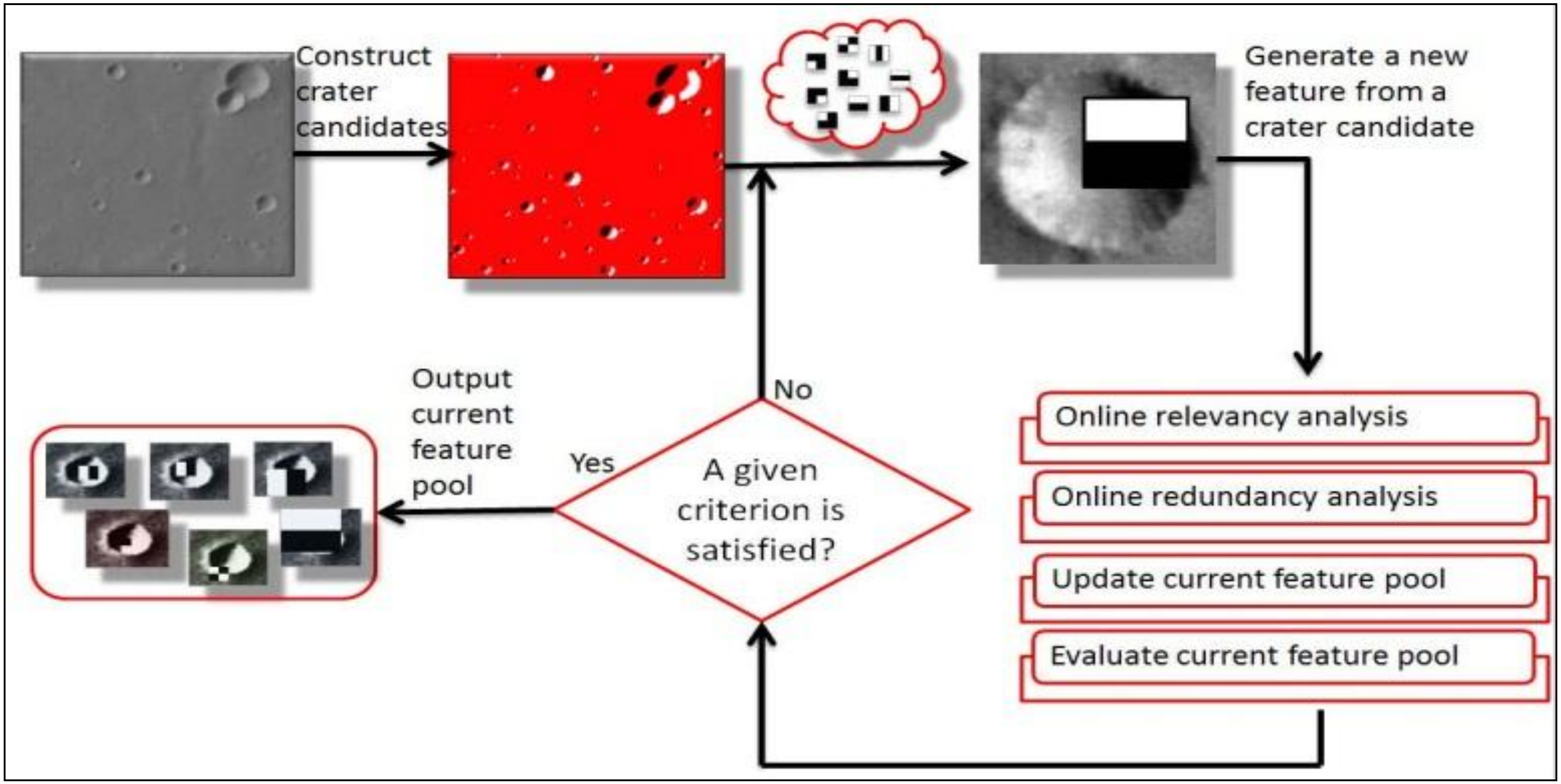
Streaming features with impact crater detection

Problem：Can we interleave feature generation and feature selection?

While rich texture features provide a tremendous source of potential features for use in crater detection tasks, they are expensive to generate and store.

A Case Study: Automatic Impact Crater Detection

A framework of streaming feature selection for crater detection



Training data and testing data

- **Training data:** consist of 204 true craters and 292 non-crater examples selected randomly from crater candidates located in the northern half of the east region.
- **Test data：**

	#samples (crater candidates)	#features
West region	6,708	1,089
Central region	2,935	1,089
East region	2,026	1,089

Experimental Results

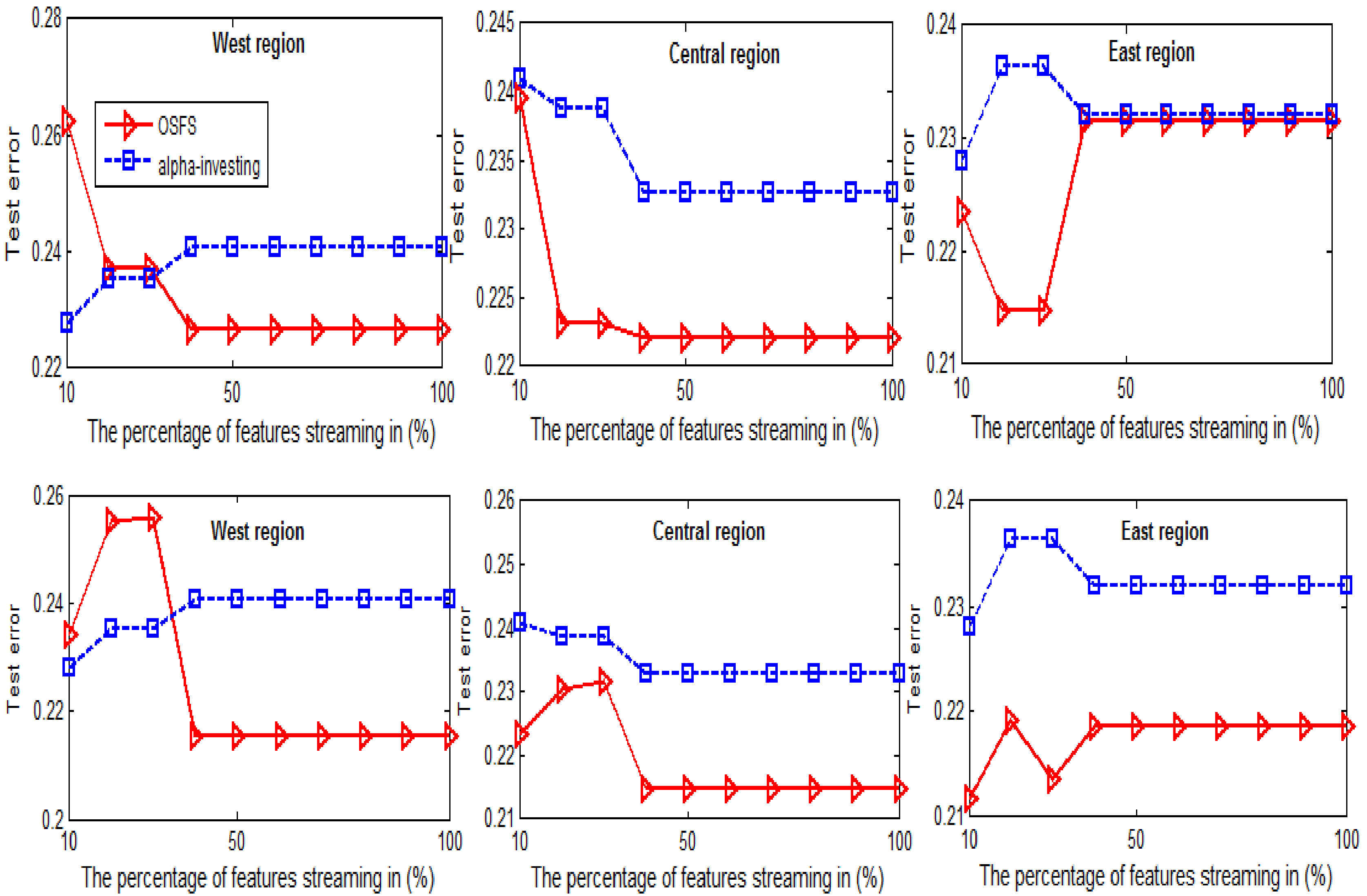
Prediction accuracy on three regions (alpha=0.01)

	#Selected features	West region	Central region	East region
OSFS	4	**0.7753**	**0.7826**	0.7725
Fast-OSFS	4	**0.7753**	**0.7826**	0.7725
Alpha-investing	16	0.7589	0.7666	**0.7730**

Prediction accuracy on three regions (alpha=0.05)

	# Selected features	West region	Central region	East region
OSFS	5	**0.7809**	**0.7874**	**0.7828**
Fast-OSFS	5	**0.7809**	**0.7874**	**0.7828**
Alpha-investing	16	0.7589	0.7666	0.7730

Prediction accuracy with # features arrived



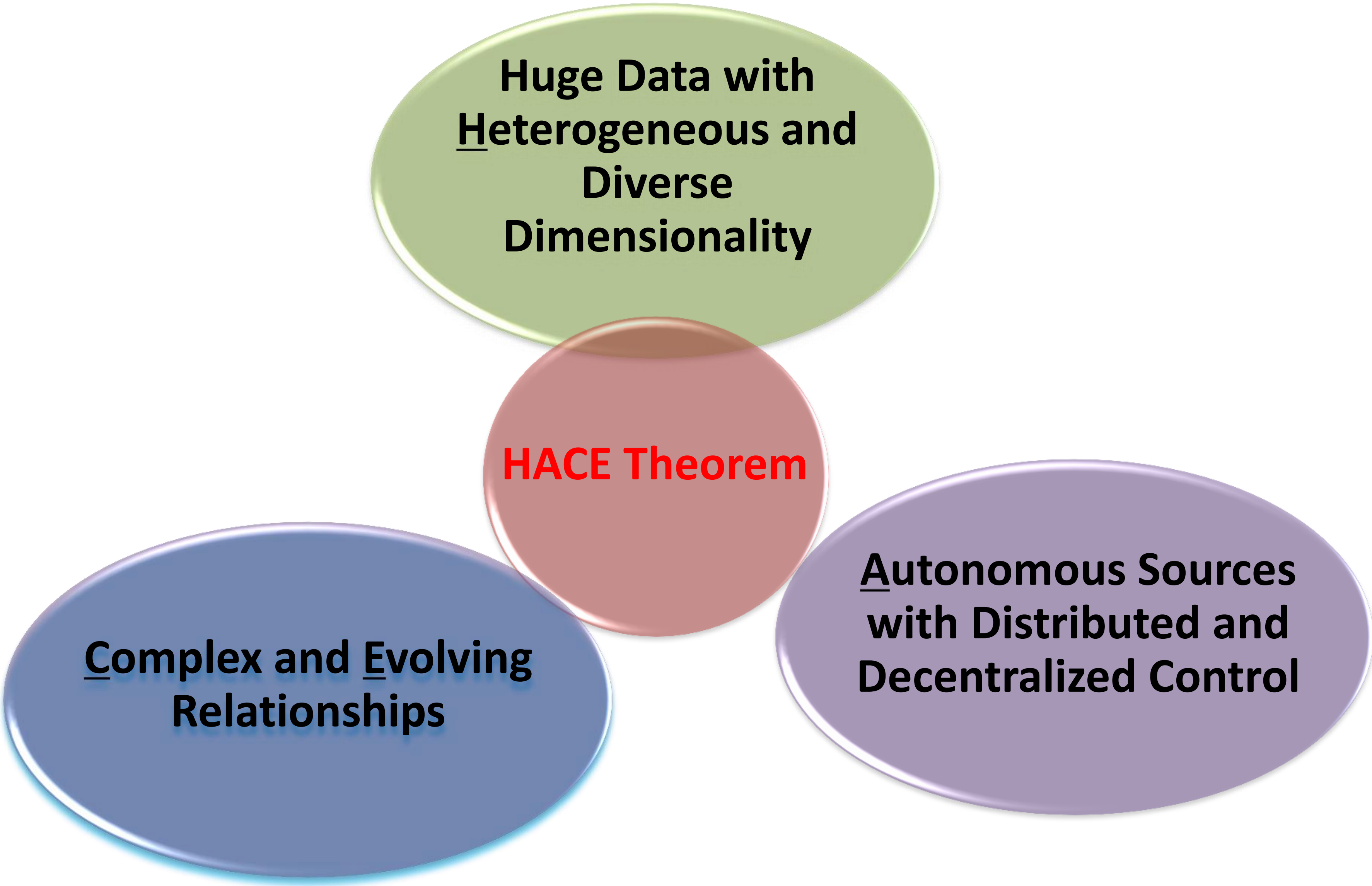
Comparison w/ Traditional Feature Selection

	#Selected features	West region	Central region	East region
OSFS	7	**0.7809**	0.7874	**0.7828**
Fast-OSFS	7	**0.7809**	0.7874	**0.7828**
HITON_PC	6	0.7749	0.7792	0.7813
LARS	6	0.7740	0.7881	0.7799
Naïve Boost	150	0.7661	**0.7888**	0.7749
No feature selection	1089	0.7303	0.7499	0.7710

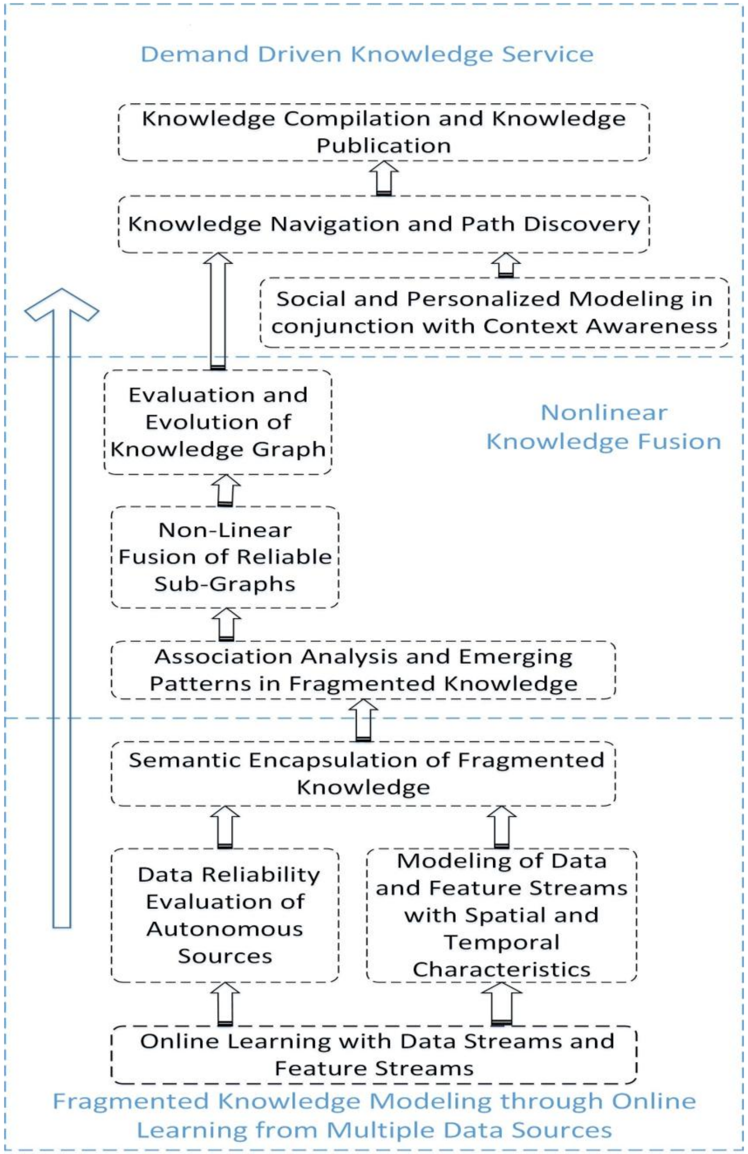
Outline

1. The Era of Big Data
2. Big Data Characteristics
3. A Big Data Processing Framework
4. Streaming Data and Streaming Features
5. **Concluding Remarks**

Conclusion: HACE Theorem w/ Big Data



From Big Data to Big Knowledge Services



- Knowledge Acquisition
  - Fragmented knowledge vs in-depth expertise
  - On-line learning with data streams & feature streams
- Knowledge Fusion
  - Knowledge graph
  - Knowledge evolution
- Knowledge Services
  - Navigation and path discovery with a knowledge graph
  - Knowledge compilation and knowledge publication

Thanks and Questions