Nov. 30, 2014

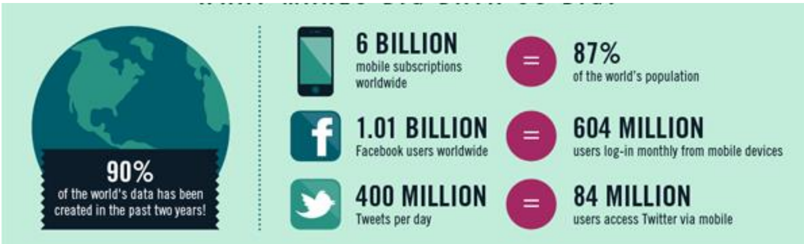
Introduction to Synchronization-based Big Data Mining

Junming Shao

Big Data Research Center, Web Sciences Center
School of Computer Science and Engineering, UESTC
Email：junmshao@uestc.edu.cn
Http://staff.uestc.edu.cn/shaojunming

BIG DATA

Media/Entertainmet



- <Dokument>
- <Patient>
<Name>Mustermann</Name>
<Vorname>Max</Vorname>
<Geburtsdatum>01.02.2009</Geburtsdatum>
</Patient>
- <Befund>
<Klassifikation>Korrespondenz</Klassifikation>
<Beschreibung>Beschreibung der Möglichkeiten von Plugin´s</Beschreibung>
<StandortID>Wien - 1210</StandortID>
<WSName>MW-PC</WSName>
<UserName>wmay</UserName>
<ErstellungsDatum>05.08.2009</ErstellungsDatum>
<ErstellungsZeit>14:34:12</ErstellungsZeit>
</Befund>
</Dokument>

Healthcare

DNA

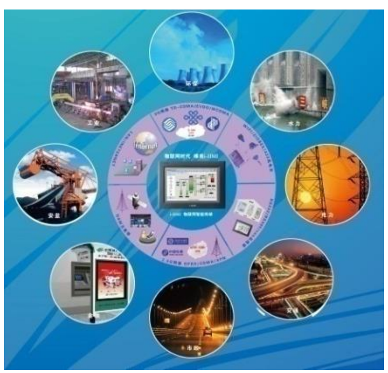
fMRI/ DTI

Messenger Watch

TCCAGGTAGTGGACGTTACACCTAC
CATGGCTCCTCCACCTAACCAGCAG
GTATGGACAGCAATATGGGCAACAA
ACCAGGTCCTCCCCCTATGGCTTAT

Gene Sequence

Industry



Sensor

Manufacture

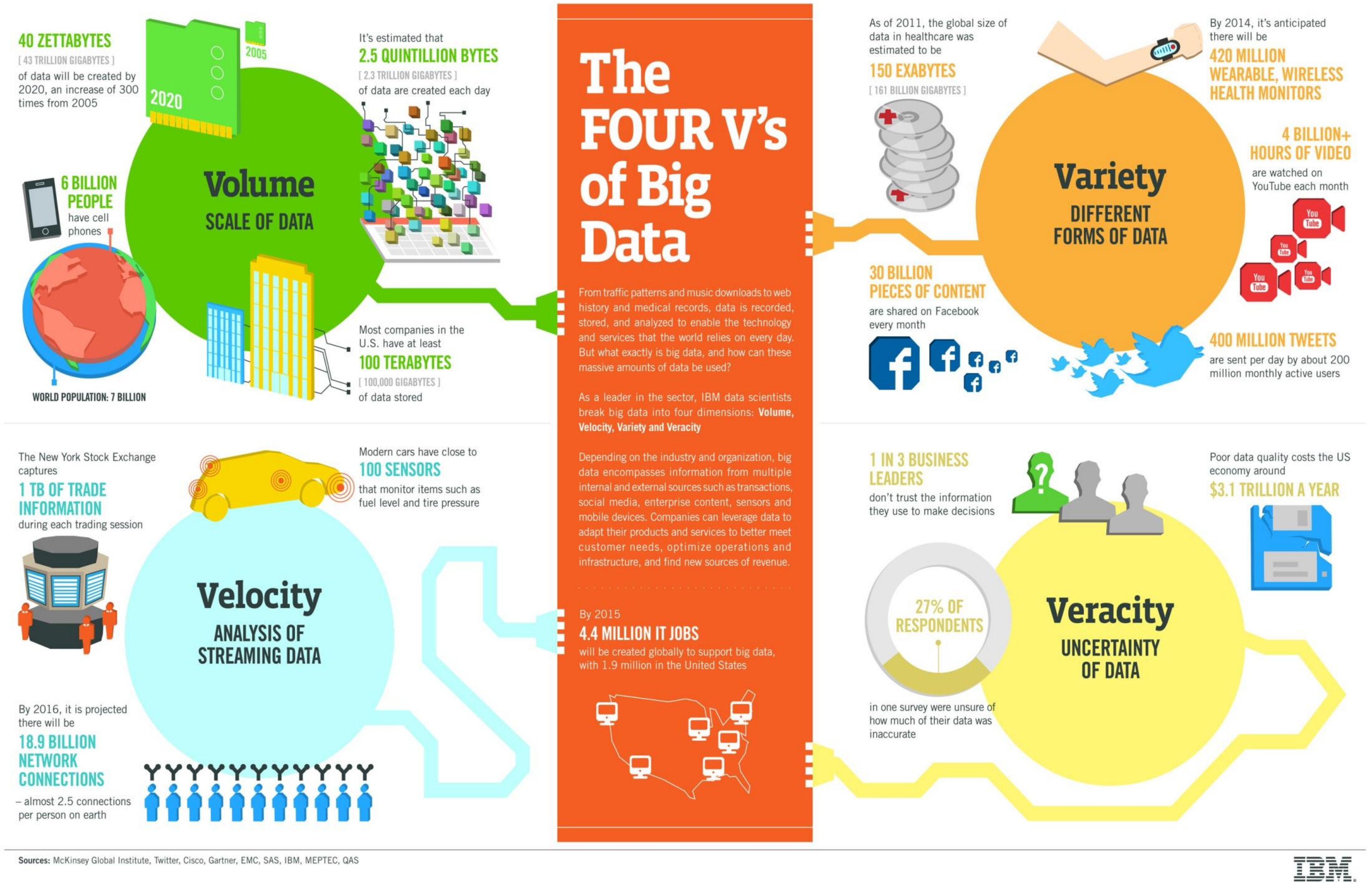
E-commerce

Wall Mart: 2.5 PB/hour

Stock Data

*Note: some pictures derived from internet

FEATURES—The FOUR V's of BIG DATA

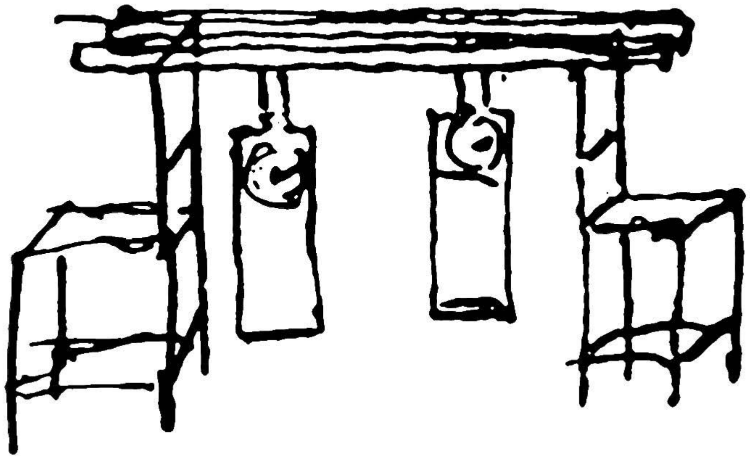


SYNCHRONIZATION

A Powerful Mechanism For Big Data Mining

Synchronization: *An universal concept in nature*.

Christian Huygens (1629–1695)



Two pendulum clocks placed on a common support had synchronized (Huygens, 1673)

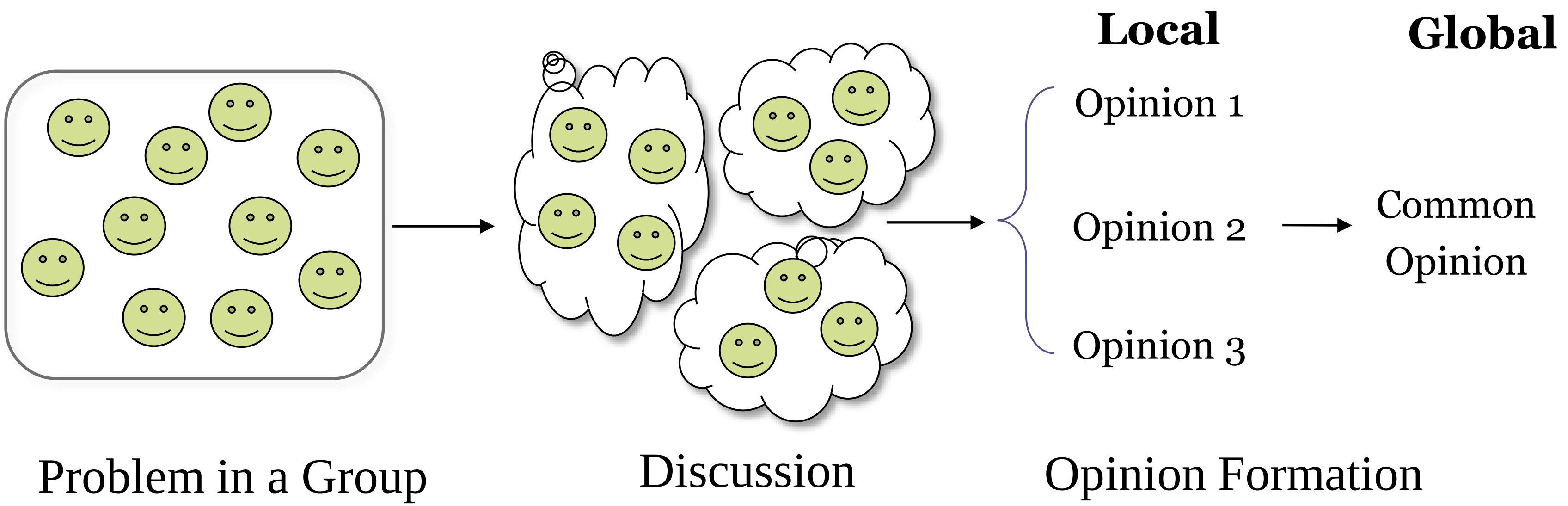
Examples

- Biology: *fireflies, crickets, yeast*
- Neuroscience: *heart, brain, menstrual cycle*
- Biochemistry: *cellular clocks, genetic circuits*
- ……

What is Synchronization?

Synchronization: is a phenomenon that a group of events spontaneously come into co-occurrence with a common rhythm, despite of the differences between individual rhythms of the events.

E.g. opinion formation



How to explore the synchronization phenomena?

— **Kuramoto Model**

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K}{N}\sum_{j=1}^{N}\sin(\theta_j - \theta_i), \qquad i = 1,\ldots, N$$

where ω_i describes the natural frequency, θ_i is the phase of *i-th* oscillator and K is the couple constant.

Properties:

- *motivated by the behavior of systems of biological oscillators.*
- *simple enough*
- *weakly-coupled, nearly identical oscillators*
- *global Synchronization*

Inspiration

- **Synchronization Phenomena**
- **Kuramoto Model**

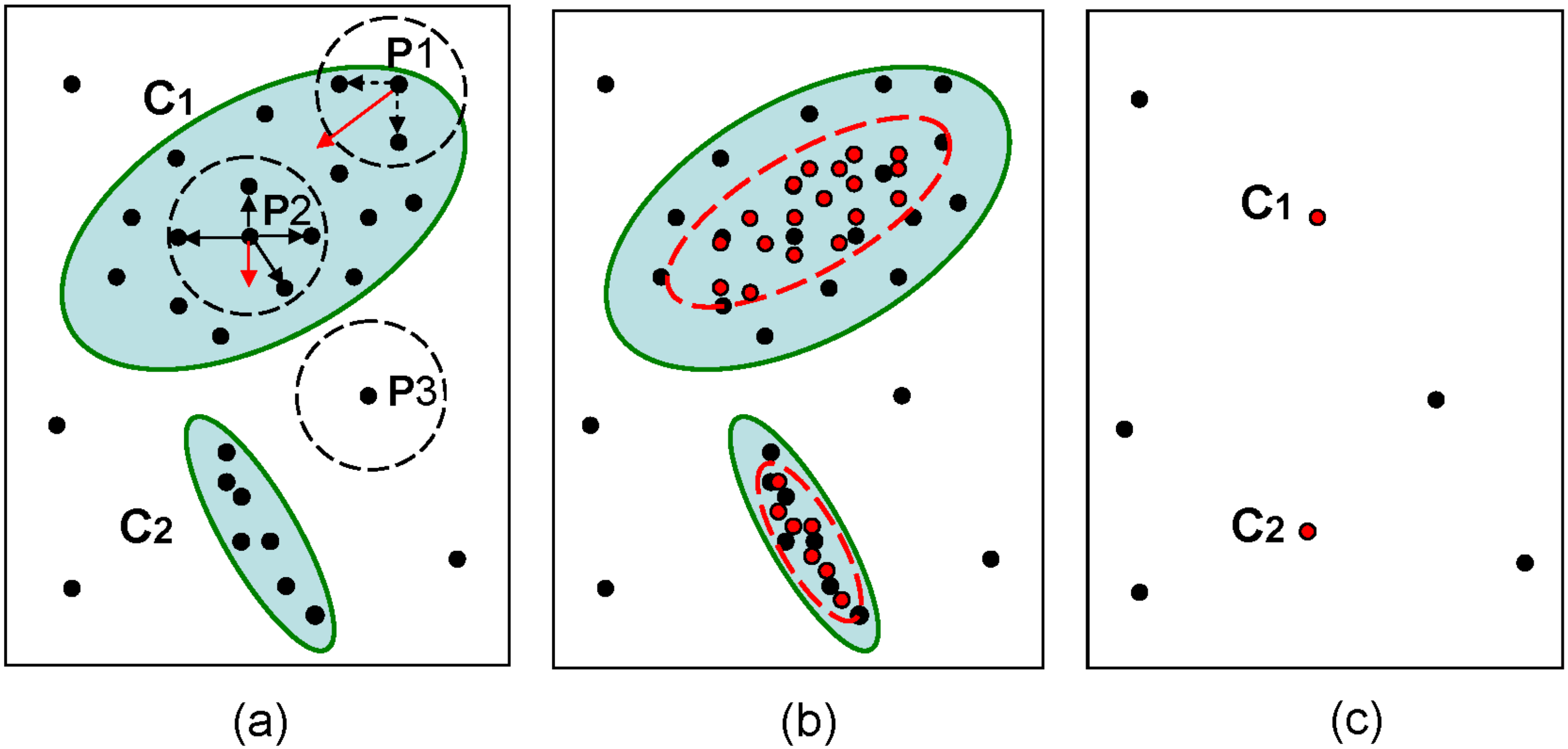
Synchronization- based Data Mining

- Sync [Clustering by Synchronization] *[KDD 2010]*
- hSync [Hierarchical Clustering] *[TKDE 2012]*
- SOD [Outlier Detection] *[PKDD/ECML 2010]*
- ORSC [Arbitrarily Oriented Synchronized Clusters] *[ICDM 2011]*
- SyncStream [Data Stream Classification] *[KDD 2014]*

Sync: Clustering by Synchronization

Basic Idea: Uncover the data structure by investigating the dynamics of objects during the process towards Synchronization.

- Each data object/node is regarded as **a phase oscillator**
- It interacts with its neighbors through an **Interaction Model** in a local fashion
- **Simulate dynamic behaviors of objects over time**
  - **Regular objects** synchronize together and form distinct clusters
  - **Outliers/ Noisy objects** tend to remain stable all the time



(a) The initial state of objects. (b) The comparison of objects states before and after one time step. (c) The final state of objects towards synchronization.

Interaction Model

Kuramoto Model: $$\frac{d\theta_i}{dt} = \omega_i + \boxed{\frac{K}{N}\sum_{j=1}^{N}\sin(\theta_j - \theta_i)}$$ ***Global Interaction***

Local Synchronization for Clustering

$$\frac{dx_i}{dt} = \omega_i + \boxed{\frac{K}{|N_\varepsilon(x)|}\sum_{y \in N_\varepsilon(x)}\sin(y_i - x_i)}$$ ***ε - Neighborhood Interaction***

Let $dt = \Delta t$, then:

$$x_i(t + \Delta t) = x_i(t) + \Delta t \cdot \omega_i + \frac{\Delta t \cdot K}{|N_\varepsilon(x(t))|} \cdot \sum_{y \in N_\varepsilon(x(t))}\sin(y_i(t) - x_i(t))$$

Let all objects have the same frequency ω, the term $\Delta t \cdot \omega_i$ is the same for each object and thus ignored. $\Delta t \cdot K$ is a constant and simply fix it as 1.

Clustering Model

Coupling Function

$$x_i(t+\Delta t) = x_i(t) + \underbrace{\frac{1}{|N_\varepsilon(x(t))|} \cdot \sum_{y \in N_\varepsilon(x(t))} \sin(\boxed{y_i(t) - x_i(t)})}_{\text{Local Interaction}}$$

i-th dimensional phase of x at time $t+\Delta t$

Local Interaction Range

Phase Diff.

Cluster Order Parameter $r_c = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|N_\varepsilon(x)|} \left(\sum_{y \in N_\varepsilon(x)} e^{-\|y-x\|} \middle| x \in D \right)$

How to find the optimal Local Interaction Range ?

Minimum Description Length (MDL) Principle

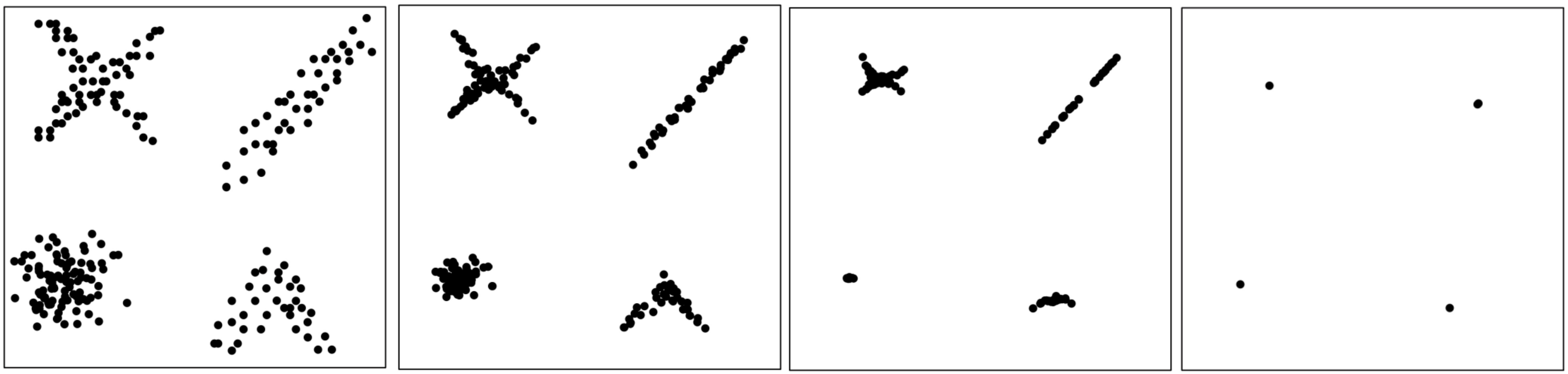
$$\mathbf{L(D,M)} = \mathbf{L(M)} + \mathbf{L(D|M)}$$

$$\sum_{i=1}^{K}\sum_{j=1}^{|C_i|}\log_2\left(\frac{N}{|C_i|}\right) + \sum_{i=1}^{K}\frac{p_i}{2}\log_2(|C_i|) \qquad -\sum_{i=1}^{K}\sum_{x\in C_i}\log_2(pdf(x))$$

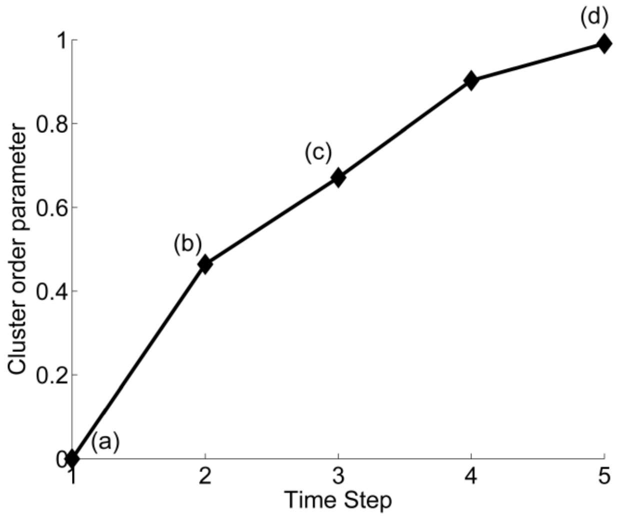
Cluster-ID   Free Parameters   Data

Clustering result with Global Minimal MDL value

Dynamical Clustering



(a) t=0 (b) t=Δt (c) t= 2Δt (d) t=4Δt



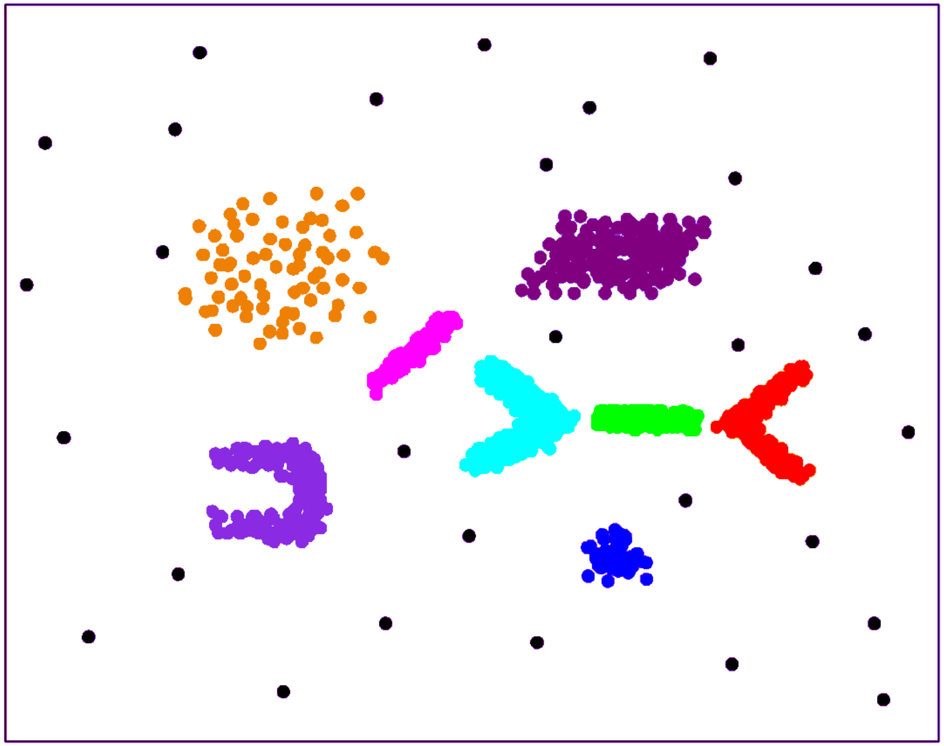
(e)

The dynamics of objects during the process of synchronization.

(a) – (d): The detail states of objects over time. (e). Corresponding cluster order parameter.

Evaluation

❖ Comparison on Synthetic Data

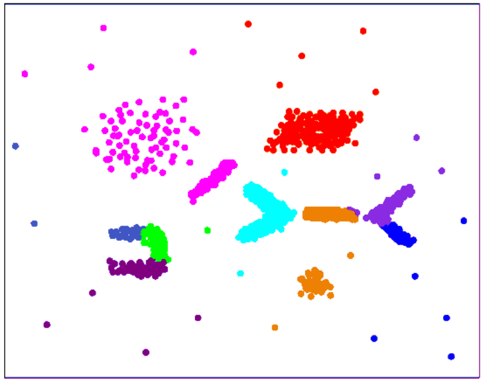


Sync

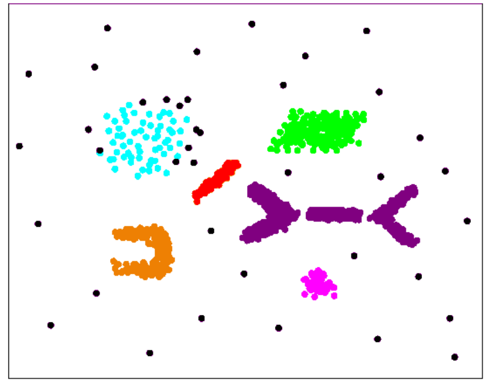
✓ Arbitrarily shapes

✓ Multiple densities

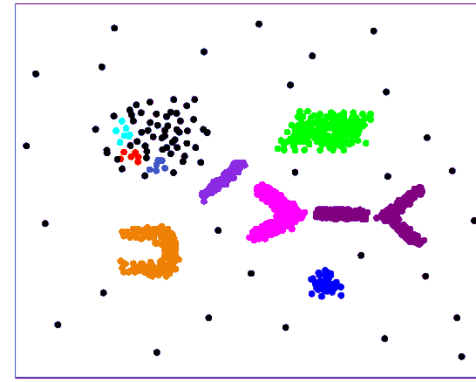
✓ In the sea of noise



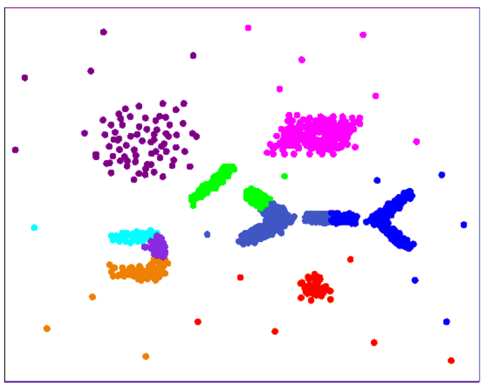
K-Means (K=9)



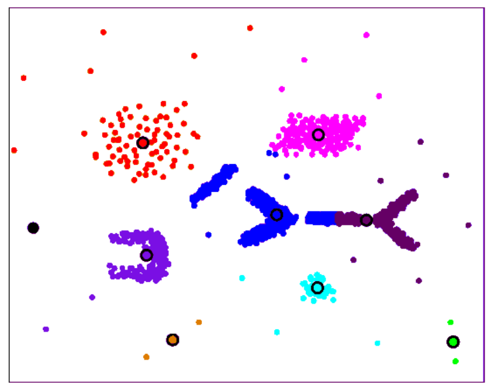
DBSCAN(ε=0.035)



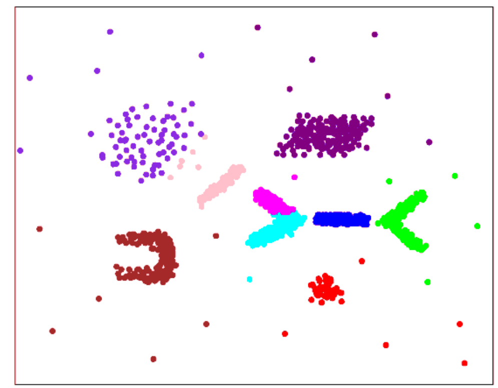
DBSCAN(ε=0.025)



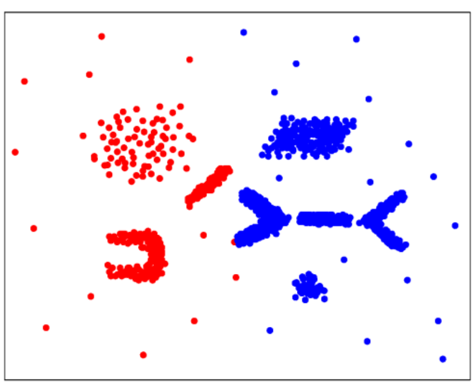
SC (K=9)



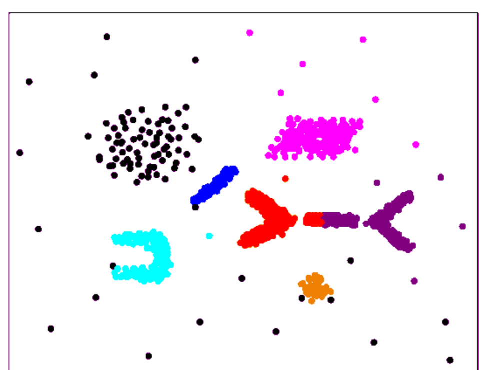
Mean-Shift(b=6.3)



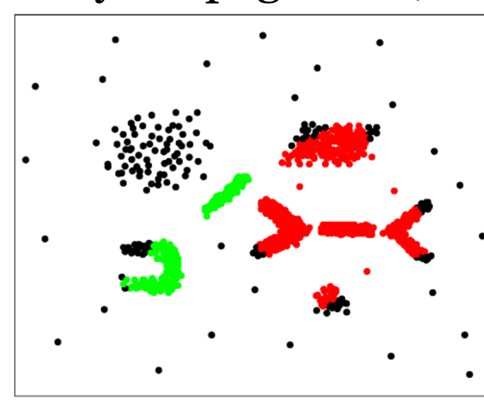
Affinity Propagation (K=9)



X-Means



RIC



OCI

❖ Real Data - Wisconsin Data

Performance:

- Find the correct number of clusters;
- Detect natural clusters (with high EC value);
- Discover almost all clusters with high recall (96.2% and 97.5%);
- All instances in each cluster match with corresponding type (with highest precision of 98.6% and 93.2%).

Table 1. Performances on Wisconsin Data

Algorithms	Sync	X-Means	RIC	OCI
EC	**0.154**	0.183	0.182	**0.154**
NMI	**0.777**	0.324	0.344	0.274
AMI	**0.777**	0.322	0.343	0.272
AVI	**0.782**	0.464	0.475	0.411

❖ Real Data - Diabetes Data

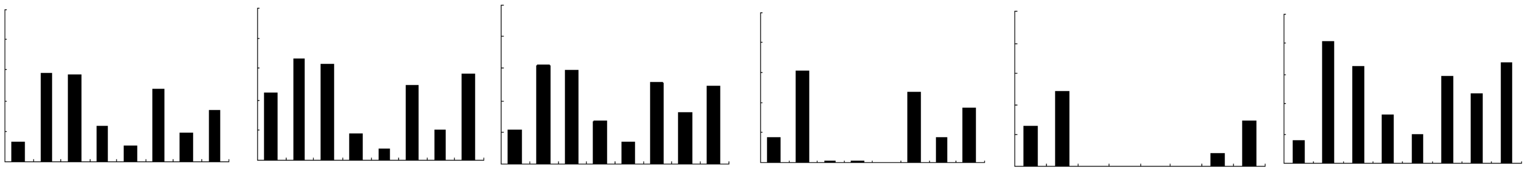


Fig. : Illustration of the result of *Sync* on diabetes data: Each bar in each of the 6 clusters indicates the mean value of different factors and is scaled to [0,1].

Table 2. Performances on Diabetes Data

Algorithms	**Sync**	**X-Means**	**RIC**	**OCI**
EC	**0.625**	0.656	0.661	0.635
NMI	**0.051**	**0.051**	0.011	0.032
AMI	0.048	**0.050**	0.009	0.031
AVI	**0.058**	0.051	0.011	0.038

Desirable properties of *Sync*

- Novel clustering notion: *Synchronization*
- Arbitrarily shaped clusters detection without data distribution assumption;
- Fully automatic clustering in combination with MDL.

hSync: Hierarchical Synchronization-based Clustering

Flat Clustering *vs* Hierarchical Clustering

Problems of existing hierarchical clustering algorithms
(e.g. Single Link, OPTICS)

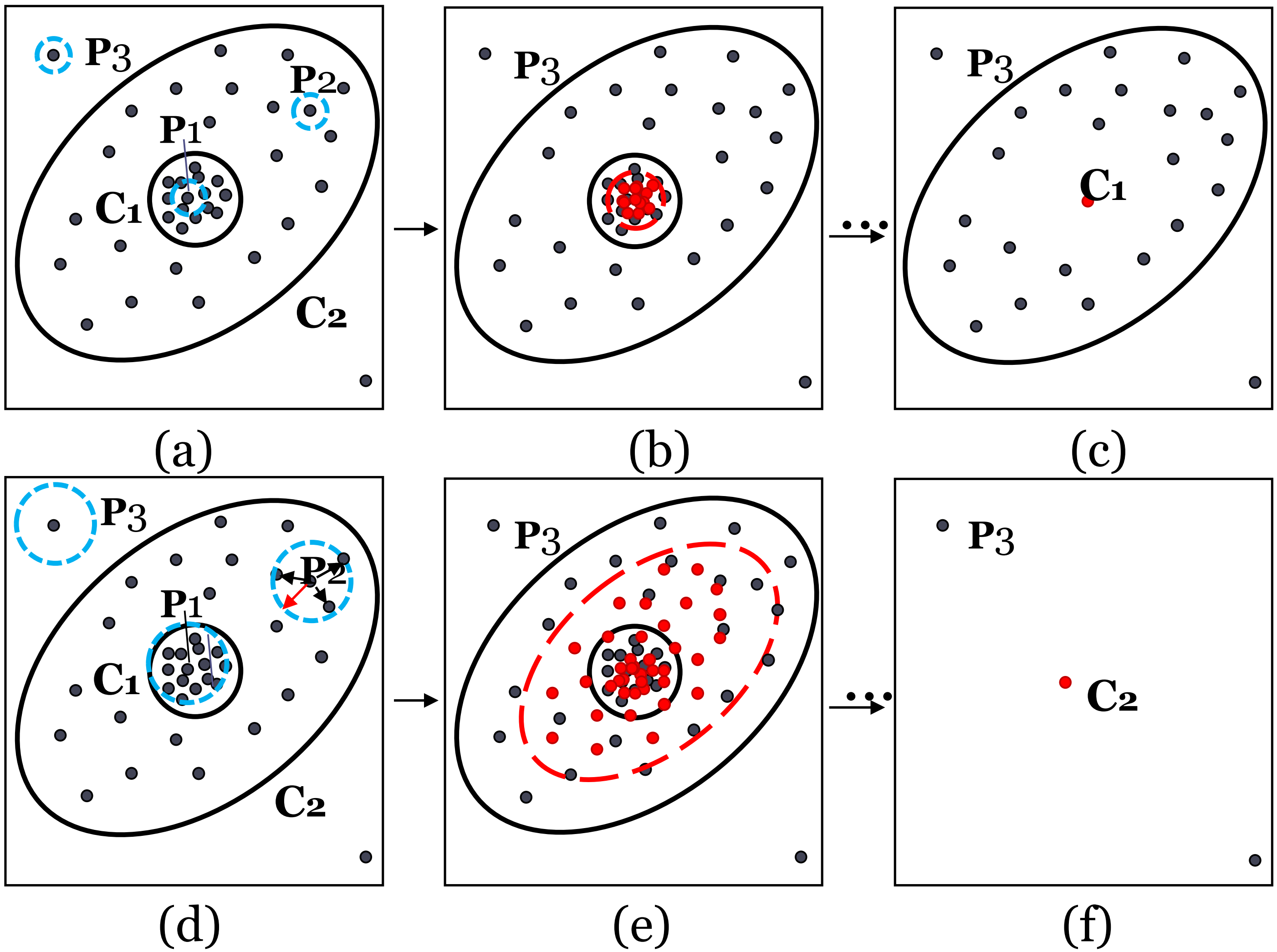
- Natural hierarchical structure detection
- Interpretation of hierarchies
- Noise / Outlier

hSync: Extending the algorithm *Sync* to hierarchical data analysis.

Intuition

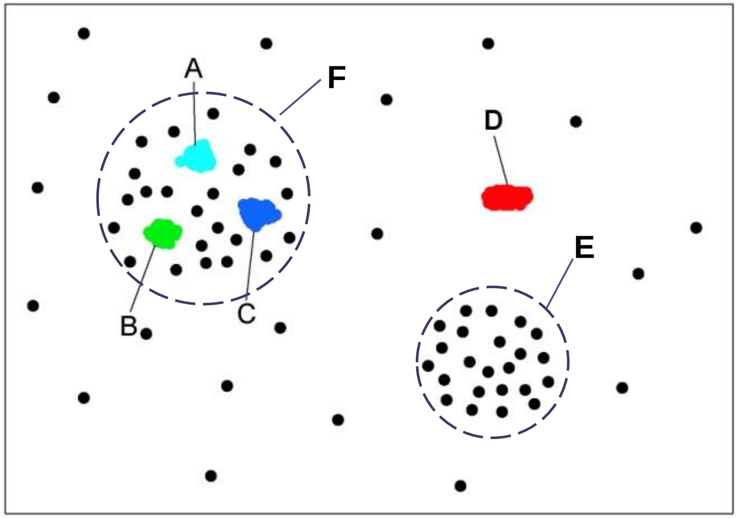
Interaction range in L1

Interaction range in L2

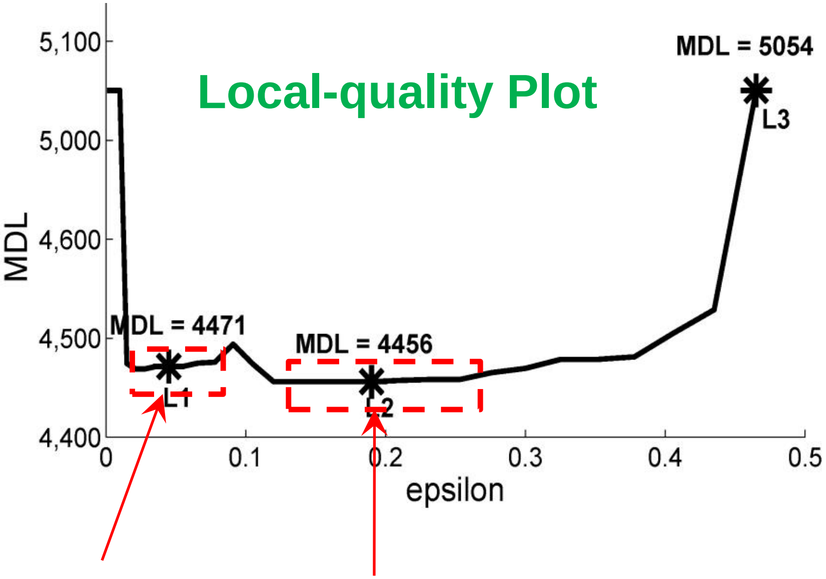


Key Observation

Key Observation: If a data set exhibits a hierarchical cluster structure, the MDL values of coding the clustering results with different interaction ranges show several distinct stable local minima in the **Local-quality Plot**.



Sample Data



First stable local minima

Second stable local minima

Illustration

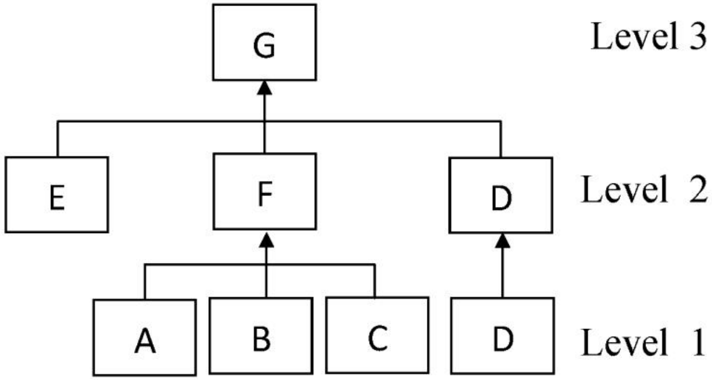
Representative Point: Middle point of Local Stable Minimal Range

Let J be the set of all intervals $J = (\varepsilon_L, \varepsilon_U)$ where MDL(ε) is sufficiently constant. Then the mean of each of these intervals defines a representative $\varepsilon^{Key} = ½ (\varepsilon_L + \varepsilon_U)$.

Cluster Structure Exploring

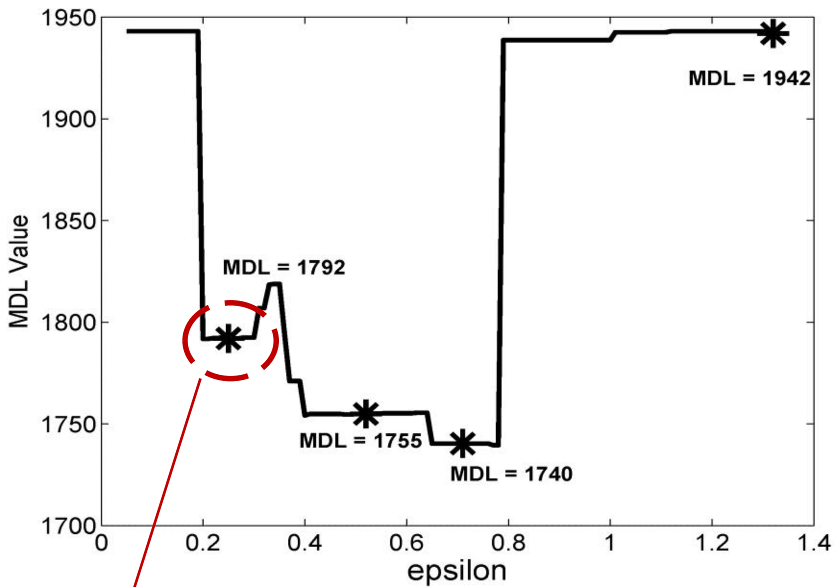
The representative points represent the hierarchical clusterings of high quality

from small-scale to large-scale by simulating the way to synchronization over different levels of locality.



Cluster Structure

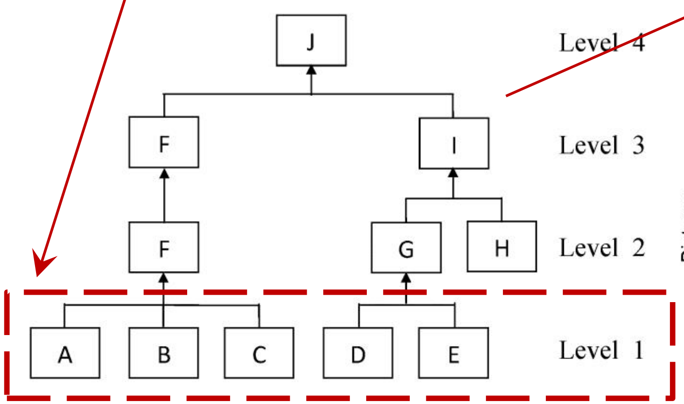
Evaluation



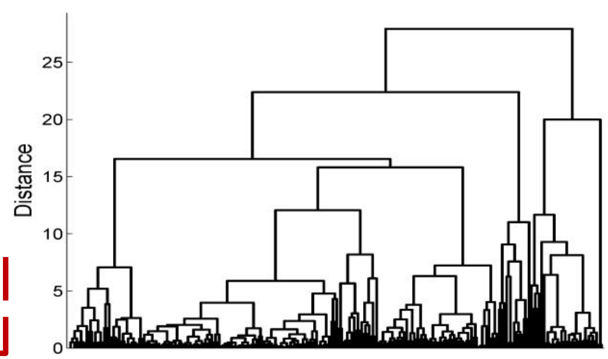
Local-quality Plot

❖ **Real Data - Glass Data**

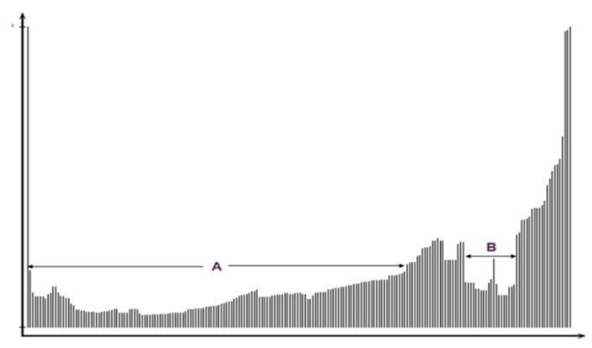
A: "building windows float processed"
B: "building windows non float processed"
C: "vehicle windows float processed"
D: "non-window glass containers"
E: "non-window glass tableware"
F: "window glass"
G: "Part of non-window glass"
H: "head-lamps"
I: "non-window glass"



Cluster Structure



Single Link

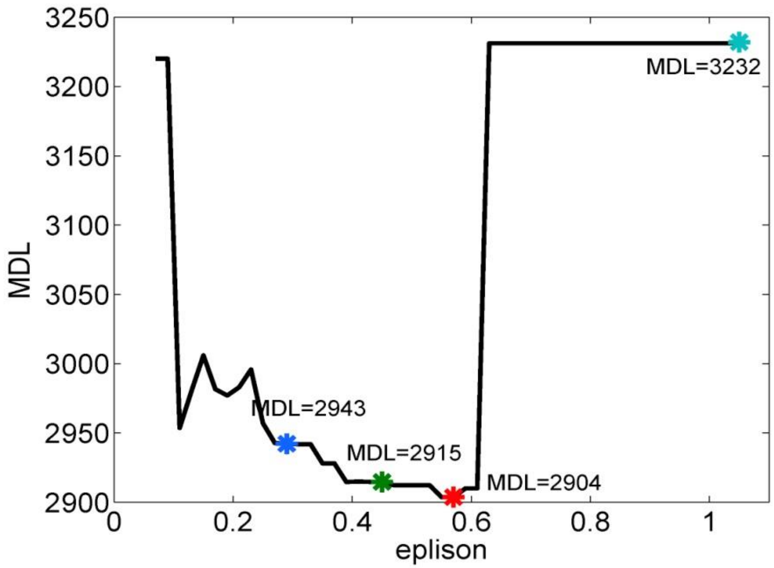


OPTICS

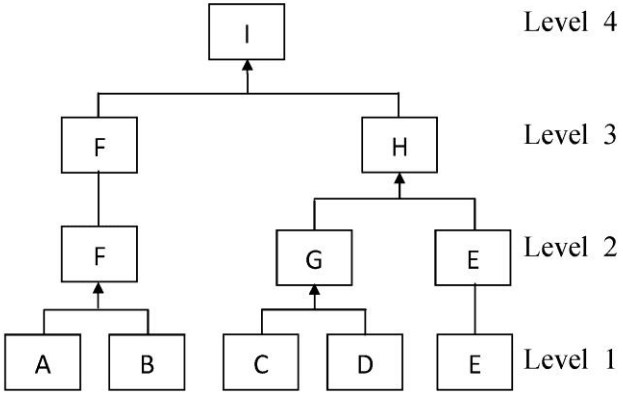
Evaluation (cont'd)

❖ **Real Data – Ecoli Data**

A: ""cytoplasm (cp)"
B: "perisplasm(pp)"
C: "inner membrane without signal sequence (im)" and "inner membrane, uncleavable signal sequence (imu)"
D: "im"
E: "outer membrane (om)"
F: "cp" and "pp"
G: "imu" and "im"
H: "imu", "im" and "om"

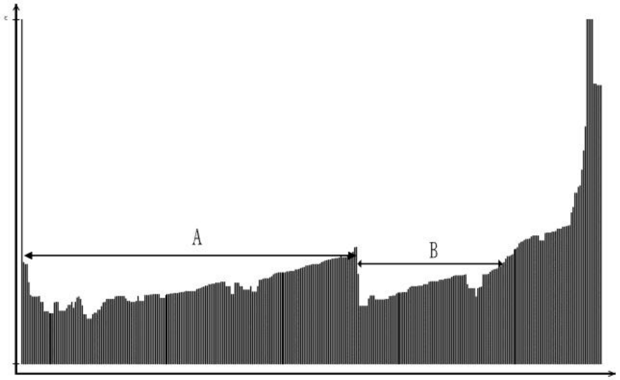


Local-quality Plot



Cluster Structure

Single Link



OPTICS

Conclusion

1. **Robust discovery of natural cluster hierarchies.** The inherent hierarchical nature of synchronization allows an intuitive and effective approach for hierarchical clustering. The algorithm *hSync* explores the hierarchical cluster structure from micro-scale to macro-scale by simulating the way to synchronization over different levels of locality.

2. **Compact and interpretable cluster hierarchies**. In combination with MDL, the algorithm *hSync* generates an interpretable cluster tree consisting of meaningful levels only, each representing a clustering of high quality. Besides the cluster tree, the output of *hSync* includes the locality-quality diagram, a visualization which allows the user to comprehensively assess the quality of the cluster hierarchy over all Levels.

SOD: Outlier Detection

Definition: "An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism." [Hawkins 1980]

Existing approaches: LOF, LOCI, CoCo, ……

Challenges:

- Data Distribution Assumption
- Data of various densities & shapes
- Interpretation

Outlier objects ⟶ ***"out of synchronization"***

Illustration

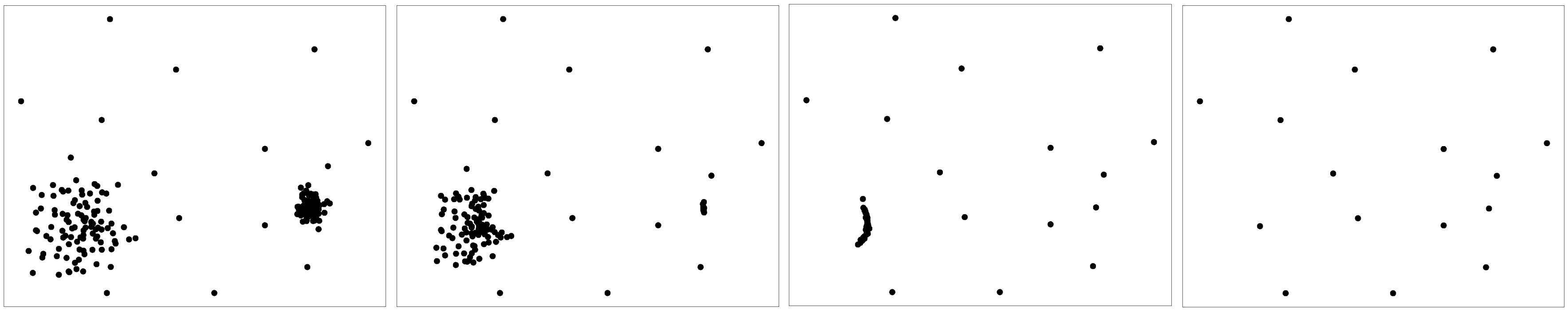
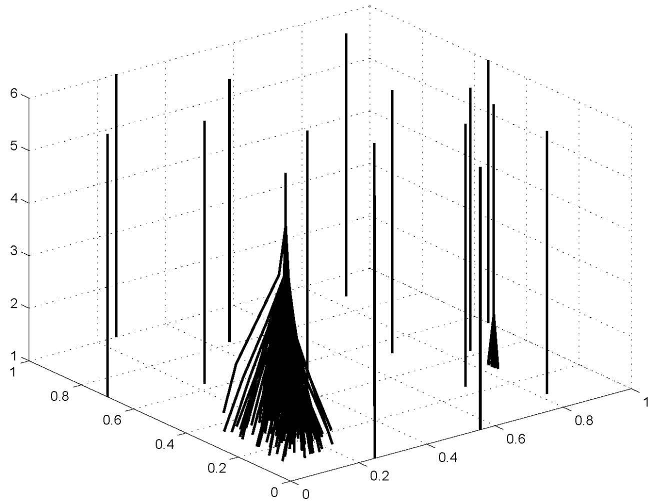


Fig. Dynamics of objects according to cluster model.



Visualization of objects' movement

How can we define a measure to flag the different dynamical behaviors between regular objects and outliers towards synchronization?

Local Synchronization Factor

Local Synchronization Factor (LSF): represents the local degree of synchronization of an object during the process of synchronization.

$$LSF(x) = \frac{1}{T}\underbrace{\sum_{t=0}^{T}}_{\text{Over time}}\left(\underbrace{\frac{1}{\left|N_{\varepsilon}(x(t))\right|}\sum_{y(t)\in N_{\varepsilon}(x(t))}\cos\left(\left\|y(t)-x(t)\right\|\right)}_{\text{Local degree of the synchronization}}\right)$$

The easier an object synchronizes with other objects, the higher of its LSF value.

LSF (cont'd)

Properties:

1. **Intuitive**: The LSF value indicates the degree of synchronization of each object. Outliers are objects which are "out of synchronization".

2. **Distinguishable**: The LSF value of regular points are close to 1 while outlier objects are nearly 0.

3. **Tight**: The range of LSF is restricted to [0 1).

4. **Interpretation**: It can be easily interpreted as the probability of each object of being an outlier, e.g. *Probability(x) = 1 - LSF(x)*.

Outlier Flagging

Outliers Flagging: K-Means(LSF, 2)

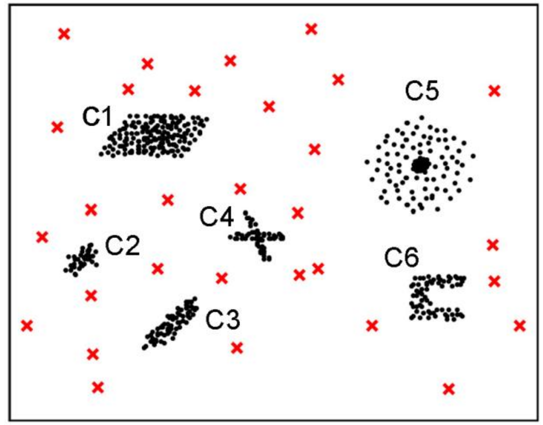
Since all outliers exhibit usually a low value in comparison to the regular objects, selecting a suitable threshold for flagging outliers could be very easy.

However, for automatically flagging, the K-Means algorithm are applied on the LSF values to split the data into two clusters: outliers and regular objects.

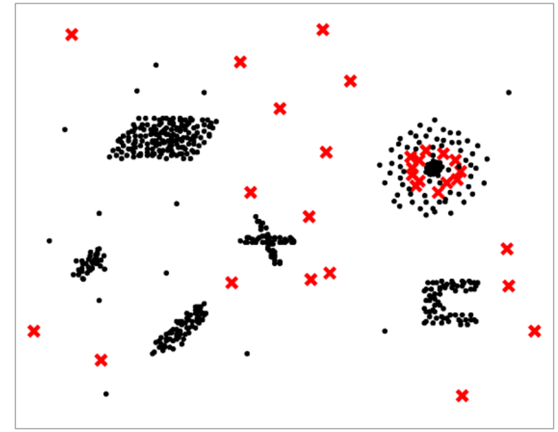
Evaluation

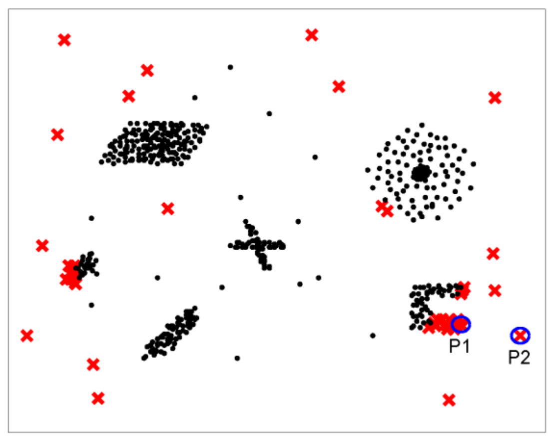
◆ Synthetic Data

- Clusters with different Shapes
- Multi-density
- Complex data
- No data distribution assumption

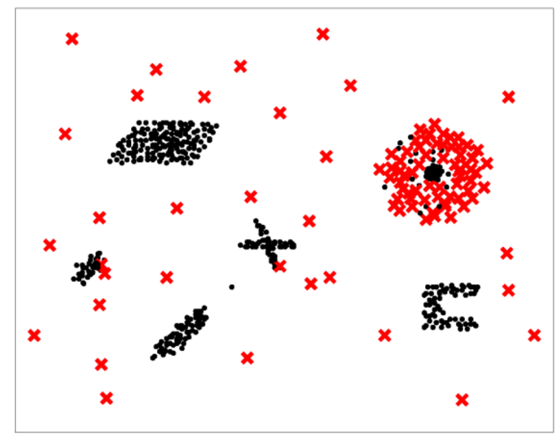


(a) SOD

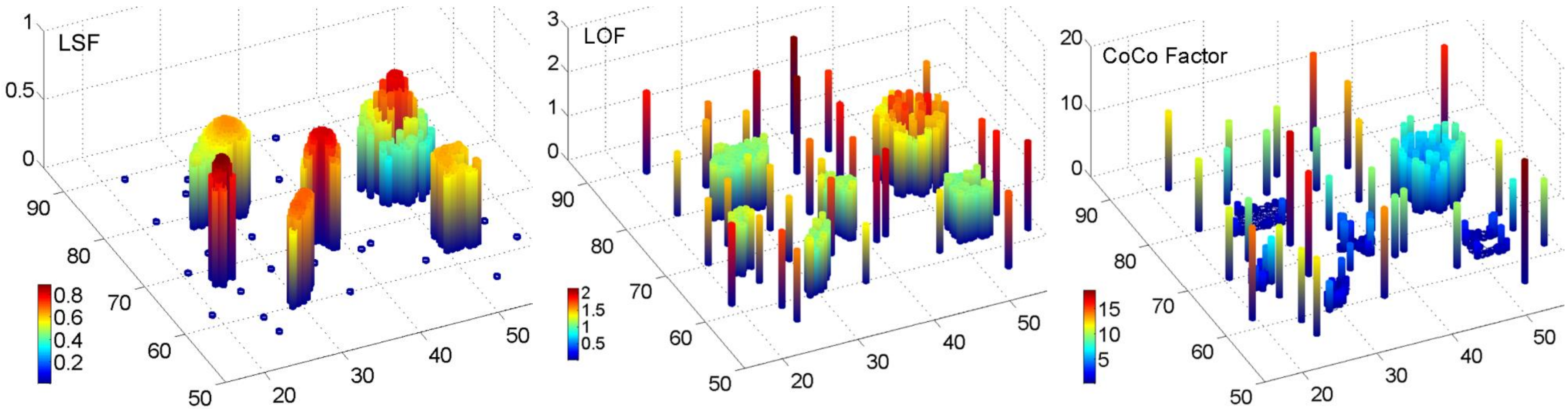

(b) LOF

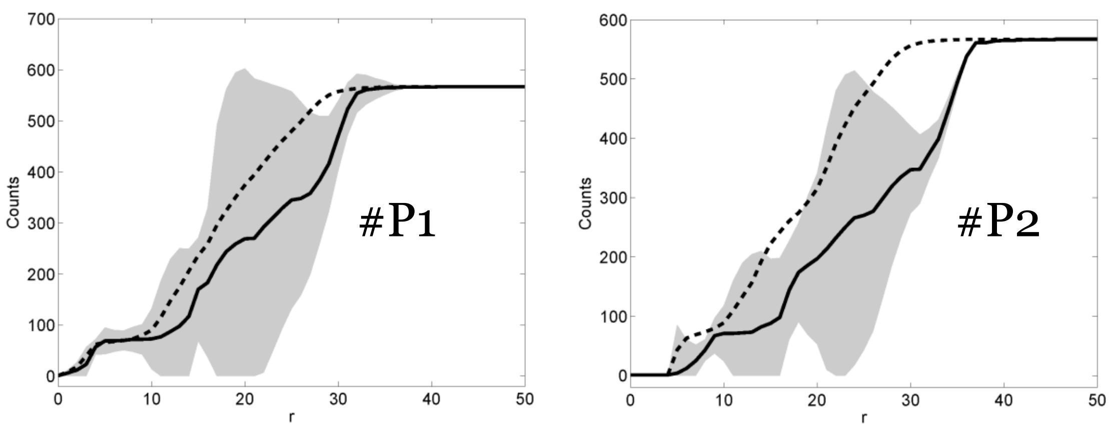


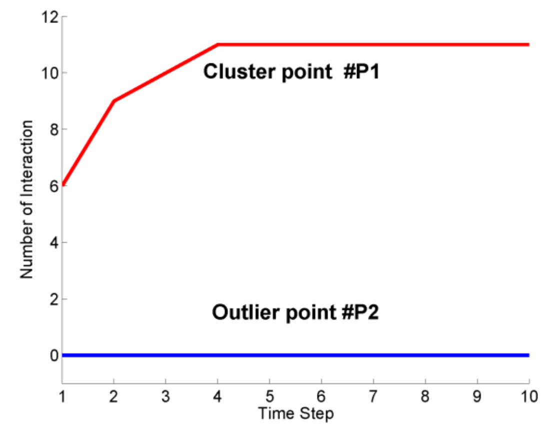
(c) LOCI


(d) CoCo

Evaluation (cont’d)







(a) LOCI Plot

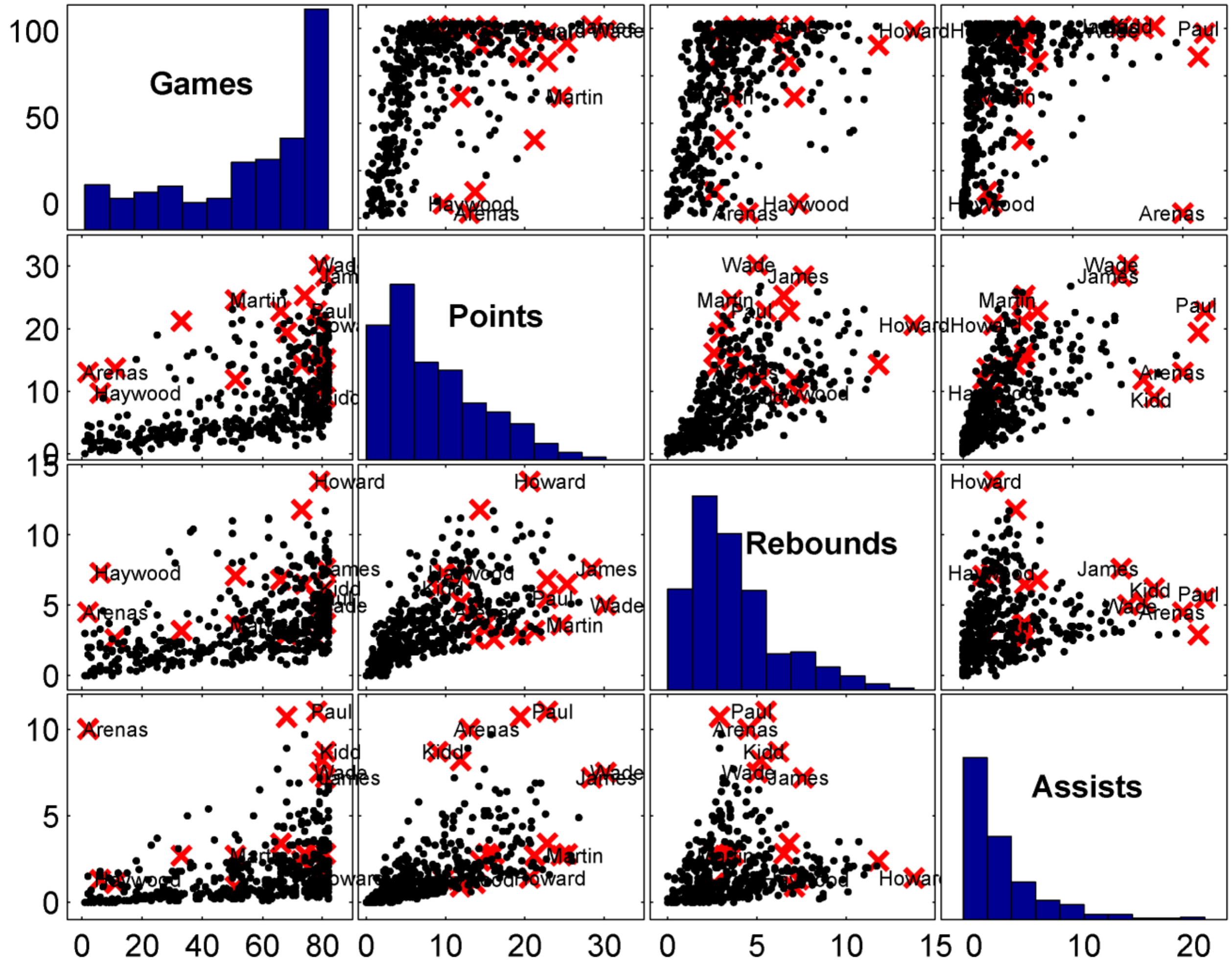
(b) Interaction Plot

Evaluation (cont'd)

◆ Real Data

NBA Performance Statistics

Season *2008/09*



ORSC: Subspace Clustering

Curse of Dimensionality

- Usually, no clusters in the full dimensional space of the data.
- Clusters are often hidden in subspaces of the data.

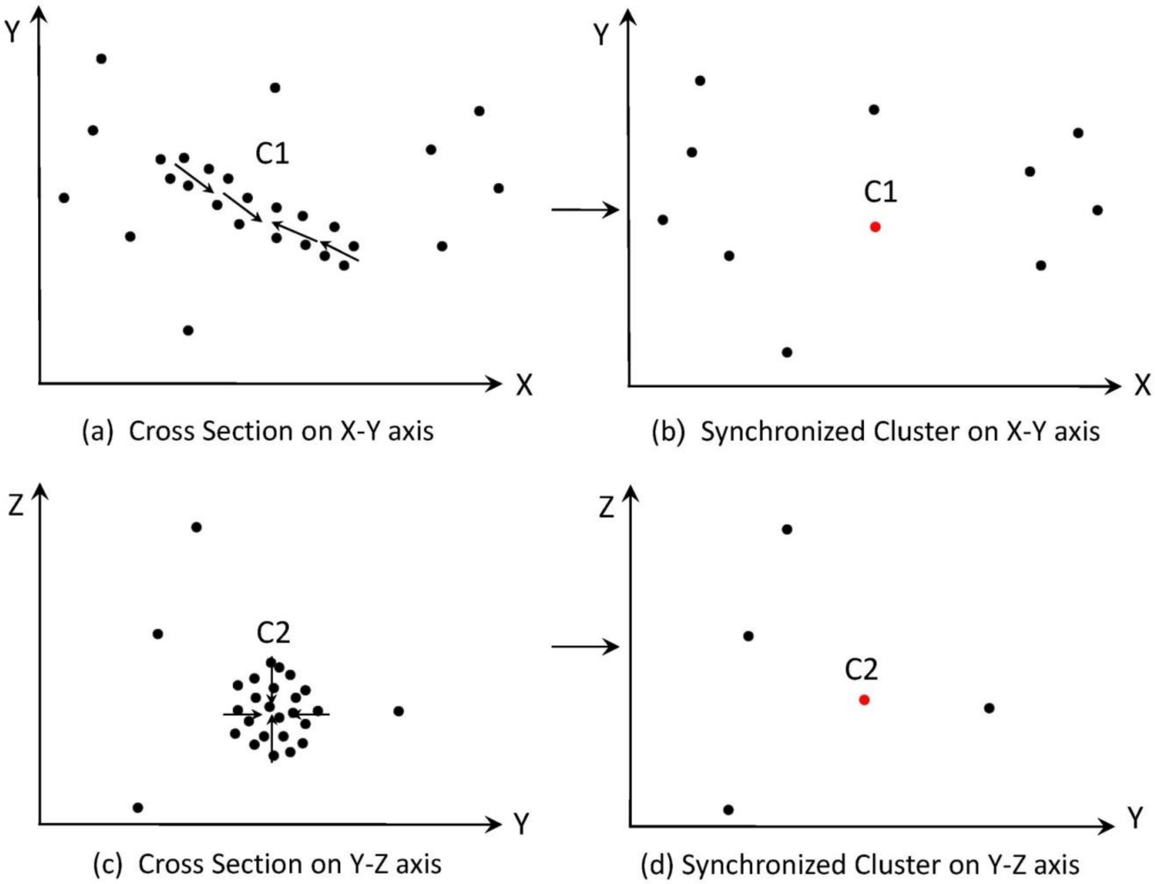
Local Feature Relevance

Different subsets of features are relevant for different clusters.

Subspace Clustering

ORSC (**Arbitrarily Oriented Synchronized Clusters**)
a novel effective and efficient method to subspace clustering inspired by synchronization.

Intuition



Arbitrarily Oriented Synchronized Clusters

Arrows indicate the main directions of movements of objects during the process of synchronization.
The red point illustrate the final states of cluster objects, which are formed as synchronized clusters in subspaces.

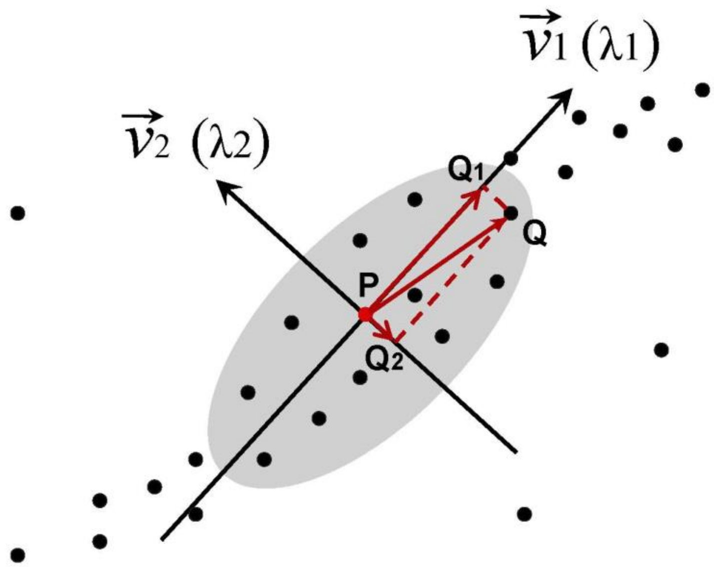
Interaction Model

For subspace clustering, Kuramoto model should be reconsidered in a different way.

1. Local Interaction Fashion. In order to exploit the hidden clusters or patterns in arbitrarily oriented subspaces, the local structure of data should be investigated.

2. Weighted Interaction. In high dimensional space, the correlations in the dimensions are often specific to data locality, which means some objects are correlated with respect to a given set of dimensions and others are correlated with respect to different dimensions. Thus, the coupling strength of objects' interactions in relevant or irrelevant dimensions should be considered with different weights.

Interaction Model (cont'd)



E.g. $WI(Q-P) = \lambda_1 \cdot \sin(\overrightarrow{Q_1P}) + \lambda_2 \cdot \sin(\overrightarrow{Q_2P})$

① ε-Neighborhood with Mahalanobis distance

$$N_\varepsilon^m(x) = \left\{ y \in D \mid \sqrt{(y-x)\cdot\Sigma_x^{-1}(y-x)^T} \le \varepsilon \right\}$$

② PCA is used to decompose the covariance matrix Σ of objects $N_\varepsilon^m(x)$

$$\Sigma = VEV^T$$

③ Weighted Interaction

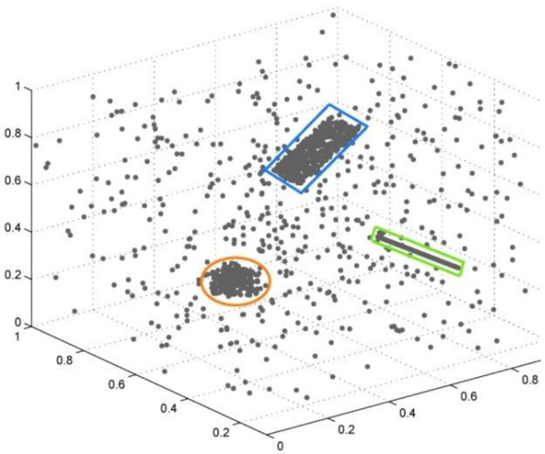
$$WI(y-x) = \sum_{k=1}^{d} \lambda_k \cdot \sin(proj(\Delta(y,x), \vec{v}_k))$$

where $proj(\Delta(y,x), \vec{v}_k) = (\Delta(y,x) \otimes \vec{v}_k)\cdot \vec{v}_k$

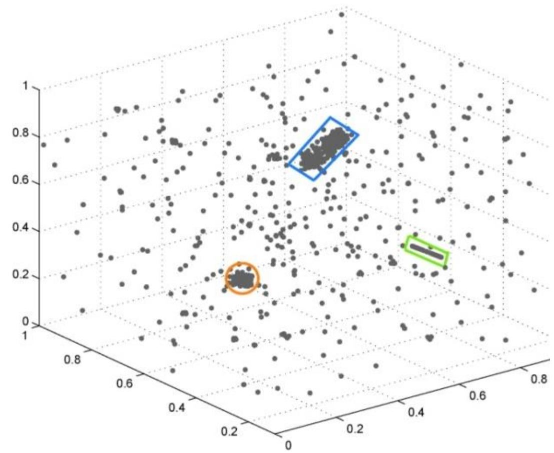
Interaction Model

$$x_i(t+1) = x_i(t) + \frac{1}{|N_\varepsilon^m(x(t))|} \sum_{y(t)\in N_\varepsilon^m(x(t))} \cdot \sum_{k=1}^{d} \lambda_k \cdot \sin(proj_{(i)}(\Delta(y,x), \vec{v}_k))$$

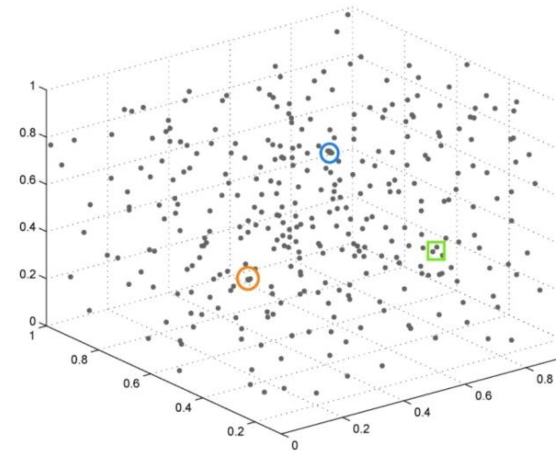
Synchronization Dynamics



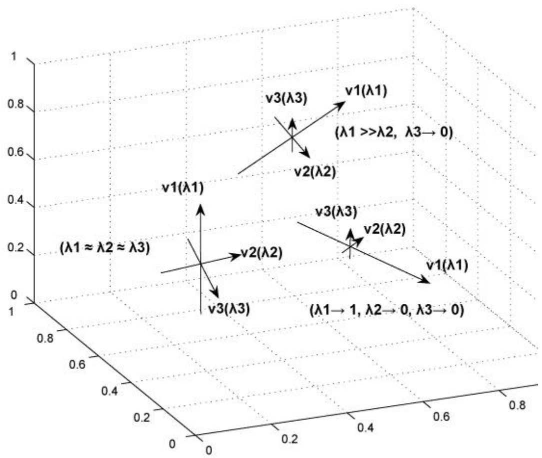
(a) t = 0



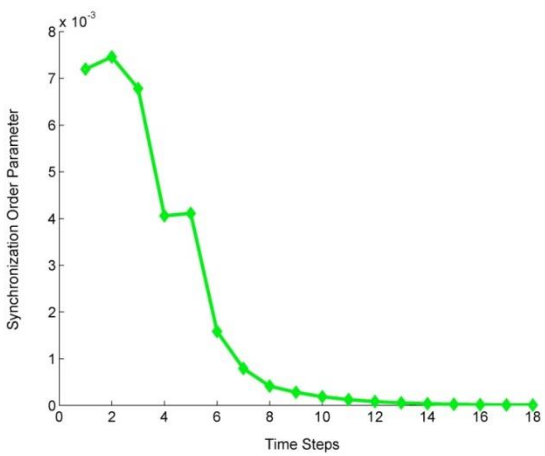
(b) t = 6



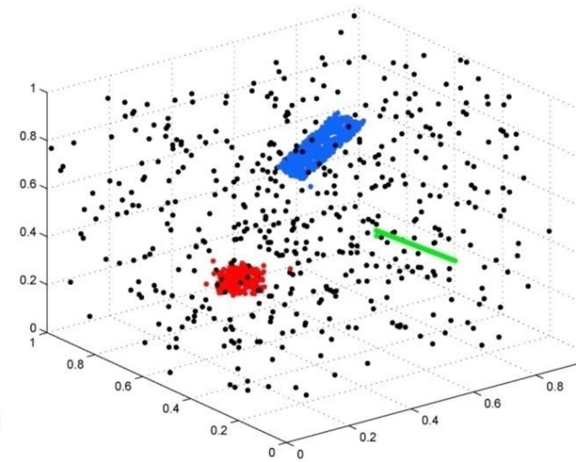
(c) t = 10



(d) Main Directions



(e) Order Parameter



(f) Synchronized Clusters

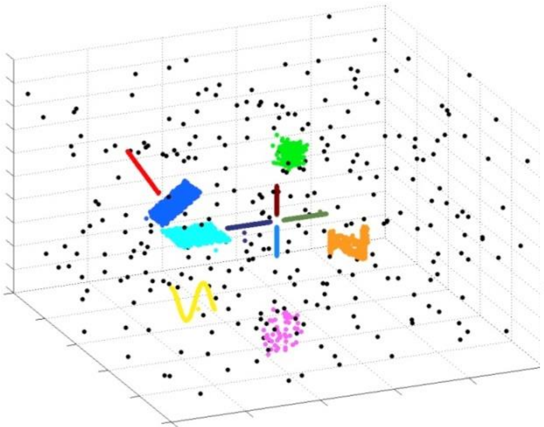
Synchronized Clusters Search

To find these synchronized clusters, the intuitive way is to find all **synchronized phases and corresponding objects**. The principle of our strategy is to consider the subspace search from **objects instead of dimensionality**.

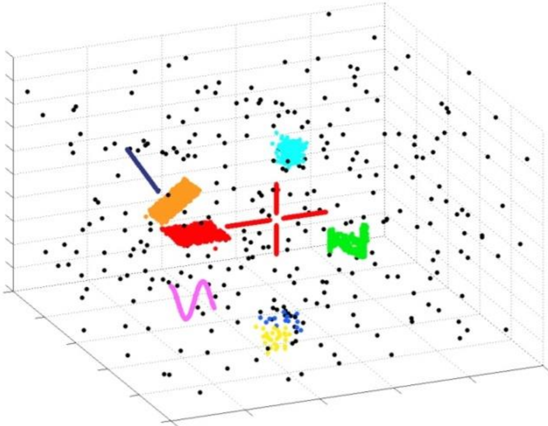
Obj.	d1	d2	d3	d4	Syn. Dim.	New Sub.	Cluster
1	0.1	0.2	0.1	0.3	1,2 4	(1,2) (4)	(1 2 3 4) (1,5)
2	0.1	0.2	0.2	0.2	1,2,4	(1,2,4)	(2, 3, 4)
3	0.1	0.2	0.7	0.2	1,2,3,4	(1,2,3,4)	(3, 4)
4	0.1	0.2	0.7	0.2	1,2,3,4	-	-
5	0.3	0.4	0.3	0.3	3	(3)	(5, 6)
6	0.9	0.5	0.3	0.1	3	-	-
7	0.7	0.6	0.4	0.5	Null	-	Noise

Evaluation

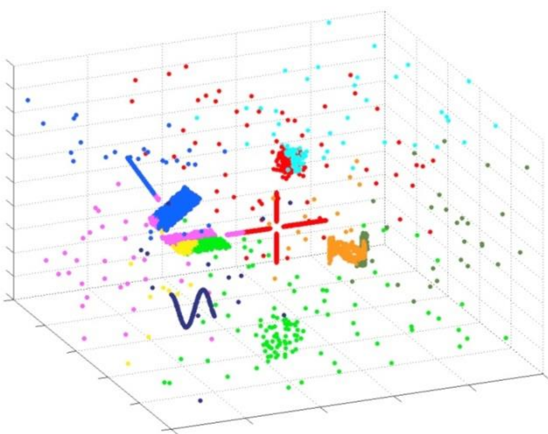
❖ 3-d Synthetic data



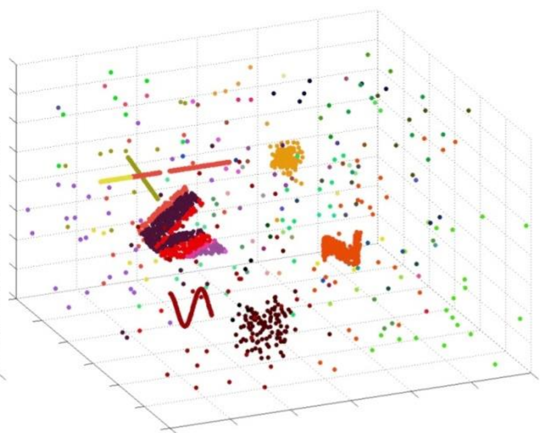
ORSC



4C

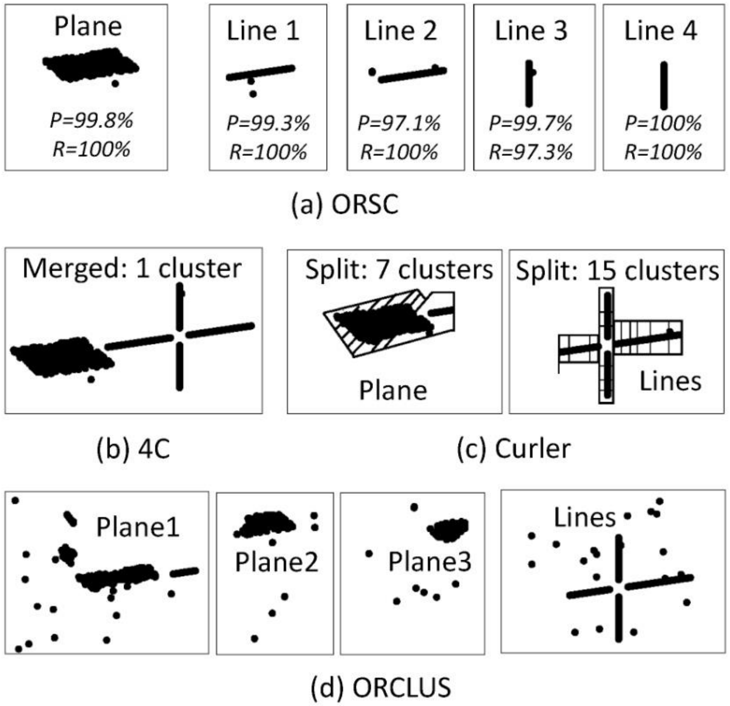


ORCLUS



Curler

Detailed view of clustering results with different algorithms on part of 3-d synthetic data.



Evaluation (cont'd)

❖ High-dimensional Synthetic data

Data	d	# C	#D	True clusters found by			
				ORCLUS	*4C*	*Curler*	*ORSC*
DS1	5	1	3	1 (Dim.: 3) P=100%; R=100%	1 (Dim. : 3) P=100%; R=100%	1 (Dim. : 3) P=99.0%; R=20.4%	1 (Dim. : 3) P=100%; R=97.0%
DS2	10	1	5	1 (Dim. : 5) P=27.8%; R=62.2%	1 (Dim. : 5) P=100%; R=94.2%	1 (Dim. : 5) P=48.4%; R=11.8%	1 (Dim. : 5) P=100%; R=97.4%
DS3	15	2	10,5	2 (Dim.: 10,10) P=16.9%; 74.4% R=14.0%;61.6%	1 (Dim. : 10) P=100%; R=99.6%	2 (Dim.: 10,5) P=14.5%; 12.0% R=19.3%;16.0%	2 (Dim.: 10,5) P=100%; 99.6% R=99.6%;100%
DS4	20	2	10,10	2 (Dim.: 10,10) P=17.9%; 100% R=13.2%,74.0%	2 (Dim. : 10,10) P=100%; 100% R=100%,100%	2 (Dim.: 10,10) P=12.5%; 100% R=12.5%;100%	2 (Dim.: 10,10) P=100%; 99.6% R=100%;99.6%
DS5	30	3	20,15,10	3 (Dim.: 20,15,10) P=21.7%,12.3%,13.2% R=100%,67.5%,72.5%	1 (Dim. : 20) P=100%; R=98.5%	3 (Dim.: 20,15,10) P=100%,99.5%,76.9% R=99.0%,100%,5%	3 (Dim.: 20,15,10) P=100%,98.5%,99.6% R=100%,99.5%,100%

Evaluation (cont'd)

❖ Real data sets — Ecoli data & Wine data

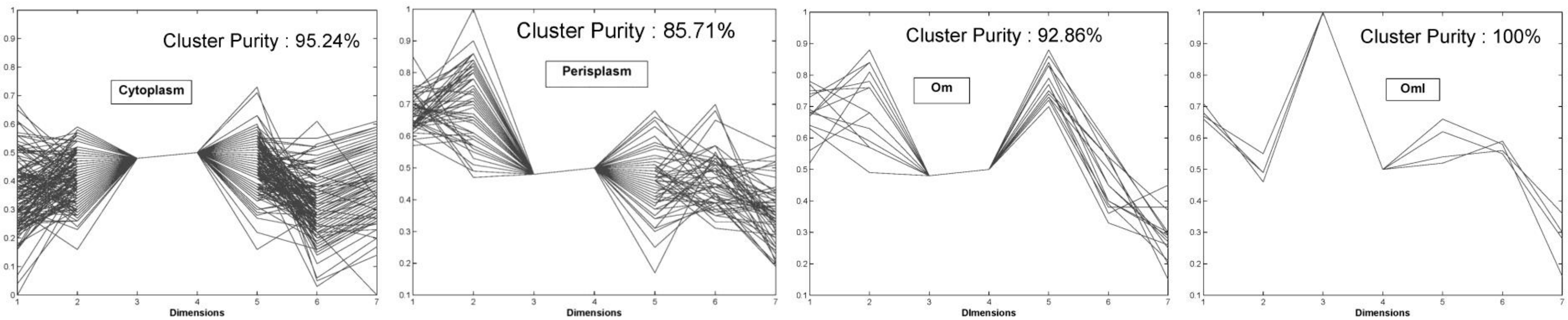


Fig. ORSC on the Ecoli data set.

Tab. Clusters found by ORSC on wine data

C_ID	T1	T2	T3	Pre.	Rec.
1	58	3	0	95.2%	98.3%
2	0	53	0	100%	73.6%
3	0	5	48	90.6	100%
4	0	4	0	100%	5.6%

Tab. Validation measures on two data.

Method	Ecoli data		Wine data	
	NMI	AMI	NMI	AMI
ORSC	**0.682**	**0.670**	**0.701**	**0.695**
4C	0.338	0.328	0.474	0.469
ORCLUS	0.452	0.430	0.191	0.182
Curler	0.060	0.049	0	0

Desirable properties of *ORSC*

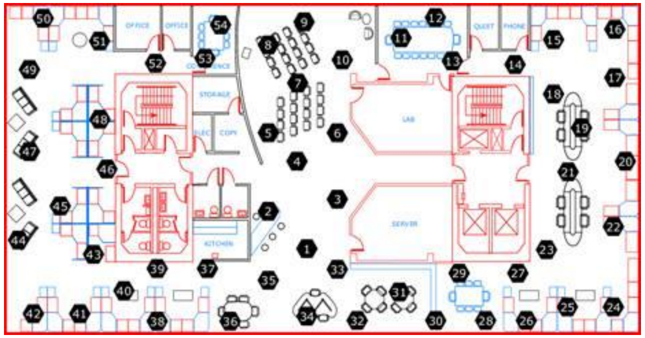
- Natural data structure exploring.
- Detection of arbitrarily shaped correlation clusters.
- Outlier detection.
- Efficient subspace searching

SyncStream: Prototype-based learning on concept-drifting data streams

Motivation

Surveillance

Smart Phone



Sensors

Data Stream: **(a) Infinite Length (b) Evolving Nature**

Challenges:

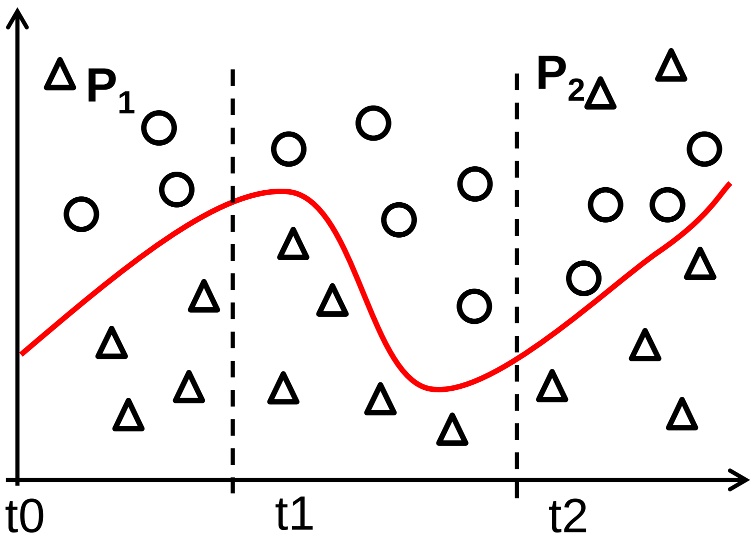
- Single Pass Handling
- Low Time Complexity
- Memory Limitation
- Concept Drift

Motivation

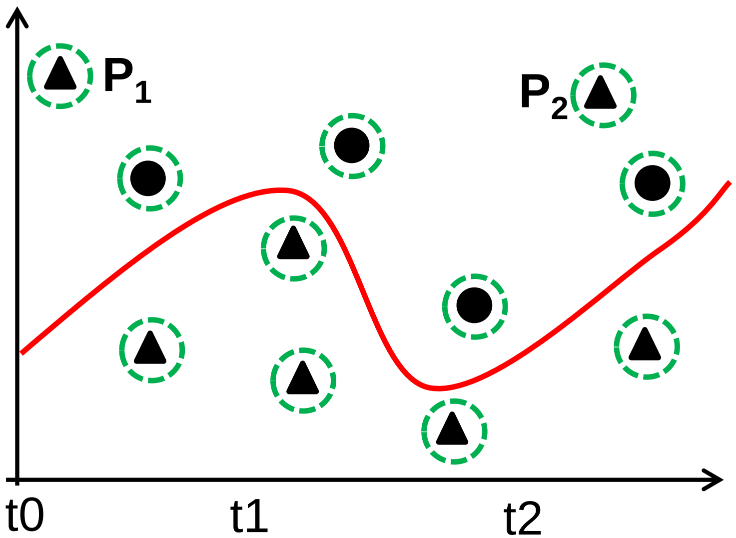
Single model learning: Learn and update a classification model by training on a fixed or adaptive window of recent incoming examples, suffers in the presence of **concept drift**.

Ensemble learning: Train a number of base classifiers to capture evolving concepts.

1. **Black-box Fashion**
2. **Data Selection for Training**



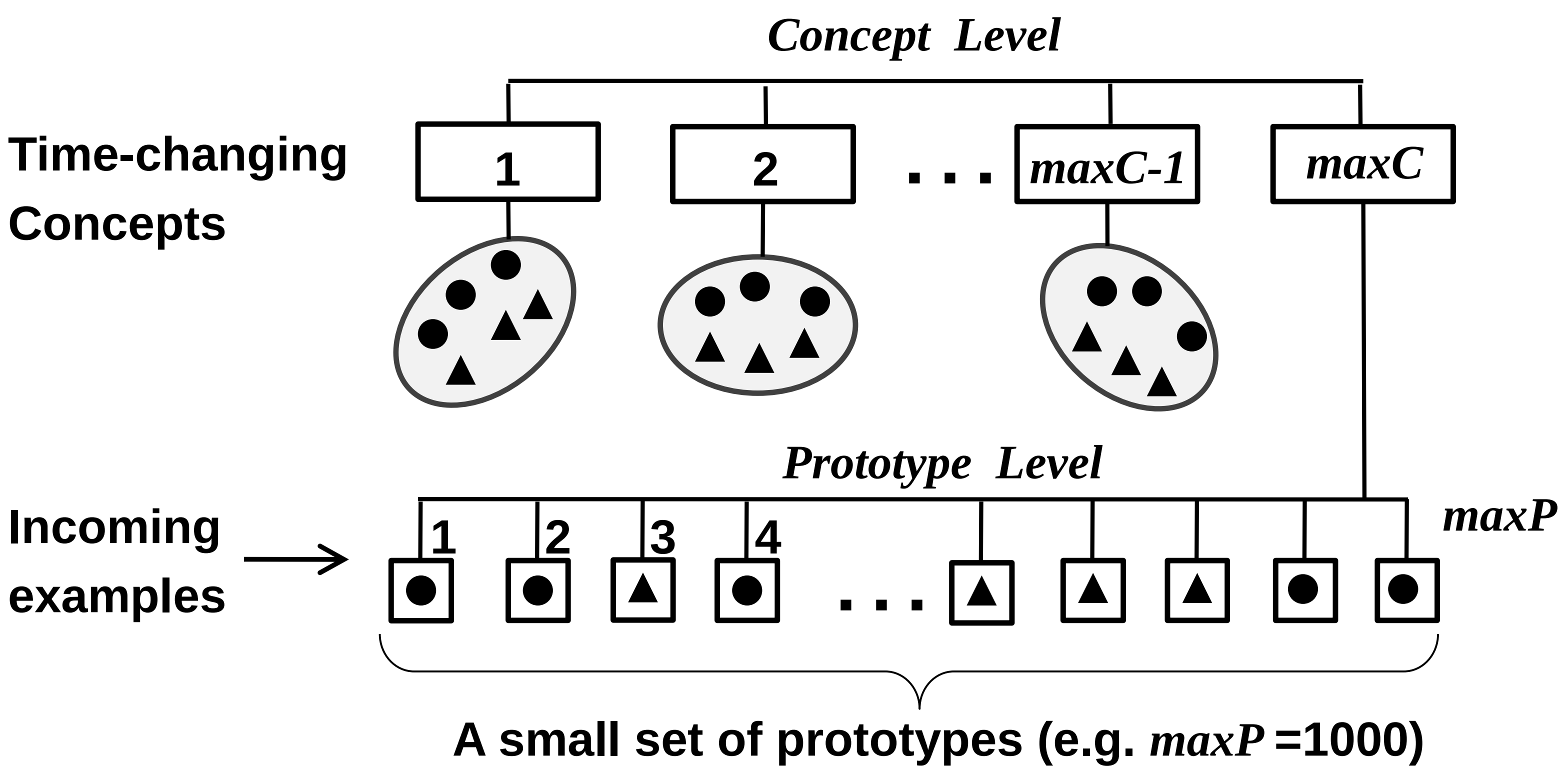
Basic Idea



Prototype-based Learning: An intuitive way is to dynamically select the short-term and/or long-term representative examples to capture the trend of time-changing concepts.

- Online Data Maintenance: **P-Tree**
- Prototypes Selection: **Error-driven representativeness learning** and **synchronization-inspired constrained clustering**
- Sudden Concept Drift: **PCA** and **Statistics**
- **Lazy Learning:** KNN

Online Data Maintenance: P-TREE



P-Tree is additionally updated:

- Maximum boundary (Synchronization-based data representation)

- Sudden concept drift (Rebuild the Prototype Level)

Error-driven Representativeness Learning

How to dynamically select the short-term and/or long-term representative examples?

Basic idea: Leverage the prediction performance of test examples to infer the representativeness of examples by lazy learning: nearest neighbor classifier.

$$\mathbf{Rep}(y) = \mathbf{Rep}(y) + \mathbf{Sign}(x_{pl}, x_l)$$

where $Sign(x, y)$ is the sign function, and 1 if x equals y, -1 otherwise.

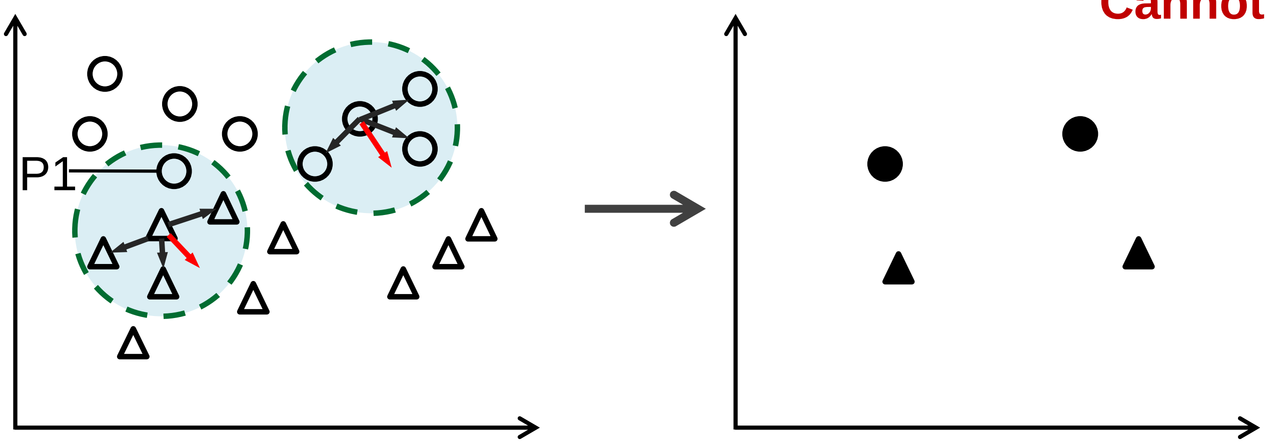
- ◆ High representativeness —— Keep
- ◆ Low representativeness —— Delete
- ◆ Unchanged representativeness? —— Summarization

Data Summarization by synchronization

Summarization: Constrained Clustering by Synchronization

$$x_i(t+\Delta t) = x_i(t) + \frac{1}{|N_\varepsilon(x(t))|} \cdot \sum_{y \in N_\varepsilon(x(t)),\ eq(lx\ \ ,ly)} \sin(y_i(t) - x_i(t))$$

Cannot Link

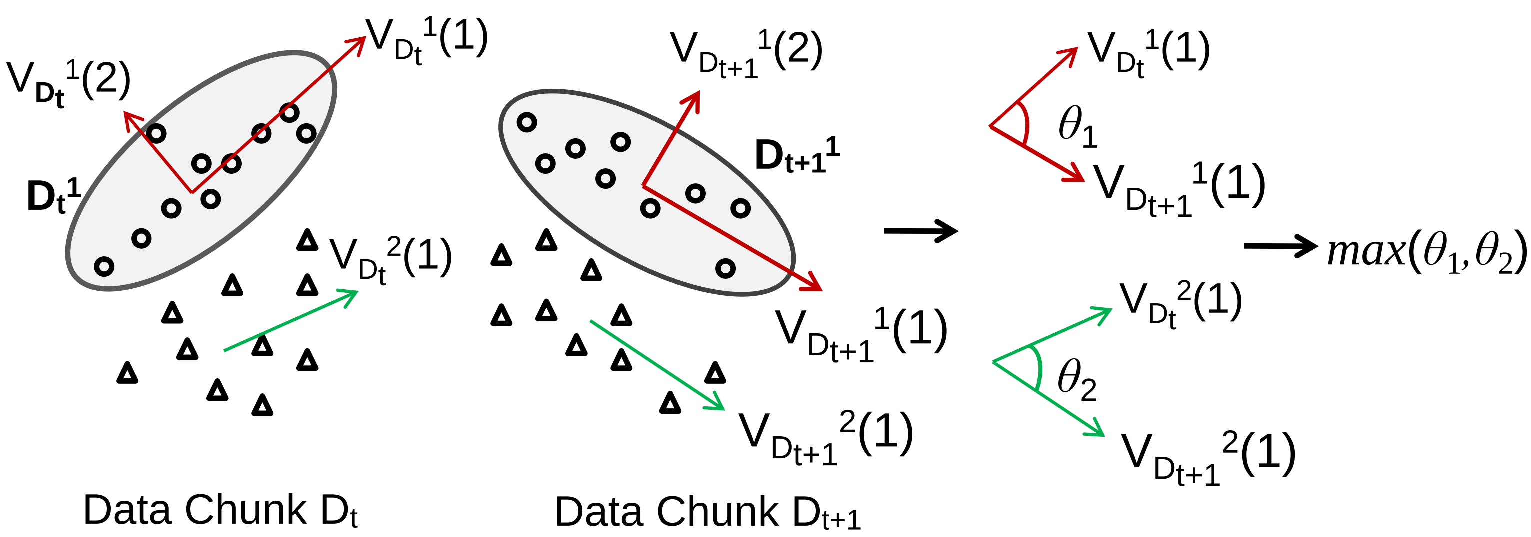


(a) Constrained clustering by synchronization

(b) Prototype-based data representation

Abrupt Concept Drift Detection

❖ **Principle Component Analysis (PCA):** Analyze the change of each class data distribution by principle component of two sets of examples.



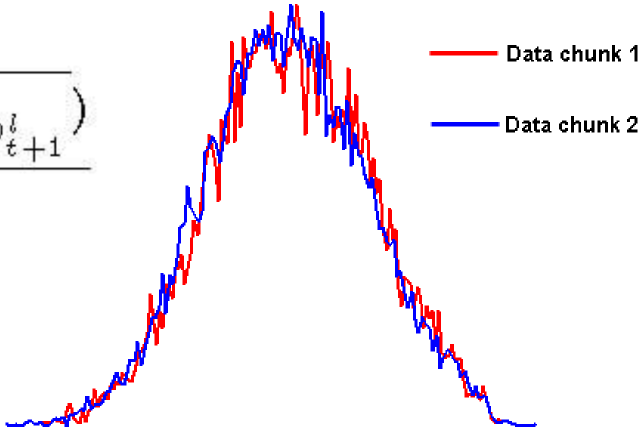
PCA-based concept drift analysis

Abrupt Concept Drift Detection

- **Statistical Analysis**: Compute a suitable statistic, which is sensitive to data class distribution changes between the two sets of examples.

$\overline{R}^{j}_{D^l_t}$ and $\overline{R}^{j}_{D^l_{t+1}}$ are the j^{th} dimensional mean ranks of examples from D^l_t and D^l_{t+1}

$$W^l_{BF} = \sqrt{\frac{|D^l_t||D^l_{t+1}|}{|D^l_t| + |D^l_{t+1}|}} \cdot \frac{\sum_{j=1}^{d}(\overline{R^j_{D^l_t}} - \overline{R^j_{D^l_{t+1}}})}{\sigma_{BF}}$$



Frequency distribution

Experiments & Results

Experiment Setup

Data sets

- Synthetic data
- Real-world data: **Spam**, **Electricity**, **Covtype**, and **Sensor**

Comparison methods

- Adaptive Hoeffding Tree - IBLStreams
- Weighted Ensemble - OzaBagAdwin
- PASC

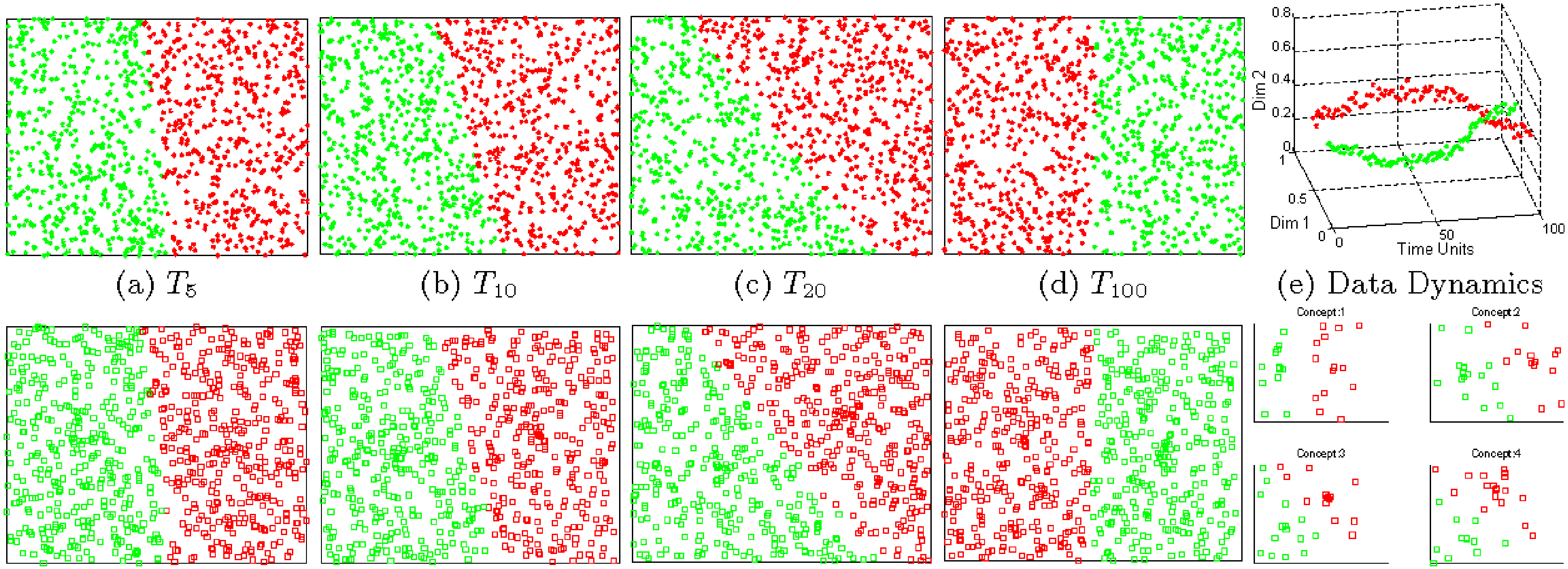
Evaluation Metrics

- Prediction performance
- Efficiency
- Sensitivity

1. Proof of Concept

– Concept Modeling

(1) Synthetic data stream with gradual concept drift

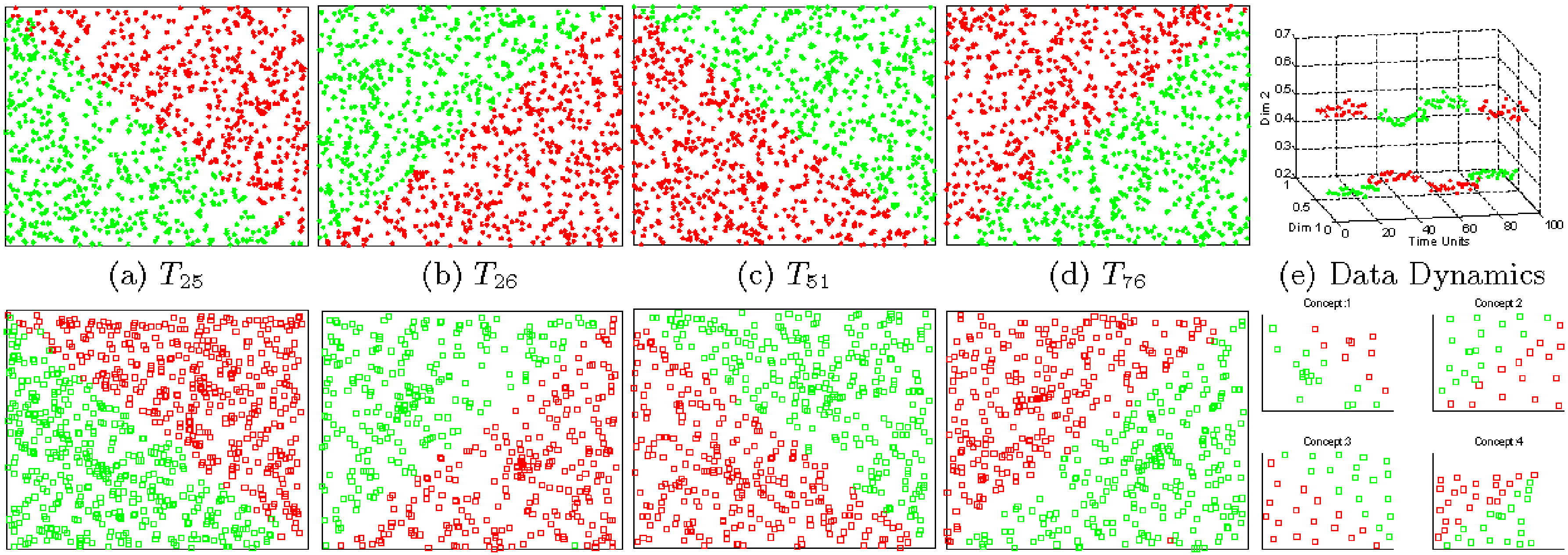


(a) T_5 (b) T_{10} (c) T_{20} (d) T_{100} (e) Data Dynamics

(f) P-Tree (T_5) (g) P-Tree (T_{10}) (h) P-Tree (T_{20}) (i) P-Tree (T_{100}) (j) P-Tree (Concepts)

1. Proof of Concept

(2) Synthetic data stream with sudden concept drift

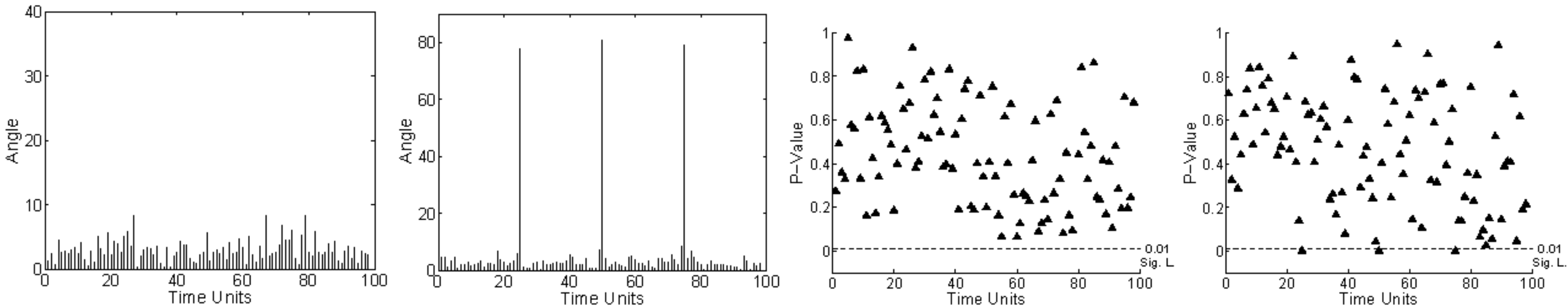


(a) T_{25} (b) T_{26} (c) T_{51} (d) T_{76} (e) Data Dynamics

(f) P-Tree (T_{25}) (g) P-Tree (T_{26}) (h) P-Tree (T_{51}) (i) P-Tree (T_{76}) (j) P-Tree (Concepts)

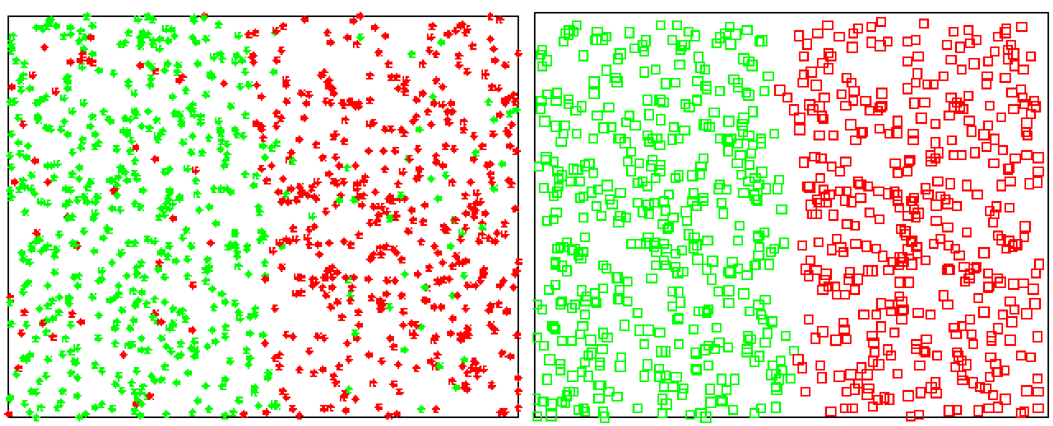
1. Proof of Concept

– Sudden concept drift detection



(a) PCA (G) (b) PCA (S) (c) Statistical Test (G) (d) Statistical Test (S)

– Prototype-based Data Representation



(a) Data stream with noise (b) Prototypes in P-Tree

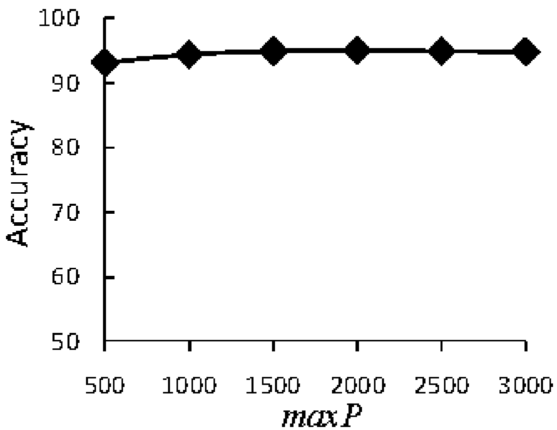
2. Prediction Performance Analysis

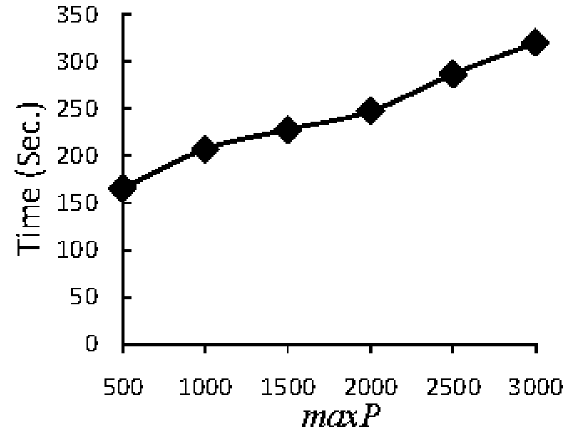
Table 1: Performance of different data stream classification algorithms on real-world data sets.

Data	#Obj	#Dim	#Class	Methods	Acc.	Prec.	Rec.	F_1	Time (ms)
Spam	9324	500	2	SyncStream (PCA)	**0.9719**	**0.9590**	**0.9665**	**0.9627**	60410
				SyncStream (Stat.)	**0.9719**	**0.9590**	**0.9665**	**0.9627**	29780
				IBLStream	0.9370	0.9070	0.372	0.9218	702632
				HoeffdingAdaTree	0.9071	0.8717	0.8935	0.8824	2252
				WeightedEnsemble	0.8629	0.8139	0.8176	0.8158	13000
				OzaBagAdwin	0.9108	0.8765	0.8973	0.8868	10848
				PASC	0.8931	0.9178	0.9415	0.9295	**2142**
Electricity	45,312	8	2	SyncStream (PCA)	0.8457	0.8423	0.8420	0.8421	3118
				SyncStream (Stat.)	**0.8459**	**0.8425**	0.8419	0.8422	3280
				IBLStream	0.7688	0.7648	0.7584	0.7616	7512
				HoeffdingAdaTree	0.8398	0.8409	0.8296	0.8352	**750**
				WeightedEnsemble	0.7092	0.7024	0.7022	0.7023	3920
				OzaBagAdwin	0.8397	0.8399	0.8302	0.8350	3810
				PASC	0.8170	0.8316	**0.8552**	**0.8432**	1327
Covtype	581,012	54	7	SyncStream (PCA)	**0.9438**	**0.8915**	**0.8980**	**0.8947**	207176
				SyncStream (Stat.)	**0.9438**	**0.8915**	**0.8980**	**0.8947**	226331
				IBLStream	0.9197	0.8620	0.8573	0.8597	3005412
				HoeffdingAdaTree	0.8087	0.7085	0.7173	0.7129	**31692**
				WeightedEnsemble	0.8033	0.7476	0.6690	0.7061	365582
				OzaBagAdwin	0.8383	0.7848	0.7722	0.7784	176000
				PASC	0.7972	0.8291	0.8348	0.8319	125387
Sensor	2,219,803	5	54	SyncStream (PCA)	0.8453	0.8508	0.8460	0.8484	244110
				SyncStream (Stat.)	0.8453	0.8508	0.8460	0.8484	246492
				IBLStream	0.1173	0.1805	0.1397	0.1575	345930
				HoeffdingAdaTree	0.6121	0.6269	0.6282	0.6276	**166600**
				WeightedEnsemble	0.6752	0.7918	0.6805	0.7319	2105133
				OzaBagAdwin	**0.8563**	**0.8660**	**0.8639**	**0.8649**	1343065
				PASC	0.7968	0.8420	0.8150	0.8283	264161

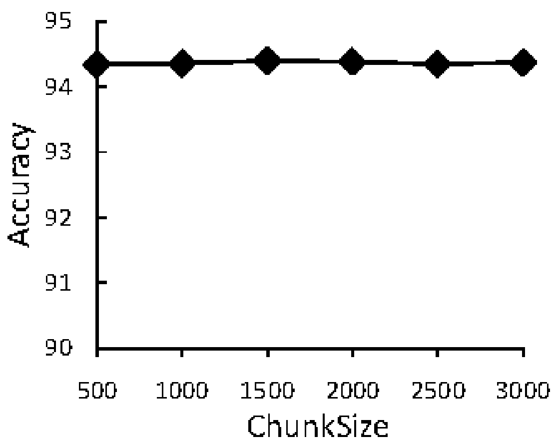
3. Sensitivity Analysis

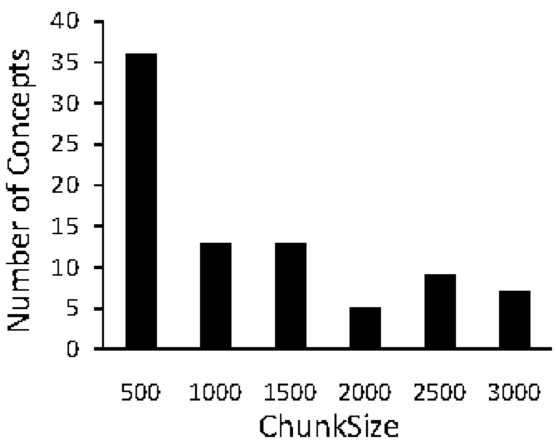
(1). Number of Prototypes





(2). Chunk Size



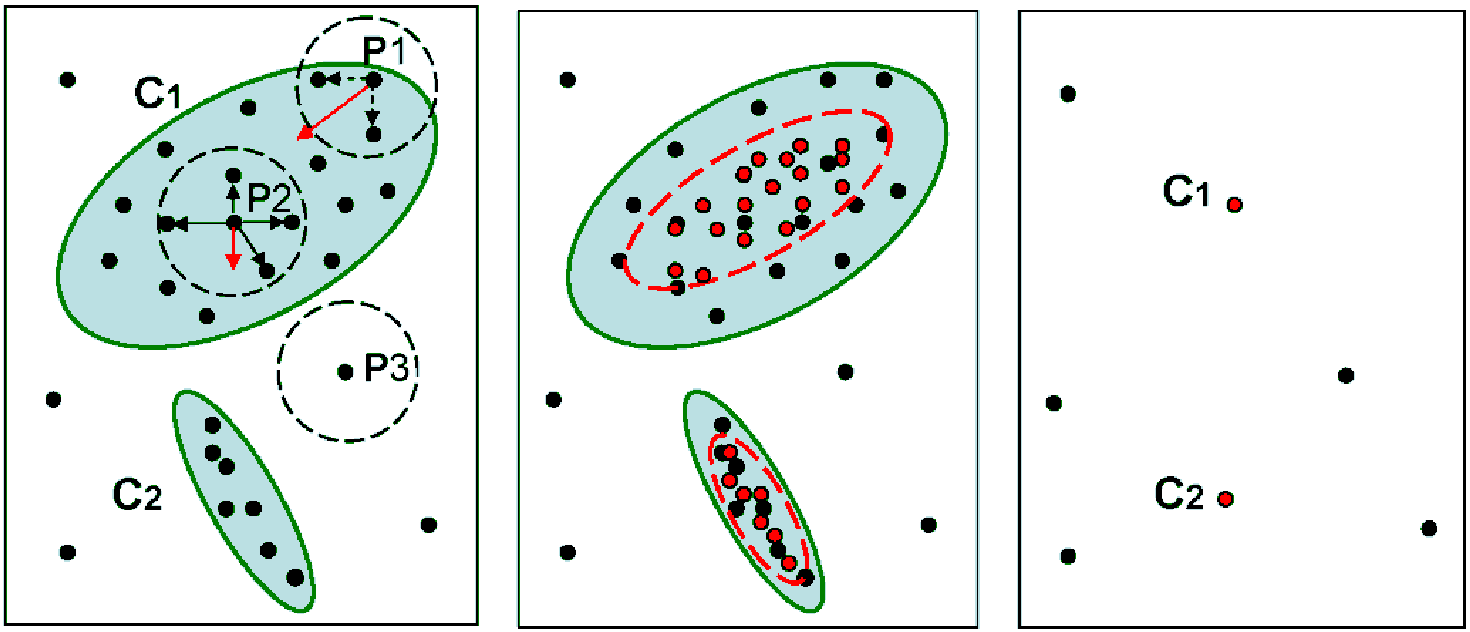


Summary

ATTACTIVE PROPERTIES:

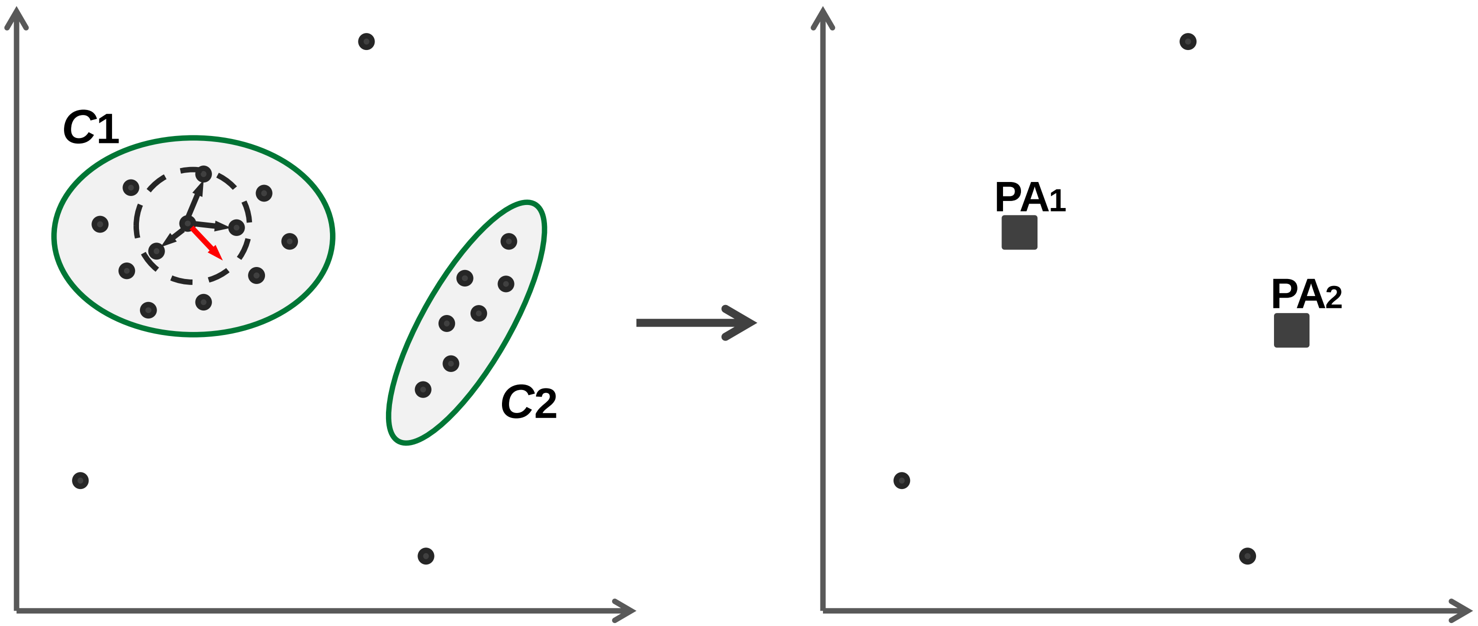
◆ **Dynamic Process**

Static vs Dynamic



Potential benefits: Simple and Intuitive, Identifying High-quality clusters driven by its local topology.

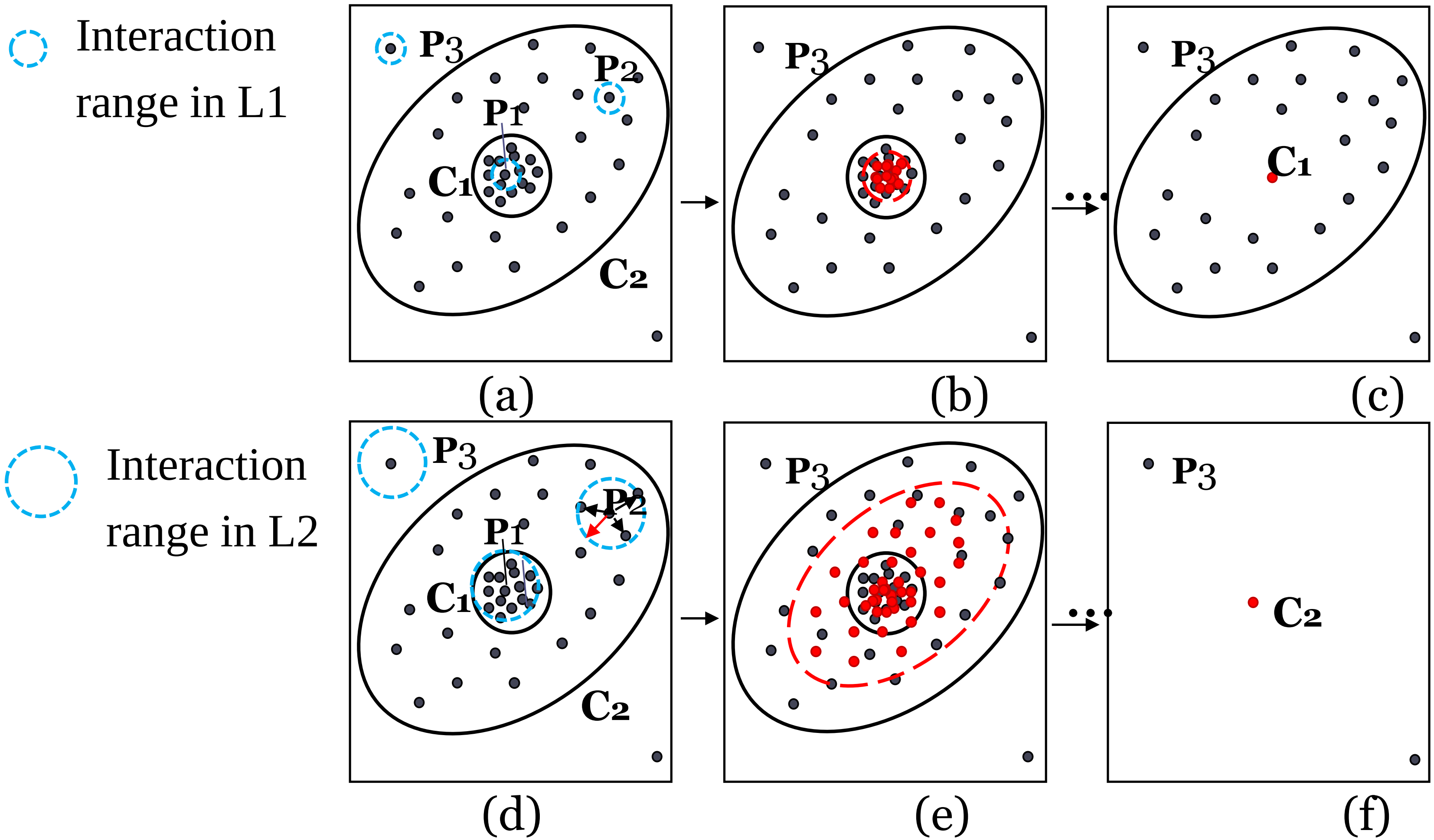
◆ Local Data Structure Preserving



(a) Synchronization-based Clustering (b) Point Attractor Representation

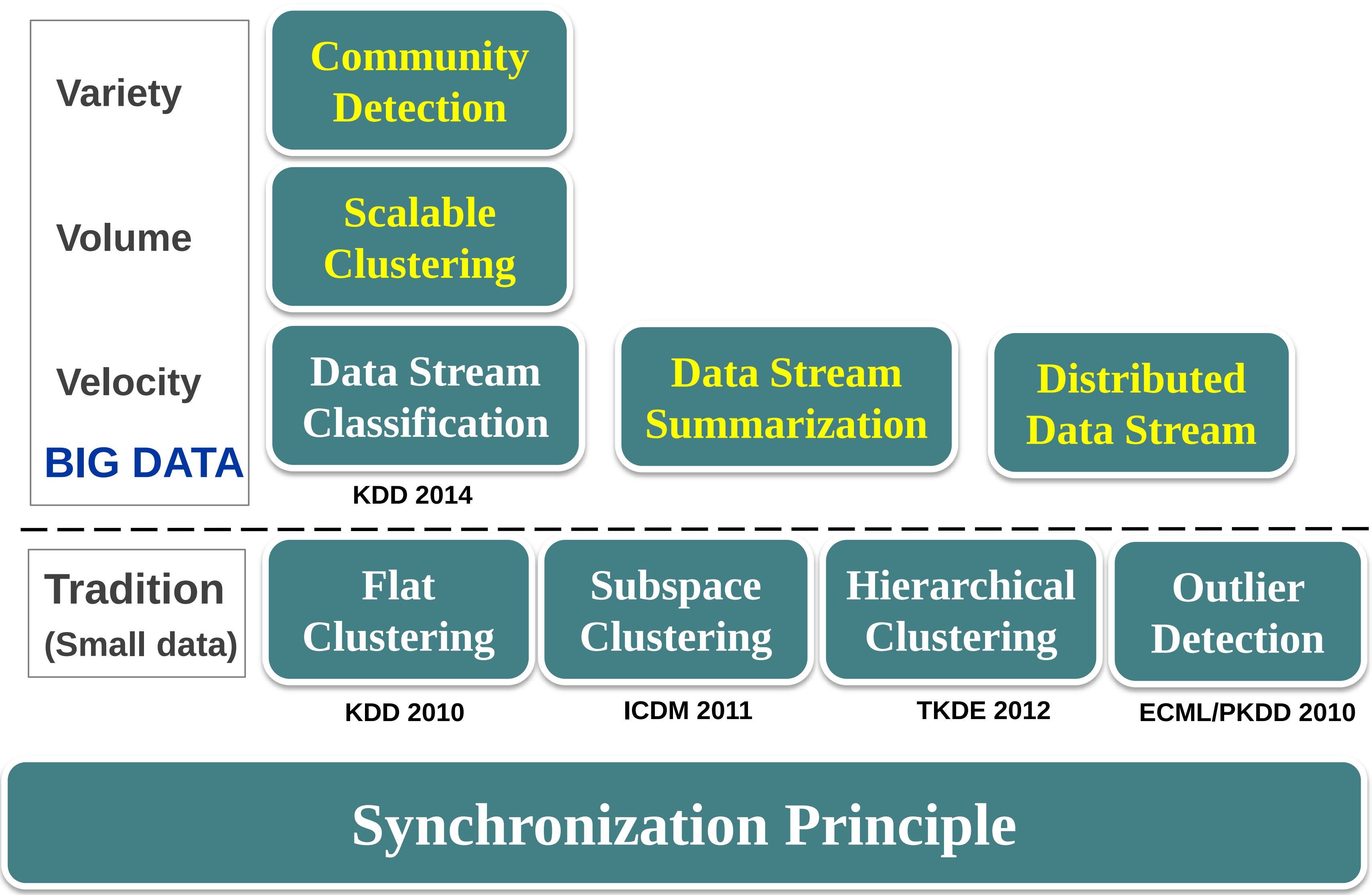
Potential benefits: summarization/visualization, scalable data mining

◆ Multi-Scale Data Representation



Potential benefits: big data handling, Multi-scale data analysis

Synchronization on Data Mining



Reference

- Boehm, C., Plant, C., Shao, J.* and Yang, Q. : Clustering by synchronization, Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2010), 583-592, 2010.

- Shao, J., Boehm, C., Yang, Q. and Plant, C. : Synchronization Based Outlier Detection, Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD 2010), 245-260, 2010.

- Shao, J., Yang, Q., Boehm, C. and Plant, C. : Detection of Arbitrarily Oriented Synchronized Clusters in High-dimensional Data, IEEE International Conference on Data Mining (ICDM), pp. 607-616, 2011.

- Shao, J., He, X., Boehm, C., Yang, Q. and Plant, C. : Synchronization-inspired Partitioning and Hierarchical Clustering, IEEE Transactions on Knowledge and Data Engineering, 25(4): 893-905. 2013.

- Shao, J., Ahmadi, Z. and Kramer, S.: Prototype-based Learning on Concept-drifting Data Streams, Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining , pp. 412-421. 2014.

A brief introduction to of our lab

Our lab, Intelligent Big Data Analysis and Mining Lab (IBDAML), is founded in Dec.2013 and led by Prof. Junming Shao. Currently we have 15 members in our lab, including graduates and undergraduates. We focus widely on data mining and machine learning, in both theoretical justification and real-world applications.

Our current research topics include:

- Clustering (scalable/subspace/hierarchical/parameter-free clustering)
- Data stream mining (Concept drift detection/clustering/classification)
- Brain network mining and applications (Mining on fMRI/DTI/EEG brain data)
- Multi-source heterogeneous data mining

For more information about out group member and research projects, please go to our home page http://staff.uestc.edu.cn/shaojunming/

Thanks for your attention！

Q & A