Hierarchical Data Stream Compression

Anne Tan

tanxiangyueer@foxmail.com

2015.7.23

Data Mining Lab, Web Sciences Center Institute of Computer Science and Technology, UESTC

Outline

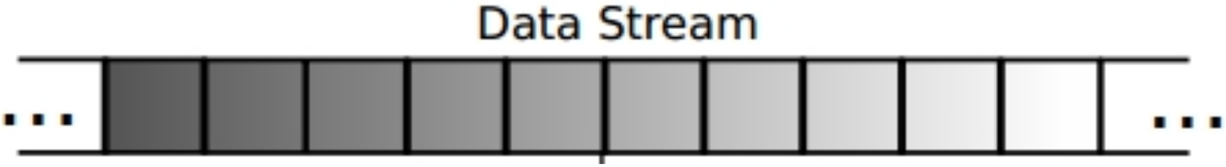
1. Introduction
2. Motivation
3. Data Stream Compression
   ① Synchronization Algorithm
   ② Exponential Power Distribution
   ③ Sync-EPD
   ④ Independent Component Analysis
4. Experiment
   ① Sync-EPD
   ② Sync-ICA-EPD
   ③ ICA-Sync-EPD
   ④ Hierarchical
5. Discussion

1. Introduction

1. Introduction

'you can never step in the same stream twice.'

——Heraclitus



Data stream:

temporally ordered , fast changing ,

massive , potentially infinite.

1. Introduction

A single pass & Limited memory & Limited time.

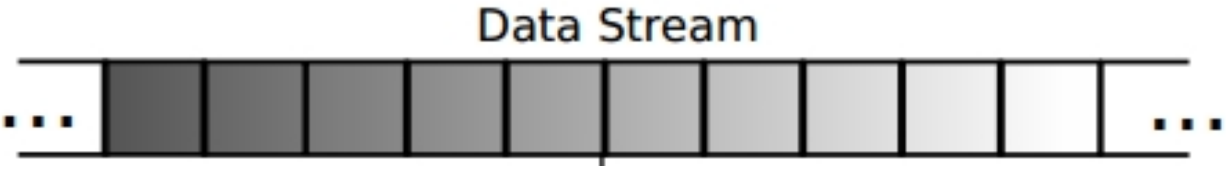
Compressed data stream

Some methods:

- Sampling,
- Sketching,
- Synopsis Data Structures,
- Aggregation,
- Sliding Window.

2. Motivation

2. Motivation



Proposing a new **hierarchical data stream compression**

Problem 1: Why we use ***hierarchical*** method**:**

Adapting to different compression requirements.

a) More interested in recent data;

b) Not interested in the details of old data;

c) a trade-off of the space and the accuracy.

So the compression ratio is different.

2. Motivation

Problem 2: How to compress streams of data?

Clustering

a) a useful technique for structuring and organizing vast amounts of continuous examples.

b) a good locality

2. Motivation

Problem 3: How to **Clustering**?

Synchronization-based clustering .

a) yields the high-quality clusters with arbitrarily shapes,

b) are robust to noise

c) allows a natural hierarchical data analysis.

And,to reflect the importance of recent data,***sliding window*** is used.

sliding window: a fixed equal lifespan;and FIFO.

2. Motivation

But just store the cluster,causing much compression loss.

Problem 4: How to do?

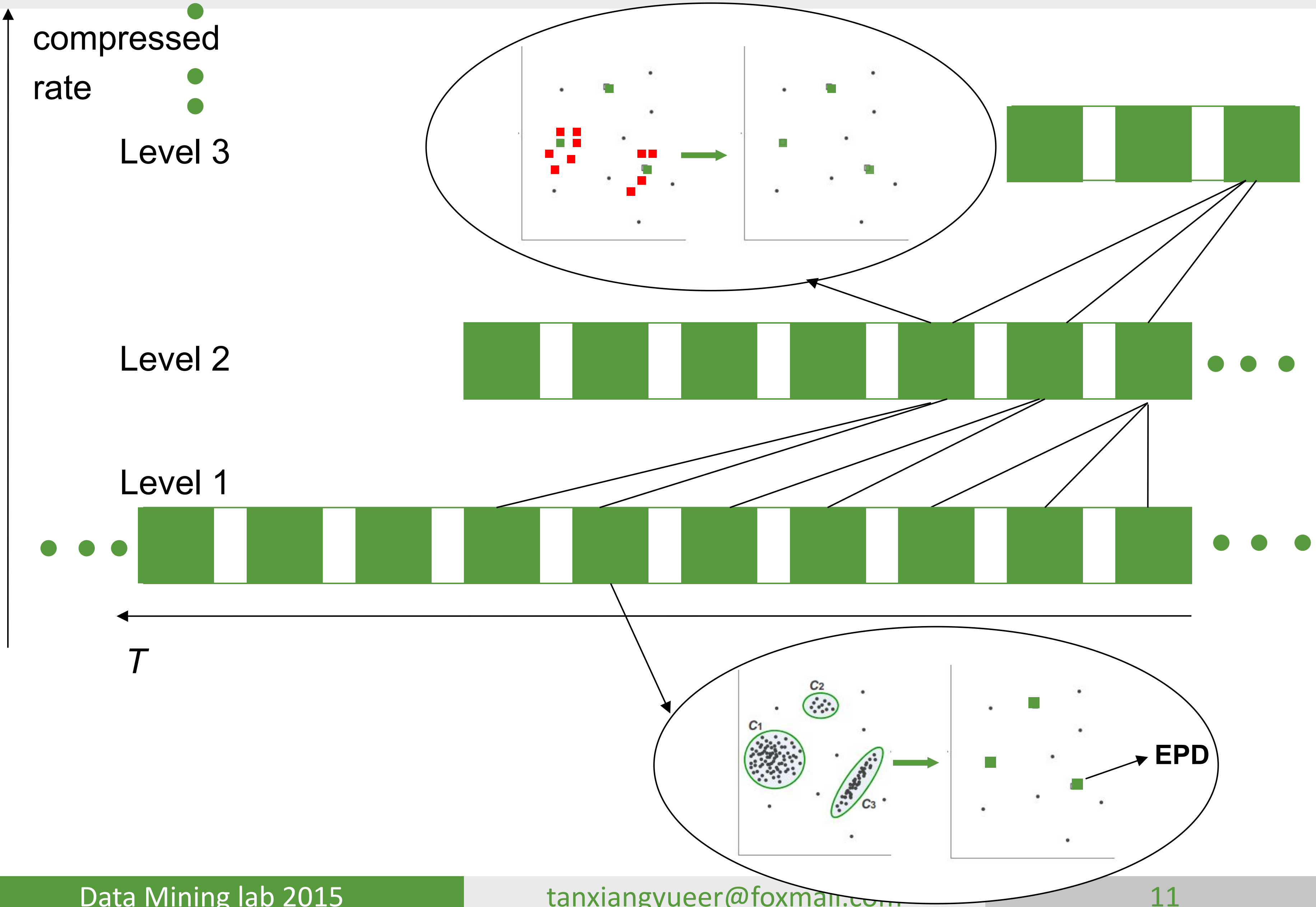
Storage distributed **parameters** in clusters.

Problem 4: Which distribution function?

EPD (Exponential Power Distribution)

To avoid the assumption of a certain data distribution,better meet the diverse data distribution.

2. Motivation



3.Data Stream Compression

3.1. Synchronization Algorithm

The key idea of clustering approaches by synchronization is to regard each data object as a phase oscillator and simulate the dynamical behaviors of the objects over time.

The dynamics of each dimension x_i of the object x over time is provided by:

$$x_i(t+1) = x_i(t) + \Delta t \omega_i + \frac{\Delta t \cdot S}{\left|Nb_\varepsilon(x(t))\right|} \sum_{y \in Nb_\varepsilon(x(t))} \sin(y_i(t) - x_i(t))$$

3.1 Synchronization Algorithm

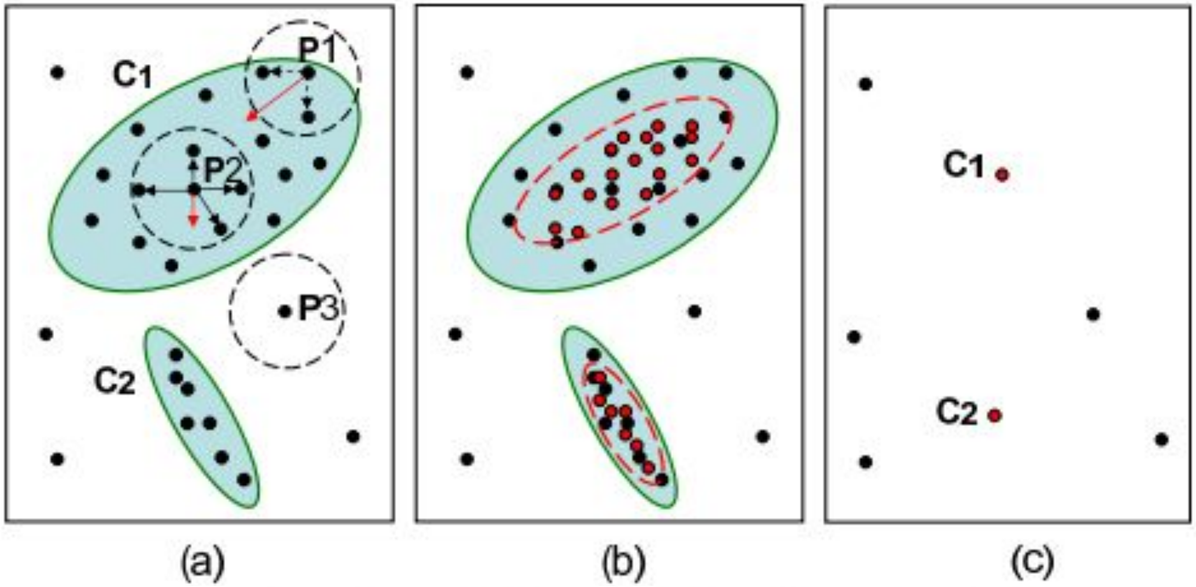


Figure 1: Clustering by synchronization.

Interaction with similar objects,the phase of an object gradually aligns with its neighborhood.

Finally, the objects in a cluster are synchronized together and have the same phase.

3.2 Exponential Power Distribution

The EPD is a family of distribution functions .

Includes:

the Gaussian distribution, the Laplacian distribution,

the uniform distribution, and so on.

Three different parameters.

the location parameter u , the scale parameter σ ,

a shape parameter p.

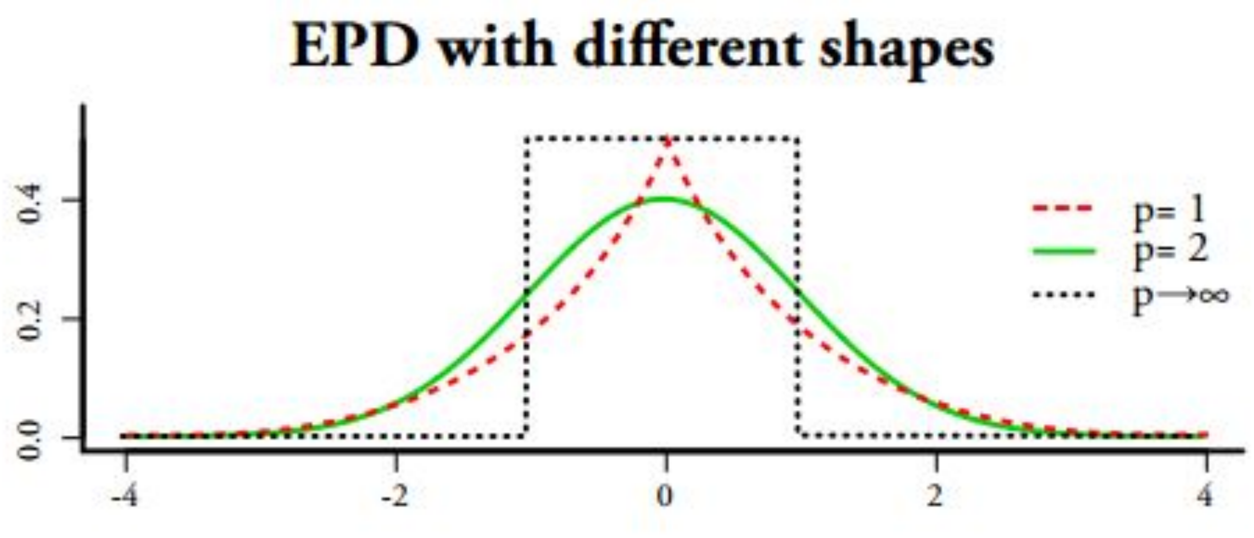
3.2 Exponential Power Distribution

For a random variable X, the EPD is defined as:

$$f_{EPD}(x;\mu,\sigma,p)=\frac{\exp\left(-\frac{\left|x-u\right|^{p}}{p\sigma^{p}}\right)}{2\sigma p^{\frac{1}{p}}\Gamma\left(1+\frac{1}{p}\right)}$$

$\Gamma(s)=\int_0^\infty t^{s-1}\exp(-t)dt$ is the gamma function

3.2 Exponential Power Distribution



shape parameter **p** determines kurtosis, or the sharpness of the distribution.

Figure 2: Different shapes of the Exponential Power Distribution for different choices of parameter p**.**

$p > 2$ & $p \to \infty$: a uniform distribution.; $p = 2$: a Gaussian distribution.

$2 < p < 0$, the EPD is leptokurtic; $p = 1$: a Laplacian distribution.

3.3 Sync-EPD

Disadvantages:

1. Every dimension is not **independent**.
2. The **covariance** is needed to consider.

However, covariance is not easy to calculate.

Thus, we apply the **ICA** to maximize non-Gaussianity as a measure of statistical independence.

3.3 Sync-EPD

Disadvantages:

1. Every dimension is not **independent**.
2. The **covariance** is needed to consider.

However, covariance is not easy to calculate.

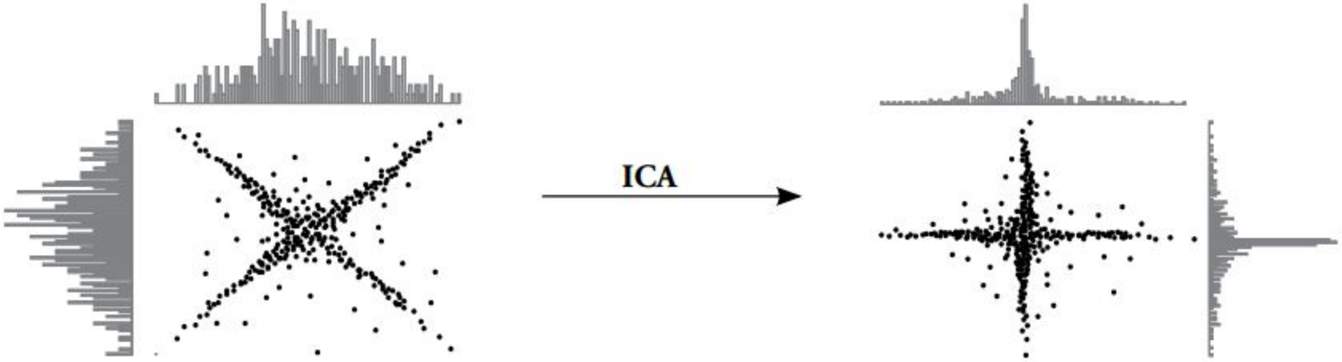
Thus, we apply the **ICA** to maximize non-Gaussianity as a measure of statistical independence.

3.4 Independent Component Analysis

The concept:

Independent component analysis (ICA) is a statistical method for transforming an observed multidimensional random vector into components that are statistically as independent from each other as possible.

The ICA enables us to process data sets which are not aligned to the orthogonal axes.



3.4 Independent Component Analysis

PCA:

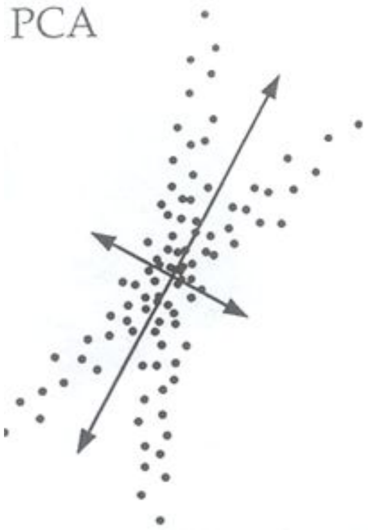
a) Focus on uncorrelated and Gaussian components

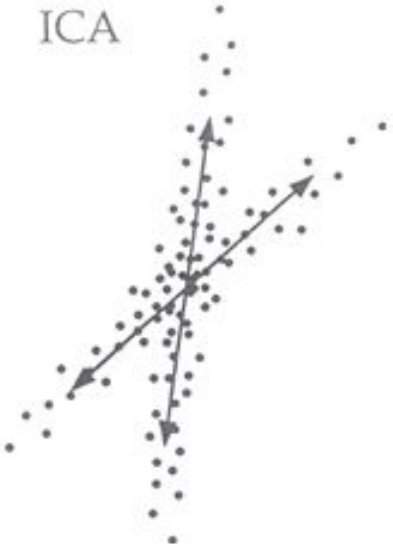
b) Orthogonal transformation

ICA:

a) Focus on independent and non-Gaussian components

b) Non-orthogonal transformation





3.4 Independent Component Analysis

The EPD in a d-dimensional space (after ICA) is defined for a point $\vec{x}$ as

$$f_{EPD}\left(x; M^{-1}, \vec{m}, \mu, \sigma, p\right) = \frac{\prod_{1 \le i \le d} f_{EPD}(z_i; \mu_i, \sigma_i, p_i)}{\left|\det\left(M^{-1}\right)\right|}$$

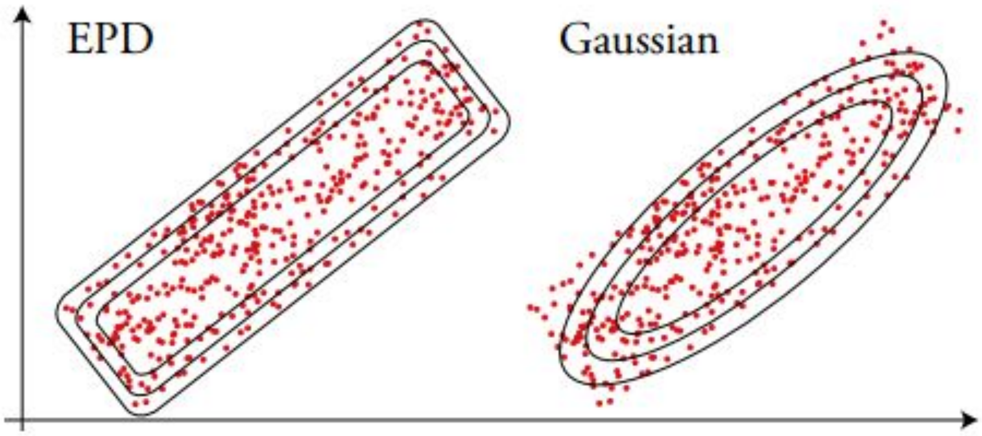


Figure 3: Data set approximated with an EPD and a Gaussian distribution.

Figure 3 illustrates the effect of the approximation of a data set with an EPD after ICA. While the approximation of the same data with a Gaussian distribution is rather inappropriate.

4 . Experiment

4 . Experiment

Context:

Intel(R) Core(TM) i5-4210M CPU @ 2.60GHz 2.60GHZ;

4.00GB RAM, Windows 7 64bit;

Eclipse Luna Release (4.4.0); jre-1.8.0。

Dataset:

Using PointPainter manually generate two-dimensional dataset, which spatial clustering feature is not obvious.There are 1299 points.

4.1 Sync-EPD

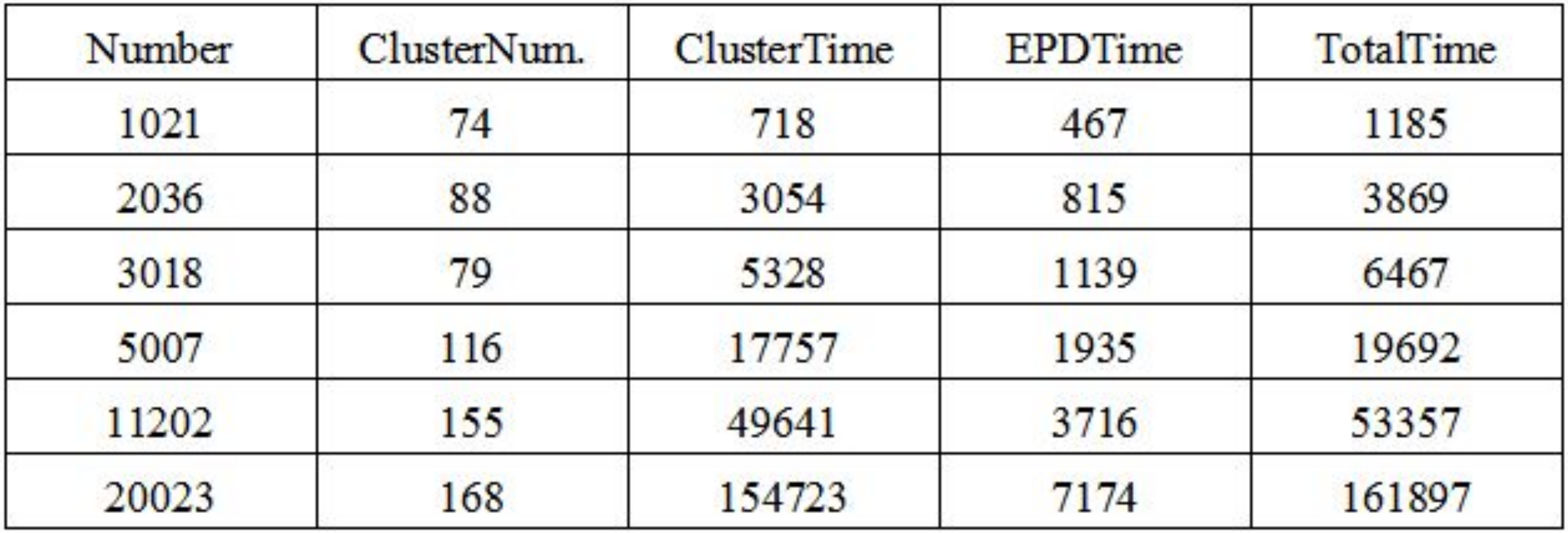
The data tuple 1:

$$\boldsymbol{S_1=(T, N, Center/ \mu, \sigma\ p, \ \varepsilon)}$$

where *T* is the timestamp, *N* is the number of points in each clusters,and center is the center of clusters. μ, σ, *p* describe the parameters of EPD. ε is the interaction range of clustering.

4.1 Sync-EPD

Table 4.1: Time consuming in different quantities of Sync-EPD algorithm.



Number	ClusterNum.	ClusterTime	EPDTime	TotalTime
1021	74	718	467	1185
2036	88	3054	815	3869
3018	79	5328	1139	6467
5007	116	17757	1935	19692
11202	155	49641	3716	53357
20023	168	154723	7174	161897

Illustration of the amount of data increasing, the time of its consumption is faster.

4.1 Sync-EPD

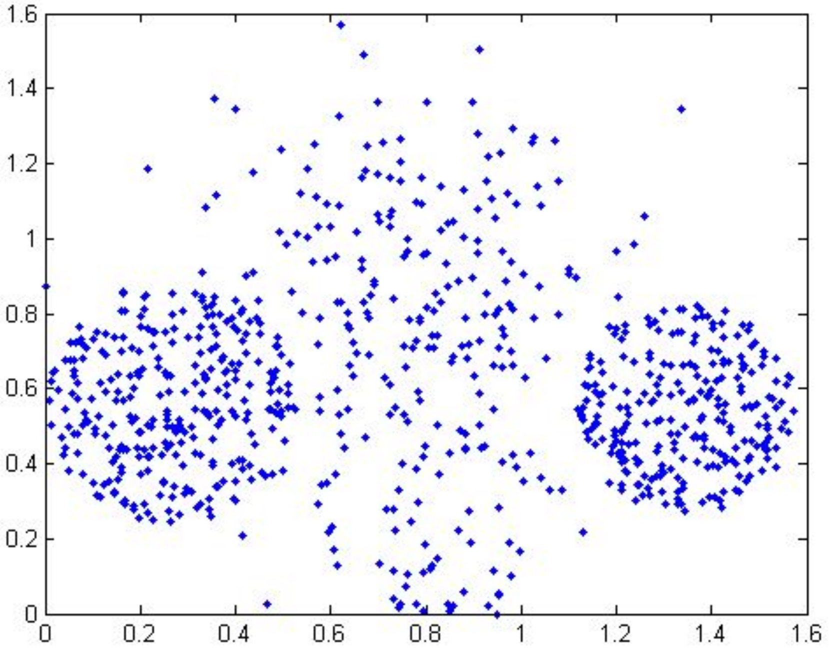


Figure 4.1 Synthetic dataset

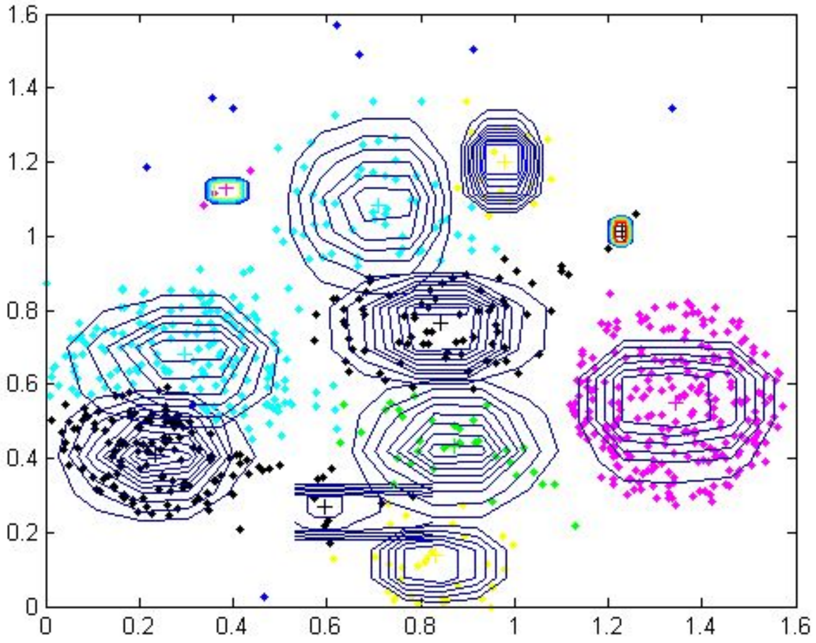


Figure 4.2 compressed result

The distribution of the data is well saved, and the noise points are preserved.

4.2 Sync-ICA-EPD

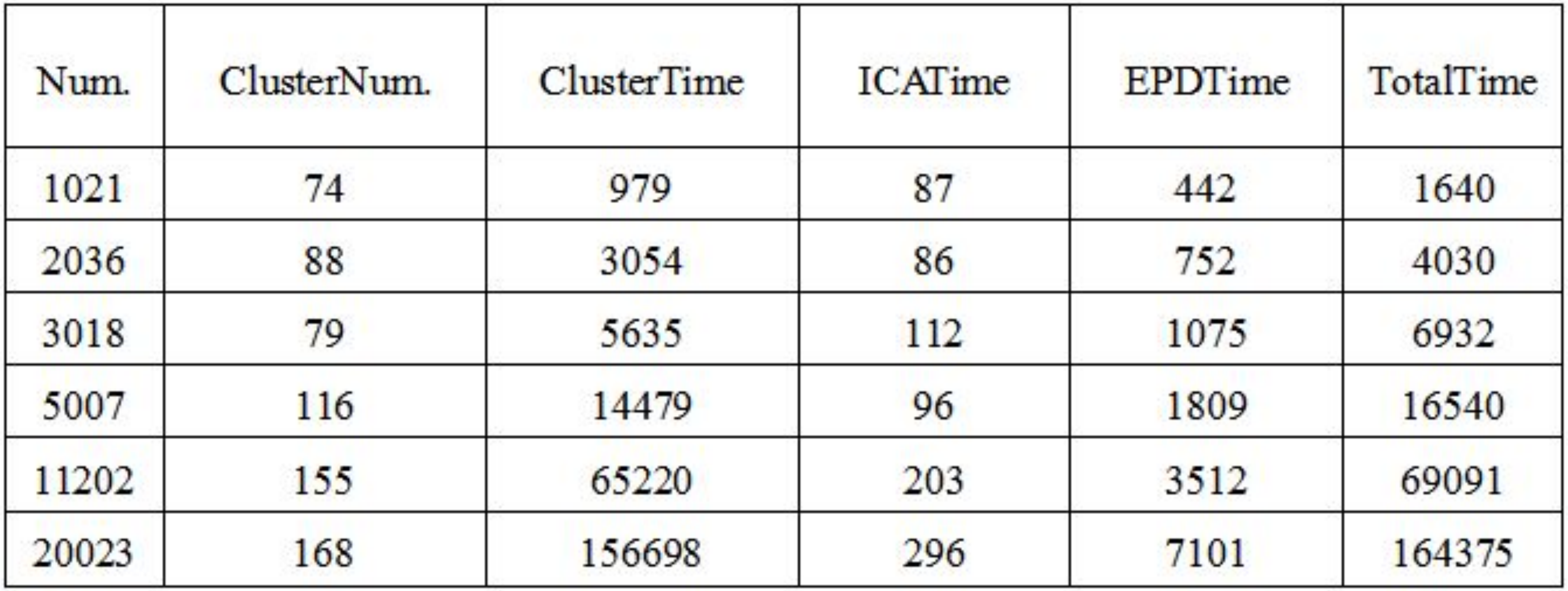
The data tuple 2:

$$S_2=(T, N, \mu, \sigma, p, \varepsilon, M, m)$$

T, N, μ, σ, p, ε is the same as $S_{1.}$ M is the de-mixing matrix, and $\vec{m}$ being the shifting vector.

4.2 Sync-ICA-EPD

Table 4.2: Time consuming of Sync-ICA-EPD algorithm.



Num.	ClusterNum.	ClusterTime	ICATime	EPDTime	TotalTime
1021	74	979	87	442	1640
2036	88	3054	86	752	4030
3018	79	5635	112	1075	6932
5007	116	14479	96	1809	16540
11202	155	65220	203	3512	69091
20023	168	156698	296	7101	164375

ICA does not take up too much time in total time-consuming .

4.2 Sync-ICA-EPD

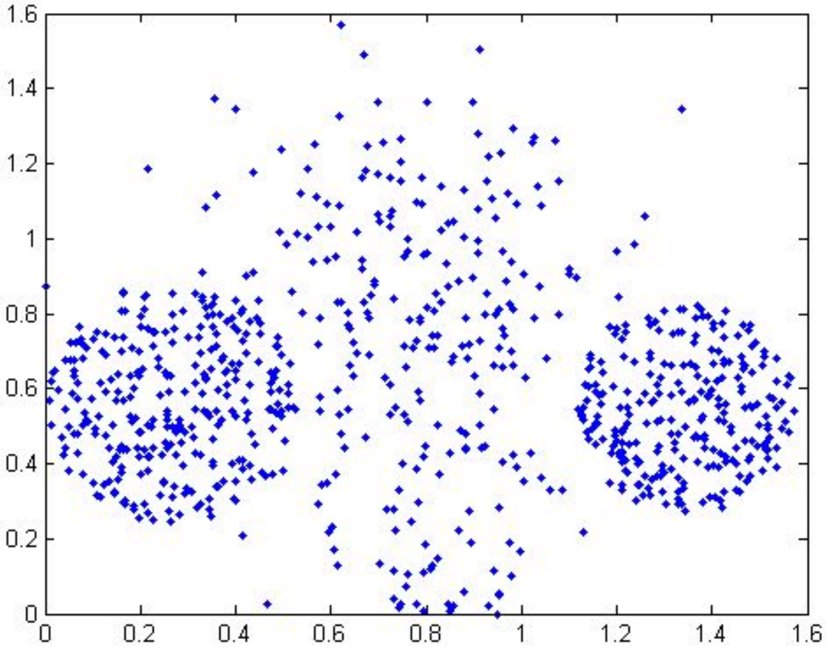


Figure 4.1 Synthetic dataset

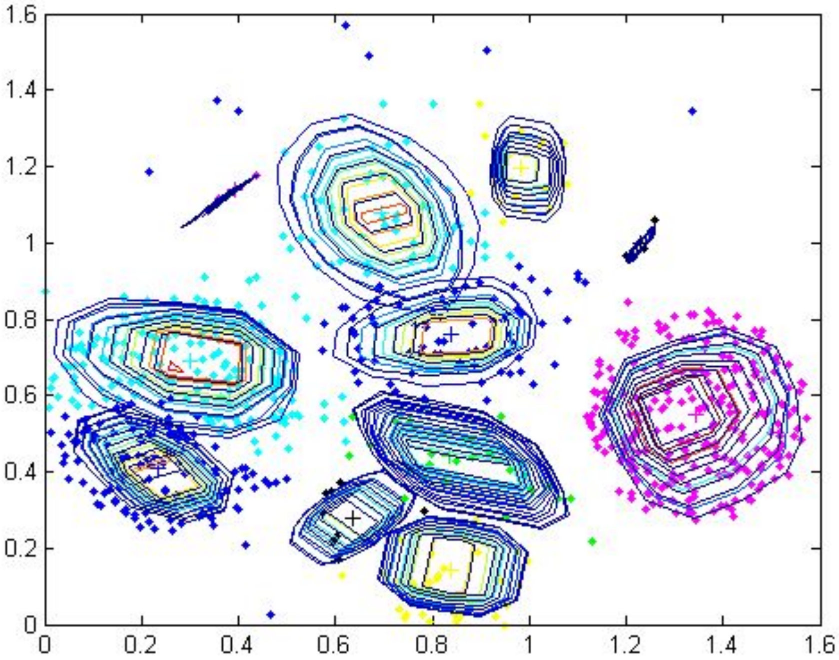


Figure 4.3 compressed result

1.The distribution of the data is saved better than Sync-EPD.

2.The fitting accuracy is higher and the effect is better.

4.3 ICA-Sync-EPD

The high space complexity.

The ICA is used at the beginning,so saved M and $\vec{m}$ once.

So,we have ICA-Sync-EPD algorithm.

The data tuple 3:

$$S_3=(T, N, Center/\mu, \sigma, p, \varepsilon)$$

Obviously, $S_3 = S_1$.

4.3 ICA-Sync-EPD

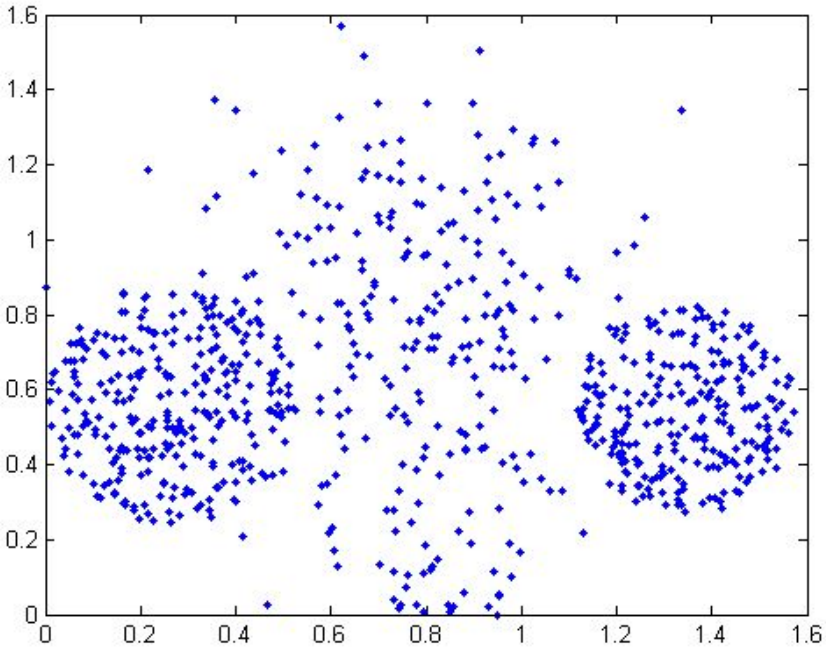


Figure 4.1 Synthetic dataset

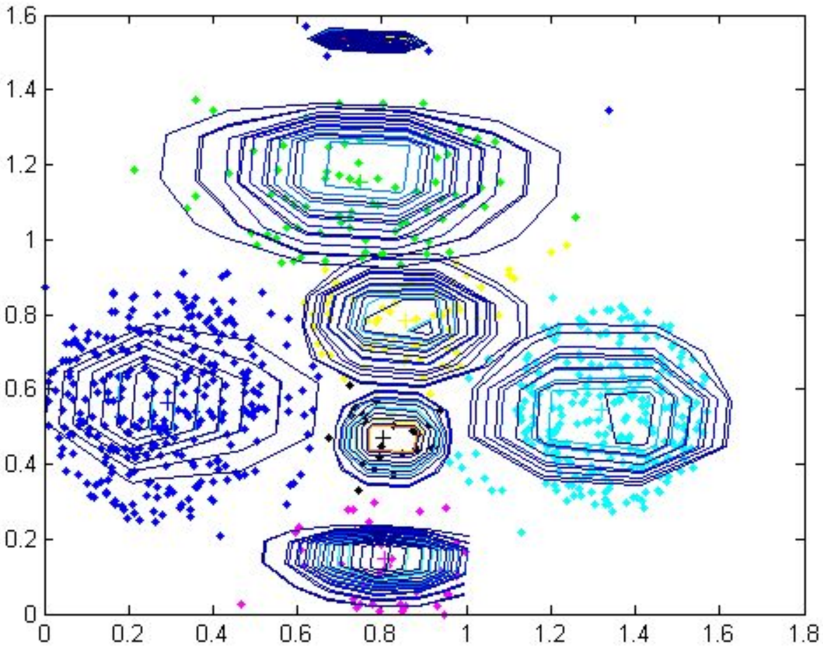


Figure 4.4 compressed result

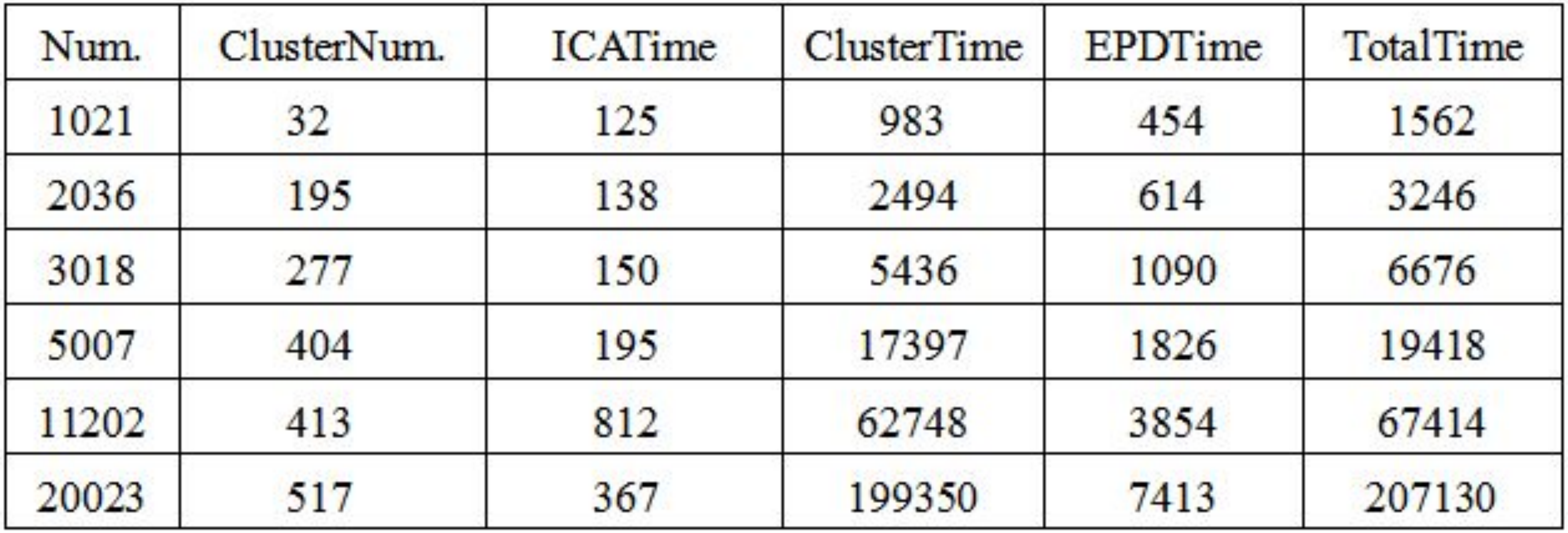
1.The distribution of the data is saved better than Sync-EPD.

2.The main direction of the distribution of the window is the same.

3. Accuracy is lower than Sync-ICA-EPD.

4.3 ICA-Sync-EPD

Table 4.3: Time consuming of ICA-Sync-EPD algorithm.



Num.	ClusterNum.	ICATime	ClusterTime	EPDTime	TotalTime
1021	32	125	983	454	1562
2036	195	138	2494	614	3246
3018	277	150	5436	1090	6676
5007	404	195	17397	1826	19418
11202	413	812	62748	3854	67414
20023	517	367	199350	7413	207130

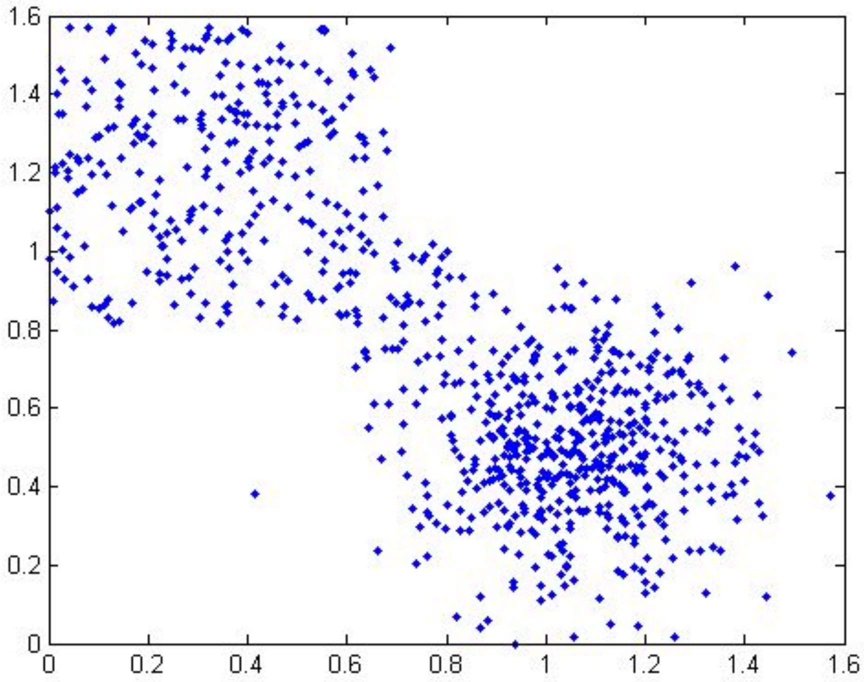
1.In the case of less number of data sets, the ISE algorithm is less time-consuming than SIE algorithm.

2.with the increase of the number of points, the time consumption of ISE algorithm is more than SIE

4.4 Hierarchical

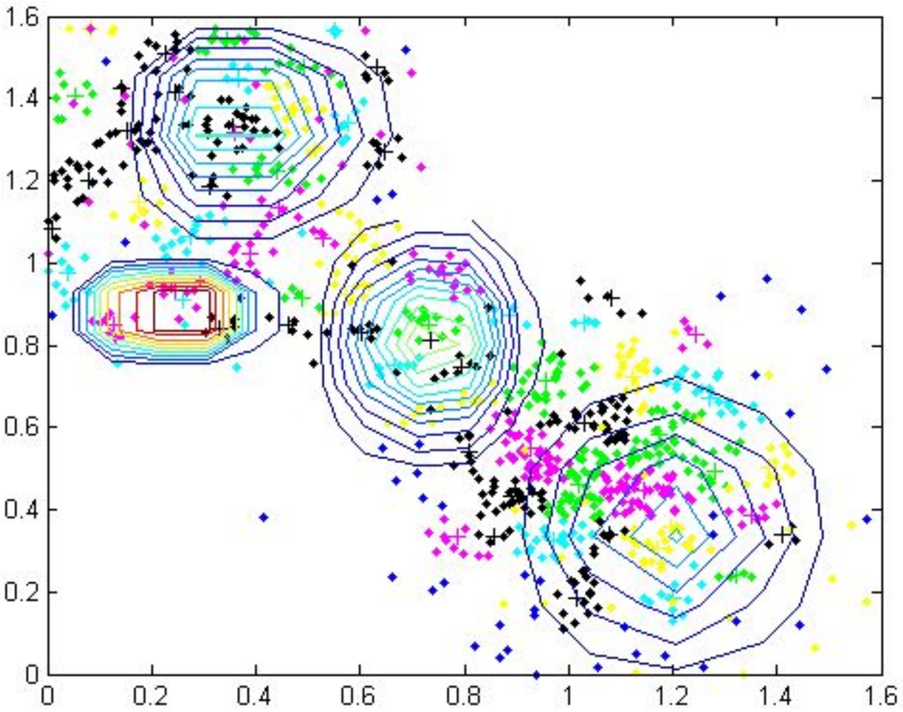
Method:

- Clustering center can be a good representative of the cluster.
- Using center as a new input

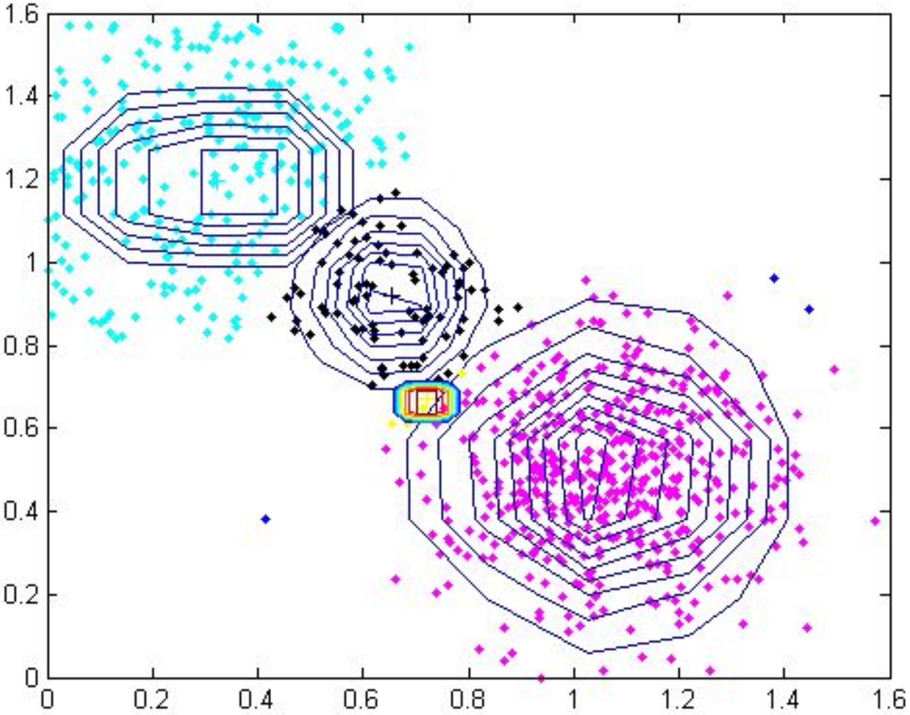


Generation has a uniform distribution and Gauss distribution of the two points set and add noise, a total of 1522 data

4.4 Hierarchical



4.4.1 Sync-EPD hierarchical compression

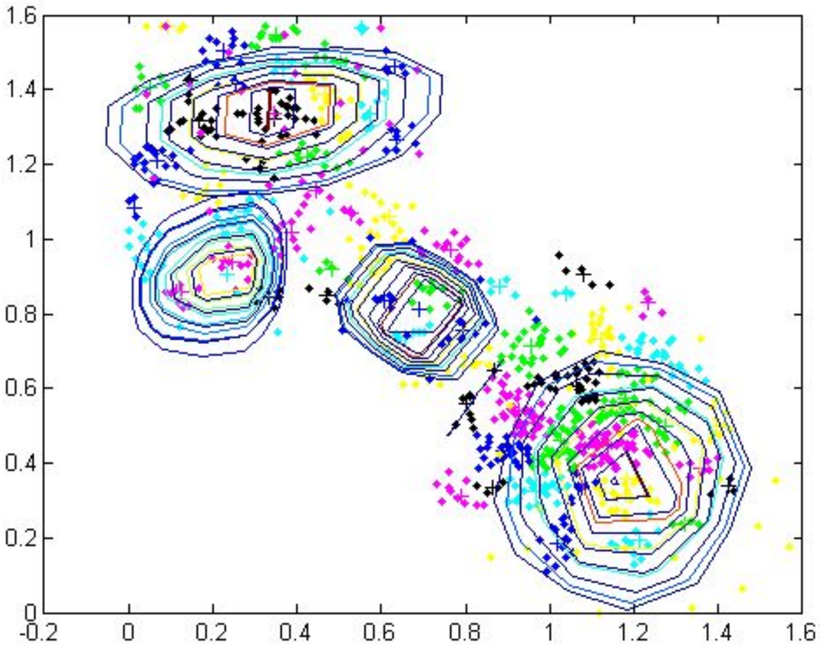


4.4.2 The best result of Sync-EPD

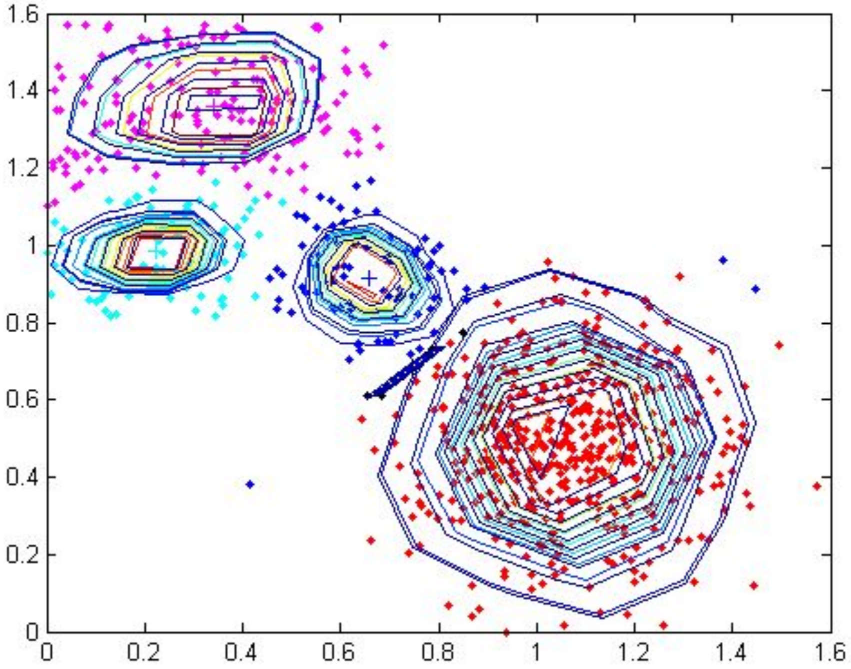
1.The result is not similarity with direct compression .

2.The distribution of the data set is well preserved.

4.4 Hierarchical



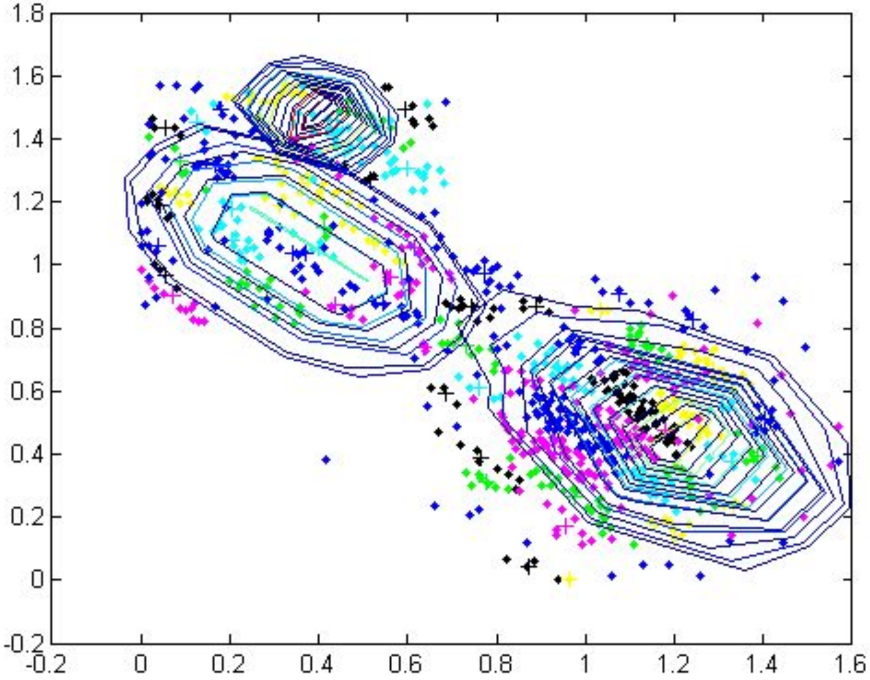
4.4.3 Sync-ICA-EPD hierarchical compression



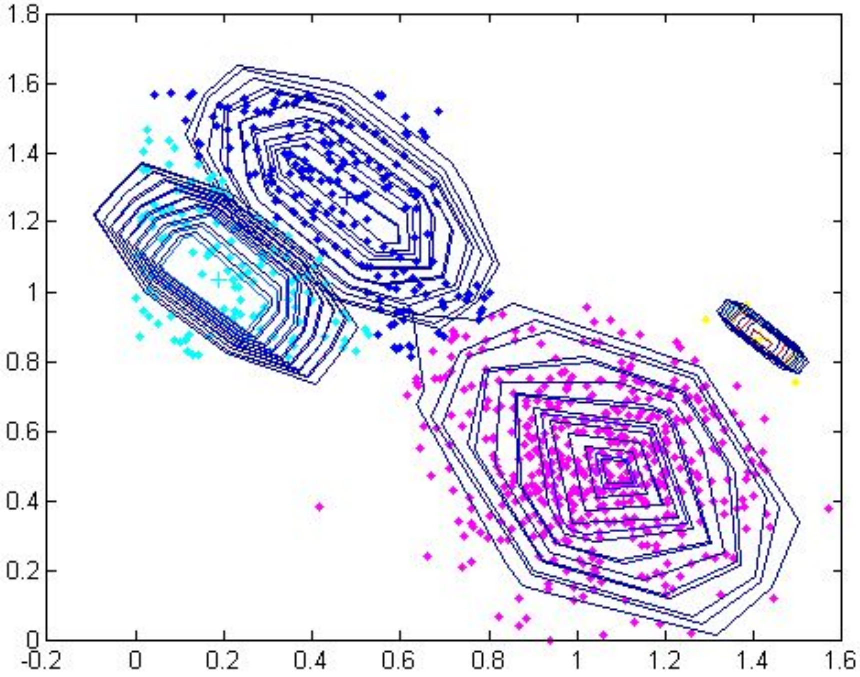
4.4.4 The best result of Sync-ICA-EPD

The fitting result of the original data is the best.

4.4 Hierarchical



4.4.3 ICA-Sync-EPD hierarchical compression



4.4.4 The best result of ICA-Sync-EPD

4.4 Hierarchical

Why Sync-ICA-EPD get better result than ICA-Sync-EPD for fitting of the original data ?

First using ICA

a) data independent in dimension;

b) data more discrete;

c) The Sync can make more clusters;

d) The number of data in clusters less;

e) Outliers more.

。

5. Discussion

5. Discussion

Problem 1 : **Additive of EPD parameters；**

a) Sum of EPD parameters;

b) Get higher level of the parameters fast;

c) The compression loss rate can be reduced;

d) The time and space complexity can be reduced .

Problem 2 **: Using ICA at higher level.**

Currently unable to apply ICA to a higher level,only use ICA at the first level.

Thank You!