Time series data mining

Outline

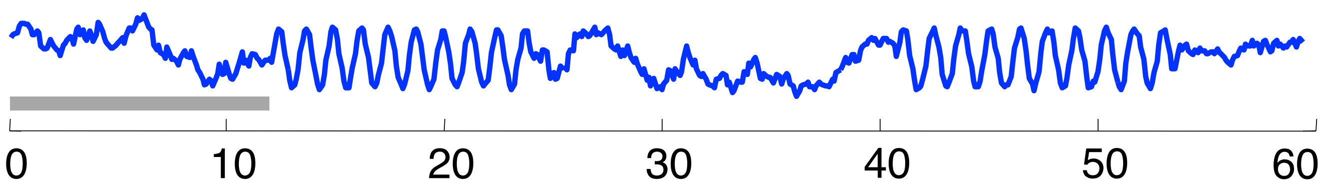
Basic Knowledge

Multi variate association

States association

What is time series data

Formally, a time series data is defined as a sequence of pairs T = [(p1, t1), (p2, t2), . . . , (pi, ti), . . . , (pn, tn)] (t1 < t2 < · · · < ti < · · · < tn), where each pi is a data point in a d-dimensional data space, and each ti is the time stamp at which the corresponding pi occurs.



Time Series Data Characteristics

1.high dimensionality

2.hierarchical nature

A time series can be analyzed by its underlying time hierarchy, such as hourly, weekly, monthly, and yearly.

3.multi-variate

Time series data analysis often studies one variable, but sometimes deals with time series data consisting of multiple related variables. For example, weather data consists of well-known measurements such as temperature, dew point, humidity, etc

Our work

Multi variate association:

A and B are highly correlated

States association :

$A = 2 \rightarrow B = 3$

Multi variate association

Extract feature

↓

Cluster the feature

↓

Analyze the clustering result

Why to extract feature

Time series are essentially high dimensional data and working directly with such data in its raw format is very expensive in terms of both processing and storage cost.

It is thus highly desirable to develop representation techniques that can reduce the dimensionality of time series, while still preserve the fundamental characteristics of it.

How to extract feature

Principles:
reduce dimension while preserve its fundamental characteristics

Split the data into fixed size window

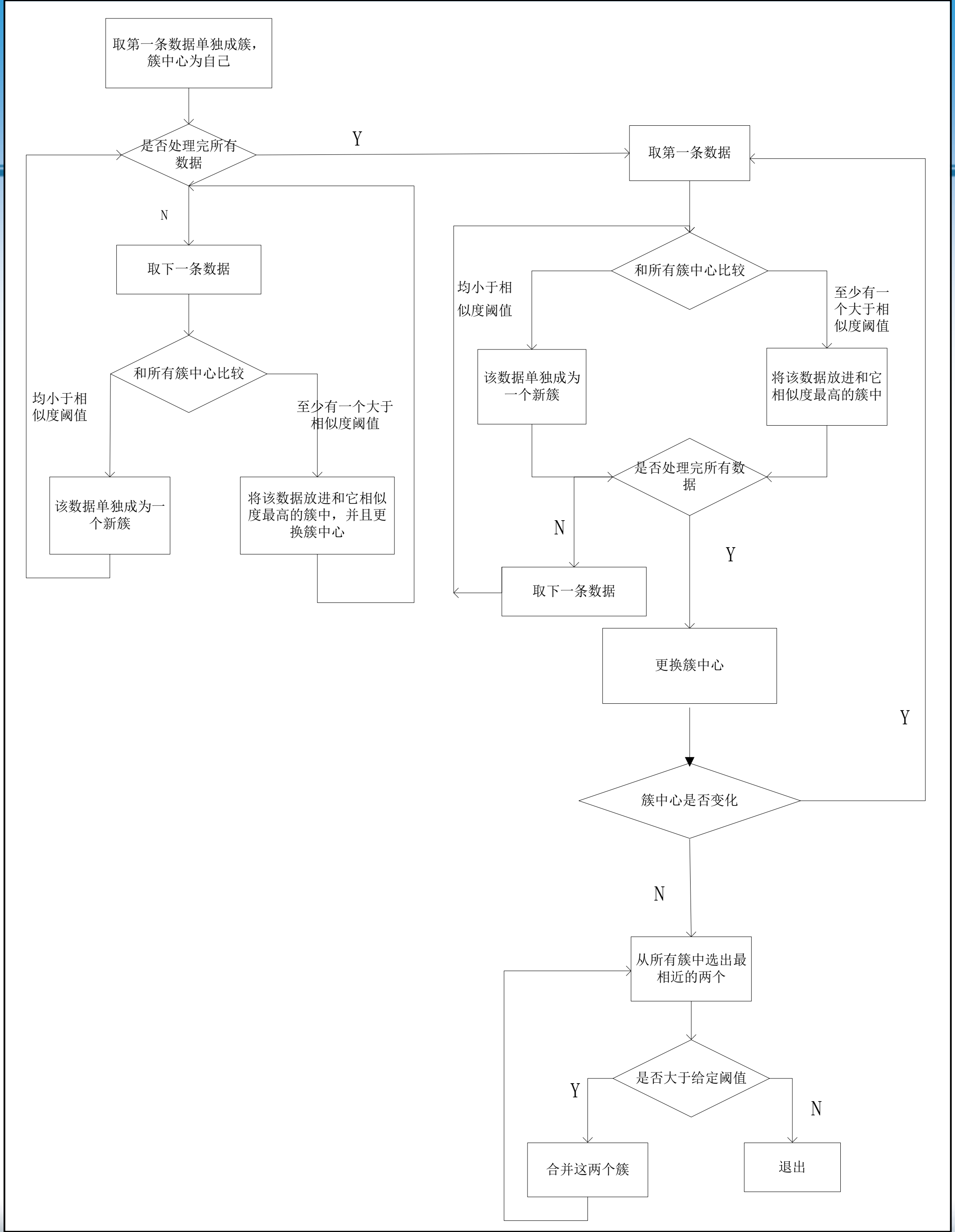
↓

Extract feature of each window
[relative time,standard deviation]

Clustering

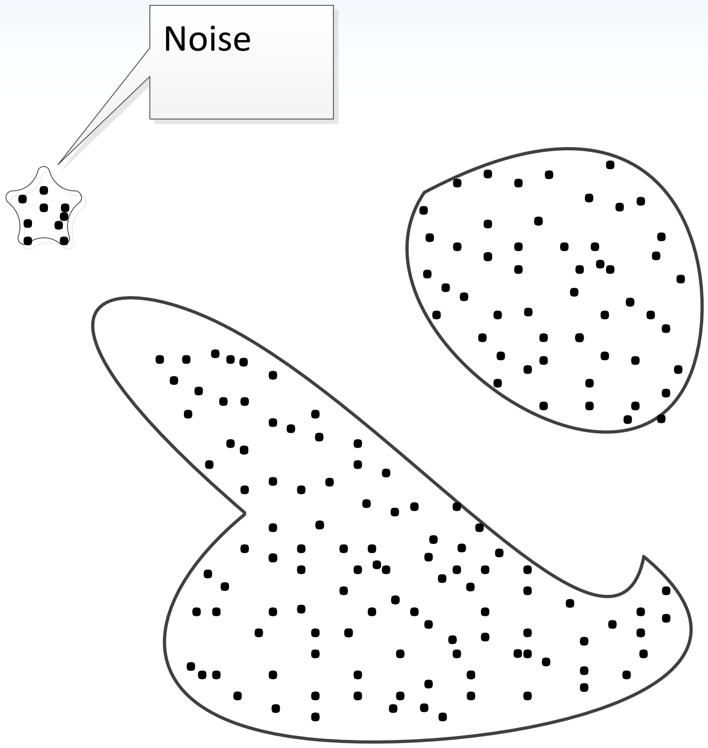
The objective is to find the most homogeneous clusters that are as distinct as possible from other clusters. More formally, the grouping should maximize intercluster variance while minimize intracluster variance.

Clustering



取第一条数据单独成簇，簇中心为自己
是否处理完所有数据
Y
N
取下一条数据
和所有簇中心比较
均小于相似度阈值
至少有一个大于相似度阈值
该数据单独成为一个新簇
将该数据放进和它相似度最高的簇中，并且更换簇中心
取第一条数据
和所有簇中心比较
均小于相似度阈值
至少有一个大于相似度阈值
该数据单独成为一个新簇
将该数据放进和它相似度最高的簇中
是否处理完所有数据
N
Y
取下一条数据
更换簇中心
Y
簇中心是否变化
N
从所有簇中选出最相近的两个
是否大于给定阈值
Y
N
合并这两个簇
退出

The analysis of clustering result



$$\sup port = \frac{\sum_{i=1}^{n} L_i}{\sum_{j=1}^{k}\sum_{i=1}^{n} L_{ij}}$$

Experiments result

```
共有3个簇:
簇的支持度为0.136363
P*186:46
P*187:47
P*183:47
P*190:46

簇的支持度为0.096712
P*221:2
P*1943:1
P*207:1
P*219:2
P*208:5
P*185:47
P*188:47
P*189:47
P*195:1
P*184:47

簇的支持度为0.766925
P*222:47
P*282:47
P*268:47
P*256:47
P*275:47
P*194:46
P*255:47
P*267:47
P*219:45
P*278:47
P*221:45
P*257:47
P*195:46
P*265:47
P*269:47
P*279:47
P*281:47
P*233:47
P*283:47
P*274:47
```

```
possible Association Parameters:
簇的支持度为0.766925
P*222:47
P*282:47
P*268:47
P*256:47
P*275:47
P*194:46
P*255:47
P*267:47
P*219:45
P*278:47
P*221:45
P*257:47
P*195:46
P*265:47
P*269:47
P*279:47
P*281:47
P*233:47
P*283:47
P*274:47
```

States Association

Our goal

Not only to find that variable A,B,C,D are related but also discover their value association.

$A = 2, D = 4 \rightarrow B = 3, C = 7$

States Association

- Transform the data into symbol

- Apriori algorithm

Data preprocessing

Split the data into fixed length window

↓

Extract the feature of each window

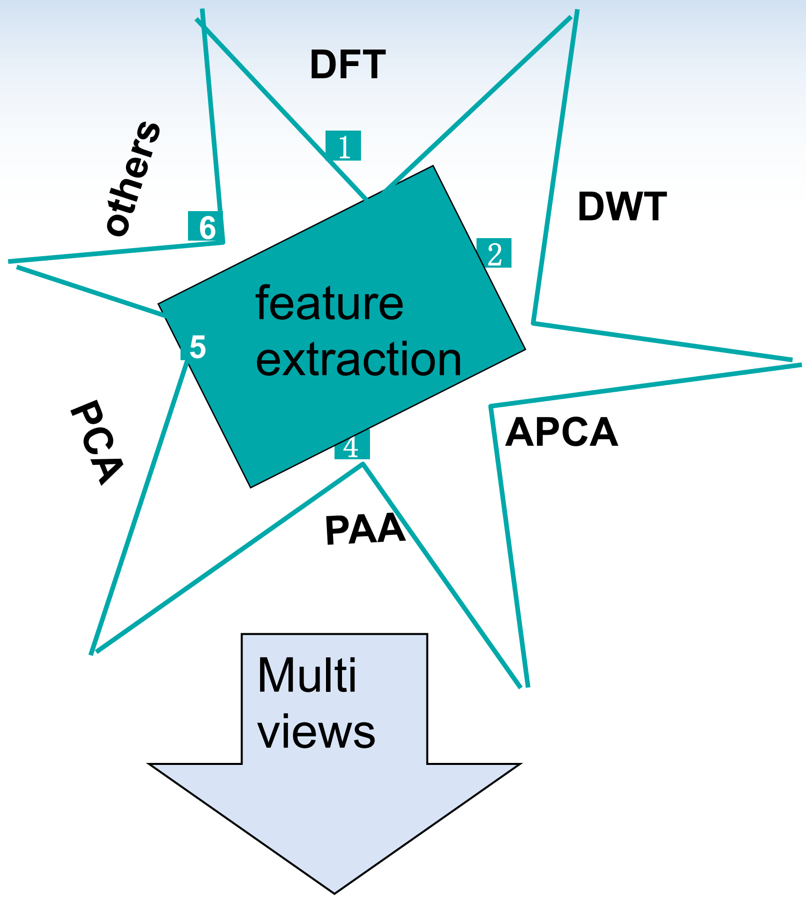
↓

Cluster the window

↓

Symbolize each cluster

Feature extraction



statistics | frequency domain | time domain | …

Clustering and Symbolization

Cluster the features extracted from each window and then each cluster is represented by a character. So the windows within a cluster is marked with the corresponding character.

data

```
2010-01-01 00:00:00 0.0871557427477
2010-01-01 00:01:00 0.104528463268
2010-01-01 00:02:00 0.121869343405
2010-01-01 00:03:00 0.13917310096
2010-01-01 00:04:00 0.15643446504
2010-01-01 00:05:00 0.173648177667
2010-01-01 00:06:00 0.190808995377
2010-01-01 00:07:00 0.207911690818
2010-01-01 00:08:00 0.224951054344
2010-01-01 00:09:00 0.2419218956
2010-01-01 00:10:00 0.258819045103
2010-01-01 00:11:00 0.275637355817
2010-01-01 00:12:00 0.292371704723
2010-01-01 00:13:00 0.309016994375
2010-01-01 00:14:00 0.325568154457
2010-01-01 00:15:00 0.342020143326
2010-01-01 00:16:00 0.358367949545
2010-01-01 00:17:00 0.374606593416
2010-01-01 00:18:00 0.390731128489
2010-01-01 00:19:00 0.406736643076
2010-01-01 00:20:00 0.422618261741
2010-01-01 00:21:00 0.438371146789
2010-01-01 00:22:00 0.45399049974
2010-01-01 00:23:00 0.469471562786
2010-01-01 00:24:00 0.484809620246
2010-01-01 00:25:00 0.5
2010-01-01 00:26:00 0.51503807491
2010-01-01 00:27:00 0.529919264233
2010-01-01 00:28:00 0.544639035015
2010-01-01 00:29:00 0.559192903471
2010-01-01 00:30:00 0.573576436351
2010-01-01 00:31:00 0.587785252292
2010-01-01 00:32:00 0.601815023152
2010-01-01 00:33:00 0.615661475326
2010-01-01 00:34:00 0.62932039105
2010-01-01 00:35:00 0.642787609687
2010-01-01 00:36:00 0.656059028991
2010-01-01 00:37:00 0.669130606359
2010-01-01 00:38:00 0.681998360062
2010-01-01 00:39:00 0.694658370459
2010-01-01 00:40:00 0.707106781187
2010-01-01 00:41:00 0.719339800339
2010-01-01 00:42:00 0.731353701619
2010-01-01 00:43:00 0.743144825477
2010-01-01 00:44:00 0.754709580223
2010-01-01 00:45:00 0.766044443119
```

→

```
babbacbabbaaaacabbccccbabaacbcabaccbabbccbbaaaccb
ccbabcabaaabbabcacaccbbaccaccbcbccbcbccbbcbaabbcbb
cbbaabbccaaaabccccabcccbaccbbacbabbccacacbccabbca
acaccbacbcabaccbbaaabaabbacabacbbaaacacacacbbcacba
acccabbccbcababbcaaaccabbaccaabacaaababccbcbcbbaca
ccbbababbbacbcbcccacbbaacbcbccacacabaabacbbacbcaac
caccacbbbbaaaacbbbbbcaabbcbaaccabbccaacacbbaacbcba
accaaabbcbaccaacbbabbabbbbcbbbcbacacbcccabbbabbbcc
abcbbbaaaabaabbbabacaccaaaacbcaccabababbcababbccbb
cbccaaaabaabbacbbbcbccbaaaaacbbcaabbaabaaccbccacca
cbcaaaabaaccccbbacbbabaaaacacbbacbcaaacaabcbcccaab
bcacccacbababcbbcbccaccaacbbacaccacaaababaabbcacaa
aaacababaababccccbbacbbbcaaabacbabcacbcbcaccacbbb
bbbbaabaaacbcaacbccccbacbbacbcabbababbabcbcbbcbbac
aaaabbbbbcbbbcaabbacbcbcacbacaacaacacabbacbacaacbb
abaacabbaacbbbccababababcbaaabccbbcacbccabbbbcababcb
acccbbaacbbcbaaabaacbccbcbcbbcaaaaccaaaccacbaabbca
bbbcccacbaabccaaaabbbbbcbbbbccaabaaacabccccbacacbc
bcabcaabbababaababcabaacbcbcaabccaacaabaababbbbcac
bccbcaabaabbacaccaaaacaacabcbcacbccbbacabccabaaccc
aabccbabcaacaaccbccaaaaabbacbcabaabcacabbcccbbabac
abccbabccabbabcccaabcaccacccaacaaaaccacbccaaacbaca
cccccbbaccbbbcbbabaccabbbcbbcaaaabacbaacaccbccbcac
cbbabacaccbcbcabbccbabbacccbcbbaabcbcabacabbbccbbb
aaabbabacabccbabccbbcccabccbacabcbaacccbbcbcbbbacb
aabbabaacabcaacaacabcccaabbbbacacccbabbccccaabaaab
cbcbaaacacabacbbbabacbaabcacababccabbcaababcabbccb
cbbbcbcbcbbccbaaaaacbacabbbaaccbaabcbcaccbacacccbc
ccbcabcaaccbabbbaccbabcabcbababcccaaabbbbcbbccaabc
bccacaacabbcccbcbaabccacbcbaaacbbcaabbcbbbbccbcabc
acccacbccccaaacbacaabccacbaacccbcccbbcabaabaacccab
ccabaabbbccbaaaabcbbaabcabacccacacbcbbbcbabcbbcbcc
bcbaacacbbbacbbaaccccbcbbcaabbacbccccacacbbccbcbab
bacbbbabccabbabbccccaaabcabcbbacaaabcccbabcbbbaaac
```

data

time	A1	A2	A3	A4	A5	A6
t1	a	d	a	b	c	a
t2	1	a	a	a	d	a
t3	2	b	b	b	c	b

Association mining

The Apriori Algorithm is an influential algorithm for mining frequent itemsets and association rules.

Association rule generation is usually split up into two separate steps:

1. First, minimum support is applied to find all frequent itemsets in a database.

2. Second, these frequent itemsets and the minimum confidence constraint are used to form rules.

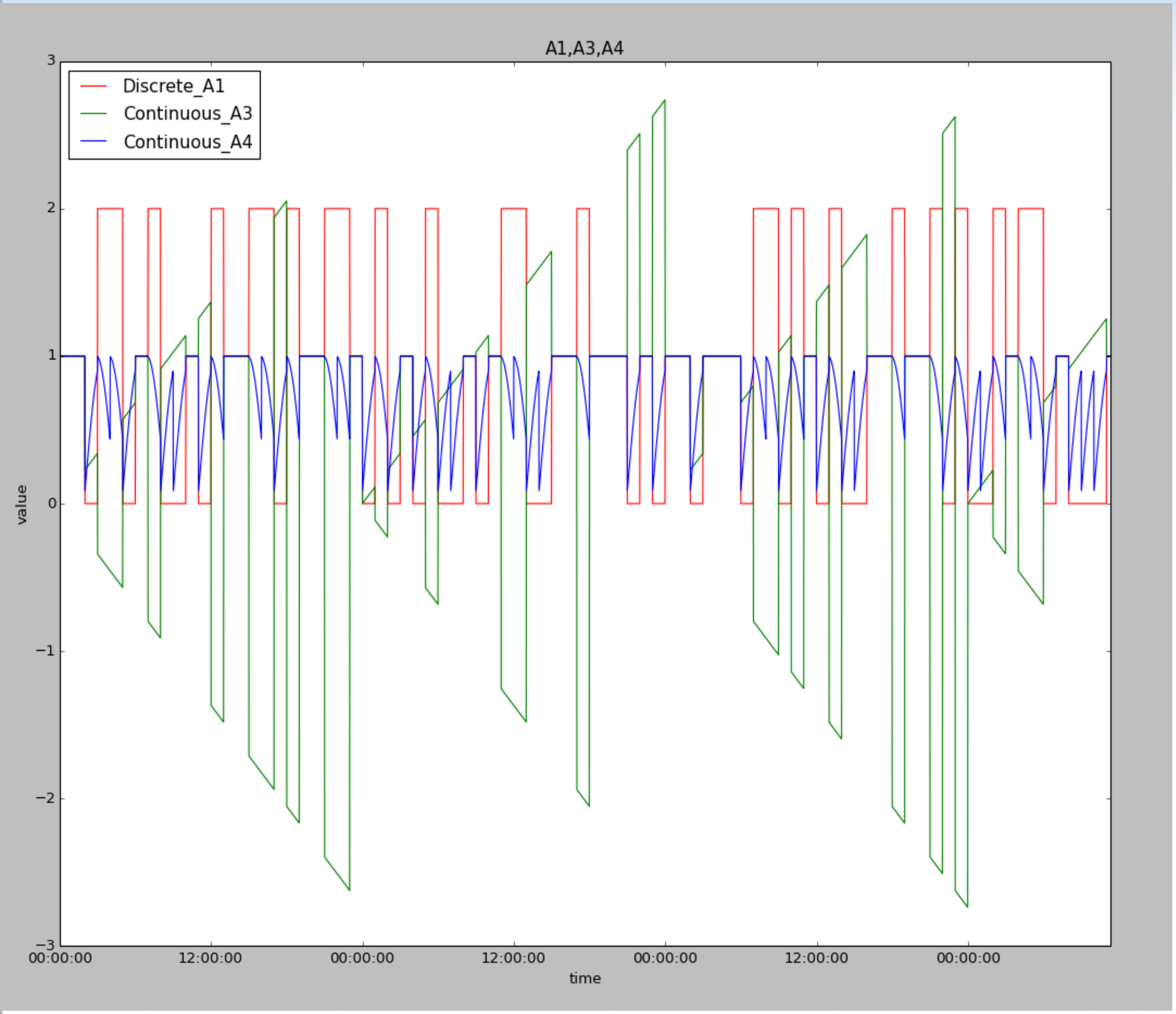
While the second step is straight forward, the first step needs more attention.

Association mining

Apriori Algorithm Pseudocode

```
procedure Apriori (T, minSupport) { //T is the database and minSupport is the minimum support
        L1= {frequent items};
        for (k= 2; L_{k-1} !=∅; k++) {
                C_k= candidates generated from L_{k-1}
                //that iscartesian product L_{k-1} x L_{k-1} and eliminating any k-1 size itemset that is not
                //frequent
                for each transaction t in database do{
                        #increment the count of all candidates in C_k that are contained in t
                        L_k = candidates in C_k with minSupport
                }//end for each
        }//end for
        return ∪_k L_k;
}
```

Experiment result

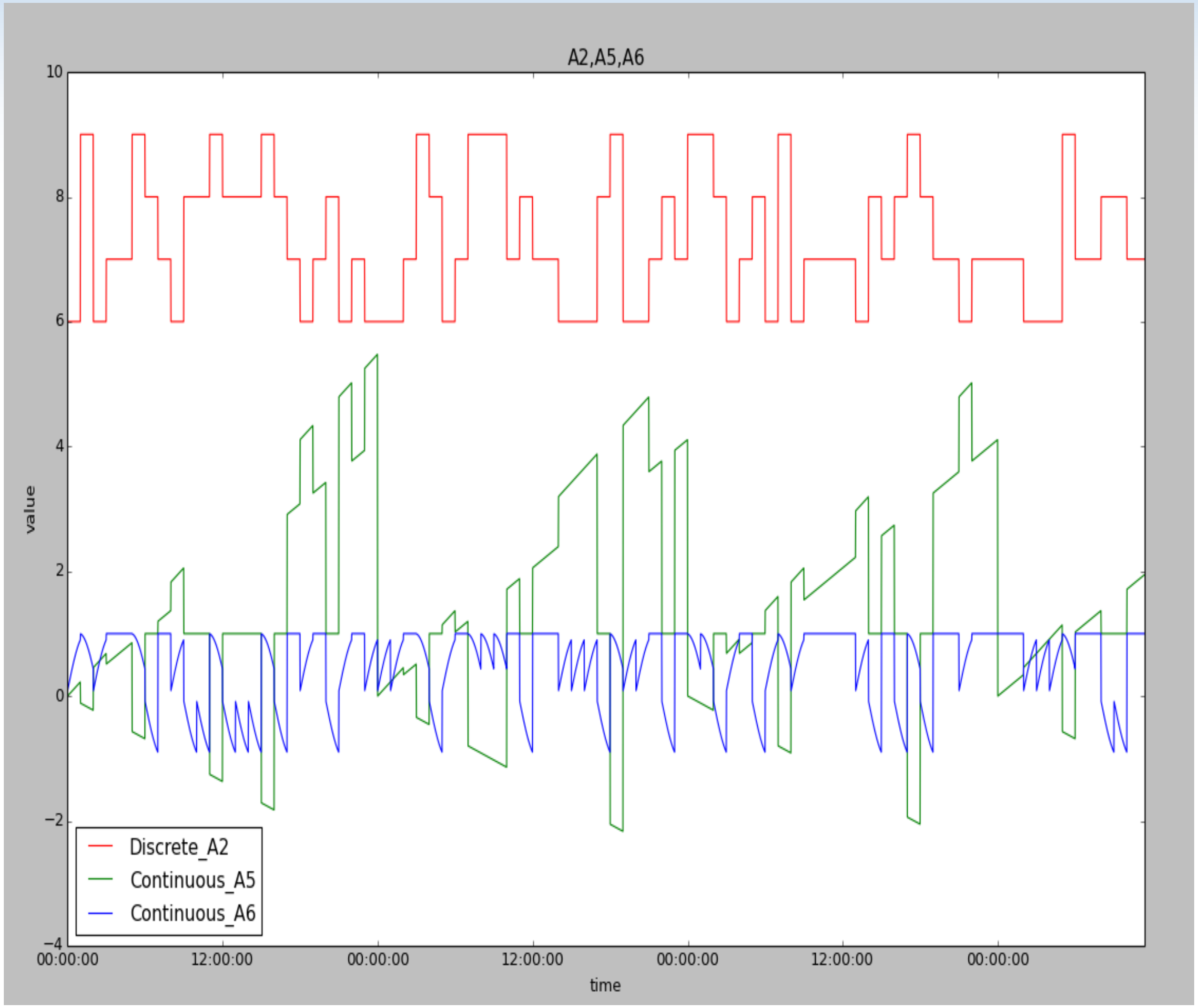


A1 = {0,1,2}

$$A_4 = \begin{cases} \sin(t+\delta) & A_1 = 0 \\ 1 & A_1 = 1 \\ -\sin(t+\delta) & A_1 = 2 \end{cases}$$

$$A_3 = \begin{cases} 1000 \times (t - t_0) / T & A_1 = 0 \\ 1 & A_1 = 1 \\ -1000 \times (t - t_0) / T & A_1 = 2 \end{cases}$$

Experiment result



A2 = {6,7,8,9}

$$A_5 = \begin{cases} 2000\times(t-t_0)/T & A_2 = 6 \\ 1500\times(t-t_0)/T & A_2 = 7 \\ 1 & A_2 = 8 \\ -1000\times(t-t_0)/T & A_2 = 9 \end{cases}$$

$$A_6 = \begin{cases} \sin(t+\delta) & A_2 = 6 \\ 1 & A_2 = 7 \\ -\sin(t+\delta) & A_2 = 8 \\ \cos(t+\delta) & A_2 = 9 \end{cases}$$

Experiment result

```
频繁1项集
[{('A3', 'a'): 2899, ('A4', 'a'): 2959, ('A2', '7'): 2175, ('A1', '2'): 2959, ('A6', 'd'): 2175,
 ('A5', '?'): 2079, ('A4', 'b'): 2902, ('A2', '8'): 2193, ('A6', 'a'): 2202, ('A1', '1'): 2899,
('A4', 'c'): 2899, ('A2', '9'): 2190, ('A6', 'b'): 2193, ('A3', '?'): 1795, ('A1', '0'): 2902, (
'A6', 'c'): 2190, ('A5', 'a'): 2193, ('A2', '6'): 2202}]
频繁2项集
[{(('A2', '7'), ('A6', 'd')): 2175, (('A1', '1'), ('A3', 'a')): 2899, (('A1', '1'), ('A4', 'c'))
: 2899, (('A3', 'a'), ('A4', 'c')): 2899, (('A1', '0'), ('A4', 'b')): 2902, (('A2', '8'), ('A6',
 'b')): 2193, (('A2', '8'), ('A5', 'a')): 2193, (('A2', '6'), ('A6', 'a')): 2202, (('A2', '9'),
('A6', 'c')): 2190, (('A1', '2'), ('A4', 'a')): 2959, (('A5', 'a'), ('A6', 'b')): 2193}]
频繁3项集
[{(('A2', '8'), ('A5', 'a'), ('A6', 'b')): 2193, (('A1', '1'), ('A3', 'a'), ('A4', 'c')): 2899}]
频繁4项集
[{}]
频繁5项集
[{}]
```

Experiment result

```
关联规则:
('A2', '7')->(('A6', 'd'),)
confidence is 1.000000
('A6', 'd')->(('A2', '7'),)
confidence is 1.000000
('A1', '1')->(('A3', 'a'),)
confidence is 1.000000
('A3', 'a')->(('A1', '1'),)
confidence is 1.000000
('A1', '1')->(('A4', 'c'),)
confidence is 1.000000
('A4', 'c')->(('A1', '1'),)
confidence is 1.000000
('A3', 'a')->(('A4', 'c'),)
confidence is 1.000000
('A4', 'c')->(('A3', 'a'),)
confidence is 1.000000
('A1', '0')->(('A4', 'b'),)
confidence is 1.000000
('A4', 'b')->(('A1', '0'),)
confidence is 1.000000
('A2', '8')->(('A6', 'b'),)
confidence is 1.000000
('A6', 'b')->(('A2', '8'),)
confidence is 1.000000
('A2', '8')->(('A5', 'a'),)
('A2', '8')->(('A5', 'a'), ('A6', 'b'))
confidence is 1.000000
('A5', 'a')->(('A2', '8'), ('A6', 'b'))
confidence is 1.000000
('A6', 'b')->(('A2', '8'), ('A5', 'a'))
confidence is 1.000000
(('A2', '8'), ('A5', 'a'))->(('A6', 'b'),)
confidence is 1.000000
(('A2', '8'), ('A6', 'b'))->(('A5', 'a'),)
confidence is 1.000000
(('A5', 'a'), ('A6', 'b'))->(('A2', '8'),)
confidence is 1.000000
('A1', '1')->(('A3', 'a'), ('A4', 'c'))
confidence is 1.000000
('A3', 'a')->(('A1', '1'), ('A4', 'c'))
confidence is 1.000000
('A4', 'c')->(('A1', '1'), ('A3', 'a'))
confidence is 1.000000
(('A1', '1'), ('A3', 'a'))->(('A4', 'c'),)
confidence is 1.000000
(('A1', '1'), ('A4', 'c'))->(('A3', 'a'),)
confidence is 1.000000
(('A3', 'a'), ('A4', 'c'))->(('A1', '1'),)
confidence is 1.000000
```

Thanks