



pcie固态硬盘

# 架构师的路上

数据挖掘，机器学习算法等领域的研究

[目录视图](#)[摘要视图](#)[RSS 订阅](#)

Follow me

<https://github.com/linyiqun>

个人资料



Android路上的人



【CSDN 技术主题月】物联网全栈开发 【评论送书】每周荐书：JVM、Nginx、小程序 CSDN日报20170609 ——《我成为程序员是别无选择，但之后却又别有洞天》

## PrefixSpan序列模式挖掘算法

标签：数据挖掘 算法 机器学习

2015-02-12 19:06

9576人阅读

[评论\(7\)](#)[收藏](#)[举报](#)[分类：](#) [机器学习 \(30\)](#) [数据挖掘 \(32\)](#) [算法 \(44\)](#)[关闭](#)

版权声明：本文为博主原创文章，未经博主允许不得转载。

[目录\(?\)](#)[\[+\]](#)

更多数据挖掘代码：<https://github.com/linyiqun/DataMiningAlgorithm>

### 介绍

与GSP一样，PrefixSpan算法也是序列模式分析算法的一种，不过与前者不同的是，在这点上可以说已经比GSP好很多了。PrefixSpan算法可以挖掘出满足阈值的序列的格式就是上文中提到过的类似于<a, b, ( de)>这种的。

### 买入口诀

- |               |                |
|---------------|----------------|
| • 1一针锥底，买股时机。 | • 9三杆通底，后市可喜   |
| • 2三军会师，看好后市。 | • 10重锤坠地，后市有戏  |
| • 3双管齐下，买进不怕。 | • 11巨阳入海，放心购买  |
| • 4白龙出水，短线可为。 | • 12布林破底，短线可取  |
| • 5五阳上阵，股价弹升。 | • 13. CR扭成团，买进 |
| • 6身抱多线，好景出现。 | • 14. 天上雨来急，地上 |
| • 7三棒撑地，行情见底。 | • 亦涨，多头应进场。    |

### 股票口诀





### 博主介绍

Apache Hadoop  
Committer，其中主要研究HDFS。毕业于HDU计算机系，研究领域分布式计算，大数据，数据挖掘，机器学习，算法。曾就职于国内女性电商平台蘑菇街，目前就职于唯品会上海研发中心，数据平台与应用部门

### 新书发布

## 算法原理

PrefixSpan算法的原理是采用后缀序列转前缀序列的方式来构造频繁序列的。举个例子，

```
abc  abc  ac  d  cf
ad   c   abcd ae
ef   ab  df  c  b
e    g   af  c  b  c
```

比如原始序列如上图所示，4条序列，1个序列中好几个项集，项集内有1个或多个元素，首先找出前缀为a的子序列，此时序列前缀为<a>,后缀就变为了：

```
a
 _bc  abc  ac  d  cf
 _d   c   abcd ae
 _b   df  c  b
 _f   c   b  c
```

"\_"下标符代表前缀为a，说明是在项集中匹配的。这就相当于从后缀序列中提取出1项加入到前缀序列中，变化的规则就是从左往右扫描，找到第1个此元素对应的项，然后做改变。然后根据此规则继续递归直到后续序列不满足最小支持度阈值的情况。所以此算法的难点就转变为了从后缀序列变为前缀序列的过程。在这个过程要分为2种情况，第1种是单个元素项的后缀提前，比如这里的a，对单个项的提前有分为几种情况，比如：

<b a c ad>，就会变为<c ad>，如果a是嵌套在项集中的情况<b c dad r>，就会变为<\_d r>，\_代表的就是a。如果a在项的最末尾，此项也会被移除<b c dda r>变为<r>。但是如果是这种情况<\_da d d>a包含在下标符中，则应该此时的a是在前缀序列所属的项集内的。

还有1个大类的分类就是对于组合项的后缀提取，可以分为2个情况，1个是从\_X中  
如在这里<a>的条件下，找出前缀<(ab)>的后缀序列

```
<<a b > > :3
 _c  abc  ac  d  cf
 _cd ae
 _df  c  b
```

第一种在\_X中寻找还有没有X=a的情况，因为\_已经代表1个a了，还有一个是判断包含有连续的aa的那个项集，然后做变换处理，与单个项集的变换规则一致。

算法的递归顺序

想要实现整个的序列挖掘，算法的递归顺序就显得非常重要了。在探索递归顺序的递归顺序是<a>---><a a>----><a a a>，假设<a a a>找不到对应的后缀模式时

关闭

### 买入口诀

- 1. 一针锥底，买股时机。
- 2. 三军会师，看好后市。
- 3. 双管齐下，买进不怕。
- 4. 白龙出水，短线可为。
- 5. 五阳上阵，股价弹升。
- 6. 身抱多线，好景出现。
- 7. 三棒撑地，行情见底。
- 9. 三杆通底，后市可喜。
- 10. 重锤坠地，后市有戏。
- 11. 巨阳入海，放心购买。
- 12. 布林破底，短线可期。
- 13. CR扭成团，买进票不商量。
- 14. 天上雨来急，地上亦涨，多头应进场。

### 股票口诀



pcie固态硬盘

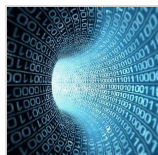


新书<<深度剖析Hadoop HDFS>>发布上市，此书源自于笔者博客，重新经过整理，完善而成，此书的定位并不是一本纯源码分析的书籍，其中有许多笔者在工作和学习中对于HDFS的一些有趣的看法和理解。链接：

淘宝

京东

#### 博客专栏



经典数据挖掘算法

文章：29篇

阅读：142886

HDFS源码分析

文章：6篇

阅读：11913

现这样会漏掉情况，为什么呢，因为如果  $\langle a \ a \rangle$  没法进行到  $\langle a \ a \ a \rangle$ ，那么就不可能会有前缀  $\langle a \ (aa) \rangle$ ，顶多会判断到  $\langle (aa) \rangle$ ，从  $\langle a \ a \rangle$  处回调的。于是我发现了这个问题，就变为了下面这个样子，经测试是对的。：

加入所有的单个元素的类似为  $a-f$ ，顺序为

$\langle a \rangle$ ，---> $\langle a \ a \rangle$ 。同时  $\langle (aa) \rangle$ ，然后  $\langle ab \rangle$  同时  $\langle (ab) \rangle$ ，就是在  $a$  添加  $a-f$  的元素的时候，检验  $a$  所属项集添加  $a-f$  元素的情况。

这样就不会漏掉情况了，用了2个递归搞定了这个问题。这个算法的整体实现可以对照代码来看会理解很多。最后提醒一点，在每次做出改变之后都会判断一下是否满足最小支持度阈值的。

### PrefixSpan实例

这里举1个真实一点的例子，下面是输入的初始序列：



$\langle (b \ d) \ c \ b \ (a \ c) \rangle$   
 $\langle (b \ f) \ (c \ e) \ b \ (f \ g) \rangle$   
 $\langle (a \ h) \ (b \ f) \ a \ b \ f \rangle$   
 $\langle (b \ e) \ (c \ e) \ d \rangle$   
 $\langle a \ (b \ d) \ b \ c \ b \ (a \ d) \rangle$

挖掘出的所有的序列模式为，下面是一个表格的形式

关闭

#### 买入口诀

- 1 一针锥底，买股时机。
- 2 三军会师，看好后市。
- 3 双管齐下，买进不怕。
- 4 白龙出水，短线可为。
- 5 五阳上阵，股价弹升。
- 6 身抱多线，好景出现。
- 7 三棒撑地，行情见底。
- 9 三杆通底，后市可喜。
- 10 重锤坠地，后市有戏。
- 11 巨阳入海，放心购买。
- 12 布林破底，短线可期。
- 13. CR扭成团，买进票不商量。
- 14. 天上雨来急，地上亦涨，多头应进场。

#### 股票口诀







pcie固态硬盘





分析  
文章：11篇  
阅读：20964



Redis源码分析  
文章：36篇  
阅读：129272

文章搜索

- 文章分类
- Jvm (2)
  - BigData (13)
  - Nosql (3)
  - Java 中间件 (1)
  - 设计模式 (19)
  - 算法 (45)
  - 数据结构 (3)
  - Linux Kernel (2)
  - JDK源码 (7)

Prefix	Projected (postfix) database	Sequential patterns
<a>	<(_c)> <(bf)abf> <(bd)bcb(ade)>	<a>, <aa>, <ab>, <aba>, <abb>
<b>	<(_d)cb(ac)> <(_f)(ce)bf> <(_f)abf> <(_e)(ce)d> <(_d)bcb(ade)>	<b>, <ba>, <bb>, <bba>, <bbc>, <bbf>, <bc>, <bca>, <bcb>, <bcba>, <bcd>, <b(ce)>, <bd>, <(bd)>, <(bd)a>, <(bd)b>, <(bd)ba>, <(bd)bc>, <(bd)c>, <(bd)ca>, <(bd)cb>, <(bd)cba>, <be>, <bf>, <(bf)>, <(bf)b>, <(bf)bf>, <(bf)f>
<c>	<b(ac)> <(_e)bf> <(_e)d> <b(ade)>	<c>, <ca>, <cb>, <cd>, <(ce)>, <cba>
<d>	<cb(ac)> <bcb(ade)>	<d>, <da>, <db>, <dc>, <dba>, <dbc>, <dca>, <dcb>
<e>	<bf> <d>	<e>
<f>	<(ce)bf> <abf>	<f>, <fb>, <ff>, <fbf>

在<b>的序列模式中少了1个序列模式。可以与后面程序算法测试的结果做对比。

算法的代码实现

代码实现同样以这个为例子，这样会显得更有说服力。

测试数据：

[java] view plain copy print ?

```
01. bd c b ac
02. bf ce b fg
03. ah bf a b f
04. be ce d
05. a bd b c b ade
```

Sequence.java:

[java] view plain copy print ?

```
01. package DataMining_PrefixSpan;
```

关闭

### 买入口诀

1一针锥底，买股时机。

2三军会师，看好后市。

3双管齐下，买进不怕。

4白龙出水，短线可为。

5五阳上阵，股价弹升。

6身抱多线，好景出现。

7三棒撑地，行情见底。

9三杆通底，后市可喜。

10重锤坠地，后市有戏。

11巨阳入海，放心购买。

12布林破底，短线可期。

13.CR扭成团，买进不商量。

14.天上雨来急，地上亦涨，多头应进场。

### 股票口诀





pcie固态硬盘



数据挖掘 (33)

机器学习 (31)

分布式系统 (34)

搜索引擎 (6)

系统运维 (17)

## 文章存档

2017年06月 (1)

2017年05月 (2)

2017年04月 (3)

2017年03月 (1)

2017年02月 (3)

展开

## 阅读排行

从Container内存监控限制

(222632)

Redis源码分析 (一) --R

(19040)

决策分类树算法之ID3 , (

(14419)

HDFS内存存储

(14396)

HDFS数据加密空间--Enc

```

02.
03. import java.util.ArrayList;
04.
05. /**
06.  * 序列类
07.  *
08.  * @author lyq
09.  *
10.  */
11. public class Sequence {
12.     // 序列内的项集
13.     private ArrayList<ItemSet> itemSetList;
14.
15.     public Sequence() {
16.         this.itemSetList = new ArrayList<>();
17.     }
18.
19.     public ArrayList<ItemSet> getItemSetList() {
20.         return itemSetList;
21.     }
22.
23.     public void setItemSetList(ArrayList<ItemSet> itemSetList) {
24.         this.itemSetList = itemSetList;
25.     }
26.
27.     /**
28.      * 判断单一项是否包含于此序列
29.      *
30.      * @param c
31.      *      待判断项
32.      * @return
33.      */
34.     public boolean strIsContained(String c) {
35.         boolean isContained = false;
36.
37.         for (ItemSet itemSet : itemSetList) {
38.             isContained = false;
39.
40.             for (String s : itemSet.getItems()) {

```

关闭

### 买入口诀

- 1 一针锥底，买股时机。
- 2 三军会师，看好后市。
- 3 双管齐下，买进不怕。
- 4 白龙出水，短线可为。
- 5 五阳上阵，股价弹升。
- 6 身抱多线，好景出现。
- 7 三棒撑地，行情见底。
- 9 三杆通底，后市可喜。
- 10 重锤坠地，后市有戏。
- 11 巨阳入海，放心购买。
- 12 布林破底，短线可期。
- 13. CR扭成团，买进票不商量。
- 14. 天上雨来急，地上亦涨，多头应进场。

### 股票口诀



pcie固态硬盘



PCIE SSD

gSpan频繁子图挖掘算法 (22)

决策分类树算法之ID3, C (16)

18大经典数据挖掘算法小 (15)

Redis源码分析 (三) ---c (13)

Redis源码分析 (二) --结 (11)

记一次DataNode慢启动 (10)

朴素贝叶斯分类算法 (7)

PrefixSpan序列模式挖掘 (7)

Redis源码分析 (三十六) (6)

Hadoop节点"慢磁盘"监控 (6)

## 推荐文章

\* 5月书讯：流畅的Python，终于等到你！

\* JSON最佳实践

\* InfiniBand技术和协议架构分析

\* Android 中解决破解签名验证之后导致的登录授权失效问题

\* 《Real-Time Rendering 3rd》提炼总结——图形渲染与视觉外观

```

41.         if (itemSet.getItems().contains("_")) {
42.             continue;
43.         }
44.
45.         if (s.equals(c)) {
46.             isContained = true;
47.             break;
48.         }
49.     }
50.
51.     if (isContained) {
52.         // 如果已经检测出包含了，直接挑出循环
53.         break;
54.     }
55. }
56.
57. return isContained;
58. }
59.
60. /**
61.  * 判断组合项集是否包含于序列中
62.  *
63.  * @param itemSet
64.  *      组合的项集，元素超过1个
65.  * @return
66.  */
67. public boolean compoentItemIsContain(ItemSet itemSet) {
68.     boolean isContained = false;
69.     ArrayList<String> tempItems;
70.     String lastItem = itemSet.getLastValue();
71.
72.     for (int i = 0; i < this.itemSetList.size(); i++) {
73.         tempItems = this.itemSetList.get(i).getItems();
74.         // 分2种情况查找，第一种从_x中找出x等于项集最后的元素，因
75.         if (tempItems.size() > 1 && tempItems.get(0).equals(
76.             && tempItems.get(1).equals(lastItem)) {
77.             isContained = true;
78.             break;
79.         } else if (!tempItems.get(0).equals("_")) {

```

关闭

### 买入口诀

• 1一针锥底，买股时机。	• 9三杆通底，后市可喜。
• 2三军会师，看好后市。	• 10重锤坠地，后市有戏。
• 3双管齐下，买进不怕。	• 11巨阳入海，放心购买。
• 4白龙出水，短线可为。	• 12布林破底，短线可取。
• 5五阳上阵，股价弹升。	• 13. CR扭成团，买进票不商量。
• 6身抱多线，好景出现。	• 14. 天上雨来急，地上亦涨，多头应进场。
• 7三棒撑地，行情见底。	

### 股票口诀



pcie固态硬盘



## Chameleon两阶段聚类算法

tianwaifeimao: "EC(Ci)表示的Ci聚簇内的边的权重和, EC(Ci,Cj)表示的是连接2个聚簇的边的权重和"这样...

## Hadoop分布式文件系统--HDFS

liuhmmjj: 很赞

## 自定义Hive Sql Job分析工具

骑猪喝咖啡: 楼主, 您好, 在我们一个集群中idone中的临时文件很多时候都不会移动到done目录中, 查看了源码, 发...

## 18大经典数据挖掘算法小结

Guanglew: 这些算法有哪些是工业中用的到的?

## 18大经典数据挖掘算法小结

Android路上的人: @linmufeifei: 抱歉各位, 笔者目前主要精力在大数据Hadoop方向, 目前尚无多余时间和精力...

## Hadoop全链路监控解决方案

Android路上的人: @hqwang4: 你好, BaseValues只是笔者为了将数据写入MySQL的二定义的, 你可以直接将...

## Hadoop全链路监控解决方案

骑猪喝咖啡: 楼主, 您好从github上下载了您的代码, 发现有很多BaseValues定义的常量, 这个文件可以提供...

## 18大经典数据挖掘算法小结

```

80. // 从没有_前缀的项集开始寻找, 第二种为从后面的后缀中找出直接找出连续字符为ab为同一项集的项
      集
81.         if (strArrayContains(tempItems, itemSet.getItems())) {
82.             isContained = true;
83.             break;
84.         }
85.     }
86.
87.     if (isContained) {
88.         break;
89.     }
90. }
91.
92. return isContained;
93. }
94.
95. /**
96.  * 删除单个项
97.  *
98.  * @param s
99.  *      待删除项
100.  */
101. public void deleteSingleItem(String s) {
102.     ArrayList<String> tempItems;
103.     ArrayList<String> deleteItems = new ArrayList<>();
104.
105.     for (ItemSet itemSet : this.itemSetList) {
106.         tempItems = itemSet.getItems();
107.         deleteItems = new ArrayList<>();
108.
109.         for (int i = 0; i < tempItems.size(); i++) {
110.             if (tempItems.get(i).equals(s)) {
111.                 deleteItems.add(tempItems.get(i));
112.             }
113.         }
114.
115.         tempItems.removeAll(deleteItems);
116.     }
117. }

```

关闭

### 买入口诀

- 1 一针锥底, 买股时机。
- 2 三军会师, 看好后市。
- 3 双管齐下, 买进不怕。
- 4 白龙出水, 短线可为。
- 5 五阳上阵, 股价弹升。
- 6 身抱多线, 好景出现。
- 7 三棒撑地, 行情见底。
- 9 三杆通底, 后市可喜。
- 10 重锤坠地, 后市有戏。
- 11 巨阳入海, 放心购买。
- 12 布林破底, 短线可期。
- 13. CR扭成团, 买进票不商量。
- 14. 天上雨来急, 地上亦涨, 多头应进场。

### 股票口诀





```

118.
119.
120.    * 提取项s之后所得的序列
121.    *
122.    * @param s
123.    *         目标提取项s
124.    */
125. public Sequence extractItem(String s) {
126.     Sequence extractSeq = this.copySequence();
127.     ItemSet itemSet;
128.     ArrayList<String> items;
129.     ArrayList<ItemSet> deleteItemSets = new ArrayList<>();
130.     ArrayList<String> tempItems = new ArrayList<>();
131.
132.     for (int k = 0; k < extractSeq.itemSetList.size(); k++) {
133.         itemSet = extractSeq.itemSetList.get(k);
134.         items = itemSet.getItems();
135.         if (items.size() == 1 && items.get(0).equals(s)) {
136.             //如果找到的是单项，则完全移除，跳出循环
137.             extractSeq.itemSetList.remove(k);
138.             break;
139.         } else if (items.size() > 1 && !items.get(0).equals("_")) {
140.             //在后续的多元素项中判断是否包含此元素
141.             if (items.contains(s)) {
142.                 //如果包含把s后面的元素加入到临时字符数组中
143.                 int index = items.indexOf(s);
144.                 for (int j = index; j < items.size(); j++) {
145.                     tempItems.add(items.get(j));
146.                 }
147.                 //将第一位的s变成下标符"_"
148.                 tempItems.set(0, "_");
149.                 if (tempItems.size() == 1) {
150.                     // 如果此匹配为在最末端，同样移除
151.                     deleteItemSets.add(itemSet);
152.                 } else {
153.                     //将变化后的项集替换原来的
154.                     extractSeq.itemSetList.set(k, new ItemSet(tempItems));
155.                 }
156.                 break;

```

关闭







```

157.         } else {
158.             deleteItemSets.add(itemSet);
159.         }
160.     } else {
161.         // 不符合以上2项条件的统统移除
162.         deleteItemSets.add(itemSet);
163.     }
164. }
165. extractSeq.itemSetList.removeAll(deleteItemSets);
166.
167.     return extractSeq;
168. }
169.
170. /**
171.  * 提取组合项之后的序列
172.  *
173.  * @param array
174.  *      组合数组
175.  * @return
176.  */
177. public Sequence extractCompoentItem(ArrayList<String> array) {
178.     // 找到目标项，是否立刻停止
179.     boolean stopExtract = false;
180.     Sequence seq = this.copySequence();
181.     String lastItem = array.get(array.size() - 1);
182.     ArrayList<String> tempItems;
183.     ArrayList<ItemSet> deleteItems = new ArrayList<>();
184.
185.     for (int i = 0; i < seq.itemSetList.size(); i++) {
186.         if (stopExtract) {
187.             break;
188.         }
189.
190.         tempItems = seq.itemSetList.get(i).getItems();
191.         // 分2种情况查找，第一种从_x中找出x等于项集最后的元素，因
192.         if (tempItems.size() > 1 && tempItems.get(0).equals
193.             && tempItems.get(1).equals(lastItem)) {
194.             if (tempItems.size() == 2) {
195.                 seq.itemSetList.remove(i);

```

关闭





集

```

196.         } else {
197.             // 把1号位置变为下标 "_" , 往后移1个字符的位置
198.             tempItems.set(1, "_");
199.             // 移除第一个的 "_" 下划线
200.             tempItems.remove(0);
201.         }
202.         stopExtract = true;
203.         break;
204.     } else if (!tempItems.get(0).equals("_")) {
205.         // 从没有_前缀的项集开始寻找, 第二种为从后面的后缀中找出直接找出连续字符为ab为同一项集的项
206.
207.         if (strArrayContains(tempItems, array)) {
208.             // 从左往右找出第一个给定字符的位置, 把后面的部分截取出来
209.             int index = tempItems.indexOf(lastItem);
210.             ArrayList<String> array2 = new ArrayList<String>();
211.
212.             for (int j = index; j < tempItems.size(); j++) {
213.                 array2.add(tempItems.get(j));
214.             }
215.             array2.set(0, "_");
216.
217.             if (array2.size() == 1) {
218.                 //如果此项在末尾的位置, 则移除该项, 否则进行替换
219.                 deleteItems.add(seq.itemSetList.get(i));
220.             } else {
221.                 seq.itemSetList.set(i, new ItemSet(a
222.             }
223.             stopExtract = true;
224.             break;
225.         } else {
226.             deleteItems.add(seq.itemSetList.get(i));
227.         }
228.     } else {
229.         // 这种情况是处理_x中x不等于最后一个元素的情况
230.         deleteItems.add(seq.itemSetList.get(i));
231.     }
232. }
233. seq.itemSetList.removeAll(deleteItems);

```

关闭

### 买入口诀

• 1一针锥底, 买股时机。	• 9三杆通底, 后市可喜。
• 2三军会师, 看好后市。	• 10重锤坠地, 后市有戏。
• 3双管齐下, 买进不怕。	• 11巨阳入海, 放心购买。
• 4白龙出水, 短线可为。	• 12布林破底, 短线可取。
• 5五阳上阵, 股价弹升。	• 13. CR扭成团, 买进票不商量。
• 6身抱多线, 好景出现。	• 14. 天上雨来急, 地上亦涨, 多头应进场。
• 7三棒撑地, 行情见底。	

### 股票口诀



```

234.
235.         return seq;
236.     }
237.
238.     /**
239.     * 深拷贝一个序列
240.     *
241.     * @return
242.     */
243.     public Sequence copySequence() {
244.         Sequence copySeq = new Sequence();
245.         ItemSet tempItemSet;
246.         ArrayList<String> items;
247.
248.         for (ItemSet itemSet : this.itemSetList) {
249.             items = (ArrayList<String>) itemSet.getItems().clone();
250.             tempItemSet = new ItemSet(items);
251.             copySeq.getItemSetList().add(tempItemSet);
252.         }
253.
254.         return copySeq;
255.     }
256.
257.     /**
258.     * 获取序列中最后一个项集的最后1个元素
259.     *
260.     * @return
261.     */
262.     public String getLastItemSetValue() {
263.         int size = this.getItemSetList().size();
264.         ItemSet itemSet = this.getItemSetList().get(size - 1);
265.         size = itemSet.getItems().size();
266.
267.         return itemSet.getItems().get(size - 1);
268.     }
269.
270.     /**
271.     * 判断strList2是否是strList1的子序列
272.     */

```

关闭





```

273.      * @param strList1
274.      * @param strList2
275.      * @return
276.      */
277.      public boolean strArrayContains(ArrayList<String> strList1,
278.          ArrayList<String> strList2) {
279.          boolean isContained = false;
280.
281.          for (int i = 0; i < strList1.size() - strList2.size() + 1; i++) {
282.              isContained = true;
283.
284.              for (int j = 0, k = i; j < strList2.size(); j++, k++) {
285.                  if (!strList1.get(k).equals(strList2.get(j))) {
286.                      isContained = false;
287.                      break;
288.                  }
289.              }
290.
291.              if (isContained) {
292.                  break;
293.              }
294.          }
295.
296.          return isContained;
297.      }
298.  }

```

ItemSet.java :

```

[java] view plain copy print ?
01.      package DataMining_PrefixSpan;
02.
03.      import java.util.ArrayList;
04.
05.      /**
06.       * 字符项聚类
07.       *
08.       * @author lyq

```

关闭

### 买入口诀

- 1 一针锥底，买股时机。
- 2 三军会师，看好后市。
- 3 双管齐下，买进不怕。
- 4 白龙出水，短线可为。
- 5 五阳上阵，股价弹升。
- 6 身抱多线，好景出现。
- 7 三棒撑地，行情见底。
- 9 三杆通底，后市可喜。
- 10 重锤坠地，后市有戏。
- 11 巨阳入海，放心购买。
- 12 布林破底，短线可期。
- 13. CR扭成团，买进票不商量。
- 14. 天上雨来急，地上亦涨，多头应进场。

### 股票口诀





```

09.  *
10.  */
11.  public class ItemSet {
12.      // 项集内的字符
13.      private ArrayList<String> items;
14.
15.      public ItemSet(String[] str) {
16.          items = new ArrayList<>();
17.          for (String s : str) {
18.              items.add(s);
19.          }
20.      }
21.
22.      public ItemSet(ArrayList<String> itemList) {
23.          this.items = itemList;
24.      }
25.
26.      public ItemSet(String s) {
27.          items = new ArrayList<>();
28.          for (int i = 0; i < s.length(); i++) {
29.              items.add(s.charAt(i) + "");
30.          }
31.      }
32.
33.      public ArrayList<String> getItems() {
34.          return items;
35.      }
36.
37.      public void setItems(ArrayList<String> items) {
38.          this.items = items;
39.      }
40.
41.      /**
42.       * 获取项集最后1个元素
43.       *
44.       * @return
45.       */
46.      public String getLastValue() {
47.          int size = this.items.size();

```

关闭

### 买入口诀

- 1 一针锥底，买股时机。
- 2 三军会师，看好后市。
- 3 双管齐下，买进不怕。
- 4 白龙出水，短线可为。
- 5 五阳上阵，股价弹升。
- 6 身抱多线，好景出现。
- 7 三棒撑地，行情见底。
- 9 三杆通底，后市可喜。
- 10 重锤坠地，后市有戏。
- 11 巨阳入海，放心购买。
- 12 布林破底，短线可期。
- 13. CR扭成团，买进票不商量。
- 14. 天上雨来急，地上亦涨，多头应进场。

### 股票口诀



pcie固态硬盘



```

48.
49.         return this.items.get(size - 1);
50.     }
51. }

```

PrefixSpanTool.java :

[java] view plain copy print ?

```

01. package DataMining_PrefixSpan;
02.
03. import java.io.BufferedReader;
04. import java.io.File;
05. import java.io.FileReader;
06. import java.io.IOException;
07. import java.util.ArrayList;
08. import java.util.Collections;
09. import java.util.HashMap;
10. import java.util.Map;
11.
12. /**
13.  * PrefixSpanTool序列模式分析算法工具类
14.  *
15.  * @author lyq
16.  *
17.  */
18. public class PrefixSpanTool {
19.     // 测试数据文件地址
20.     private String filePath;
21.     // 最小支持度阈值比例
22.     private double minSupportRate;
23.     // 最小支持度，通过序列总数乘以阈值比例计算
24.     private int minSupport;
25.     // 原始序列组
26.     private ArrayList<Sequence> totalSeqs;
27.     // 挖掘出的所有序列频繁模式
28.     private ArrayList<Sequence> totalFrequentSeqs;
29.     // 所有的单一项，用于递归枚举
30.     private ArrayList<String> singleItems;

```

关闭

### 买入口诀

- 1 一针锥底，买股时机。
- 2 三军会师，看好后市。
- 3 双管齐下，买进不怕。
- 4 白龙出水，短线可为。
- 5 五阳上阵，股价弹升。
- 6 身抱多线，好景出现。
- 7 三棒撑地，行情见底。
- 9 三杆通底，后市可喜。
- 10 重锤坠地，后市有戏。
- 11 巨阳入海，放心购买。
- 12 布林破底，短线可期。
- 13. CR扭成团，买进票不商量。
- 14. 天上雨来急，地上亦涨，多头应进场。

### 股票口诀



```

31.
32. public PrefixSpanTool(String filePath, double minSupportRate) {
33.     this.filePath = filePath;
34.     this.minSupportRate = minSupportRate;
35.     readDataFile();
36. }
37.
38. /**
39.  * 从文件中读取数据
40.  */
41. private void readDataFile() {
42.     File file = new File(filePath);
43.     ArrayList<String[]> dataArray = new ArrayList<String[]>();
44.
45.     try {
46.         BufferedReader in = new BufferedReader(new FileReader(file));
47.         String str;
48.         String[] tempArray;
49.         while ((str = in.readLine()) != null) {
50.             tempArray = str.split(" ");
51.             dataArray.add(tempArray);
52.         }
53.         in.close();
54.     } catch (IOException e) {
55.         e.printStackTrace();
56.     }
57.
58.     minSupport = (int) (dataArray.size() * minSupportRate);
59.     totalSeqs = new ArrayList<>();
60.     totalFrequentSeqs = new ArrayList<>();
61.     Sequence tempSeq;
62.     ItemSet tempItemSet;
63.     for (String[] str : dataArray) {
64.         tempSeq = new Sequence();
65.         for (String s : str) {
66.             tempItemSet = new ItemSet(s);
67.             tempSeq.getItemSetList().add(tempItemSet);
68.         }
69.         totalSeqs.add(tempSeq);

```

关闭

### 买入口诀

• 1一针锥底，买股时机。	• 9三杆通底，后市可喜。
• 2三军会师，看好后市。	• 10重锤坠地，后市有戏。
• 3双管齐下，买进不怕。	• 11巨阳入海，放心购买。
• 4白龙出水，短线可为。	• 12布林破底，短线可取。
• 5五阳上阵，股价弹升。	• 13. CR扭成团，买进票不商量。
• 6身抱多线，好景出现。	• 14. 天上雨来急，地上亦涨，多头应进场。
• 7三棒撑地，行情见底。	

### 股票口诀



```

70.     PrefixSpan序列模式挖掘算法 - 走在前往架构师的路上 - 博客频道 - CSDN.NET
71.     }
72.     System.out.println("原始序列数据:");
73.     outputSequence(totalSeqs);
74. }
75.
76. /**
77.  * 输出序列列表内容
78.  *
79.  * @param seqList
80.  *       待输出序列列表
81.  */
82. private void outputSequence(ArrayList<Sequence> seqList) {
83.     for (Sequence seq : seqList) {
84.         System.out.print("<");
85.         for (ItemSet itemSet : seq.getItemSetList()) {
86.             if (itemSet.getItems().size() > 1) {
87.                 System.out.print("(");
88.             }
89.
90.             for (String s : itemSet.getItems()) {
91.                 System.out.print(s + " ");
92.             }
93.
94.             if (itemSet.getItems().size() > 1) {
95.                 System.out.print(")");
96.             }
97.         }
98.         System.out.println(">");
99.     }
100. }
101.
102. /**
103.  * 移除初始序列中不满足最小支持度阈值的单项
104.  */
105. private void removeInitSeqsItem() {
106.     int count = 0;
107.     HashMap<String, Integer> itemMap = new HashMap<>();
108.     singleItems = new ArrayList<>();

```

关闭

### 买入口诀

- 1 一针锥底，买股时机。
- 2 三军会师，看好后市。
- 3 双管齐下，买进不怕。
- 4 白龙出水，短线可为。
- 5 五阳上阵，股价弹升。
- 6 身抱多线，好景出现。
- 7 三棒撑地，行情见底。
- 9 三杆通底，后市可喜。
- 10 重锤坠地，后市有戏。
- 11 巨阳入海，放心购买。
- 12 布林破底，短线可期。
- 13. CR扭成团，买进票不商量。
- 14. 天上雨来急，地上亦涨，多头应进场。

### 股票口诀





```

109.
110.
111.   for (Sequence seq : totalSeqs) {
112.       for (ItemSet itemSet : seq.getItemSetList()) {
113.           for (String s : itemSet.getItems()) {
114.               if (!itemMap.containsKey(s)) {
115.                   itemMap.put(s, 1);
116.               }
117.           }
118.       }
119.
120.   String key;
121.   for (Map.Entry entry : itemMap.entrySet()) {
122.       count = 0;
123.       key = (String) entry.getKey();
124.       for (Sequence seq : totalSeqs) {
125.           if (seq.strIsContained(key)) {
126.               count++;
127.           }
128.       }
129.
130.       itemMap.put(key, count);
131.
132.   }
133.
134.   for (Map.Entry entry : itemMap.entrySet()) {
135.       key = (String) entry.getKey();
136.       count = (int) entry.getValue();
137.
138.       if (count < minSupport) {
139.           // 如果支持度阈值小于所得的最小支持度阈值，则删除该项
140.           for (Sequence seq : totalSeqs) {
141.               seq.deleteSingleItem(key);
142.           }
143.       } else {
144.           singleItems.add(key);
145.       }
146.   }
147.

```

关闭

### 买入口诀

• 1一针锥底，买股时机。	• 9三杆通底，后市可喜。
• 2三军会师，看好后市。	• 10重锤坠地，后市有戏。
• 3双管齐下，买进不怕。	• 11巨阳入海，放心购买。
• 4白龙出水，短线可为。	• 12布林破底，短线可取。
• 5五阳上阵，股价弹升。	• 13. CR扭成团，买进票不商量。
• 6身抱多线，好景出现。	• 14. 天上雨来急，地上亦涨，多头应进场。
• 7三棒撑地，行情见底。	

### 股票口诀



```

148. Collections.sort(singleItems);
149. }
150.
151. /**
152.  * 递归搜索满足条件的序列模式
153.  *
154.  * @param beforeSeq
155.  *       前缀序列
156.  * @param afterSeqList
157.  *       后缀序列列表
158.  */
159. private void recursiveSearchSeqs(Sequence beforeSeq,
160.     ArrayList<Sequence> afterSeqList) {
161.     ItemSet tempItemSet;
162.     Sequence tempSeq2;
163.     Sequence tempSeq;
164.     ArrayList<Sequence> tempSeqList = new ArrayList<>();
165.
166.     for (String s : singleItems) {
167.         // 分成2种形式递归, 以<a>为起始项, 第一种直接加入独立项集遍历<a, a>, <a, b> <a, c>..
168.         if (isLargerThanMinSupport(s, afterSeqList)) {
169.             tempSeq = beforeSeq.copySequence();
170.             tempItemSet = new ItemSet(s);
171.             tempSeq.getItemSetList().add(tempItemSet);
172.
173.             totalFrequentSeqs.add(tempSeq);
174.
175.             tempSeqList = new ArrayList<>();
176.             for (Sequence seq : afterSeqList) {
177.                 if (seq.strIsContained(s)) {
178.                     tempSeq2 = seq.extractItem(s);
179.                     tempSeqList.add(tempSeq2);
180.                 }
181.             }
182.
183.             recursiveSearchSeqs(tempSeq, tempSeqList);
184.         }
185.
186.         // 第二种递归为以元素的身份加入最后的项集内以a为例<(aa)>

```

关闭





```

187. // a在这里可以理解为一个前缀序列，里面可能是单个元素或者已经是多元素的项集
188. tempSeq = beforeSeq.copySequence();
189. int size = tempSeq.getItemSetList().size();
190. tempItemSet = tempSeq.getItemSetList().get(size - 1);
191. tempItemSet.getItems().add(s);
192.
193. if (isLargerThanMinSupport(tempItemSet, afterSeqList)) {
194.     tempSeqList = new ArrayList<>();
195.     for (Sequence seq : afterSeqList) {
196.         if (seq.compoentItemIsContain(tempItemSet)) {
197.             tempSeq2 = seq.extractCompoentItem(tempItemSet
198.                 .getItems());
199.             tempSeqList.add(tempSeq2);
200.         }
201.     }
202.     totalFrequentSeqs.add(tempSeq);
203.
204.     recursiveSearchSeqs(tempSeq, tempSeqList);
205. }
206. }
207. }
208.
209. /**
210.  * 所传入的项组合在所给定序列中的支持度是否超过阈值
211.  *
212.  * @param s
213.  *         所需匹配的项
214.  * @param seqList
215.  *         比较序列数据
216.  * @return
217.  */
218. private boolean isLargerThanMinSupport(String s, ArrayList<Sequence> seqList) {
219.     boolean isLarge = false;
220.     int count = 0;
221.
222.     for (Sequence seq : seqList) {
223.         if (seq.strIsContained(s)) {
224.             count++;
225.         }
226.     }

```

关闭





```

226.     }
227.
228.     if (count >= minSupport) {
229.         isLarge = true;
230.     }
231.
232.     return isLarge;
233. }
234.
235. /**
236.  * 所传入的组合项集在序列中的支持度是否大于阈值
237.  *
238.  * @param itemSet
239.  *         组合元素项集
240.  * @param seqList
241.  *         比较的序列列表
242.  * @return
243.  */
244. private boolean isLargerThanMinSupport(ItemSet itemSet,
245.     ArrayList<Sequence> seqList) {
246.     boolean isLarge = false;
247.     int count = 0;
248.
249.     if (seqList == null) {
250.         return false;
251.     }
252.
253.     for (Sequence seq : seqList) {
254.         if (seq.compoentItemIsContain(itemSet)) {
255.             count++;
256.         }
257.     }
258.
259.     if (count >= minSupport) {
260.         isLarge = true;
261.     }
262.
263.     return isLarge;
264. }

```

关闭

### 买入口诀

- 1 一针锥底，买股时机。
- 2 三军会师，看好后市。
- 3 双管齐下，买进不怕。
- 4 白龙出水，短线可为。
- 5 五阳上阵，股价弹升。
- 6 身抱多线，好景出现。
- 7 三棒撑地，行情见底。
- 9 三杆通底，后市可喜。
- 10 重锤坠地，后市有戏。
- 11 巨阳入海，放心购买。
- 12 布林破底，短线可期。
- 13. CR扭成团，买进票不商量。
- 14. 天上雨来急，地上亦涨，多头应进场。

### 股票口诀





```

265.
266.  /**
267.   * 序列模式分析计算
268.   */
269. public void prefixSpanCalculate() {
270.     Sequence seq;
271.     Sequence tempSeq;
272.     ArrayList<Sequence> tempSeqList = new ArrayList<>();
273.     ItemSet itemSet;
274.     removeInitSeqsItem();
275.
276.     for (String s : singleItems) {
277.         // 从最开始的a, b, d开始递归往下寻找频繁序列模式
278.         seq = new Sequence();
279.         itemSet = new ItemSet(s);
280.         seq.getItemSetList().add(itemSet);
281.
282.         if (isLargerThanMinSupport(s, totalSeqs)) {
283.             tempSeqList = new ArrayList<>();
284.             for (Sequence s2 : totalSeqs) {
285.                 // 判断单一项是否包含于在序列中, 包含才进行提取操作
286.                 if (s2.strIsContained(s)) {
287.                     tempSeq = s2.extractItem(s);
288.                     tempSeqList.add(tempSeq);
289.                 }
290.             }
291.
292.             totalFrequentSeqs.add(seq);
293.             recursiveSearchSeqs(seq, tempSeqList);
294.         }
295.     }
296.
297.     printTotalFreSeqs();
298. }
299.
300. /**
301.  * 按模式类别输出频繁序列模式
302.  */
303. private void printTotalFreSeqs() {

```

关闭

### 买入口诀

• 1一针锥底, 买股时机。	• 9三杆通底, 后市可喜。
• 2三军会师, 看好后市。	• 10重锤坠地, 后市有戏。
• 3双管齐下, 买进不怕。	• 11巨阳入海, 放心购买。
• 4白龙出水, 短线可为。	• 12布林破底, 短线可取。
• 5五阳上阵, 股价弹升。	• 13. CR扭成团, 买进。
• 6身抱多线, 好景出现。	• 14. 天上雨来急, 地上。
• 7三棒撑地, 行情见底。	• 亦涨, 多头应进场。

### 股票口诀



```

304. System.out.println("序列模式挖掘结果：");
305.
306. ArrayList<Sequence> seqList;
307. HashMap<String, ArrayList<Sequence>> seqMap = new HashMap<>();
308. for (String s : singleItems) {
309.     seqList = new ArrayList<>();
310.     for (Sequence seq : totalFrequentSeqs) {
311.         if (seq.getItemSetList().get(0).getItem().get(0).equals(s)) {
312.             seqList.add(seq);
313.         }
314.     }
315.     seqMap.put(s, seqList);
316. }
317.
318. int count = 0;
319. for (String s : singleItems) {
320.     count = 0;
321.     System.out.println();
322.     System.out.println();
323.
324.     seqList = (ArrayList<Sequence>) seqMap.get(s);
325.     for (Sequence tempSeq : seqList) {
326.         count++;
327.         System.out.print("<");
328.         for (ItemSet itemSet : tempSeq.getItemSetList()) {
329.             if (itemSet.getItems().size() > 1) {
330.                 System.out.print("(");
331.             }
332.
333.             for (String str : itemSet.getItems()) {
334.                 System.out.print(str + " ");
335.             }
336.
337.             if (itemSet.getItems().size() > 1) {
338.                 System.out.print(")");
339.             }
340.         }
341.         System.out.print(">, ");
342.

```

关闭

### 买入口诀

• 1一针锥底，买股时机。	• 9三杆通底，后市可喜。
• 2三军会师，看好后市。	• 10重锤坠地，后市有戏。
• 3双管齐下，买进不怕。	• 11巨阳入海，放心购买。
• 4白龙出水，短线可为。	• 12布林破底，短线可取。
• 5五阳上阵，股价弹升。	• 13. CR扭成团，买进票不商量。
• 6身抱多线，好景出现。	• 14. 天上雨来急，地上亦涨，多头应进场。
• 7三棒撑地，行情见底。	

### 股票口诀



```

343.         // 每5个序列换一行
344.         if (count == 5) {
345.             count = 0;
346.             System.out.println();
347.         }
348.     }
349.
350. }
351.
352.
353. }
```

调用类Client.java:

```

[java] view plain copy print ?

01. package DataMining_PrefixSpan;
02.
03. /**
04.  * PrefixSpan序列模式挖掘算法
05.  * @author lyq
06.  *
07.  */
08. public class Client {
09.     public static void main(String[] args){
10.         String filePath = "C:\\Users\\lyq\\Desktop\\icon\\in";
11.         //最小支持度阈值率
12.         double minSupportRate = 0.4;
13.
14.         PrefixSpanTool tool = new PrefixSpanTool(filePath, m);
15.         tool.prefixSpanCalculate();
16.     }
17. }
```

输出的结果：

```

[java] view plain copy print ?

01. 原始序列数据：
```

关闭

### 买入口诀

• 1一针锥底，买股时机。	• 9三杆通底，后市可喜。
• 2三军会师，看好后市。	• 10重锤坠地，后市有戏。
• 3双管齐下，买进不怕。	• 11巨阳入海，放心购买。
• 4白龙出水，短线可为。	• 12布林破底，短线可取。
• 5五阳上阵，股价弹升。	• 13. CR扭成团，买进票不商量。
• 6身抱多线，好景出现。	• 14. 天上雨来急，地上亦涨，多头应进场。
• 7三棒撑地，行情见底。	

### 股票口诀



```

02. <(b d )c b (a c )>
03. <(b f )(c e )b (f g )>
04. <(a h )(b f )a b f >
05. <(b e )(c e )d >
06. <a (b d )b c b (a d e )>
07. 序列模式挖掘结果：
08.
09.
10. <a >, <a a >, <a b >, <a b a >, <a b b >,
11.
12.
13. <b >, <b a >, <b b >, <b b a >, <b b c >,
14. <b b f >, <b c >, <b c a >, <b c b >, <b c b a >,
15. <b c d >, <b (c e )>, <b d >, <(b d )>, <(b d )a >,
16. <(b d )b >, <(b d )b a >, <(b d )b c >, <(b d )c >, <(b d )c a >,
17. <(b d )c b >, <(b d )c b a >, <b e >, <b f >, <(b f )>,
18. <(b f )b >, <(b f )b f >, <(b f )f >,
19.
20. <c >, <c a >, <c b >, <c b a >, <c d >,
21. <(c e )>,
22.
23. <d >, <d a >, <d b >, <d b a >, <d b c >,
24. <d c >, <d c a >, <d c b >, <d c b a >,
25.
26. <e >,
27.
28. <f >, <f b >, <f b f >, <f f >,

```

经过比对，与上述表格中的结果完全一致，从结果中可以看出他的递归顺序正是

## 算法实现时的难点

我在实现这个算法时确实碰到了不少的问题，下面一一列举。

- 1、Sequence序列在判断或者提取单项和组合项的时候，情况少考虑了，还有考
  - 2、递归的顺序在最早的时候考虑错了，后来对递归的顺序进行了调整。
  - 3、在算法的调试时遇到了，当发现某一项出现问题时，不能够立即调试，因为里
- 手算此情况下的前缀，后缀序列，然后自己模拟出1个Seq调试，在纠正extract方

## 我对PrefixSpan算法的理解

关闭

### 买入口诀

• 1一针锥底，买股时机。	• 9三杆通底，后市可喜。
• 2三军会师，看好后市。	• 10重锤坠地，后市有戏。
• 3双管齐下，买进不怕。	• 11巨阳入海，放心购买。
• 4白龙出水，短线可为。	• 12布林破底，短线可取。
• 5五阳上阵，股价弹升。	• 13.CR扭成团，买进票不商量。
• 6身抱多线，好景出现。	• 14.天上雨来急，地上亦涨，多头应进场。
• 7三棒撑地，行情见底。	

### 股票口诀





pcie固态硬盘

实现了这个算法之后，再回味这个算法，还是很奇妙的，一个序列，通过从左往右的扫描，通过各个项集的子集，能够组合出许许多多的序列模式，然后进行挖掘，PrefixSpan通过递归的形式全部找出，而且效率非常高，的确是个很强大的算法。

## PrefixSpan算法整体的特点

首先一点，他不会产生候选序列，在产生投影数据库的时候(也就是产生后缀子序列),他的规模是不断减小的。PrefixSpan采用分治法进行序列的挖掘，十分的高效。唯一比较会有影响的开销就是在构造后缀子序列的过程，专业上的名称叫做构造投影数据库的时候。

顶  
2

踩  
1

上一篇 GSP序列模式分析算法

下一篇 CBA算法---基于关联规则进行分类的算法

关闭

### 相关文章推荐

- 序列模式PrefixSpan算法介绍
- PrefixSpan序列模式挖掘算法
- 数据挖掘中的模式发现（七）GSP算法、SPADE算法..
- 4种序列模式挖掘算法的比较分析
- 机器学习：序列模式挖掘算法
- 序列模式挖掘（A
- 序列模式挖掘——
- 序列模式挖掘算法
- FreeSpan 和 Pref

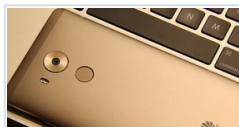
### 买入口诀

• 1一针锥底，买股时机。	• 9三杆通底，后市可喜
• 2三军会师，看好后市。	• 10重锤坠地，后市有戏
• 3双管齐下，买进不怕。	• 11巨阳入海，放心购买
• 4白龙出水，短线可为。	• 12布林破底，短线可取
• 5五阳上阵，股价弹升。	• 13。CR扭成团，买进
• 6身抱多线，好景出现。	• 14。天上雨来急，地上
• 7三棒撑地，行情见底。	• 亦涨，多头应进场。

### 股票口诀



pcie固态硬盘



国产手机前十名



翻盖



架构师



50元的智能手机



二手i5处理器



新款手机排名

## 参考知识库



### Java 知识库

27989 关注 | 3746 收录



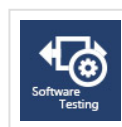
### Java EE知识库

19043 关注 | 1408 收录



### Java SE知识库

26752 关注 | 578 收录



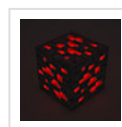
### 软件测试知识库

4890 关注 | 318 收录



### MySQL知识库

22810 关注 | 1581 收录



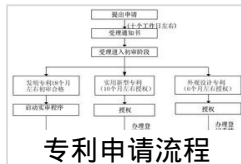
### 算法与数据结构知识库

16800 关注 | 2320 收录

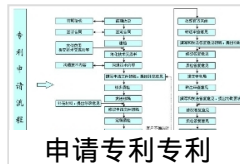
## 猜你在找



努比亚手机怎么样



专利申请流程



申请专利专利

国有银行各期限存款利率(%)				
银行	三个月	六个月	一年	两年
基准利率	1.1	1.3	1.5	2.1
央行	1.35-1.41	1.55-1.69	1.75-2.1	2.25-2.71
农行	1.35-1.41	1.55-1.69	1.75-1.95	2.25-2.71
工行	1.35-1.41	1.5-1.69	1.75-2	2.25-2.71
建行	1.35-1.41	1.55-1.69	1.75-1.95	2.25-2.71

存款利率

## 查看评论

4楼 shanpi6361 2017-03-21 00:02发表

请问这个程序怎么把多个字符识别为序列中的一个元素啊...

关闭

### 买入口诀

- 1一针锥底，买股时机。
- 2三军会师，看好后市。
- 3双管齐下，买进不怕。
- 4白龙出水，短线可为。
- 5五阳上阵，股价弹升。
- 6身抱多线，好景出现。
- 7三棒撑地，行情见底。
- 9三杆通底，后市可喜。
- 10重锤坠地，后市有戏。
- 11巨阳入海，放心购买。
- 12布林破底，短线可取。
- 13.CR扭成团，买进票不商量。
- 14.天上雨来急，地上亦涨，多头应进场。

### 股票口诀



pcie固态硬盘



3楼 [abcdefgh123321](#) 2016-05-02 16:35发表



想问下 每个频繁序列的支持度 想计算, 怎么改代码啊

2楼 [baidu\\_31887577](#) 2016-04-16 23:46发表



您好, 请问这个程序是在哪个软件运行的?

Re: [Android路上的人](#) 2016-04-17 20:22发表



回复baidu\_31887577: 在Eclipse上跑就可以了, 简单的java程序

1楼 [abcdefgh123321](#) 2016-03-22 16:57发表



您好, 想请问下最大项集个数您没有设定么? 没有找到相关的参数。

Re: [Android路上的人](#) 2016-03-22 19:44发表



回复abcdefgh123321: 好像没有设置吧, 这个程序写的比较久了, 你可以再仔细查查我的程序。

Re: [abcdefgh123321](#) 2016-03-23 11:33发表



回复Android路上的人: 好的 谢谢楼主 我试着加加参数

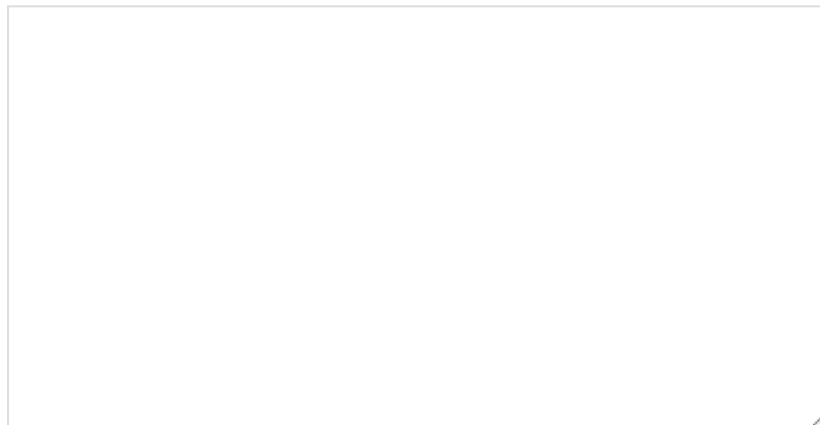
#### 发表评论

用户名: haijunz

评论内容:

关闭





提交

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

#### 核心技术类目

全部主题   Hadoop   AWS   移动游戏   Java   Android   iOS   Swift   智能硬件   Docker   OpenStack   VPN  
 Spark   ERP   IE10   Eclipse   CRM   JavaScript   数据库   Ubuntu   NFC   WAP   jQuery   BI   卜  
 Spring   Apache   .NET   API   HTML   SDK   IIS   Fedora   XML   LBS   Unity   Splashtop   UML  
 components   Windows Mobile   Rails   QEMU   KDE   Cassandra   CloudStack   FTC   coremail   OPh  
 CouchBase   云计算   iOS6   Rackspace   Web App   SpringSide   Maemo   Compuware   大数据   aptech  
 Perl   Tornado   Ruby   Hibernate   ThinkPHP   HBase   Pure   Solr   Angular   Cloud Foundry   Redi:  
 Scala   Django   Bootstrap

关闭

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服

杂志客服

微博客服

webmaster@csdn.net

400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved



### 买入口诀

- 1 一针锥底，买股时机。
- 2 三军会师，看好后市。
- 3 双管齐下，买进不怕。
- 4 白龙出水，短线可为。
- 5 五阳上阵，股价弹升。
- 6 身抱多线，好景出现。
- 7 三棒撑地，行情见底。
- 9 三杆通底，后市可喜。
- 10 重锤坠地，后市有戏。
- 11 巨阳入海，放心购买。
- 12 布林破底，短线可期。
- 13. CR扭成团，买进票不商量。
- 14. 天上雨来急，地上亦涨，多头应进场。

### 股票口诀