浙江大学

题目	股票分类
课程名称	模式识别与机器学习
指导老师	韦巍、项基
姓名学号	庄周 21610152
年级专业	2016 级电气工程硕士
所在学院	电气工程学院

目录

股票分类

1. 分类要求

上证股票分类，对 3000 多个股票设计分类器，类别小于 20 个，并分析分类后的均值，方差，以及类别间的差别。（数据中的空文档和时间日期不对的文档可以忽略不在算法的考虑样本范围内）

1.1 股票数据初步分析

在给出的股票数据文档中，每只股票共有 7 个维度的信息，包括日期，以及在日期当天对应的开盘价、最高价、最低价、收盘价、交易量以及成交额。由于每一股票的历史交易日期天数不尽相同，并且数据量比较大，我们需要首先选择合适的特征来表征每一只股票。由于一天内的数据量比较少，只有四个，并且以每一天的数据作为一维特征，那么特征的维数就是日期天数，显然这么做既会使维数无限增加，而且不同股票的交易天数并不一样，这使得将每一天的数据作为一维变量的做法并不可取。联想到股票的 K 线图，我们自然而然想到了分别用开盘价、收盘价随日期变化的曲线特征作为我们对于股票分类的依据。值得一提的是，股票的日震荡幅度是一个很重要的特征，生活经验告诉我们，一些股票的每日振荡很小，比如像中石化、中国银行这种大盘股，而有些股票，比如暴风科技这样的小盘股往往震荡得比较剧烈，所以股票的日震荡幅度随日期变化的曲线可以作为我们对股票进行分类的依据。而成交量与成交额反映了股票交易的活跃程度，其随日期变化的曲线也可以作为我们分类的特征。

仔细观察数据文档，我们发现每个文档的数据量都不相同，这意味着在我们绘制变量随时间变化的曲线时候，在时间维度上会产生很大的差异，有的股票具有一百多天的交易信息，而有的股票具有一千多天的交易信息，那么在不同时间长度上比较不同股票的趋势变化并不合适，不过幸运的是，大多数股票具有相近的起始日期，在剔除掉部分空文档以及时间错误的文档之后，我们可以选择起始日期之后的一段时间作为我们最终的数据来源。

1.2 分类方法初步分析

面对股票交易价格和交易量的时间序列，每一只股票并没有给出特定的标签，我们希望将这些股票进行分类，这是一个无监督分类问题，而目前对于此类无监督分类问题，我们通常使用的是聚类的方法。

聚类分析（Clustering Analysis）是将数据对象的集合分组由类似的对象组成的多个类、组或者簇这样的过程。对一组无序的没有分出类别的样本集合，按照样本之间的相似程度进行分类，这个过程称为聚类，就是无监督分类。在聚类方法中又有 K 均值聚类，ISODATA 聚类，分层聚类以及模糊聚类等。

聚类分析作为数据挖掘的一种方式，其中的一个最为重要的步骤就是获取数据，包括对数据进行筛选、整理、规范化和预处理，也就是合理处理待分类对象的特征。本文将用不同的特征提取方法和分类方法对于股票进行分类。

2. 股票特征提取方法

2.1 股票序列拟合的特征提取

对于上述提到的股票各个随日期变化的变量，我们对于这样的时间序列，可以采用拟合的方式，提取出数据的趋势性，我们选择了每个交易日的开盘价、收盘价、股票震荡（最高价与最低级的差值除以前一日的收盘价）以及成交量作为我们的拟合对象，并将获得拟合函数的系数作为刻画各只股票的特征向量。

2.1.1 拟合函数选择

在股票交易时间序列的拟合中我们选择了多项式拟合和高斯拟合函数，并且分析了在不同阶数下的拟合效果。我们以上证 SH600677 股票从 3 月 2 日至 12 月 21 日的开盘价格时间序列为例，研究股票交易时间序列在不同拟合函数下拟合效果。由于篇幅的缘故，我将这些函数拟合出来的实际函数，以及拟合效果的评价指标在附录一中给出。

1. 多项式函数拟合

1） 上证 SH600677 一阶多项式 $y = a_1 x + a_0$ 的拟合效果如图 2-1 所示。

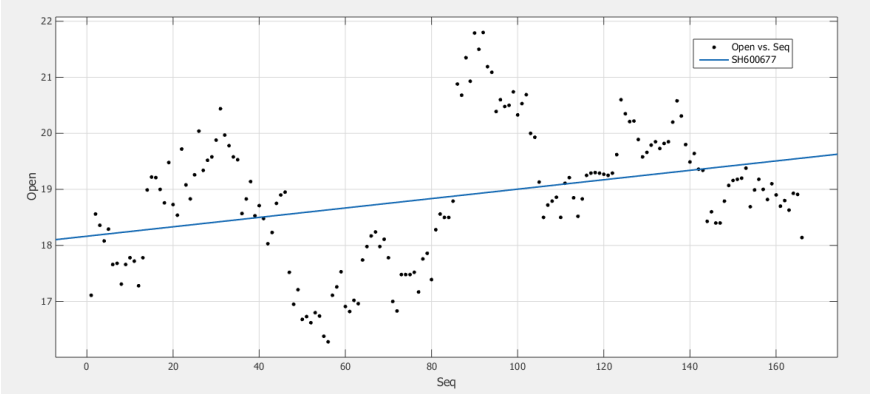


图 2-1 上证 SH600677 一阶多项式拟合图

2） 上证 SH600677 三阶多项式 $y = a_3 x^3 + a_2 x^2 + a_1 x + a_0$ 拟合效果如图 2-2 所示。

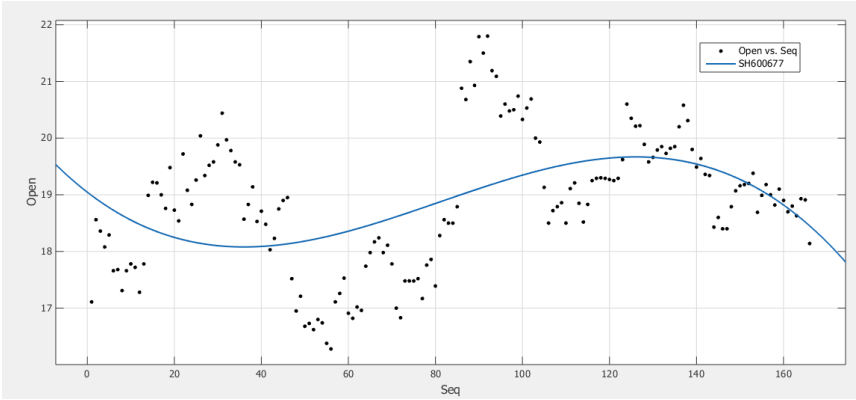


图 2-2 上证 SH600677 三阶多项式拟合图

3） 上证SH600677 五阶多项式 $y = a_5 x^5 + a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0$ 拟合效果如图 2-3 所示。

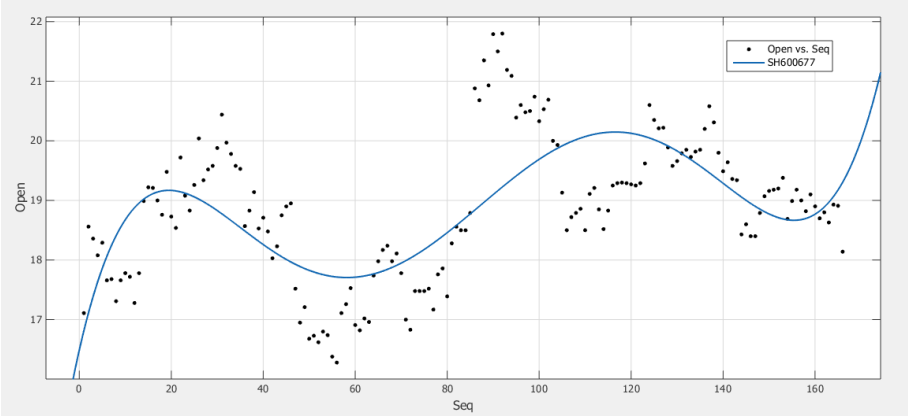


图 2-3 上证 SH600677 五阶多项式拟合图

2. 高斯函数拟合

1) 上证 SH600677 一阶高斯 y= a1*exp(-((x-b1)/c1)^2)拟合效果如图 2-4 所示。

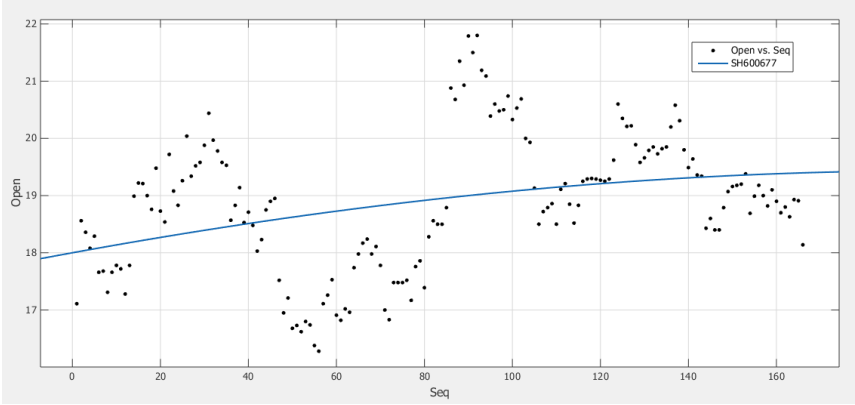


图 2-4 上证 SH600677 一阶高斯函数拟合图

2) 上证 SH600677 三阶高斯 y= a1*exp(-((x-b1)/c1)^2) + a2*exp(-((x-b2)/c2)^2) + a3*exp(-((x-b3)/c3)^2)拟合效果如图 2-5 所示。

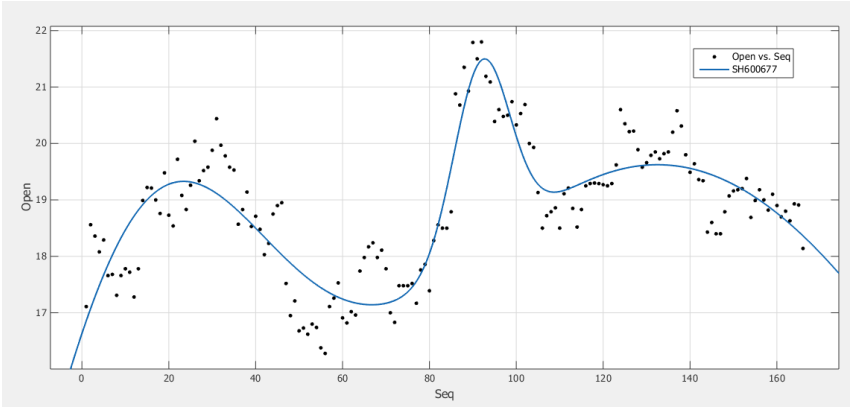


图 2-5 上证 SH600677 三阶高斯函数拟合图

3) 上证 SH600677 五阶高斯 a1*exp(-((x-b1)/c1)^2) + a2*exp(-((x-b2)/c2)^2) + a3*exp(-((x-b3)/c3)^2) + a4*exp(-((x-b4)/c4)^2) + a5*exp(-((x-b5)/c5)^2)拟合效果如图 2-6 所示。

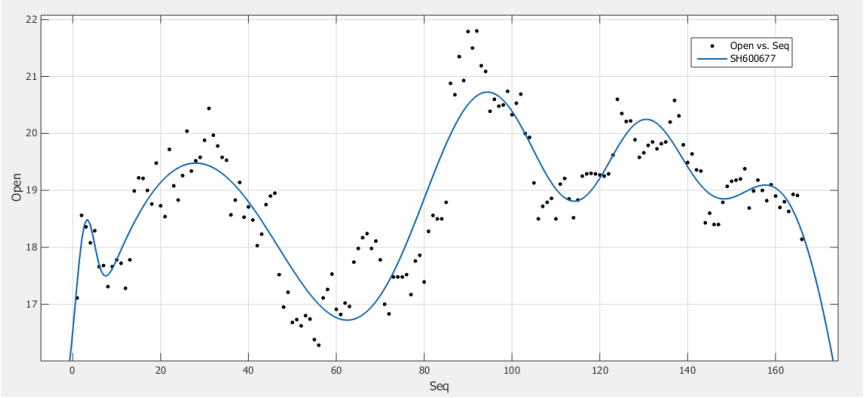


图 2-6 上证 SH600677 五阶高斯函数拟合图

2.1.2 拟合结果分析

由多项式函数和高斯函数的拟合结果，我们可以发现使用三阶高斯函数可以基本准确描述股票交易曲线，并且其系数项的数量也不太多，这使得我们之后的特征向量空间的维数大大降低。

2.1.3 股票特征提取

在剔除掉部分空文档以及时间错误的文档之后，我们可以选择所有股票起始日期之后的 100 个交易日作为我们最终的数据来源。

我们首先采用三阶高斯拟合，获得各个股票，开盘，收盘，日震荡以及交易量随时间变化的曲线的高斯拟合函数系数。将各种股票的高斯系数作为股票的特征值，即使用股票开盘价，收盘价以及震荡幅度还有交易量分别对应的拟合函数系数 a1,b1,c1,a2,b2,c2,a2,b3,c3。总共为 9*4=36 维的特征向量来表征一只股票。

2.1.1 股票特征归一化

下图是开盘价格拟合函数对应的系数矩阵，我们发现存在很多大数，为了避免这些极端数字的影响，我们采用对数函数转换的方式,将系数矩阵进行对数归一化，我们采用反正切函数来进行归一化操作，即 y=arctan(x)，将系数矩阵归一化。图 2-7 为归一化之前特征向量矩阵，图 2-8 为归一化之后特征向量矩阵。

	1	2	3	4	5	6	7	8	9
1	15.7370	0.6612	1.2853	78.4072	32.1804	10.0718	240.2949	7.8260	14.1373
2	87.6081	0.3546	9.0526	412.5215	29.8358	5.7134	214.9521	4.9562	81.1524
3	8.1872	3.0210	6.3262	89.3729	45.8173	-15.0509	61.3983	16.6088	44.9064
4	3.6005	1.7391	462.0333	89.7000	56.9121	3.7771e+03	17.5024	9.3103	2.0286e+03
5	3.9901	2.3298	2.0266e+14	93.8068	22.7782	-758.1574	72.4281	32.8654	133.3428
6	2.3931	1.3401	27.9455	96.9695	22.6060	19.9531	7.6059	10.0813	213.8936
7	242.9436	0.2605	1.4373	1.9004e+03	36.0400	-25.0525	850.9431	8.8312	90.6337
8	1.9597	-0.8811	7.8260	98.8263	48.6186	38.9216	16.8530	17.5707	91.9839
9	17.2944	2.4944	4.3146	182.1020	34.4039	5.2792	228.3069	7.9390	25.0651
10	10.8620	4.7958	0.4639	160.8896	6.7481	7.0939	162.9744	60.8598	10.7572
11	16.5417	1.1416	1.5523	577.8968	38.5587	8.5048	626.6255	19.3932	14.9198
12	0.3858	0.7189	4.3608	95.4219	27.7096	5.8774	8.6206	10.4615	260.7058
13	3.9131e+11	0.4661	0.3632	4.0570e+04	27.6256	9.5815	8.0852e+03	8.8452	9.2977
14	4.7608e+04	0.4975	1.5064	7.7755e+03	34.2237	6.1941	2.5200e+03	12.8482	38.0974
15	20.3244	0.3320	1.9285	567.8038	35.7368	-14.6065	313.8351	10.5985	86.1805
16	5.2430	0.6948	1.3301	106.8085	30.7877	-1.1464	184.3079	17.0864	24.7499
17	7.4588	0.6677	4.1411	164.9497	32.6983	3.3586	149.8401	3.3900	55.2060
18	5.1087	0.7674	1.4067	117.9125	25.9519	-17.7052	241.8332	18.1863	30.1571
19	17.3860	0.4311	0.1961	2.5129e+03	32.0700	7.2241	2.1183e+03	12.2732	8.3411
20	2.1401	7.0104	2.1032	96.2694	68.2392	0.0735	8.2142	85.2402	25.1069
21	16.8664	2.8862	6.1722e+14	173.8052	25.1208	-4.3234e+03	221.3691	18.5168	759.8968
22	5.4353	0.4123	3.3303	121.5508	34.9440	5.2957	117.2339	10.2075	46.0366

图 2-7 归一化前特征向量矩阵

经过反正切函数归一化之后。

1	2	3	4	5	6	7	8	9
1.4848	1.4544	1.5708	1.5598	1.5328	-1.5701	1.5451	1.5433	1.5668
1.3400	0.5118	1.4062	1.5619	1.5371	-1.4049	1.5514	1.5156	1.5624
1.3659	1.3235	1.5659	1.5602	1.5392	-1.5706	1.5145	1.5037	1.5706
1.3194	1.2595	1.5290	1.5601	1.5301	-1.5677	1.4520	1.5076	1.5700
0.7141	1.2054	1.5708	1.5598	1.5215	1.5707	1.4083	1.5311	1.5704
1.4792	1.4144	1.4481	1.5595	1.5377	-1.4980	1.5539	1.5200	1.5411
1.4171	1.2475	1.5038	1.5627	1.5313	-1.5698	1.5478	1.5145	1.5703
1.5708	1.2400	-1.2768	-1.5708	1.5395	-1.4324	1.5706	1.4762	1.5425
0.5512	0.5732	1.5708	1.5646	1.5417	1.5708	1.5480	1.4455	1.5707
1.4930	1.3532	1.5708	1.5610	1.5352	-1.5707	1.5588	1.5184	1.5700
1.3198	0.7154	1.5616	1.5633	1.5432	-1.5691	1.5602	1.5024	1.5679
1.4801	1.3717	1.5708	1.5607	1.5349	-1.5700	1.5592	1.5241	1.5662
1.5627	1.5350	1.5708	1.5663	1.4875	-1.5683	1.5599	1.5605	1.5564
1.3428	1.0855	1.5708	1.5593	1.5295	-1.5690	1.5596	1.5301	1.5604
1.3834	1.2293	1.3888	1.5599	1.5426	-1.5241	1.5448	1.5421	1.5546

图 2-8 归一化后特征向量矩阵

2.2 股票序列的时频域和混沌特征提取

由于我们分类的对象是股票，从股票交易情况的长时间序列的性质和特点出发，选择时间序列的时域和频域特性，我们一共选择了以下特性进行统计，分别为趋势、序列相关性（自相关和偏相关）、幅值平方、方差、峰谷、偏度、混沌性。

2.2.1 趋势项

趋势项是时间序列的一般特征，是股票交易时间序列的一个非常直观的特征。我们这里将趋势项定义为股票交易时间序列（包括开盘价、收盘价、震荡幅度和成交量）的一次函数拟合的一次项系数，其反应股票长期增长趋势，如图 2-9 所示。

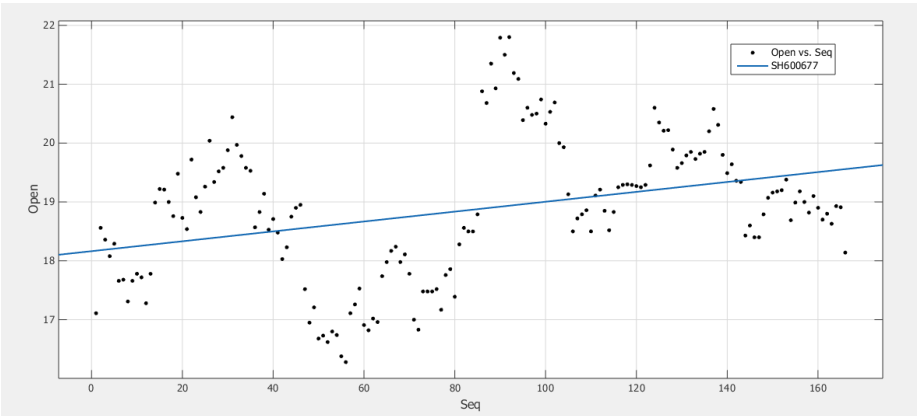


图 2-9 股票价格趋势

2.2.2 相关性

统计学中用相关系数来描述变量之间的相关性，即两个变量之积的数学期望，称为相关性，它表征了这两个变量之间的关联程度。在时间序列中，相关系数描述了两个平稳时间序列或者一个平稳时间序列自身不同时刻的相似程度，通过相关性分析可以发现时间序列中许多规律的信息。自相关性是指一个时间序列某个时间 t 时刻的值与延迟 k 时刻后的同一时间序列在 t+k 时刻的值的相关程度。

相关系数的定义为：

$$r_{XY}=\frac{\sum_{i=1}^{N}(X_i-\overline{X})(Y_i-\overline{Y})}{\sqrt{\sum_{i=1}^{N}\left(X_i-\overline{X}\right)^2}\sqrt{\sum_{i=1}^{N}\left(Y_i-\overline{Y}\right)^2}} \tag{2-1}$$

我们可以计算股票间的相关系数，可以获得如图 2-10 的系数矩阵。

	1	2	3	4	5	6	7	8
1	1	0.1367	-0.1314	0.0374	0.0708	0.0077	0.1287	0.2163
2	0.1367	1	0.0837	0.5085	0.6124	-0.0322	0.2458	0.5126
3	-0.1314	0.0837	1	0.5927	-0.1541	-0.6695	-0.3097	-0.5398
4	0.0374	0.5085	0.5927	1	-0.0567	-0.8038	-0.4454	-0.2353
5	0.0708	0.6124	-0.1541	-0.0567	1	0.4475	0.5971	0.6174
6	0.0077	-0.0322	-0.6695	-0.8038	0.4475	1	0.5545	0.5349
7	0.1287	0.2458	-0.3097	-0.4454	0.5971	0.5545	1	0.5833
8	0.2163	0.5126	-0.5398	-0.2353	0.6174	0.5349	0.5833	1

图 2-10 相关性系数矩阵

2.2.3 偏度

表征概率分布密度曲线相对于平均值不对称程度的特征数。直观看来就是密度函数曲线尾部的相对长度。对于单变量时间序列，其偏度的系数的计算公式为：

$$S=\frac{1}{n\sigma^3}\sum_{t=1}^{n}(x_t-\overline{x_t})^3 \tag{2-2}$$

这里 $\overline{x_t}$ 为时间序列的均值，σ 为标准偏差，n 为数据点的个数。

2.2.4 峰度

又称峰态系数。表征概率密度分布曲线在平均值处峰值高低的特征数。直观看来，峰度反映了峰部的尖度，其计算公式为：

$$K=\frac{1}{n\sigma^4}\sum_{t=1}^{n}(x_t-\overline{x_t})^4 \qquad (2\text{-}3)$$

这里 $\overline{x_t}$ 为时间序列的均值，σ 为标准偏差，n 为数据点的个数。

方差、幅值平方和、峰值与谷值概念比较直观，这里不做赘述。

2.2.5 分形维数计算

分维的概念，在欧氏空间中，人们习惯把空间看成三维的，平面或球面看成二维，而把直线或曲线看成一维，也可以稍加推广，认为点是零维的，还可以引入更高维空间，如把时间加进空间，就是四维，通常人们习惯于整数的维数。分形理论把维数视为分数，这类维数是物理学家在研究混沌吸引子等理论时需要引入的重要概念，为了定量地描述客观事物的“非规则”程度，1919 年，数学家从测度的角度引入了维数概念，将维数从整数扩大到分数，从而突破了一般拓扑集维数为整数的界限。

运用混沌分形方法与理论研究股票市场波动，已成为目前金融学的前沿领域，通过对股票的价格变动的分形维数计算，我们可以对股票波动复杂程度进行定量的判断，从而获取股票波动性的特征。

计盒维数法是一种常用的计算分形图形分维数的实用方法。取边长为 r 的小盒子，把分形曲线覆盖起来。则有些小盒子是空的，有些小盒子覆盖了曲线的一部分。计数多少小盒子不是空的，所得的非空盒子数记为 N（r）。然后缩小盒子的尺寸，所得 N（r）自然要增大，当 r→0 时，得到分形维数为

$$D=-\lim_{r\to 0}\frac{\log N(r)}{\log(r)} \qquad (2\text{-}4)$$

2.3 股票特征提取总结

本节主要讨论了股票特征的选择和提取方法，我们首先提出了股票序列拟合的特征提取方法，即通过多项式函数与高斯函数拟合股票交易曲线的方式，将拟合系数向量作为股票的特征向量。其次，我们又从股票交易曲线的实际意义出来，讨论了使用股票本身的统计特征，即股票的趋势、序列相关性（自相关和偏相关）、幅值平方、方差、峰谷、偏度、混沌性等统计值，作为股票的特征向量。上述两种特征提取的方式，将在后续的分类中分别使用。

3. 股票分类方式

3.1 K-means 聚类

K-means 聚类算法。也称为 k-平均或 k-均值聚类算法，是一种得到最广泛使用的聚类算法。它将各个聚类子集内的所有数据样本的均值作为该聚类（簇）的代表点，算法的主要思想是通过迭代过程把数据基划分为不同的类别，使得评价聚类性能的准则函数达到最优，从而使得生成的每个聚类内紧凑，类之间独立。

3.1.1 K-means 算法

输入：簇的数目 k 和包含 n 个对象的数据库

输出：k 个簇，使平方误差准则最小。

- 算法步骤：

1) 为每一个聚类确定一个初始聚类中心，这样就有 k 个初始聚类中心。
2) 将样本按照最小距离原则分配到最邻近聚类。
3) 使用每个聚类中的样本均值作为新的聚类中心。
4) 重复步骤 2 和 3 直到聚类中心不再变化。
5) 结束，得到 k 个聚类。

3.1.2 K-means 聚类簇数目的确定

由于我们在股票分类前并不知道分类的数量，其类型数目假定已知为 C。对于 C 未知时，可以令 C = 1, 2,……逐渐增加，使用 K-means 算法，误差平方和随 C 的增加而单调减少。最初，由于 C 较小，类型的分裂会使误差平方和迅速减小，但当 C 增加到一定数值时，误差平方和的减小速度会减慢，直到 C = N 时，误差平方和为 0。

3.1.3 K-means 聚类结果分析

我们首先分析股票单个开盘价、收盘价、震荡幅度以及交易量函数拟合系数作为特征向量输入之后的聚类结果，然后分析综合输入后的聚类结果，最后使用主成分法，查看特征向量降维之后的输入的聚类结果。

在只使用开盘价（如图 3-1）、收盘价（如图 3-2）、震荡幅度（如图 3-3）以及交易量函数（如图 3-4）拟合曲线系数作为特征情况下的分类情况，我们可以看到一开始分类的误差平方和随着聚类中心数量的增加快速减少，当聚类簇数增加到 9 类以上，误差平方和的变化基本平缓下来。

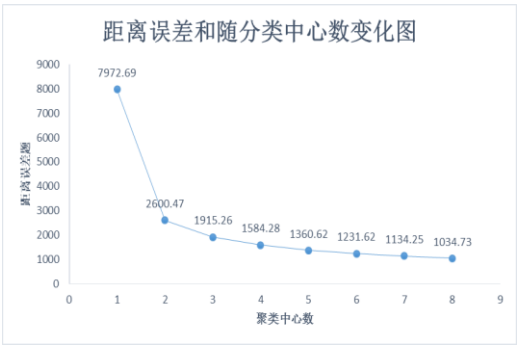


图 3-1 开盘价聚类结果

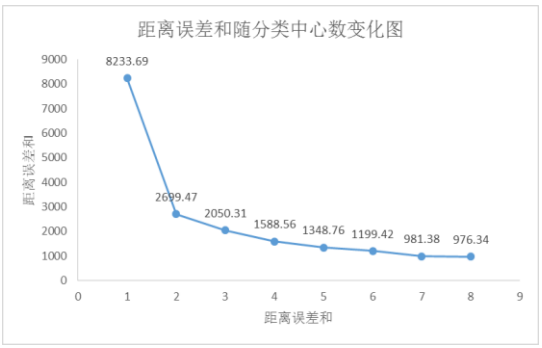


图 3-2 收盘价聚类结果

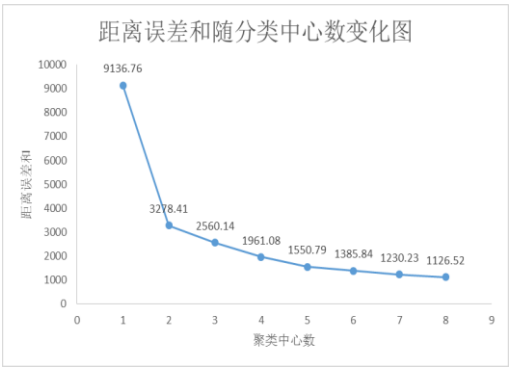


图 3-3 震荡幅度聚类结果

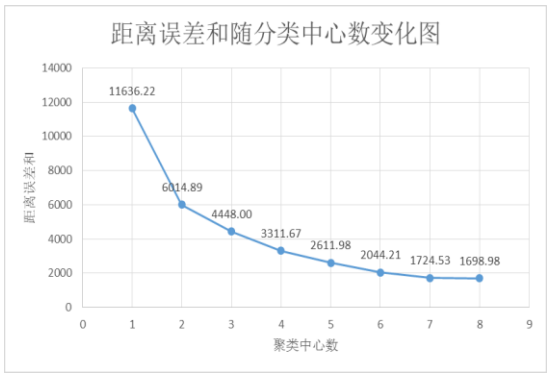


图 3-4 交易量聚类结果

在上述以开盘价，收盘价，日震荡幅度和交易量拟合函数系数聚类的基础上，我们把四类进行合并，构成一个具有 36 维特征向量的矩阵，并重新进行聚类，发现在聚类中心数量增加到 9 类以上的时候，误差平方和曲线基本平缓下来。

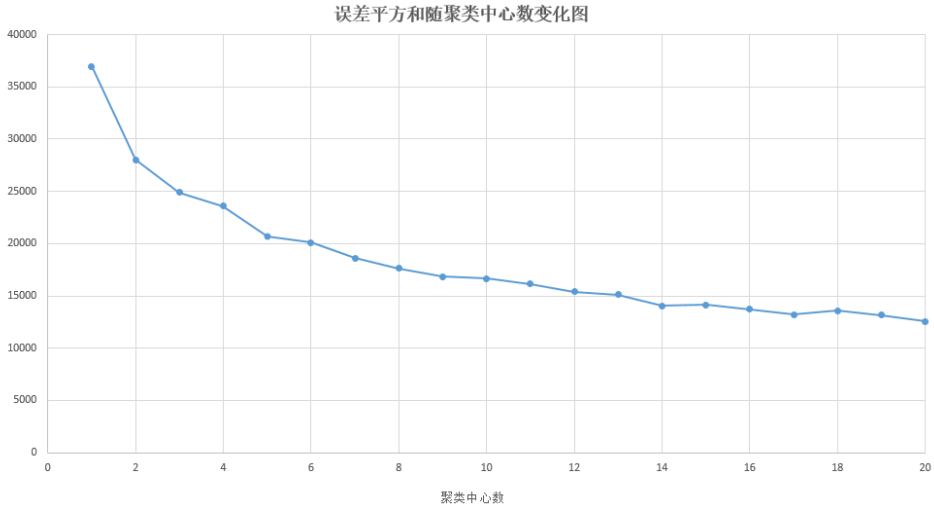


图 3-5 综合拟合系数特征值聚类结果

过多的维数使得我们在进行分类后的均值，方差，以及类别间的差别分析的时候，变得非常麻烦，我们进一步采用主成分分析的方法，进行降维，并且观察在特征向量降维之后，对于分类效果产生了哪些影响。

我们采用主成分分析（PCA）对于之前的 36 维特征向量进行降维处理，通过 PCA 分析，我们发现该 36 维特征向量中的前两维占据了 50%的比重，我们选择前两维特征向量进行 k-means 聚类，误差平方和随聚类簇数变化结果如图 3-6 所示，当聚类簇增加到 9 之后，误差平方和变化基本稳定，聚类簇数为 9 时股票分布结果如图 3-7 所示。

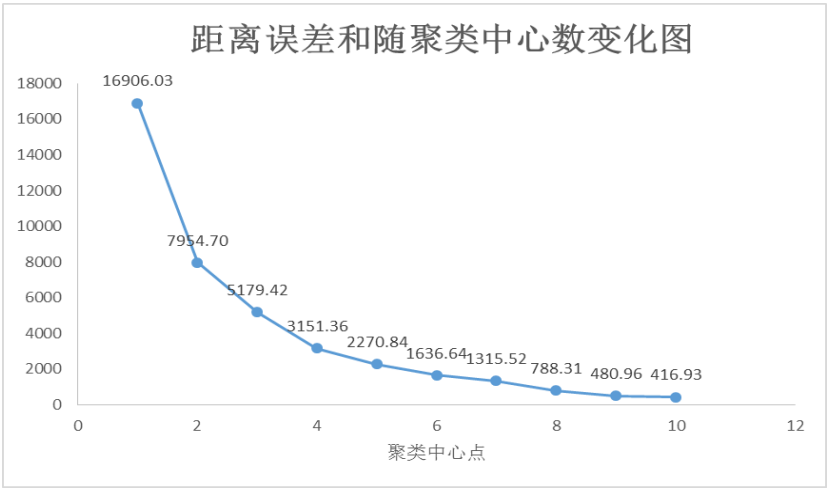


图 3-6 误差平方和随聚类中心数变化曲线

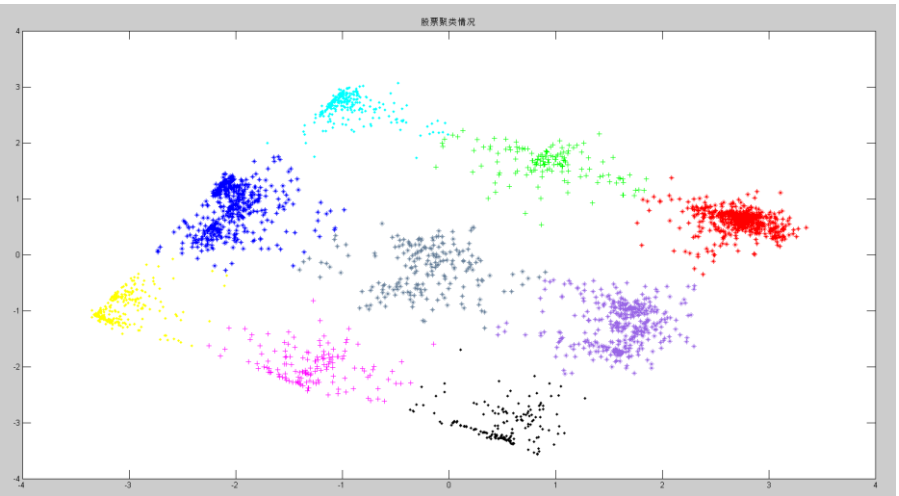


图 3-7 PCA 方法聚类结果

- 计算均值和方差。

每一类的平均值如表 3-1 所示。

表 3-1 聚类簇类平均值

类别	第一维分量均值	第二维分量均值
1	2.7272	0.5957
2	0.9194	1.6230
3	0.5111	-3.0115
4	-3.0921	-0.9683
5	-2.0259	0.8250
6	-1.2745	-1.9946
7	-0.9236	2.6569
8	-0.2228	-0.2623
9	1.6159	-1.2537

每一类的方差如表 3-2 所示。

表 3-2 聚类簇内方差

类别	类误差平方和	类中点数	占比	簇内方差
1	55.7537	564	21.32%	0.098854
2	36.6800	144	5.44%	0.254722
3	31.8206	166	6.28%	0.19169
4	34.2956	308	11.64%	0.111349
5	104.8858	470	17.77%	0.223161
6	33.7694	151	5.71%	0.223638
7	28.5968	255	9.64%	0.112144
8	64.8775	202	7.64%	0.321176
9	90.2776	385	14.56%	0.234487

- 计算分类的类间差别

类别间的差别我们采用 Silhouette 系数来进行度量， Silhouette 系数，即轮廓系数是对聚类结果有效性的解释和验证，由 Peter J. Rousseeuw 于 1986 提出。

轮廓系数算法：

1) 计算样本到同簇其他样本的平均距离。平均距离越小，说明样本越应该被聚类到该簇。将平均距离称为样本的簇内不相似度。簇 C 中所有样本的平均距离均值称为簇 C 的簇不相似度。

2) 计算样本到其他某簇的所有样本的平均距离，称为样本与簇的不相似度。定义为样本的簇间不相似度，样本到其他某簇的所有样本的平均距离的最小值越大，说明样本越不属于其他簇。

3) 根据样本 i 的簇内不相似度 ai 和簇间不相似度 bi ，定义样本 i 的轮廓系数：

$$s(i)=\frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \qquad s(i)=\begin{cases}1-\dfrac{a(i)}{b(i)}, & a(i)<b(i)\\ 0, & a(i)=b(i)\\ \dfrac{b(i)}{a(i)}-1, & a(i)>b(i)\end{cases}$$

4) 判断：

- si 接近 1，则说明样本 i 聚类合理；
- si 接近-1，则说明样本 i 更应该分类到另外的簇；
- 若 si 近似为 0，则说明样本 i 在两个簇的边界上。

所有样本的 si 的均值称为聚类结果的轮廓系数，是该聚类是否合理、有效的度量。

根据轮廓系数的算法，我们获得了上述分类的各个类别的 Silhouette 系数分布图，如图 3-8 所示。我们可以看出第 1 类，第 4 类，第 7 类的分类效果最好，基本系数都接近 1，而第 5 类，第 8 类，第 9 类的系数也基本上处于 0.8 以上，整体看来，分类效果较好。整体轮廓系数为 0.8692，属于分类情况较好的情况。

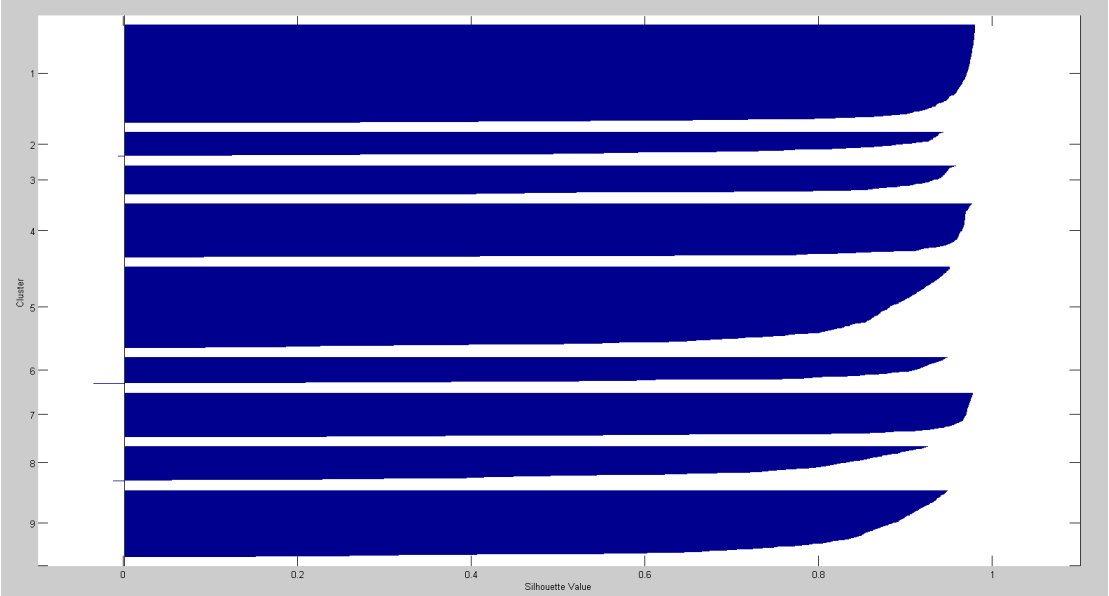


图 3-8 分类后各类轮廓系数

3.1.4 K-means 聚类总结

在使用 K-means 方法对股票进行分类的过程中，我们利用了三阶高斯函数的拟合系数作为股票的特征，在分类前，先对这些股票的特征进行了反正切归一化处理，然后采用 K-means 聚类，每一次固定聚类中心数量之后，改变初始输入位置，多次计算，选择其中平方误差和最小的一次聚类，作为结果输出。在确定聚类中心数 K 的时候，我采用了 Elbow 方法，观察误差平方和随聚类中心数的变化曲线，来确定 K 值，而对于转折不明显的变化曲线，我们采用了轮廓系数来比较分类结果，发现在使用主成分（PCA）法降维之后，当分类数为 9 的时候，分类效果比较好。

但是由于 K-means 算法，对于边界点的处理非此即彼，这意味着一个数据集点只能属于一个簇，这样的硬分类对于股票这类高维特征的时间序列的描述并不准确，我们在接下来使用模糊聚类的方法，尝试从数据集中生成有重叠的簇。在

模糊聚类中任何点都属于不止一个簇，而且该点到这些簇之间都有一定大小的吸引度。这种吸引度与该点到这个簇中心距离成比例。

3.2 FCM 模糊聚类

3.2.1 FCM 算法

传统的聚类分析是一种硬划分，它把每个待识别的对象严格地划分到某类之中，具有“非此即彼”的性质，因此这种类别的界限是分明的。而实际上大多数对象没有严格的属性，它们在性态和类属方面存在着中介性，因此适合于软划分。模糊集理论的提出为这种软划分提供了有力的工具。由于模糊聚类得到了样本属于各个类别的不确定程度，表达了样本类属的中介性，即建立起了样本对于类别的不确定性描述，更能反映现实世界。

FCM 算法的基本思想：在之前聚类的基础之上，引入了样本属于不同类别的隶属度函授矩阵（又称为模糊划分矩阵）和模糊系数 m。从而数据的分类，类心的计算，以及目标函数都较之前一般的聚类算法进行了修整。

- FCM 算法步骤如下：

初始化：给定聚类类别数 C，$2 \le C \le N$，N 是数据个数，设定迭代停止的阈值ε，初始化聚类原型模型P^0，设置迭代计算器$b=0$；

步骤一：用下式计算或者更新划分矩阵U^b：

对于$\forall i,k$，如果$\exists d_{ik}^{(b)} > 0$，则有

$$\mu_{ik}^{(b)} = \left\{ \sum_{j=1}^{c} \left[\left(\frac{d_{ik}^{(b)}}{d_{jk}^{(b)}} \right)^{\frac{2}{m-1}} \right] \right\}^{-1} \qquad (3\text{-}1)$$

对于$\forall i,r$，如果$d_{ik}^{(b)} = 0$，则有

$$\mu_{ir}^{(b)} = 1，\text{且对 } j \ne r，\ \mu_{ij}^{(b)} = 0$$

步骤二：用下式更新聚类原型模式矩阵 P^{b+1}：

$$P^{b+1}=\frac{\sum_{k=1}^{N}\mu_{ik}^{(b+1)}\bullet x_k}{\sum_{k=1}^{N}\mu_{ik}^{(b+1)}}\quad i=1,2,\ldots,C \tag{3-2}$$

步骤三：如果 $\| P_i^{b+1}-P_i^b \|<\varepsilon$，则算法停止并输出划分矩阵 U 和聚类原型 P，否则令 $b=b+1$，转向步骤一。其中 $\|\bullet\|$ 为某种合适的矩阵范数。

3.2.2 FCM 特征向量选择

我们在模糊聚类中，采用第 2 节中提及的股票序列的时域与混沌特性作为样本的特征向量，即采用开盘价、最高价、最低价、收盘价和成交量曲线的趋势、序列相关性（自相关和偏相关）、方差、峰谷、偏度、混沌性作为我们分类的依据。

3.2.3 FCM 分类结果

我们采用 FCM 分类的方法，结合主成分分析法，将特征向量维数降低，并确定聚类中心数，我们依然首先采用 FCM 优化函数随聚类中心数变化的曲线来确定，如图 3-9 所示。

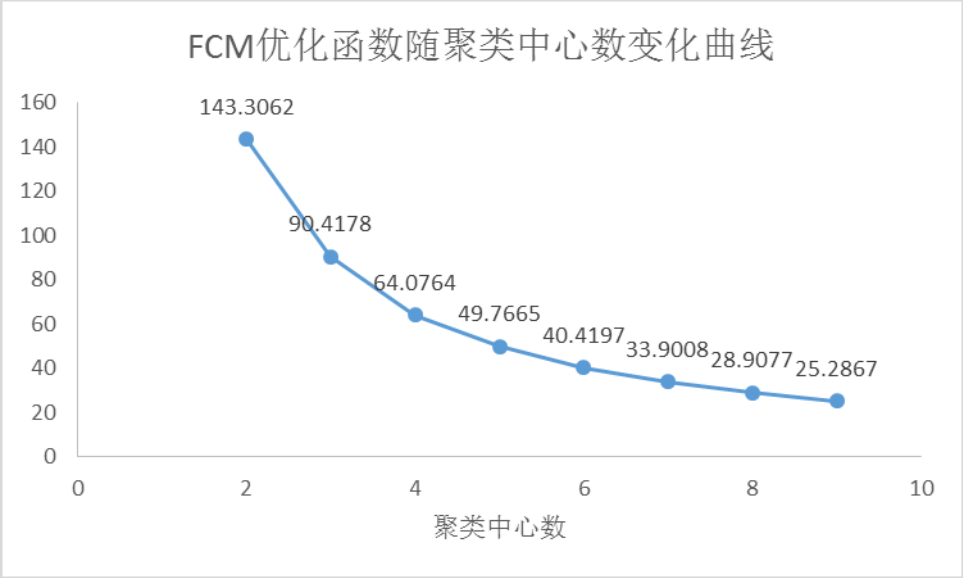


图 3-9 FCM 优化函数随聚类中心数变化曲线

图 3-10 至图 3-17 给出了 FCM 聚类在不同聚类中心数下的分类的情况。

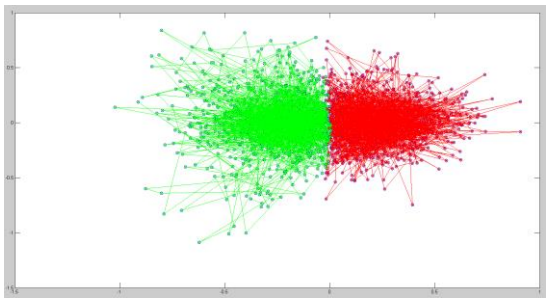


图 3-10 FCM 二分类图

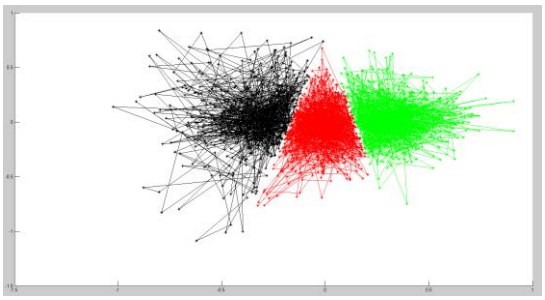


图 3-11 FCM 三分类图

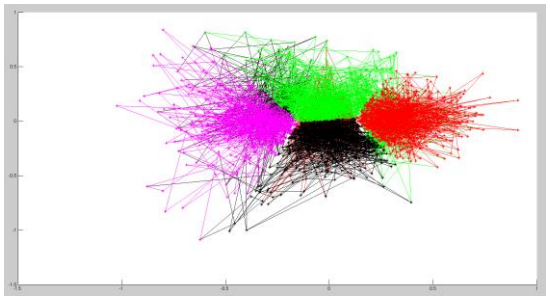


图 3-12 FCM 四分类图

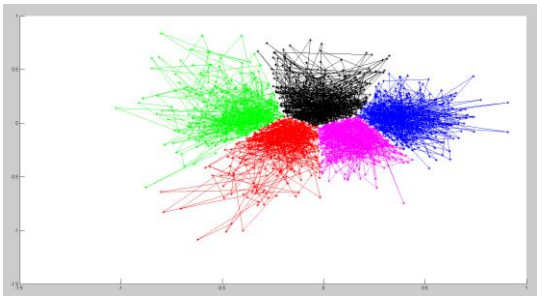


图 3-13 FCM 五分类图

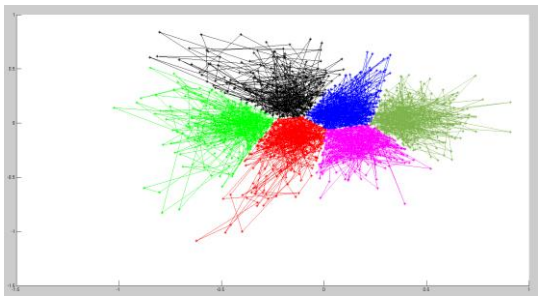


图 3-14 FCM 六分类图

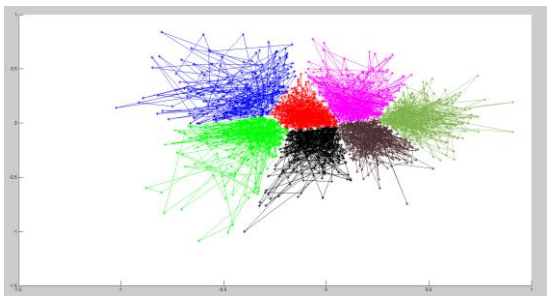


图 3-14 FCM 七分类图

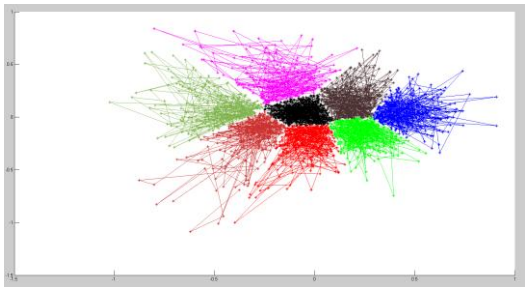


图 3-15 FCM 八分类图

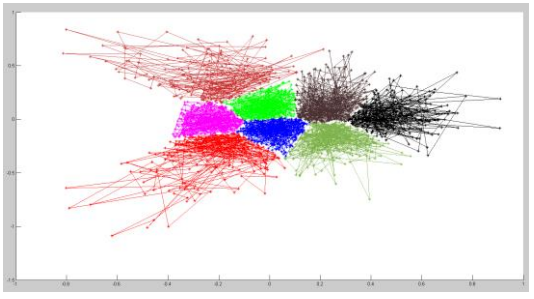


图 3-16 FCM 九分类图

由于上述特征空间中样本分类非常密集，并且各个类之间并没有明显的界限，所以分类效果并不理想，我们计算二分类的轮廓系数，如图 3-17 所示，也可以看出分类效果并不理想。整个分类的轮廓系数仅仅为 0.63，低于 0.8 水平。

这也反应了模糊聚类的特性，每一个样本不仅仅属于一个类别，所以在按照硬分类的指标来评价计算时，也就难怪会获得不好的效果。

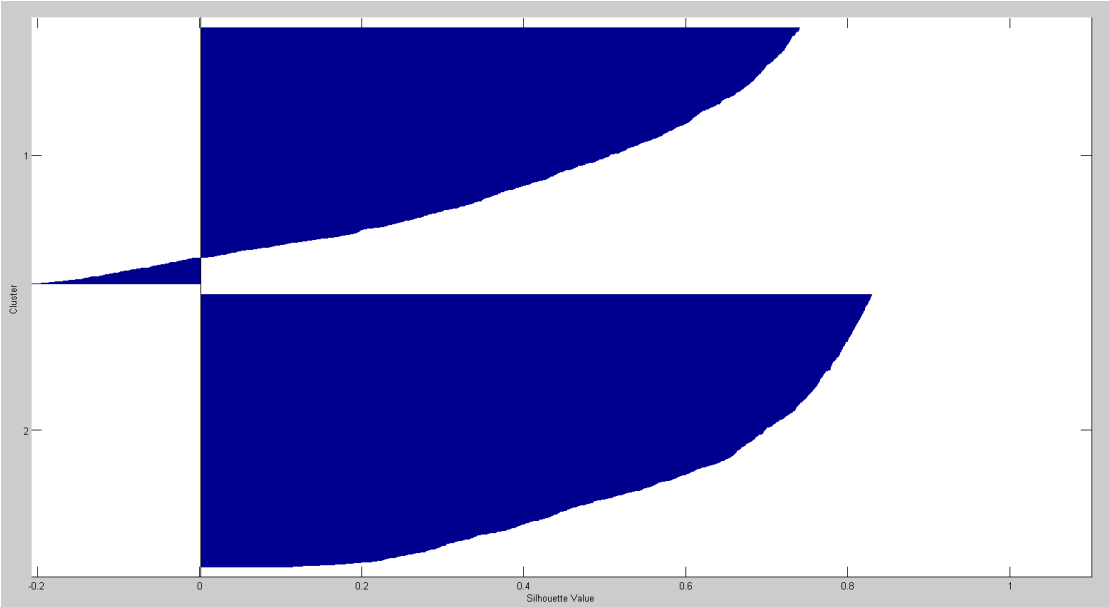


图 3-17 FCM 二分类结果轮廓系数

我们计算在二分类情况下的聚类平均值和方差，可以得到两个聚类中心点为[-0.2288 ， -0.0009]和[0.2178 -0.0011]，方差为 78.436 和 98.623。但是实际上，模糊聚类输出的结果为隶属度函数矩阵，如图 3-18 所示。

	1	2	3	4	5	6	7	8	9
1	0.6058	0.3294	0.1471	0.0528	0.8965	0.2281	0.4580	0.0628	0.2561
2	0.1846	0.2245	0.5944	0.3874	0.0536	0.6090	0.2074	0.3774	0.5777
3	0.1548	0.3967	0.1456	0.1366	0.0416	0.1159	0.2327	0.4732	0.1086
4	0.0548	0.0494	0.1129	0.4232	0.0083	0.0469	0.1019	0.0866	0.0575

图 3-18 FCM 四分类隶属度矩阵

3.2.4 FCM 聚类总结

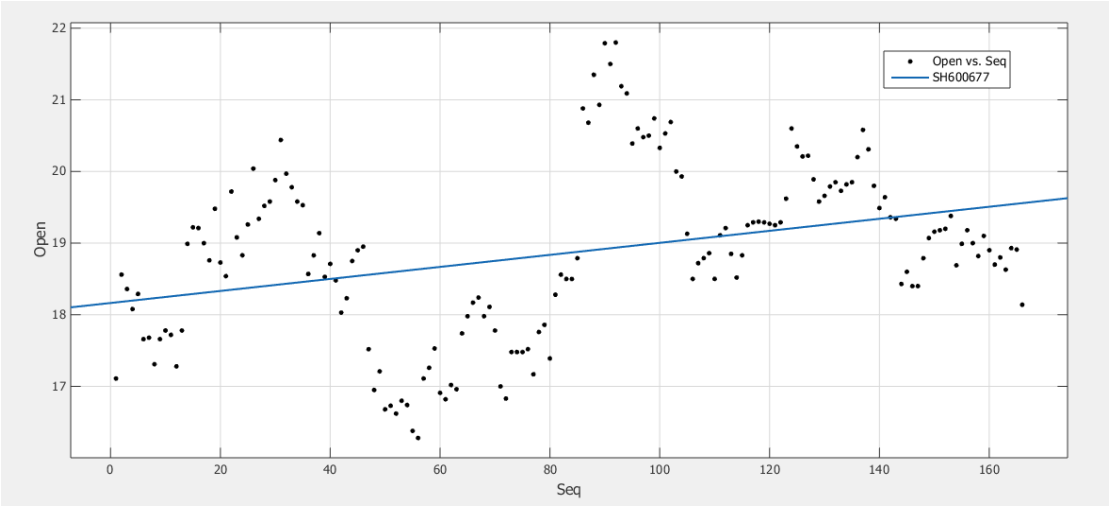
在进行 FCM 聚类的过程中，我们采用了时频域和混沌特性作为特征向量，在 FCM 聚类中我们获得了各个股票的属于各类的隶属度函数，我们采用了最大隶属度函数的方法，由于数据量庞大，我们不能很直观地表示出各只股票与各个类的隶属度关系，所以采用了将股票归为最大隶属度函数的那一类，这样简单的归类方式实际上违背了模糊聚类的初衷，应该将模糊聚类最终得到的隶属度关系矩阵作为结果，分类结果也应该反应各个样本在各类所占的比重。

4. 股票分类总结

在完成本次股票分类的大作业的过程中，我深刻体会到了聚类分析的作用，也掌握了一些聚类分析的理论和方法。面对股票这一充满了随机性和波动性的金融时间序列，虽然最终聚类的结果并不是那么惊艳，但是我在整个过程中深深感受到了其中的魅力，非常欣喜有这样一次深入学习聚类算法的机会。

附录一

上证 SH600677 一阶多项式拟合效果：



Linear model Poly1:

f(x) = p1*x + p2

Coefficients (with 95% confidence bounds):

p1 = 0.008395 (0.004825, 0.01197)

p2 = 18.16 (17.82, 18.51)

Goodness of fit:

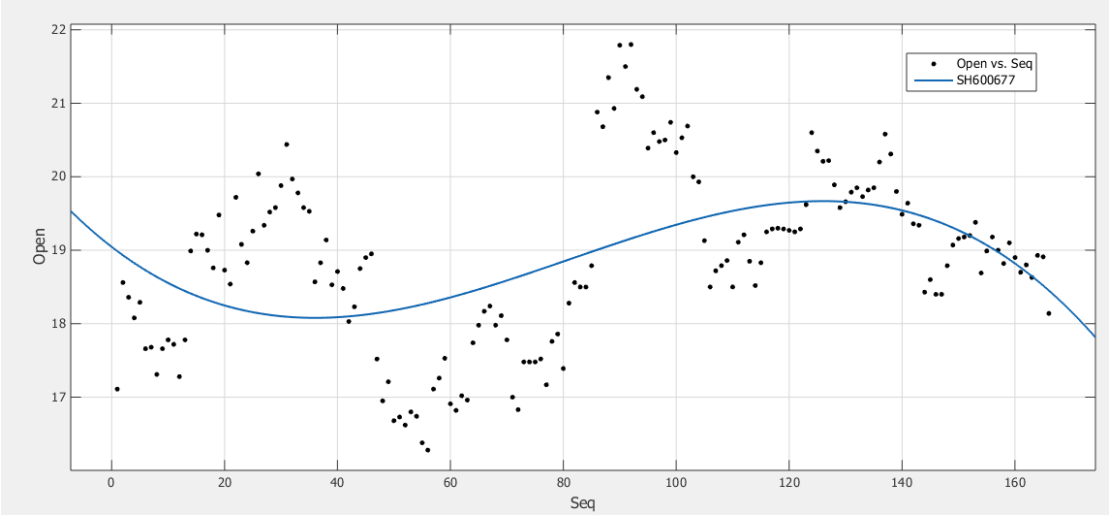
SSE: 204.4

R-square: 0.1162

Adjusted R-square: 0.1108

RMSE: 1.116

上证 SH600677 三阶多项式拟合效果：



Linear model Poly3:

f(x) = p1*x^3 + p2*x^2 + p3*x + p4

Coefficients (with 95% confidence bounds):

p1 = -4.355e-06 (-6.225e-06, -2.485e-06)

p2 = 0.001059 (0.0005837, 0.001534)

p3 = -0.05931 (-0.09351, -0.02512)

p4 = 19.05 (18.38, 19.71)

Goodness of fit:

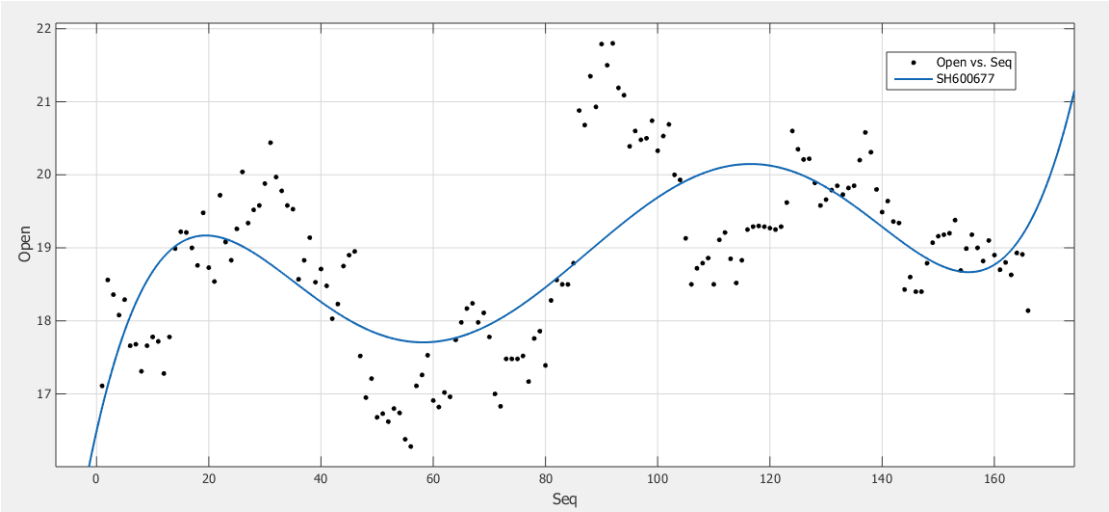
SSE: 180.2

R-square: 0.221

Adjusted R-square: 0.2066

RMSE: 1.055

上证 SH600677 五阶多项式拟合效果：



Linear model Poly5:

f(x) = p1*x^5 + p2*x^4 + p3*x^3 + p4*x^2 + p5*x + p6

Coefficients (with 95% confidence bounds):

p1 =	3.312e-09	(2.396e-09, 4.228e-09)
p2 =	-1.447e-06	(-1.832e-06, -1.063e-06)
p3 =	0.0002228	(0.0001645, 0.0002811)
p4 =	-0.01419	(-0.01805, -0.01034)
p5 =	0.3397	(0.2351, 0.4443)
p6 =	16.49	(15.62, 17.36)

Goodness of fit:

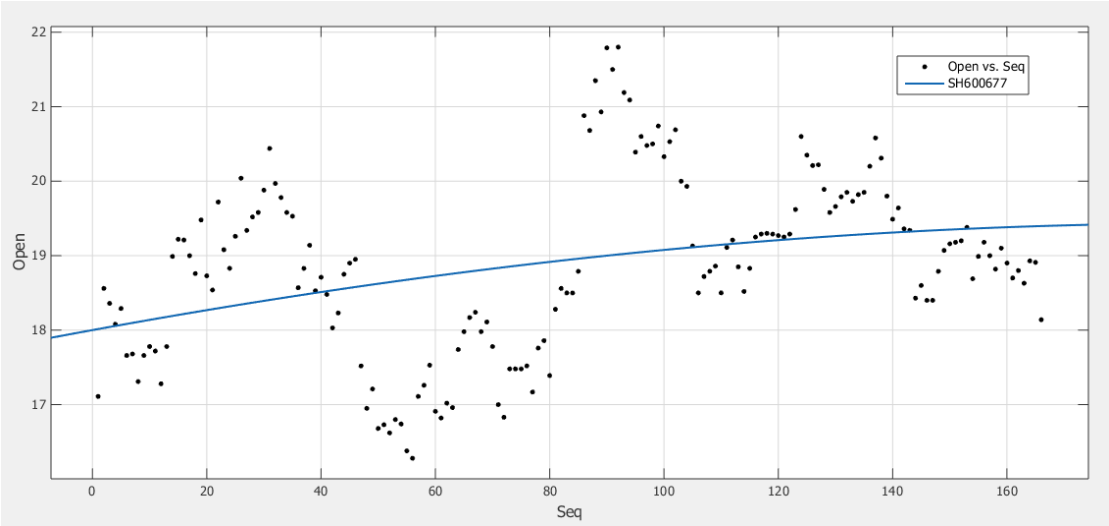
SSE: 129.8

R-square: 0.4389

Adjusted R-square: 0.4213

RMSE: 0.9006

上证 SH600677 一阶高斯拟合效果：



General model Gauss1:

f(x) = a1*exp(-((x-b1)/c1)^2)

Coefficients (with 95% confidence bounds):

a1 =	19.43	(18.43, 20.43)
b1 =	196.8	(-59.92, 453.6)
c1 =	711.1	(-85.93, 1508)

Goodness of fit:

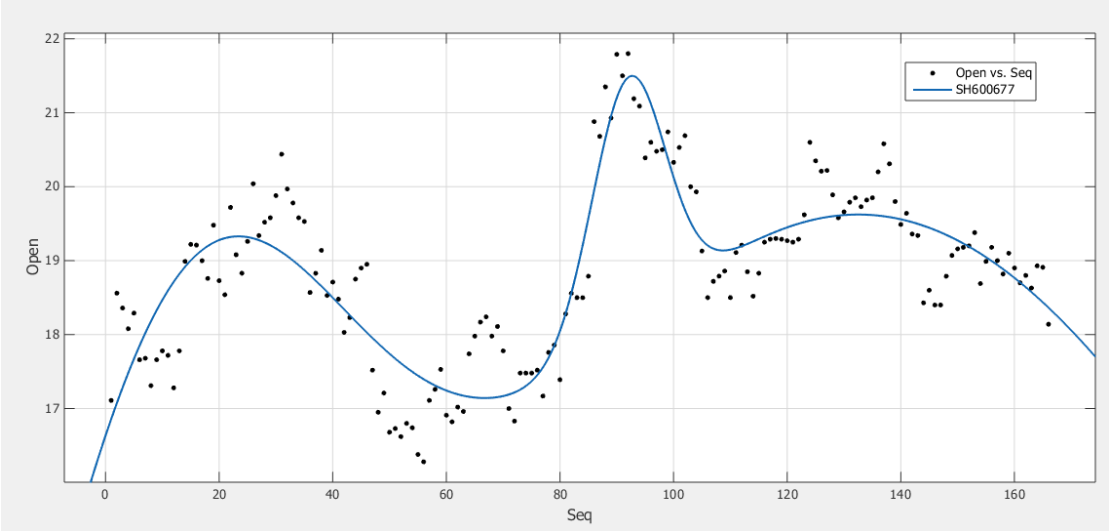
SSE: 203.5

R-square: 0.12

Adjusted R-square: 0.1092

RMSE: 1.117

上证 SH600677 三阶高斯拟合效果：



General model Gauss3:

f(x) =

a1*exp(-((x-b1)/c1)^2) + a2*exp(-((x-b2)/c2)^2) +
a3*exp(-((x-b3)/c3)^2)

Coefficients (with 95% confidence bounds):

a1 =	3.394	(2.862, 3.926)
b1 =	92	(91.05, 92.94)
c1 =	9.116	(7.347, 10.89)
a2 =	19.62	(19.43, 19.81)
b2 =	132.7	(128.5, 136.9)
c2 =	129.4	(103.4, 155.4)
a3 =	10.5	(7.479, 13.51)
b3 =	11.45	(8.681, 14.22)
c3 =	43.01	(34.49, 51.53)

Goodness of fit:

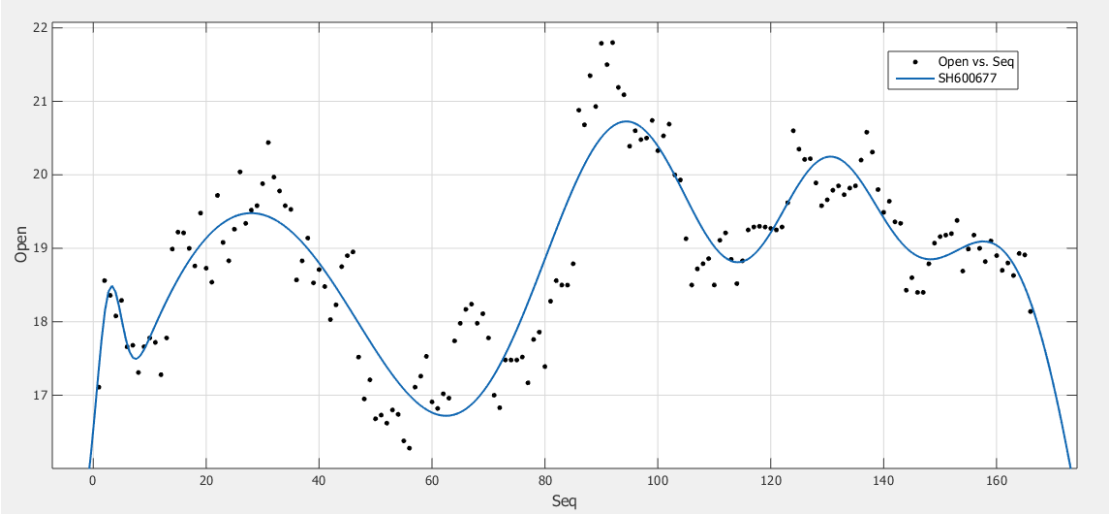
SSE: 52.75

R-square: 0.7719

Adjusted R-square: 0.7603

RMSE: 0.5796

上证 SH600677 五阶高斯拟合效果：



General model Gauss5:

f(x) =

a1*exp(-((x-b1)/c1)^2) + a2*exp(-((x-b2)/c2)^2) +
a3*exp(-((x-b3)/c3)^2) + a4*exp(-((x-b4)/c4)^2) +
a5*exp(-((x-b5)/c5)^2)

Coefficients (with 95% confidence bounds):

a1 =	15.68	(12.48, 18.88)
b1 =	98.79	(95.26, 102.3)
c1 =	28.78	(23.73, 33.83)
a2 =	7.893	(-7.444, 23.23)
b2 =	129.9	(125.1, 134.7)
c2 =	16.34	(7.661, 25.02)
a3 =	19.44	(19.21, 19.68)
b3 =	27.43	(25.54, 29.32)
c3 =	58.19	(47.98, 68.4)
a4 =	18.52	(17.23, 19.82)
b4 =	161.2	(154.6, 167.8)
c4 =	30.8	(-1.087, 62.68)
a5 =	2.183	(1.135, 3.231)
b5 =	2.773	(1.66, 3.886)
c5 =	3.056	(0.9817, 5.13)

Goodness of fit:

SSE: 48.92

R-square: 0.7885

Adjusted R-square: 0.7689

RMSE: 0.5692

附录二

```
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%
% file=dir('H:\patrecognition\stock\*.txt'); %获取当前目录下的.txt 文件名
% textnum=length(file);
% f=fullfile('H:','patrecognition','fulldata_2886.mat');%获取上述文件对应的路径
% Info=importdata(f); %使用 importdata 来处理含标题的文本
%     for k=1:textnum
%     [M,N]=size(Info{k}.data);
%     Sequence=1:M;Sequence=Sequence.';
%     Open=Info{k}.data(:,2);
%     High=Info{k}.data(:,3);
%     Low=Info{k}.data(:,4);
%     Close=Info{k}.data(:,5);
%     Amount=Info{k}.data(:,6);
%     Volume=Info{k}.data(:,7);
%
% %数据预处理 preprocessing
% Vibra=High - Low;
% for i=1:M-1
%     Vibra(i)=Vibra(i+1)/Close(i);
% end
% Vibra(M)=[]; %删除最后一行，降一维，获得振幅
% SequVib=1:M-1;SequVib=SequVib.';

%拟合求解系数
% [fitresult1, gof1]=createFit( Sequence,Open);
% % [fitresult2, gof2]=createFit( Sequence,Close);
% % [fitresult3, gof3]=createFit( SequVib,Vibra);
% %开盘系数
% Open_a1(k)=fitresult1.a1;
% Open_b1(k)=fitresult1.b1;
% Open_c1(k)=fitresult1.c1;
%     Open_a2(k)=fitresult1.a2;
%     Open_b2(k)=fitresult1.b2;
%     Open_c2(k)=fitresult1.c2;
% Open_a3(k)=fitresult1.a3;
% Open_b3(k)=fitresult1.b3;
% Open_c3(k)=fitresult1.c3;
```

```
%     Open_a4(k)=fitresult1.a4;
%     Open_b4(k)=fitresult1.b4;
%     Open_c4(k)=fitresult1.c4;
% Open_a5(k)=fitresult1.a5;
% Open_b5(k)=fitresult1.b5;
% Open_c5(k)=fitresult1.c5;
%     Open_a6(k)=fitresult1.a6;
%     Open_b6(k)=fitresult1.b6;
%     Open_c6(k)=fitresult1.c6;
% %收盘系数
% Close_a1(k)=fitresult2.a1;
% Close_b1(k)=fitresult2.b1;
% Close_c1(k)=fitresult2.c1;
%     Close_a2(k)=fitresult2.a2;
%     Close_b2(k)=fitresult2.b2;
%     Close_c2(k)=fitresult2.c2;
% Close_a3(k)=fitresult2.a3;
% Close_b3(k)=fitresult2.b3;
% Close_c3(k)=fitresult2.c3;
%     Close_a4(k)=fitresult2.a4;
%     Close_b4(k)=fitresult2.b4;
%     Close_c4(k)=fitresult2.c4;
% Close_a5(k)=fitresult2.a5;
% Close_b5(k)=fitresult2.b5;
% Close_c5(k)=fitresult2.c5;
%     Close_a6(k)=fitresult2.a6;
%     Close_b6(k)=fitresult2.b6;
%     Close_c6(k)=fitresult2.c6;
% %振荡系数
% Vib_a1(k)=fitresult3.a1;
% Vib_b1(k)=fitresult3.b1;
% Vib_c1(k)=fitresult3.c1;
%     Vib_a2(k)=fitresult3.a2;
%     Vib_b2(k)=fitresult3.b2;
%     Vib_c2(k)=fitresult3.c2;
% Vib_a3(k)=fitresult3.a3;
% Vib_b3(k)=fitresult3.b3;
% Vib_c3(k)=fitresult3.c3;
%     Vib_a4(k)=fitresult3.a4;
%     Vib_b4(k)=fitresult3.b4;
%     Vib_c4(k)=fitresult3.c4;
% Vib_a5(k)=fitresult3.a5;
% Vib_b5(k)=fitresult3.b5;
% Vib_c5(k)=fitresult3.c5;
```

```
%     Vib_a6(k)=fitresult3.a6;
%     Vib_b6(k)=fitresult3.b6;
%     Vib_c6(k)=fitresult3.c6;
%取平均
%     k
%     end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %%
%
MM=[Open_a1;Open_b1;Open_c1;Open_a2;Open_b2;Open_c2;Open_a3;Open_b3;Open_c3;Open_a4...
% ;Open_b4;Open_c4;Open_a5;Open_b5;Open_c5;Open_a6;Open_b6;Open_c6];
% MM=MM.';
% [Idx,C,sumD,D]=kmeans(MM,3,'dist','sqEuclidean','rep',4);
% %%



% %用 cell 结构把当前文档中的.txt 文件数据全部读取出来
% clc;clear;
% file=dir('H:\patrecognition\stock\*.txt'); %获取当前目录下的.txt 文件名
% textnum=length(file);
% Fulldata={};T={};
% for j=1:textnum
%     f=fullfile('H:','patrecognition\stock',file(j).name);
%     Fulldata=[Fulldata;importdata(f)];
%     j
% end
%
% for i=1:textnum
%     T{i}=Fulldata{i}.data(1:100,:);
%     i
% end

%% 三阶高斯拟合
Info=T; M=100;
textnum=2645;
for k=1:textnum

    Sequence=1:M;Sequence=Sequence.';
    Open=Info{1,k}(:,2);
    High=Info{1,k}(:,3);
```

```
    Low=Info{1,k}(:,4);
    Close=Info{1,k}(:,5);
    Amount=Info{1,k}(:,6);
    Volume=Info{1,k}(:,7);

% %数据预处理 preprocessing
    Vibra=High - Low;
        for i=1:M-1
            Vibra(i)=Vibra(i+1)/Close(i);
        end
Vibra(M)=[]; %删除最后一行，降一维，获得振幅
SequVib=1:M-1;SequVib=SequVib.';

% 拟合求解系数
[fitresult1, gof1]=Gauss3( Sequence,Open);
[fitresult2, gof2]=Gauss3( Sequence,Close);
[fitresult3, gof3]=Gauss3( SequVib,Vibra);
[fitresult4, gof4]=Gauss3( Sequence,Amount);
%开盘系数
Open_a1(k)=fitresult1.a1;
Open_b1(k)=fitresult1.b1;
Open_c1(k)=fitresult1.c1;
    Open_a2(k)=fitresult1.a2;
    Open_b2(k)=fitresult1.b2;
    Open_c2(k)=fitresult1.c2;
Open_a3(k)=fitresult1.a3;
Open_b3(k)=fitresult1.b3;
Open_c3(k)=fitresult1.c3;
%
% %收盘系数
Close_a1(k)=fitresult2.a1;
Close_b1(k)=fitresult2.b1;
Close_c1(k)=fitresult2.c1;
    Close_a2(k)=fitresult2.a2;
    Close_b2(k)=fitresult2.b2;
    Close_c2(k)=fitresult2.c2;
Close_a3(k)=fitresult2.a3;
Close_b3(k)=fitresult2.b3;
Close_c3(k)=fitresult2.c3;
%
% %振荡系数
Vib_a1(k)=fitresult3.a1;
Vib_b1(k)=fitresult3.b1;
Vib_c1(k)=fitresult3.c1;
```

```
    Vib_a2(k)=fitresult3.a2;
    Vib_b2(k)=fitresult3.b2;
    Vib_c2(k)=fitresult3.c2;
Vib_a3(k)=fitresult3.a3;
Vib_b3(k)=fitresult3.b3;
Vib_c3(k)=fitresult3.c3;

%成交量
Amount_a1(k)=fitresult4.a1;
Amount_b1(k)=fitresult4.b1;
Amount_c1(k)=fitresult4.c1;
    Amount_a2(k)=fitresult4.a2;
    Amount_b2(k)=fitresult4.b2;
    Amount_c2(k)=fitresult4.c2;
Amount_a3(k)=fitresult4.a3;
Amount_b3(k)=fitresult4.b3;
Amount_c3(k)=fitresult4.c3;


% 取平均
    k
end

%%
%kmeans 分类
OpenM=[Open_a1;Open_a2;Open_a3;Open_b1;Open_b2;Open_b3;Open_c1;Open_c
2;Open_c3];
OpenM=OpenM.';
OpenMarctan=atan(OpenM);
[Idx,C,sumD,D]=kmeans(OpenMarctan,3,'dist','sqEuclidean','rep',4);

CloseM=[Close_a1;Close_a2;Close_a3;Close_b1;Close_b2;Close_b3;Close_c
1;Close_c2;Close_c3];
CloseM=CloseM.';
CloseMarctan=atan(CloseM);
[Idx,C,sumD,D]=kmeans(CloseMarctan,1,'dist','sqEuclidean','rep',6);

AmountM=[Amount_a1;Amount_a2;Amount_a3;Amount_b1;Amount_b2;Amount_b3;
Amount_c1;Amount_c2;Amount_c3];
AmountM=AmountM.';
AmountMactan=atan(AmountM);
[Idx,C,sumD,D]=kmeans(AmountMactan,1,'dist','sqEuclidean','rep',6);

VibM=[Vib_a1;Vib_a2;Vib_a3;Vib_b1;Vib_b2;Vib_b3;Vib_c1;Vib_c2;Vib_c3]
```

```
;
VibM=VibM.';
VibMactan=atan(VibM);
[Idx,C,sumD,D]=kmeans(VibMactan,1,'dist','sqEuclidean','rep',6);
FourM=[OpenM CloseM VibM AmountM];%36 维特征向量
FourMarctan=atan(FourM);
[Idx,C,sumD,D]=kmeans(FourMarctan,2,'dist','sqEuclidean','rep',6);
[coeff,score,latent,tsquared,explained]= pca(FourMarctan);%主成分分析
```

```
function [fitresult, gof] = Gauss3(M, Open)
%CREATEFIT(M,OPEN)
%  Create a fit.
%
%  Data for 'SH600677' fit:
%      X Input : M
%      Y Output: Open
%  Output:
%      fitresult : a fit object representing the fit.
%      gof : structure with goodness-of fit info.
%
%  See also FIT, CFIT, SFIT.

%  Auto-generated by MATLAB on 13-Jan-2017 15:36:14


%% Fit: 'SH600677'.
[xData, yData] = prepareCurveData( M, Open );

% Set up fittype and options.
ft = fittype( 'gauss3' );
opts = fitoptions( ft );
opts.Display = 'Off';
opts.Lower = [-Inf -Inf 0 -Inf -Inf 0 -Inf -Inf 0];
opts.StartPoint = [21.8 92 18.048072036116 20.4397615998153 31
8.87344025921902 18.5595303475873 2 19.6190999738226];
opts.Upper = [Inf Inf Inf Inf Inf Inf Inf Inf Inf];

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );

% % Plot fit with data.
% figure( 'Name', 'SH600677' );
% h = plot( fitresult, xData, yData );
% legend( h, 'Open vs. M', 'SH600677', 'Location', 'NorthEast' );
% % Label axes
% xlabel( 'M' );
% ylabel( 'Open' );
% grid on
```

```
%%计算相关系数
Open=[];High=[];Low=[];Close=[];Amount=[];Volume=[];

for i=1:2645
    Open=[Open T{1,i}(:,2)];
 end
 OpenRe=corr(Open);

 for i=1:2645
    High=[High T{1,i}(:,3)];
 end
 HighRe=corr(High);

  for i=1:2645
    Low=[Low T{1,i}(:,4)];
 end
 LowRe=corr(Low);

 for i=1:2645
    Close=[Close T{1,i}(:,5)];
 end
 CloseRe=corr(Close);

  for i=1:2645
    Amount=[Amount T{1,i}(:,6)];
 end
 AmountRe=corr(Amount);

   for i=1:2645
    Volume=[Volume T{1,i}(:,7)];
 end
 VolumeRe=corr(Volume);
```

```
%计算时间序列分形维数。
function D=FractalDim(y,cellmax)
if cellmax<length(y)
    error('cellmax must be larger than input signal!')
end
L=length(y);
y_min=min(y);
y_min=min(y);
y_shift=y-y_min;
x_ord=[0:L-1]./(L-1);
xx_ord=[0:cellmax]./(cellmax);
y_interp=interp1(x_ord,y_shift,xx_ord);
ys_max=max(y_interp);
factory=cellmax/ys_max;
yy=abs(y_interp*factory);
t=log2(cellmax)+1;
for e=1:t
    Ne=0;
    cellsize=2^(e-1);
    NumSeg(e)=cellmax/cellsize;
       for j=1:NumSeg(e)
           begin=cellsize*(j-1)+1;
           tail=cellsize*j+1;
           seg=[begin:tail];
           yy_max=max(yy(seg));
           yy_min=min(yy(seg));
           up=ceil(yy_max/cellsize);
           down=floor(yy_min/cellsize);
           Ns=up-down;
           Ne=Ne+Ns;
       end
       N(e)=Ne;
end

r=-diff(log2(N));
id=find(r<=2&r>=1);
Ne=N(id);
e=NumSeg(id);
P=polyfit(log2(e),log2(Ne),1);
D=P(1);
```

```
clc;
clear;
%%
%在加载 corrmatrix_2645_100,mat 文件之后
length=2645;
N=1:100;
N=N.';
Sequence=1:length;
%峰值
OpenMAX=max(Open);
OpenMAX=OpenMAX.';
CloseMAX=max(Close);
CloseMAX=CloseMAX.';
HighMAX=max(High);
HighMAX=HighMAX.';
LowMAX=max(Low);
LowMAX=LowMAX.';
AmountMAX=max(Amount);
AmountMAX=AmountMAX.';
%%
%谷值
OpenMIN=min(Open);
OpenMIN=OpenMIN.';
CloseMIN=min(Close);
CloseMIN=CloseMIN.';
HighMIN=min(High);
HighMIN=HighMIN.';
LowMIN=min(Low);
LowMIN=LowMIN.';
AmountMIN=min(Amount);
AmountMIN=AmountMIN.';
%%
%峰度
OpenKUR=kurtosis(Open);
OpenKUR=OpenKUR.';
CloseKUR=kurtosis(Close);
CloseKUR=CloseKUR.';
HighKUR=kurtosis(High);
HighKUR=HighKUR.';
LowKUR=kurtosis(Low);
LowKUR=LowKUR.';
AmountKUR=kurtosis(Amount);
AmountKUR=AmountKUR.';
%%
```

```
%偏度
OpenSKE=skewness(Open);
OpenSKE=OpenSKE.';
CloseSKE=skewness(Close);
CloseSKE=CloseSKE.';
HighSKE=skewness(High);
HighSKE=HighSKE.';
LowSKE=skewness(Low);
LowSKE=LowSKE.';
AmountSKE=skewness(Amount);
AmountSKE=AmountSKE.';
%%
%趋势
for i=1:length
    [fitresult1, gof1] = ODpoly(N, Open(:,i));
    OpenTrend(i)=fitresult1.p1;
    [fitresult1, gof1] = ODpoly(N, Close(:,i));
    CloseTrend(i)=fitresult1.p1;
    [fitresult1, gof1] = ODpoly(N, High(:,i));
    HighTrend(i)=fitresult1.p1;
    [fitresult1, gof1] = ODpoly(N, Low(:,i));
    LowTrend(i)=fitresult1.p1;
        [fitresult1, gof1] = ODpoly(N, Amount(:,i));
    AmountTrend(i)=fitresult1.p1;
    i
end

OpenTrend=OpenTrend.';
CloseTrend=CloseTrend.';
HighTrend=HighTrend.';
LowTrend=LowTrend.';
AmountTrend=AmountTrend.';
%%
%分形维数
cellmax=128;
for i=1:length
   OpenFD(i)=FractalDim(Open(:,i),cellmax) ;
   CloseFD(i)=FractalDim(Close(:,i),cellmax)  ;
   HighFD(i)=FractalDim(High(:,i),cellmax) ;
   LowFD(i)=FractalDim(Low(:,i),cellmax) ;
   AmountFD(i)=FractalDim(Amount(:,i),cellmax)  ;
   i
end
OpenFD=OpenFD.';
```

```
CloseFD=CloseFD.';
HighFD=HighFD.';
LowFD=LowFD.';
AmountFD=AmountFD.';
%%
%%按每一维归一化

%峰值归一化

OpenMAX1=mapminmax(OpenMAX.',0,1);

CloseMAX1=mapminmax(CloseMAX.',0,1);

HighMAX1=mapminmax(HighMAX.',0,1);

LowMAX1=mapminmax(LowMAX.',0,1);

AmountMAX1=mapminmax(AmountMAX.',0,1);

MAX1M=[OpenMAX1.' CloseMAX1.' HighMAX1.' LowMAX1.' AmountMAX1.'];
%%
%谷值归一化

OpenMIN1=mapminmax(OpenMIN.',0,1);

CloseMIN1=mapminmax(CloseMIN.',0,1);

HighMIN1=mapminmax(HighMIN.',0,1);

LowMIN1=mapminmax(LowMIN.',0,1);

AmountMIN1=mapminmax(AmountMIN.',0,1);

MIN1M=[OpenMIN1.' CloseMIN1.' HighMIN1.' LowMIN1.' AmountMIN1.'];
%%
%峰度归一化

OpenKUR1=mapminmax(OpenKUR.',0,1);

CloseKUR1=mapminmax(CloseMIN.',0,1);

HighKUR1=mapminmax(HighKUR.',0,1);

LowKUR1=mapminmax(LowKUR.',0,1);
```

```
AmountKUR1=mapminmax(AmountKUR.',0,1);

KUR1M=[OpenKUR1.' CloseKUR1.' HighKUR1.' LowKUR1.' AmountKUR1.'];
%%
%偏度归一化

OpenSKE1=mapminmax(OpenSKE.',0,1);

CloseSKE1=mapminmax(CloseSKE.',0,1);

HighSKE1=mapminmax(HighSKE.',0,1);

LowSKE1=mapminmax(LowSKE.',0,1);

AmountSKE1=mapminmax(AmountSKE.',0,1);

SKE1M=[OpenSKE1.' CloseSKE1.' HighSKE1.' LowSKE1.' AmountSKE1.'];
%%
%趋势归一化

OpenTrend1=mapminmax(OpenTrend.',0,1);

CloseTrend1=mapminmax(CloseTrend.',0,1);

HighTrend1=mapminmax(HighTrend.',0,1);

LowTrend1=mapminmax(LowTrend.',0,1);

AmountTrend1=mapminmax(AmountTrend.',0,1);

Trend1M=[OpenTrend1.'    CloseTrend1.'    HighTrend1.'    LowTrend1.'
AmountTrend1.'];
%%
%分形维数归一化

OpenFD1=mapminmax(OpenFD.',0,1);

CloseFD1=mapminmax(CloseFD.',0,1);

HighFD1=mapminmax(HighFD.',0,1);

LowFD1=mapminmax(LowFD.',0,1);
```

```
AmountFD1=mapminmax(AmountFD.',0,1);

FD1M=[OpenFD1.' CloseFD1.' HighFD1.' LowFD1.' AmountFD1.'];
%%
FiveM=[MAX1M MIN1M KUR1M SKE1M Trend1M FD1M];

%%

% [CENTER1, U1, OBJ_FCN1] = fcm(FiveM, 1);
% [CENTER2 ,U2, OBJ_FCN2] = fcm(FiveM, 3);
% [CENTER3 ,U3, OBJ_FCN3] = fcm(FiveM, 3);
% [CENTER4 ,U4, OBJ_FCN4] = fcm(FiveM, 4);
% [CENTER5 ,U5, OBJ_FCN5] = fcm(FiveM, 5);
% [CENTER6 ,U6, OBJ_FCN6] = fcm(FiveM, 6);
% [CENTER7 ,U7, OBJ_FCN7] = fcm(FiveM, 7);
%[CENTER8,U8, OBJ_FCN8] = fcm(FiveM, 8);

[coeff,score,latent,tsquared,explained]= pca(FiveM);
% FiveMCPA=score(:,1:3);
%%
% %plot3(FiveMCPA(:,1),FiveMCPA(:,2),FiveMCPA(:,3),'o');

%%
FiveMCPA=score(:,1:2);
plot(FiveMCPA(:,1),FiveMCPA(:,2),'o');
hold on;
%%
[CENTER2 ,U2, OBJ_FCN2] = fcm(FiveMCPA, 9);
maxU = max(U2);
index1 = find(U2(1,:) == maxU);
index2 = find(U2(2,:) == maxU);
index3 = find(U2(3,:) == maxU);
index4 = find(U2(4,:) == maxU);
index5 = find(U2(5,:) == maxU);
index6 = find(U2(6,:) == maxU);
index7 = find(U2(7,:) == maxU);
index8 = find(U2(8,:) == maxU);
index9 = find(U2(9,:) == maxU);
line(FiveMCPA(index1,1),FiveMCPA(index1,2),'marker','*','color','g');
line(FiveMCPA(index2,1),FiveMCPA(index2,2),'marker','*','color','r');
line(FiveMCPA(index3,1),FiveMCPA(index3,2),'marker','*','color','k');
line(FiveMCPA(index4,1),FiveMCPA(index4,2),'marker','*','color','m');
line(FiveMCPA(index5,1),FiveMCPA(index5,2),'marker','*','color','b');
line(FiveMCPA(index6,1),FiveMCPA(index6,2),'marker','*','color',[0.5,
```

```
0.7,0.3]);
line(FiveMCPA(index7,1),FiveMCPA(index7,2),'marker','*','color',[0.3,
0.2,0.2]);
line(FiveMCPA(index8,1),FiveMCPA(index8,2),'marker','*','color',[0.8,
0.2,0.2]);
line(FiveMCPA(index9,1),FiveMCPA(index8,2),'marker','*','color',[0.8,
1,0.4]);
% plot(CENTER2(1,1),CENTER2(1,2),'*','color','k');
hold off
%%
```

```
%计算时间序列趋势
function [fitresult, gof] = ODpoly(Sequence, AmountSKE)
%CREATEFIT(SEQUENCE,AMOUNTSKE)
%  Create a fit.
%
%  Data for 'untitled fit 1' fit:
%      X Input : Sequence
%      Y Output: AmountSKE
%  Output:
%      fitresult : a fit object representing the fit.
%      gof : structure with goodness-of fit info.
%
%  See also FIT, CFIT, SFIT.

%  Auto-generated by MATLAB on 17-Jan-2017 18:49:13


%% Fit: 'untitled fit 1'.
[xData, yData] = prepareCurveData( Sequence, AmountSKE );

% Set up fittype and options.
ft = fittype( 'poly1' );
opts = fitoptions( ft );
opts.Lower = [-Inf -Inf];
opts.Upper = [Inf Inf];

% Fit model to data.
[fitresult, gof] = fit( xData, yData, ft, opts );

% % Plot fit with data.
% figure( 'Name', 'untitled fit 1' );
% h = plot( fitresult, xData, yData );
% legend( h, 'AmountSKE vs. Sequence', 'untitled fit 1', 'Location',
'NorthEast' );
% % Label axes
% xlabel( 'Sequence' );
% ylabel( 'AmountSKE' );
% grid on
```