基于时间和成本分析的蔬菜类商品定价与补货决策

摘要

本文主要通过建立 SARIMA 时间序列模型、人群搜索模型优化求解满足收益最大化的单目标规划模型、熵权法-灰色关联法优化的 Topsis 模型、满意度及收益最大化的多目标规划模型、基于商品品质下降和信誉，存储等损失的多目标规划模型，帮助生鲜超市制定合理的定价-补货决策。

针对问题一，首先，依靠统计分析得到了品类销量的数值变动规律；再利用斯皮尔曼系数确定了各品类在销量上呈现的关联关系。其后，依照赤池准则，运用网格搜索法得到最优的 SARIMA 模型参数，得到销量变化的季节与时间因子，最后运用高斯混合聚类进行类别检验，最后结合 ADF 检验和周期性指标分析出各品类及单品销量的宏观季节性及周期性规律。

针对问题二，依据对三年商超销售数据进行一元线性拟合，运用灰色预测模型对数据进行有序化处理最后应用 EWMA 指数加权移动平均预测未来 7 天补货数据。依据人群搜索法优化的算法求解得到合理的以品类为补货单位的补货计划使超市未来一周的最大日平均收益为 2137.42 元。

针对问题三，对蔬菜单品进行评价指标提取得到总利润、保鲜率、销售次数、损耗率等。其后构建熵权法-灰色关联法-Topsis 模型对各蔬菜单品进行评价，其后对各单品进行周期性一元线性拟合得到各单品之间相关性系数, 构建 kano 模型量化顾客对各单品满意度, 从而构建提高满意度并且实现超市收益最大化的多目标函数，运用粒子群算法优化求解出 7 月 1 日的补货单品为 33 个其补货量与定价策略为：净藕（1）定价 15.75 元/kg，补货量 10.5kg 等，当日超市收益为 1850.93 元。

针对问题四，通过采集数据引入存储损失，沉没成本，品质下降成本和信誉损失四个新的考虑维度，依托定价折扣和销售量增量的线性关系，构造最大化销量增量函数和最小化利润衰减函数的目标，用遗传算法和粒子群算法确定模型参数。同时，依托单品品质下降函数确定补货周期间隔，从而构造存货量的分段函数，根据不同的补货形式构造存货量的期望函数，损失期望。最终得到收益期望。构造最大化收益期望和销量增加，最小化利润衰减的多目标规划模型，以解决多因素影响的市场决策问题。

一、 问题重述

1.1 问题的背景

在生鲜超市中，通常蔬菜类商品的保鲜期均相对较短，且商品质量随上架时间的增加而变差，过半数品类如若当日滞销，隔日便无法再售。因此，超市会依据商品的历史营销和需求数据进行合理的补货。

由于超市售卖的蔬菜品类甚多、产地各异，而其采购时间通常在凌晨 3:00-4:00，为此商家须在不清楚具体品类和采购价格的情况下，做出当天各蔬菜的补货策略。蔬菜的定价原则为“成本加成定价”，另外超市对运损与品相差的商品采用打折销售策略。稳妥的市场需求分析，对补货策略及定价策略极其重要。从需求侧分析，蔬菜类商品的销售量与时间有一定的关联；从供给侧分析，蔬菜的供应品类在 4 月至 10 月较多，超市售卖空间的局限让良好的销售组合变得尤为重要。

1.2 问题的提出

根据某超市经销的 6 个蔬菜品类信息，该商超 2020 年 7 月 1 日至 2023 年 6 月 30 日三年各商品的销售流水明细与批发价格的相关数据，及各商品近期的损耗率数据，结合实际情况建立数学模型解决以下问题：

问题 1：在蔬菜类商品中不同品类及单品之间存在一定联系，请依照蔬菜各品类及单品销售量的规律确定其相互关系及分布规律。

问题 2：请以品类为补货单位，结合各蔬菜品类的营销总量与成本加成定价的关系，预测各蔬菜品类未来一周的日补货总量和定价策略，使得超市收益最大化。

问题 3：由于蔬菜类商品的销售空间有限，需要进一步优化单品补货计划，限制可售单品总数 27-33 个，各单品订购量达到最小陈列量 2.5 千克。依据 2023 年 6 月 24-30 日的可售品种，确定 7 月 1 日的单品补货量及定价策略，在满足市场需求的前提下使得超市收益最大化。

问题 4：为了制定更完备的蔬菜商品的补货和定价策略，请对超市仍需采集的相关数据进行补充并且给出意见和理由。

二、 问题的分析

2.1 问题一的分析

针对问题一，为了得到蔬菜类商品不同品类及单品之间的相互关系和分布规律，首先对各蔬菜品类及单品销售量的长期变动趋势进行平稳性检验，其后构建由季节变动、

循环变动、不规则变动的时间序列乘积模型。而后将所有单品每月销售数据进行处理，利用网格搜索法找到最优的模型。最后利用最优模型求解出各个单品的周期性指标，并且基于这些周期性指标对各个单品进行高斯混合聚类，将聚类结果和单品所属品类进行比对得到匹配度，进而检验同一类单品销量在时间上的一致性。最终结合 ADF 考量及单品周期性指标分析出相互关系及分布规律。

2.2 问题二的分析

针对问题二，以品类为补货单位结合营销量及定价关系预测蔬菜品类未来一周的补货量和定价来让超市收益最大化。首先我们结合现实考量提出四点模型假设：保证运用供大于求，打折区间满足售价大于进价，假设 2023 年 6 月 30 日刚好售完所有单品，规定加成定价处于一定弹性区间。由此列出相应数学公式。其后将六类蔬菜品类的成本加成定价与品类销售总量用 Matlab 进行一元线性回归拟合得到拟合曲线及其 R 方值用于判断拟合效果，其后将拟合曲线简化为条件公式。用灰色预测模型将大量离散的数据量进行有序化处理进而构建出 6 大品类的 GM(1,1) 模型。最后用 EWMA 指数加权移动平均模型基于前七天的补货量，跟据距离标本点的时间跨度确定系数矩阵的相对权重，构建收获量函数。最后建立目标收益函数及条件函数求解。

2.3 问题三的分析

针对问题三，需要进一步优化补货单品的补货计划，制定 7 月 1 日具体补货计划。首先对各蔬菜品类单品进行评价指标的提取，其后建立熵权法-灰色关联法优化的 Topsis 模型为每个评价指标打分其后为每个蔬菜单品进行具体打分。对之前提取出的各个蔬菜单品的评价指标以周期为 7 天的数据进行相似性分析，得到每个不同蔬菜品类中各单品之间的相关性系数。其后构建 kano 模型量化该超市对蔬菜购买的满意度，算出各个单品的满意度评分。最后构建函数求解出在满意度最高的情况下实现超市收益最大化的函数，运用多目标粒子群算法优化求解出最优定价策略、补货单品及补货量。

2.4 问题四的分析

针对问题四，需要采集相关数据更好的优化补货定价策略。这里我们 5 个采集数据即：单品存储费用、沉没成本、单品腐败规律曲线、单品开始变质时间、单品达不到售卖标准时间、信誉损失量。构建满足销量增加最大，利润衰减最小及收益期望最大的多目标规划模型实现超市收益最大化。

三、 模型假设

1. 在考虑超市运营实际情况下，保证供大于求，不会出现缺货断货等现象。

2. 在满足超市正常运营的情况下，打折区间始终满足售价大于进货价要求。

3. 成本加成定价处于一定的弹性区间且存在上下界。

4. 不计未知突发事件因素对销售单品的影响。

四、 符号说明

符号	意义
T	长期变动趋势
S	季节变动规律
C	循环变动规律
I	不规则变动（随机干扰项）
α	水平平滑参数
β	趋势平滑参数
X	补货量
W	成本加成定价
n	销售量
V	成本
c	产品种类
a	是否打折（0 为不打折，1 为打折）
φ	打折力度
S	损耗率

五、 模型的建立与求解

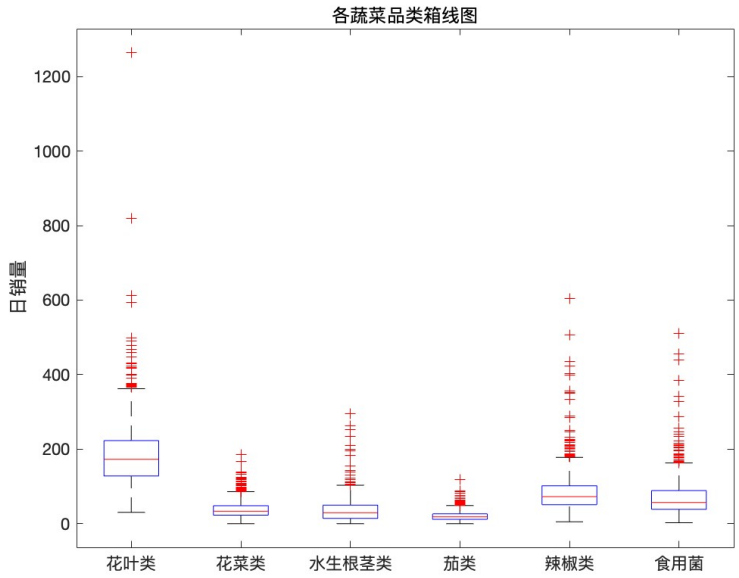
5.1 数据预处理

1. 删除缺失值：对附件的数据进行处理，将存在缺失值的数据进行删除。

2. 重复值处理：遍历附件 2 对重复数据进行检验并且删除。

3. 负值数据处理：在对附件 2 的检验中发现销售量的值存在负数，此情况认为是存在购买者退货的情况，将该数据删去。

4. 对异常值和离群值检测，并用临近数据的平均值进行替换。



5.2 问题一的求解

5.2.1 问题一的求解思路

为了更好的确定蔬菜各品类的销售量分布规律和相互关系，对各品类蔬菜的销售量进行统计学分析和相关性检验。为了更好的确定蔬菜各单品销售量的分布规律和相互关系，可以采用时间顺序模型进行分析。时间顺序模型主要基于过去事物所涵盖具体指标的数值依照时间顺序排列，用以分析其变化规律。

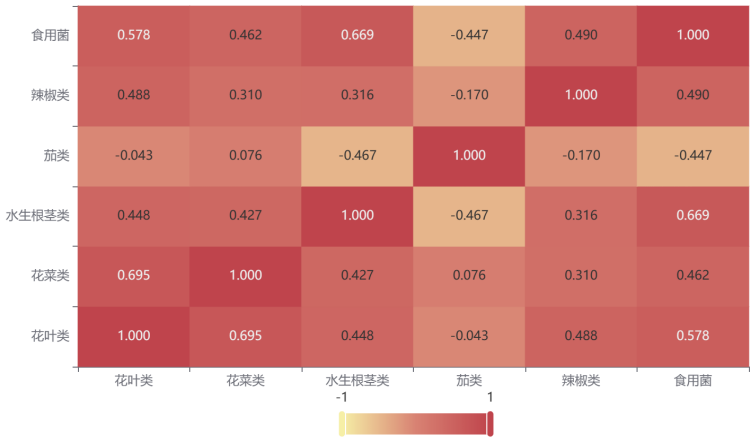
5.2.2 蔬菜各品类销售量的分布规律及其相互关系

斯皮尔曼相关系数是衡量两个变量的依赖性的非参数指标，它利用单调方程评价两个统计变量的相关性，并且当两个变量完全单调相关时，斯皮尔曼相关系数为 +1 或 −1。[2] 对于样本的原始数据转化为等级间的相关系数 ρ，样本容量为 n，d 为 x_i 和 y_i 之间的等级差。

$$\rho = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2\sum_{i=1}^{n}(y_i-\bar{y})^2}} = 1-\frac{6\sum d_i^2}{n(n^2-1)}$$

上述相关性系数的绝对值越接近于 1，相关性越大；越接近于 0，相关性越小。

ρ 的绝对值	相关程度
0-0.2	极弱（无相关性）
0.2-0.4	弱相关
0.4-0.6	中等相关
0.8-1	极强相关



通过分析销售量数据能够明显发现不同品类蔬菜的销售量存在相互关系，例如花菜类和花叶类的相关性系数接近于 0.695，说明花菜类的销售量会随着花叶类销售量的增长而增长。茄类和水生根茎的相关性系数为-0.467，则表示茄类的销量会随着水生根茎类销量的增加而减少。

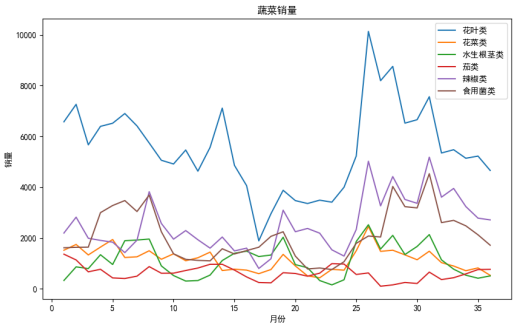


图 1　各品类蔬菜每月销售量

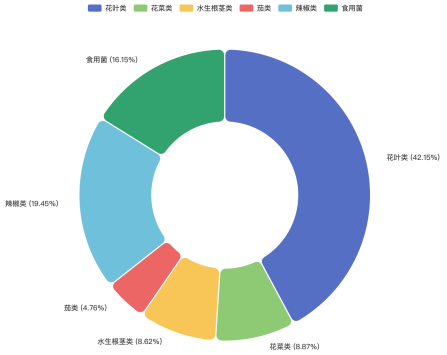
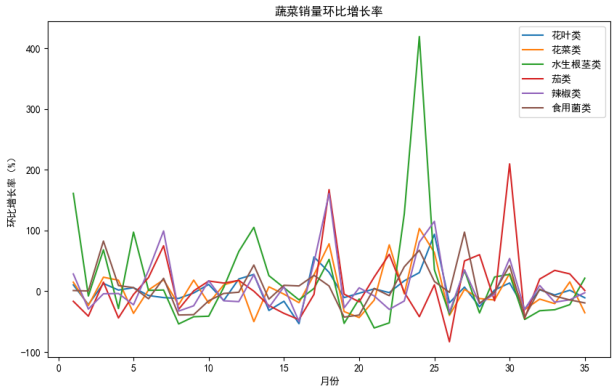


图 2　各品类蔬菜销售总量

由上图所示，花叶类蔬菜销售总量比例最高，茄类蔬菜销售总量比例最低。



由 36 个月蔬菜环比增长率对照图看出水生根茎类蔬菜在第 20 个月份时环比增长

最为显著，茄类在第 30 个月环比增长率最显著。

5.2.3 各单品蔬菜的销售量分布规律及其相互关系

5.2.2.1 模型的准备

时间序列数据本质上反映的是某个或者某些随机变量随时间不断变化的趋势，而时间序列预测方法的核心就是从数据中挖掘出这种规律，并利用其对将来的数据做出估计。[1] 时间顺序模型主要构成要素为：长期变动趋势，季节变动规律，循环变动规律和随机干扰项。

Step1: 首先对各蔬菜品类及单品销售量的长期变动趋势进行**平稳性检验**。将该超市的蔬菜类各品类及销售量随时间的变化情况设置成时间序列 $\{X_t\}$，时间间隔为 s。

1.$\{X_t\}$ 的均值

$E(X_t) = E(X_{t-s}) = c$, c 为常数

由此看出时间序列 $\{X_t\}$ 的均值大致相同，且趋近于固定常数。

2.$\{X_t\}$ 的方差

$Var(X_t) = Var(X_{t-s}) = \sigma^2$

由此看出时间序列 $\{X_t\}$ 有较为稳定的方差。

3.$\{X_t\}$ 的协方差

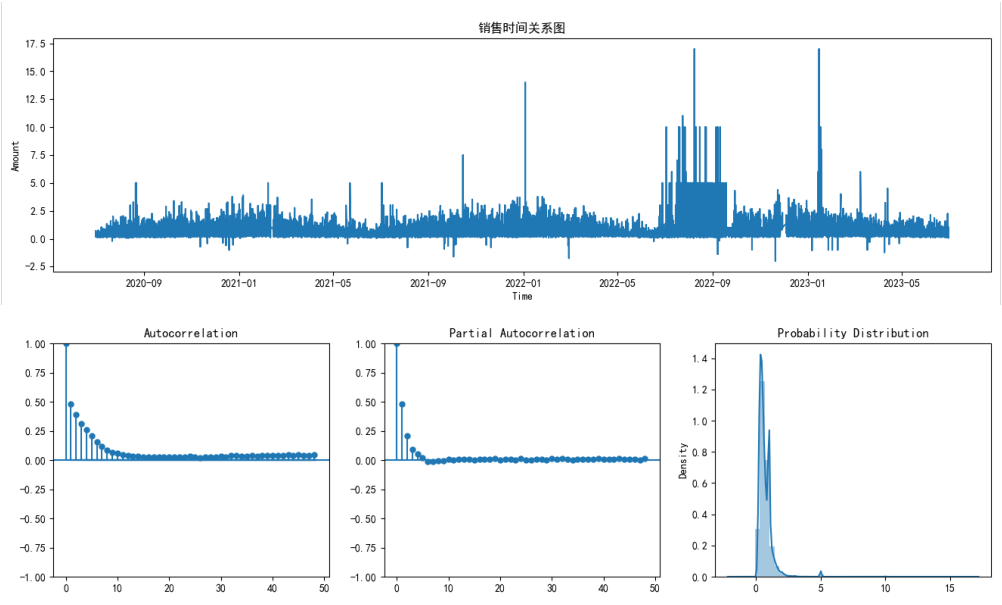
$Cov(X_t, X_{t-s}) = \gamma_s$

由此看出，协方差只与时间间隔有关而与选取的时间点无关。

根据以上三点条件的结果认为 $\{X_t\}$ 为协方差平稳（弱平稳）

Step2: 其次对附件二中不同品类的蔬菜类商品进行 ADF 检验。

以水生根茎为例绘出销售时间关系图如下：



根据销售时间关系图可知在 95% 的时间区域内，我们可以认为销量收到干扰性因素 I 的影响相对较低，呈现出较平稳的周期性分布。同时在 ADF 检验表明：统计结果的 P 值处于低于 0.05 的水平，则可初步认定模型是可接受的。观察自相关图（Autocorrel）可以发现，模型在 12 个滞后周期以内的相关性是显著的，判定模型具有较强的季节相关性。同时，偏自相关函数 PCFA 给出了平稳时间序列与其自身滞后值偏相关。PCFA 消除了其它滞后的相关贡献，在不受其它滞后因素影响的情况下给出了两个滞后的纯相关，在滞后周期达到 12 以后，PCFA 平稳趋近于 0，说明对每个商品的周期分布趋于平稳。对其余五项分类进行 ADF 分析，也通过了平稳性检验。

5.2.2.2 模型的假设

1. 根据模型准备中的分析结果认为 $\{X_t\}$ 为协方差平稳，即在考虑蔬菜类品类及单品销售量规律和联系时不考虑长期变动趋势因素 T。

2. 根据对三年销售流水明细数据的分析可以发现时间序列 $\{X_t\}$ 有明显的季节性 S。

3. 根据对销售流水数据每日时刻分析发现时间序列 $\{X_t\}$ 有明显周期性 C。

5.2.2.3 模型的建立

依据上述分析，决定时间序列的数值变化的 T,S,C,I 之间存在相互影响的关系，故应该使用乘积模型，又根据上述假设不考虑 T 长期趋势变动乘积模型如下所示：

$$Y = S \times C \times I$$

依据上述分析结果构建差分析回归移动平均模型（ARIMA）

1.AR 模型

自回归：将 1 至 p 阶的滞后项作为自变量来参与回归, 用以处理之前的观测值对当前值的影响。建立 p 阶自回归模型：

$$X_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i X_{t-i} + \varepsilon_t$$

上式中 ε_t 为方差的白噪声序列。

2.MA 模型

移动平均过程能够消除时间序列预测中不规则的波动，通过构建 q 阶移动平均过程模型：

$$X_t = \varepsilon_t + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i}$$

3.ARIMA 模型

在用差分过程将数据转换为平稳化（即消除长期趋势及季节变动规律）之后结合自回归与移动平均过程，协同模拟出预测结果。

$$ARIMA(p,d,q) : X_t' = \alpha_0 + \sum_{i=1}^{p} \alpha_i X_{t-i}' + \varepsilon_t + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i}$$

$$X_t' = \Delta^d X_t = (1-L)^d X_t \qquad \text{(d 为差分阶数)}$$

$$(1-\sum_{i=1}^{p}\alpha_i L^i)(1-L)^d X_t = \alpha_0 + (1+\sum_{i=1}^{q}\beta_i L^i)\varepsilon_t$$

4.SARIMA 模型

为了能够考虑季节因素对时间序列模型的影响，考虑周期性的季节影响因素，公式如下：

$$SARIMA \quad (p,d,q)(P,D,Q)_m$$

上式中 (p,d,q) 为非季节因素，(P,D,Q) 为季节因素，m 为周期数。

5.2.2.4 SARIMA 模型的建立

首先把所有的单品在每月的销售数据进行了累计和处理，利用网格搜索法对数据进行分析，对 p,d,q 和 P,D,Q 进行不同参数的组合搜索。我们知道，SARIMA 模型的选取依赖于赤池信息量准则（AIC）在一般情况下，AIC（赤池信息量准则）可以表示为以下公式：

$$AIC = 2k - 2ln(L)$$

上式中，k 为模型的参数数量，L 为似然函数。

假设模型的误差服从独立正态分布。

如果 n 表示观测数，RSS 表示残差平方和，那么 AIC 可以表示为：

$$AIC = nln(\frac{RSS}{n})$$

增加自由参数的数量能够提高拟合的优良性，但 AIC 鼓励在保持拟合优良性的同时尽量避免过度拟合的情况。因此，首选 AIC 值最小的模型，即寻找可以最好地解释数据特性但包含最少自由参数的模型。

通过网格搜索，我们得到的最优模型为 $(1,1,1)*(1,1,1,12)$

5.2.2.5 模型的求解对 SARIMA 模型进行求解，得到其周期性指标 ar.L1,ma.L1,ar.S.L12,ma.S.L12. 以下附 0-20 商品的周期性指标，完整数据表格详见（coef.xlsx）

单品序号	**ar.L1**	**ma.L1**	**ar.S.L12**	**ma.S.L12**
0	-0.76151	-0.25241	-0.78013	-0.99359
1	0.567522	-1	-0.84844	0.99995
2	0.431597	-1.00001	0.009273	0.9999
3	0.626815	-1.00003	0.474148	0.999886
4	0.382797	-1	-0.86445	-1.0015
5	0.411638	-1.00001	-1.01062	-1.0009

单品序号	**ar.L1**	**ma.L1**	**ar.S.L12**	**ma.S.L12**
6	-0.6174	0.999993	-0.98799	0.999969
7	0.520128	0.105964	0.001739	0.332377
8	1.312281	-0.00058	-2.66495	0.022364
9	-0.94111	1.000399	-0.1497	0.397004
10	0.649184	-1.00001	-0.00012	-0.05351
11	-0.18629	0.99999	0.069118	-0.10577
12	0.370745	-1.00001	-0.02987	-0.04578
13	0.54716	-1.00001	-0.42084	0.999871
14	0.777548	-1	-0.18965	1.000481
15	1.640318	-0.82702	-0.87243	-1.04551
16	-0.74397	1.100448	-0.21667	0.274374
17	-0.26894	-1	-0.61938	-1.00074
18	0.012881	-1	-0.00131	-0.25638
19	0.433388	0.949276	1.495945	0.9908
20	0.000237	-1.8E-10	0.001023	-6.7E-12

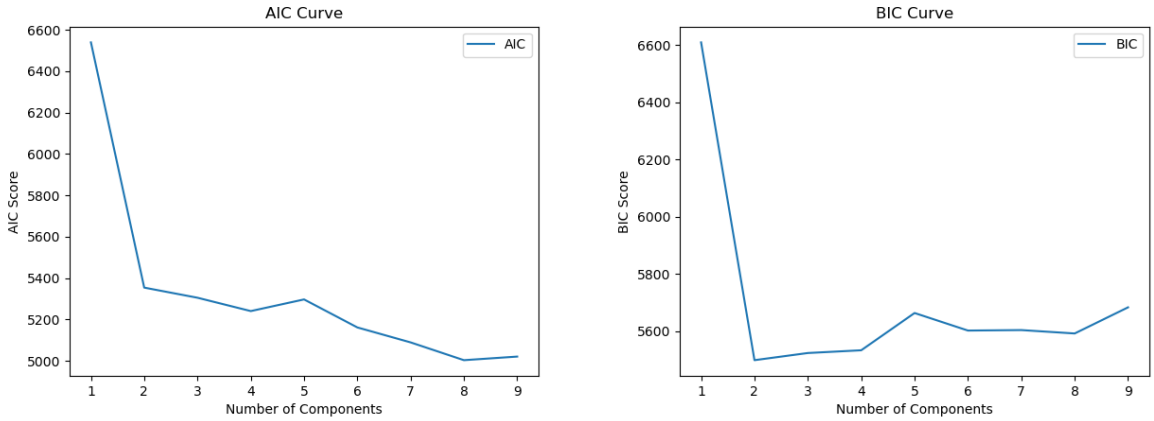
表 2

依据上表中四种周期性指标进行高斯混合聚类分析，首先假设该单品数据集符合 k 个高斯分布（$k \in [1, 251]$），高斯混合聚类的概率密度函数公式如下：

$$P(x) = \sum_{i=1}^{k} \alpha_i p(x|\mu_i, \Sigma_i)$$

上式中 k 为聚类个数,α_i 为第 i 个高斯分布，μ_i 为均值向量，Σ_i 为协方差矩阵。

其后将商品划分于概率密度最大的组簇内，用最大似然估计更新高斯分布参数，不断迭代得到最优组件数。由此过程绘出曲线 AIC 和 BIC 如下:

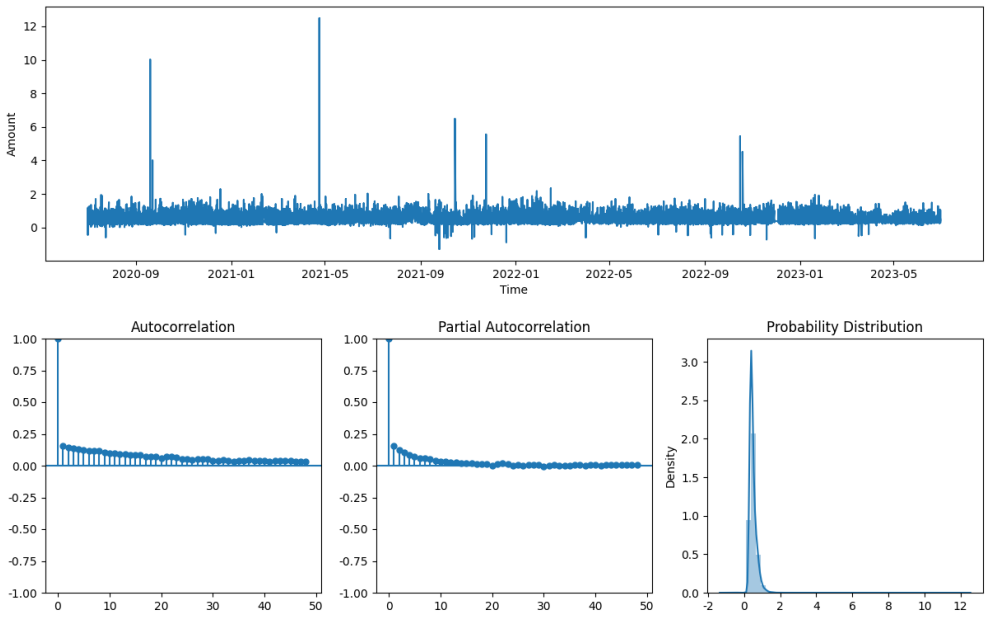


综合考虑得到的最优组件数为 6 时最合适。设定聚类组件数为 6(即 k 为 6) 对单品进行聚类后与附件 1 中各蔬菜品类的分类情况进行比对，发现通过时间序列分析得到的聚类结果与单品所属品类的匹配度高达 92.4%，可以认为同一类别的单品销量在时间上呈现了高度的一致性。

具体分析季节因子和时间因子的衰减系数，同时结合前文的 ADF 考量，我们可以得到以下结论：

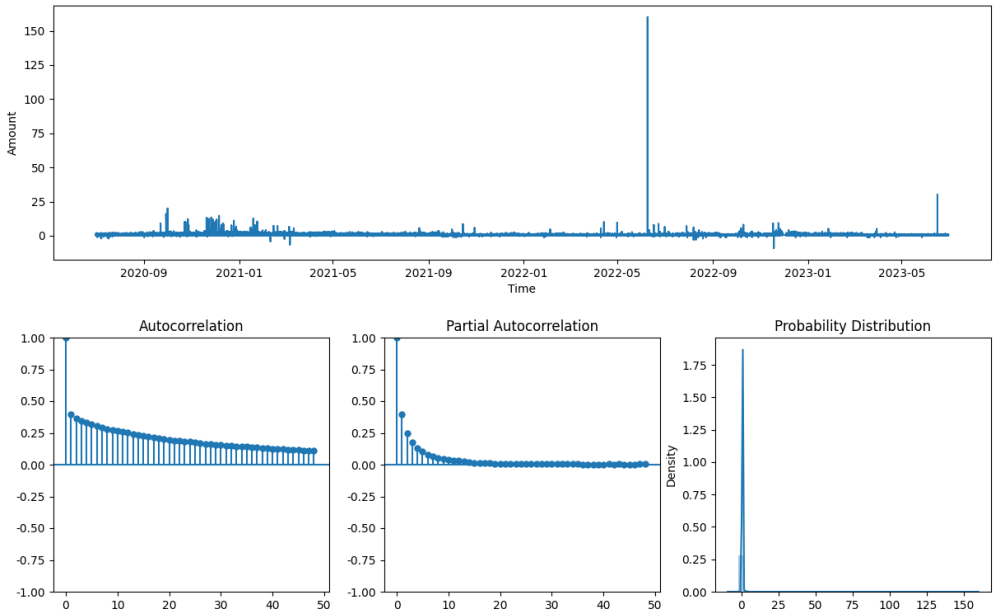
1. 对花菜类而言，ar.L1 整体维持在较高水平，呈现规律性的随季节的波动分布，ma.L1 的数值较高，再考虑到其 PCFA 在低滞后周期内有较高水平，可以解释其在每年的五月和九月有销量突变的形成。从具体时刻看，花菜类单品销量在一天的早中晚时段呈现出线性的下降趋势。

图 3　花菜类蔬菜销售时间关系图



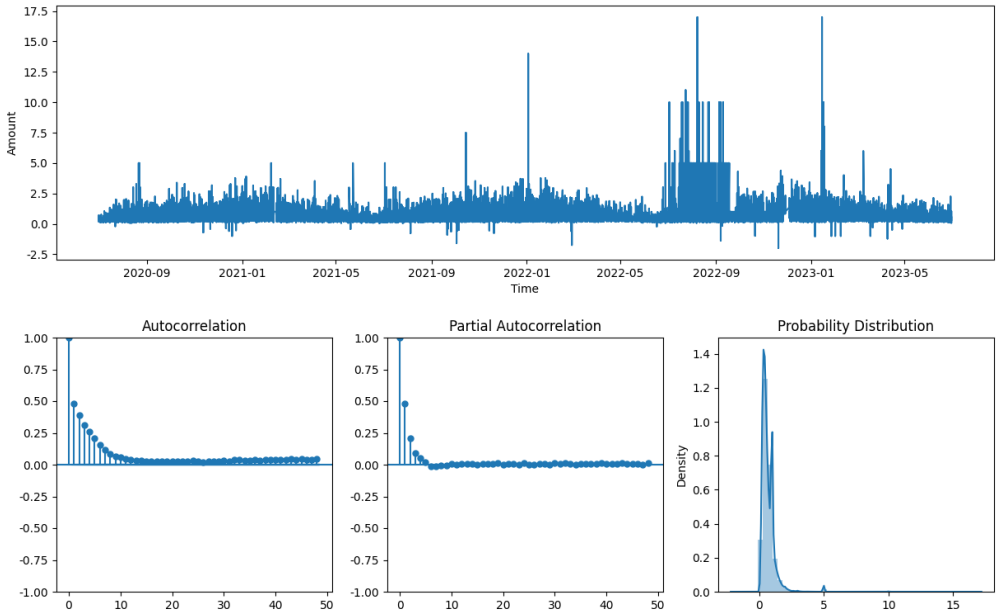
2. 对花叶类而言，ar.L1 和 ar.S.L12 都维持在较高水平，反应在数据上，每个月的花叶类单品销量都趋于稳定，前后相关性很强，其自相关系数在整个时间序列中都维持在 0.25 左右的高水平，说明数据的波动性不高。从具体时刻来看，花菜类单品的销量在早上和中午维持在较高水平且波动性较强，在晚上销量有明显的下降。

图 4　花叶类蔬菜销售时间关系图



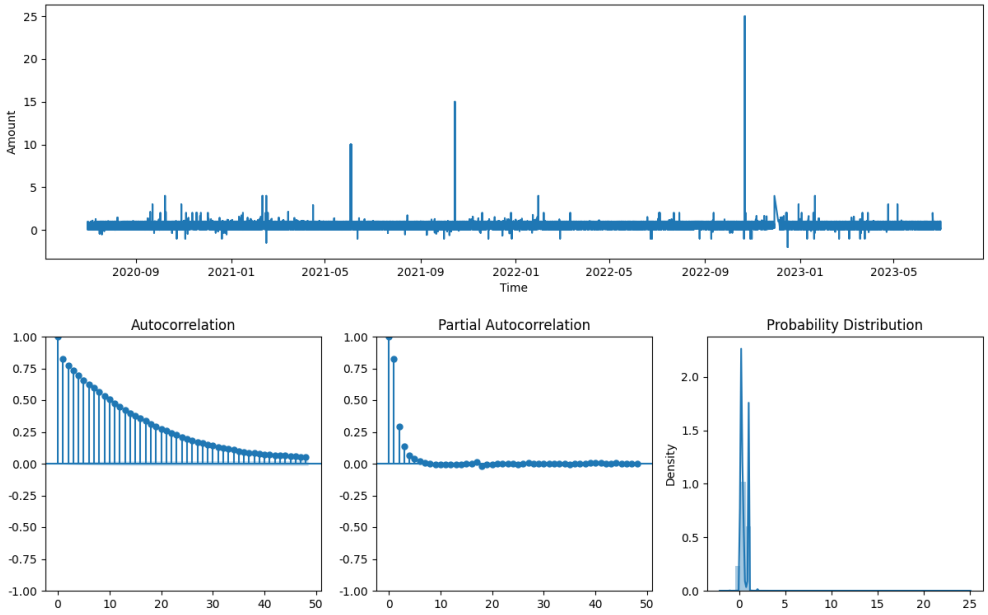
3. 对水生根茎类而言，它们的 ma.L1 和 ma.S.L12 相对于 ar 系数普遍处于较高水平，虽然依然随着季度，数据呈现了周期性的变化，但是收到的干扰系数 I 较高，所以在 1 月和 9 月容易发生销量的突变。它们在不同时刻的销量差别不大，有轻微的波动。

图 5　水生根茎类蔬菜销售时间关系图



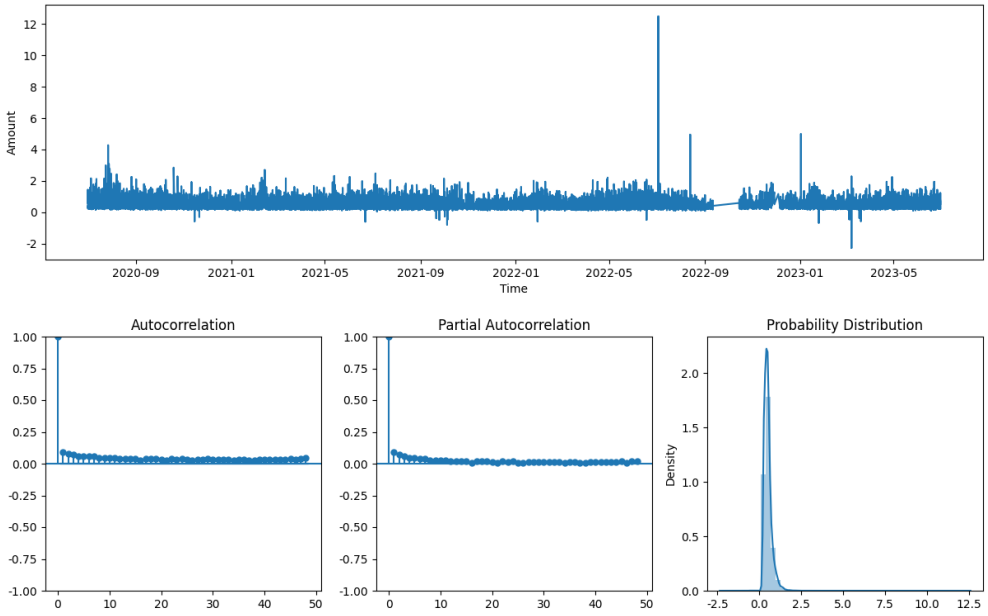
4. 对食用菌类而言,ar.L1 和 ma.L1 很高而 ma.S.L12,ar.S.L12 都在较低水平，说明食用菌类的销量很容易收到相邻季节的影响而在相同季节（较长间隔）下有很好的周期性，同时在 ADF 检验中也发现它的自相关系数是六个种类中维持在最高水平的，这很好地解释了它能够长期趋于平稳且在相邻时间段内数据极其稳健的原因。具体到每一天，食用菌类单品的销量依然在早中晚时段保持了这种稳定性。

图 6　食用菌类蔬菜销售时间关系图



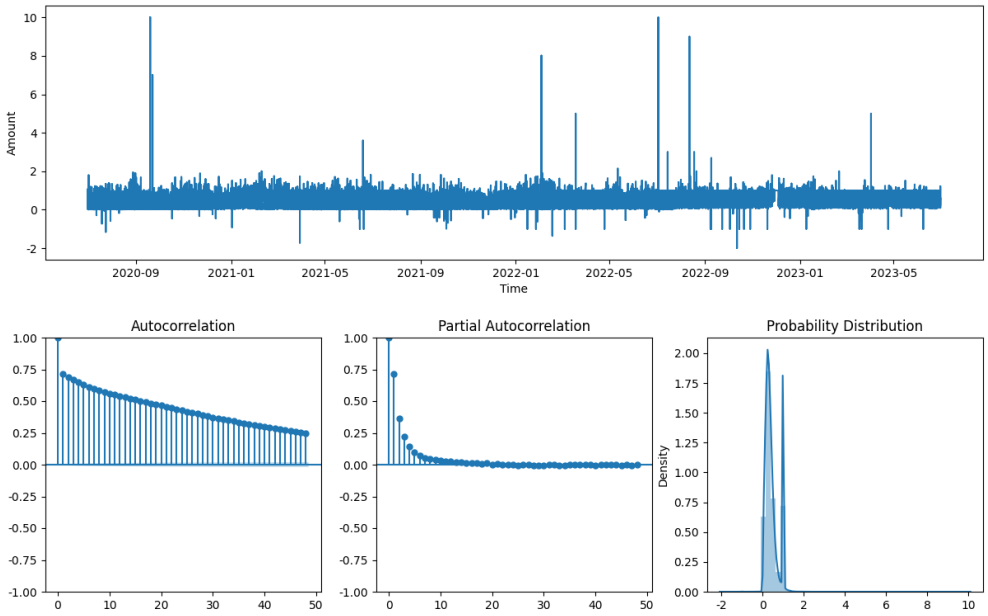
5. 对茄类而言，它的自相关系数较低，ma 系数较高而 ar 系数较低，说明茄类很容易在时间序列中产生 I 类系数引起的波动，观察图表也可以发现茄类随着季节变化销量不稳定的特性。在每天的具体时段里，茄类一般在早晚迎来销量高峰。

图 7　茄类蔬菜销售时间关系图



6. 对辣椒类而言，自相关系数较高的同时，在较短的滞后区间内 PCFA 也较高，从 SARIMA 所得到的指标结果来看：L12 造成的影响也大于 L1。这说明短期内辣椒类单品滞后区间造成的误差已经大于了它的自相关系数带来的稳定性收益，导致它的波动性较大，而长期来看，它的波动较高的区间却稳定在 5 月和 9 月。这应证了辣椒类单品销量环比的稳定。在每天的具体时段里，辣椒类的销量在早中晚并没有明显差别，趋于稳定。

图 8　辣椒类蔬菜销售时间关系图



5.3 问题二的求解

5.3.1 问题二的求解思路

首先对各蔬菜品类的销售总量与成本加成定价进行回归分析，并根据以往的销售数据对未来一周各蔬菜品类的进货成本和顾客预期需求量进行预测。最后以超市收益最大化为目标建立日补货总量和定价策略决策模型。

5.3.2 问题二的符号说明

符号	意义
X	补货量
W	成本加成定价
n	销售量
V	成本
c	产品种类
i	日期
a_{ic}	是否打折（0 为不打折，1 为打折）
φ_{ic}	打折力度
S_{ic}	损耗率

γ_c	价格弹性系数

表 3　模型变量参照表

5.3.3 模型的假设

为了以品类为补货单位结合营销总量及成本加成定价关系使超市营销收益最大化需要满足以下假设：

1. 在考虑超市运营实际情况下，保证供大于求，不会出现缺货断货等现象。即满足下列公式：

$$X_{(i-1)c} - n_{(i-1)c} + X_{ic} \geq n_{ic} \tag{1}$$

上式中 X_{ic} 代表 2023 年 7 月 i 日 c 蔬菜品类的补货量，n_{ic} 代表 2023 年 7 月 i 日 c 蔬菜品类的销售量。

2. 在满足超市正常运营的情况下，打折区间始终满足售价大于进货价要求。

$$W_{ic} \geq V_{ic} \tag{2}$$

上式中 W_{ic} 代表 2023 年 7 月 i 日 c 蔬菜品类的成本加成定价，V_{ic} 代表 2023 年 7 月 i 日 c 蔬菜品类的成本。

3. 假设 2023 年 6 月 30 日刚好售完所有单品。

4. 成本加成定价处于一定的弹性区间且存在上下界。

完全成本定价法，其中定价即为成本与加成的和。采用边际成本定价法 [3]，用以达到薄利多销的目的。

$$W = \frac{V\gamma_c}{(1 - a_{ic}\varphi_{ic})(1 - S_{ic})} \tag{3}$$

上式中的 γ_c 即为价格弹性系数，因为蔬菜类各单品的定价受行业等诸多因素影响，经过查阅大量资料找到 γ_c 较准确的波动区间，汇总于以下表格。

蔬菜品类	γ_{cmin}	γ_{cmax}
花叶类	1.1	1.8
花菜类	1.2	1.7
茄类	1.4	2.2
水生根茎类	1.5	2.6
辣椒类	1.4	2.3

食用菌类	1.8	3.2

接下来以 SVM，LDA 和 Decisio Tree 为基分类器 (且权重为 0.5:0.3:0.2) 构建 Voting 集成学习算法来预测 a_{ic},（0 为不打折，1 为打折）。

由此构建 W 成本加成定价的弹性变化区间函数如下：

$$\frac{V\gamma_{cmin}}{(1-a_{ic}\varphi_{ic})(1-S_{ic})} \leq W_{ic} \leq \frac{V\gamma_{cmax}}{(1-a_{ic}\varphi_{ic})(1-S_{ic})} \tag{4}$$

5.3.4 模型的构建

Step1: 将蔬菜类商品的品类作为基本单位，分别计算出每月该品类的销售总量 h[kg] 及每月的成本加成定价值 w[元/kg]（即 $w=\sum_{i=1}^{n} m_i p_i/h$, 其中 m 为销售量，p 为销售单价）。根据 w 和 h 的分布用 Matlab 进行一元非线性回归拟合，得到 6 个不同蔬菜品类的成本加成定价与品类销售总量的拟合函数如下：

1. 花叶类蔬菜：　$h(w)=72.74w^3-1115w^2+4905w+1.277$　　R 方：0.9103
2. 花菜类蔬菜：　$h(w)=1.455w^3-51.88w^2+485.8w+0.1323$　R 方：0.8839
3. 水生根茎类：　$h(w)=2.193w^3-72.93w^2+617.1w-0.008345$　R 方：0.8463
4. 茄类蔬菜：　$h(w)=2.354w^3-57.75w^2+395.3w-0.01305$　　R 方：0.9072
5. 辣椒类蔬菜：　$h(w)=3.477w^3-108.7w^2+955.9w+10.11$　　R 方：0.8583
6. 食用菌类蔬菜：　$h(w)=5.354w^3-136.7w^2+999.6w+0.2373$　　R 方：0.9133

R 方值（也称为决定系数）是用来表示拟合优度的量，其取值在 0 到 1 间。当 R 方值越接近 1，表示模型对数据的拟合得越好；而 R 方值越接近 0，则说明模型对数据的拟合效果较差。上述你和函数的 R 方均在 0.8 以上可以认为拟合程度良好。

将上述函数简化为如下公式：

$$n_{ic}=h_c(w_{ic}) \tag{5}$$

上式中 $h(w)$ 为成本加成定价与品类销售总量的拟合函数。

Step2: 对各品类蔬菜近七天的成本加成定价进行级比检验，经检验相关数据能通过级比检验。下表以水生根茎类为例。

索引项	原始值	级比值
1	8.802	-
2	9.565	0.92
3	9.734	0.983

Table 5 continued from previous page

索引项	原始值	级比值
4	10.567	0.921
5	8.912	1.186
6	10.581	0.842
7	10.553	1.003

表 5　水生根茎类成本加成级比对照表

从上表分析可以得到，原序列的所有级比值都位于区间（0.779，1.284））内，说明原序列适合构建灰色预测模型。

其后进行灰色预测模型的构建，首先生成灰色算子将无序的序列弱化并将其有序化用于规律分析。其后累加生成的算子求均值，进行可行性分析知道级比满足 $\sigma(k) \in (e^{-\frac{2}{n+1}}, e^{\frac{2}{n+1}})$ 则新算子 $V^{(0)}$ 和 $Z^{(1)}$ 可建立 GM(1,1) 模型。由此构建模型：

$$V_{ic}^{(0)} + aZ_{ic}^{(1)} = b \tag{6}$$

用微分方程确定估计 a,b 的值。

$$\frac{dv^{(1)}(t)}{dt} + av^{(1)}(t) = b$$

$$v^{(1)}(t) = (v^{(0)}(1) - \frac{b}{a})e^{-a(t-1)} + \frac{b}{a}$$

各品类蔬菜的灰色预测模型的各相关指标如下：

蔬菜品类	发展系数 a	灰色作用量 b	后验差比 c 值
茄类	0.024	4.505	0.032
水生根茎类	-0.018	9.236	0.131
花菜类	0.04	6.338	0.072
花叶类	0.031	5.007	0.109
辣椒类	0.028	4.321	0.003
食用菌类	-0.029	2.488	0.051

一般后验差比值 c 值小于 0.35 则模型精度高，c 值小于 0.5 说明模型精度合格，c 值小于 0.65 说明模型精度基本合格，如果 c 值大于 0.65，则说明模型精度不合格。

从上表分析可以得到，各品类蔬菜的成本加成定价的后差比值均小于 0.35，模型精度高。

Step3:EWMA 指数加权移动平均模型建立

$$X_{ic} =< A, X > \tag{7}$$

上式中 A 为基于前七天补货量数据确定的系数矩阵，即 $[\alpha_1, \alpha_2, \ldots, \alpha_7]$，且满足:

$$\sum_{i=1}^{7} \alpha_i = 1$$

X 为前七天的补货量矩阵，即 $[X_{(i-7)c}, X_{(i-6)c}, \ldots, X_{(i-1)c}]$，在系数确定的情况下用 N_i 表示已知数据（2023 年 6 月 24 日-2023 年 6 月 30 日）距离标本点的时间跨度，（其中距离标本点越近的数据点所对应的向量间系数权重越大）则以 $coef(i) = \frac{2}{N_i+1}$ 修正并且确定 A 矩阵的 7 个向量间的相对权重，而后进行单位化处理。

5.3.5 目标函数的构建

为了使超市收益最大化，建立收益函数 $f(X, W)$

$$max \quad f(X, W) = \sum_{i=1}^{7} \sum_{c=1}^{6} (n_{ic} W_{ic} - X_{ic} V_{ic})$$

综合上述公式（1～7）得到条件函数:

$$s.t. \begin{cases} X_{(i-1)c} - n_{(i-1)c} + X_{ic} \geq n_{ic} \\ W_{ic} \geq V_{ic} \\ W = \frac{V\gamma_c}{(1-a_{ic}\varphi_{ic})(1-S_{ic})} \\ n_{ic} = h_c(w_{ic}) \\ V_{ic}^{(0)} + aZ_{ic}^{(1)} = b \\ \frac{V\gamma_{cmin}}{(1-a_{ic}\varphi_{ic})(1-S_{ic})} \leq W_{ic} \leq \frac{V\gamma_{cmax}}{(1-a_{ic}\varphi_{ic})(1-S_{ic})} \\ X_{ic} =< A, X > \end{cases}$$

5.3.6 用人群搜索模型对算法求解过程进行优化

人群搜索模型能够直接智能地进行搜索行为，在传统搜索算法基础上更加智能、简便、准确。它能有效的运用描述、不确定性推理与模糊逻辑等限制规则来提升最终求解的准确度。

Step1: 搜索步长的确定

首先采用高斯隶属函数表示搜索步长 (模糊变量) 函数:

$$\mu(x) = e^{\frac{-(x-\mu)^2}{2\delta^2}}$$

上式中 $\mu(x)$ 为 x 变量的高斯隶属度，μ, δ 为隶属函数参数。

在上述步长函数输出值取值为 $[\mu-3\delta,\mu+3\delta]$ 之间时认为为最佳位置，且满足以下关系：

$$\begin{cases}\mu_i=\mu_{max}-\frac{S-I_i}{S-I}(\mu_{max}-\mu_{min}) & i=1,2,\ldots,S\\ \mu_{ij}=rand(\mu_j,1) & j=1,2,\ldots,D\end{cases}$$

上式中，μ_{ij} 为在第 j 惟搜索空间目标函数 i 地隶属度，I 为蔬菜单品排列编号。

由此进行模糊处理得到步长 α:

$$\alpha_{ij}=\delta_{ij}\sqrt{-ln(\mu_{ij})}$$

$$\delta_{ij}\text{即高斯隶属函数参数}\quad \delta_{ij}=\frac{(T_{max}-t)\left|x_{min}-x_{max}\right|}{T_{max}}$$

上式中，x_{max}、x_{min} 分别为同一子群的最大最小值，t 为迭代次数，T_{max} 为最大迭代次数。

Step2: 搜索方向的确定

首先对利己利他及预动的行为进行分析通过以下公式算出各行为方向：

$$\text{利己方向}\quad \vec{d}_{i,ego}(t)=\vec{p}_{i,best}-\vec{x}_i(t)$$

$$\text{利他方向}\quad \vec{d}_{i,alt}(t)=\vec{g}_{i,best}-\vec{x}_i(t)$$

$$\text{预动方向}\quad \vec{d}_{i,pro}(t)=\vec{x}_i(t_1)-\vec{x}_i(t_2)$$

综合各种因素考量，利用随机加权几何平均确定该次搜索的具体方向：

$$\vec{d}_i(t)=sign(\omega\vec{d}_{i,pro}+\iota_1)\vec{d}_{i,ego}+\iota_2\vec{d}_{i,alt}$$

上式中，$\vec{x}_i(t)$ 为 $\{\vec{x}_i(t-1),\vec{x}_i(t),\vec{x}_i(t+1)\}$ 中的最佳位置，g 为搜索个体所在邻域的集体最佳位置，p 为搜索个体所在的最佳位置。

Step3: 搜索位置更新

在确定完步长及方向后根据以下公式更新变动位置。

$$x_{ij}(t+1)=x_{ij}(t)+\alpha_{ij}(t)d_{ij}(t)$$

求解结果如下表

表 7　未来 7 天日补货量（单位：kg）

日期	花叶类	花菜类	水生根茎类	茄类	辣椒类	食用菌
1	211.2538632	29.28831568	38.67237	22.28013272	133.8801953	88.825727
2	210.9893832	32.13607365	38.08772	23.390424	133.6576146	87.32761283
3	212.7399964	34.29769328	34.42682	19.93066184	130.0198001	88.07388986

Table 7 continued from previous page

日期	花叶类	花菜类	水生根茎类	茄类	辣椒类	食用菌
4	213.2150539	33.4767588	34.54998	19.66254923	133.1047757	91.16814695
5	212.8418747	31.85310523	39.28932	21.02739804	125.6141906	88.58590515
6	210.1612189	32.04869407	37.40249	20.01283748	124.8833674	86.93454343
7	214.262705	30.30650594	35.42157	20.12786611	126.5184536	89.86645845

表 8　未来 7 天日定价（单位：元/kg）

日期	花叶类	花菜类	水生根茎类	茄类	辣椒类	食用菌
1	6.173785109	11.01939773	12.57725	8.554897026	13.34556008	7.661487616
2	5.754395356	11.61281552	13.32669	9.242079795	13.43740775	8.166194084
3	6.375909069	12.88737227	13.67294	8.959759827	12.68652817	7.882385336
4	5.838389812	11.84978817	12.55654	9.111842012	14.40612926	8.345406838
5	6.180787562	11.69959473	13.33946	9.026011768	13.59761041	7.607432216
6	6.24014315	10.49521177	14.87661	9.438853906	13.73740775	8.288874925
7	5.938243364	10.14362845	14.079	10.23777205	13.33740775	8.460542397

5.4 问题三的求解

5.4.1 问题三的求解思路

首先对 6 月 24 日-6 月 30 日的销售数据进行处理，找出近期商超的可售品种。对所有的可售品种建立评价模型找出对商超经营保障最重要的 33 个蔬菜单品。其后对此 33 个单品于各自品类内做偏最小二乘回归分析，找出各单品销售总量和相同品类内所有单品成本加成定价的关系。以顾客满意度最大化和收益最大化为目标建立多目标规划模型，确立商超 7 月 1 日单品补货量与定价策略。

5.4.2 评价指标的确定

需要给出各蔬菜单品的盈利分数，首先需要根据数据分析提取出个单品的评价指标，在此我们给出 6 个评价指标：销售次数（f）、损失率（S）、折扣率（$n(\varphi)=\frac{n(a_{cm}=1)}{n(a_{cm})}$）、总利润（$\sum_{i=1}^{T} w_{cm}$）、交易变动率（$\eta_n=\frac{n_t-n_{t-1}}{n_{t-1}}$）、利润增长率（$\eta_w=\frac{w_t-w_{t-1}}{w_{t-1}}$）。

5.4.3 构建熵权法-灰色关联法优化的 Topsis 模型

要确定具体售货单品需要对各蔬菜类单品进行收益性评价，通过熵权法得到单品的权重，再运用灰色关联法计算评价各蔬菜单品的关联性，最后应用优劣解距法对蔬菜单

品进行排序。[4]

Step1: 对原始数据进行级差化处理

构建个蔬菜类单品的评价指标决策矩阵 $A=(a_{ij})$，其中 a_{ij} 表示第 i 个蔬菜单品的第 j 个评价指标（$i\in[1,m],j\in[1,n]$）。其后对矩阵 A 中数据进行级差化处理得到新的矩阵 $S=(s_{ij})_{m\times n}$。

$$s_{ij}=\max\left\{\frac{a_{ij}-\min_i[a_{ij}]}{\max_i[a_{ij}]-\min_i[a_{ij}]},\frac{\max_i[a_{ij}]-a_{ij}}{\max_i[a_{ij}]-\min_i[a_{ij}]}\right\}$$

Step2: 应用熵权法计算个评价指标的熵权

假设 f_{ij} 为第 i 个单品的第 j 个评价指标权重，h_j 为第 j 个评级指标的熵值，（m 为蔬菜单品的数目）计算公式如下:

$$f_{ij}=\frac{s_{ij}}{\sum_{i=1}^{m}s_{ij}}$$

$$h_j=-\frac{1}{lnm}\sum_{j=1}^{m}f_{ij}ln(f_{ij})$$

计算第 j 个评价指标的熵权 w_j:

$$w_j=\frac{1-h_j}{\sum_{j=1}^{n}(1-h_j)}=\frac{1-h_j}{n-\sum_{j=1}^{n}h_j}$$

得到评价指标加权后的集合 $Z=\{z_{ij}\}$ 其中 $z_{ij}=s_{ij}\times w_j$ 且 $i\in[1,m],j\in[1,n]$

Step3: 确定最优方案解和最劣方案解

定义: 选取蔬菜单品的最优方案矩阵为 Z^+, 最劣方案矩阵为 Z^-, 第 i 个评价对象与最大值的距离为 d_i^+(这里运用理想解欧氏距离表示), 最小值的距离为 d_i^-，N_i' 为第 i 个评价对象未归一化的得分,N_i 为第 i 个评价对象归一化之后的得分:

$$Z^+=(max\{z_{11},z_{21},\ldots,z_{m1}\},max\{z_{12},z_{22},\ldots,z_{m2}\},\ldots,max\{z_{1n},z_{2n},\ldots,z_{mn}\})$$

$$Z^-=(min\{z_{11},z_{21},\ldots,z_{m1}\},min\{z_{12},z_{22},\ldots,z_{m2}\},\ldots,min\{z_{1n},z_{2n},\ldots,z_{mn}\})$$

上式中 Z^+ 表示 z_{ij} 第 i 个蔬菜品类评价指标最大值，Z^- 表示 z_{ij} 第 i 个蔬菜品类评价指标最劣值。

$$d_i^+=\sqrt{\sum_{j=1}^{n}(Z_j^+-z_{ij})^2},\qquad d_i^-=\sqrt{\sum_{j=1}^{n}(Z_j^--z_{ij})^2}$$

$$N_i'=\frac{d_i^-}{d_i^++d_i^-},\qquad N_i=\frac{N_i'}{\sum_{i=1}^{m}N_i'}$$

Step4: 计算蔬菜单品评价方案与理想解的灰色关联度

定义：正向理想解指标绝对值差矩阵为 $R^+=(r_{ij}^+)_{m\times n}$，负向理想解指标绝对值差矩阵为 $R^-=(r_{ij}^+)_{m\times n}$，且 ρ 为分辨系数。

$$r_{ij}^+=\frac{\min_i\min_j\left|Z_j^+-z_{ij}\right|+\rho\max_i\max_j\left|Z_j^+-z_{ij}\right|}{\left|Z_j^+-z_{ij}\right|+\rho\max_i\max_j\left|Z_j^+-z_{ij}\right|}=\frac{\rho w_j}{w_j-z_{ij}+\rho w_j}$$

$$r_{ij}^{-}=\frac{\min_i\min_j\left|Z_j^{-}-z_{ij}\right|+\rho\max_i\max_j\left|Z_j^{-}-z_{ij}\right|}{\left|Z_j^{-}-z_{ij}\right|+\rho\max_i\max_j\left|Z_j^{-}-z_{ij}\right|}=\frac{\rho w_j}{z_{ij}+\rho w_j}$$

知道分辨系数 ρ 的取值为 $\rho\in(0,1)$, 且 ρ 越接近于 0 分辨能力越显著，通常认为 $\rho=0.5$ 的时候为最佳取值。

计算最优评价方案与理想解的关联程度，见以下公式：

$$R_i^{+}=\frac{1}{n}\sum_{j=1}^{n}r_{ij}^{+};R_i^{-}=\frac{1}{n}\sum_{j=1}^{n}r_{ij}^{-}$$

将上述结果进行无量纲操作：

定义：D_i^{+} 和 D_i^{-} 为欧氏距离的无量纲符号；R_i^{+} 和 R_i^{-} 用以表示关联度无量纲符号。

$$D_i^{+}=\frac{d_i^{+}}{\max_i l_i^{+}}\quad D_i^{-}=\frac{d_i^{-}}{\max_i d_i^{-}}\quad R_i^{+}=\frac{r_i^{+}}{\max_i r_i^{+}}\quad R_i^{-}=\frac{r_i^{-}}{\max_i r_i^{-}}$$

Step5: 计算评价方案与理想方案的贴近成度

用 $\eta_i^{+}=\kappa D_i^{+}+\varphi R_i^{-}$和$\eta_i^{-}=\kappa D_i^{-}+\varphi R_i^{+}$ 来衡量方案的接近度，其中 κ 和 φ 表示决策位置及样本数据偏好度。有以下公式 (t 越小则评价方案越优秀)

$$t_i^{+}=\frac{\eta_i^{+}}{\eta_i^{+}+\eta_i^{-}}$$

评价结果如下表所示：

表 9　蔬菜单品种类选择结果

分类名称	名称	评分
水生根茎类	菱角	0.16281926
	高瓜 (1)	0.24496213
	洪湖藕带	0.49155012
	净藕 (1)	0.52611758
食用菌	海鲜菇 (包)	0.21673604
	金针菇 (盒)	0.32085503
	双孢菇 (盒)	0.34911212
	西峡花菇 (1)	0.79973526
茄类	青茄子 (1)	0.13513941
	长线茄	0.36811198
	紫茄子 (2)	0.58491016
辣椒类	七彩椒 (2)	0.13438196
	姜蒜小米椒组合装 (小份)	0.24098078
	小皱皮 (份)	0.27575016

Table 9 continued from previous page

分类名称	名称	评分
	红椒 (2)	0.28356637
	螺丝椒 (份)	0.39868248
	螺丝椒	0.58422336
	芜湖青椒 (1)	0.7958546
	小米椒 (份)	0.82701378
花叶类	菠菜	0.10795481
	菠菜 (份)	0.20252995
	红薯尖	0.25040906
	上海青	0.25378272
	小青菜 (1)	0.2693461
	木耳菜	0.31996536
	奶白菜	0.32133033
	苋菜	0.39252678
	娃娃菜	0.44521919
	竹叶菜	0.59505038
	云南油麦菜 (份)	0.67756919
	云南生菜 (份)	0.99090248
花菜类	枝江青梗散花	0.25119591
	西兰花	1

5.4.4 构建 kano 模型量化市场对蔬菜的满意度

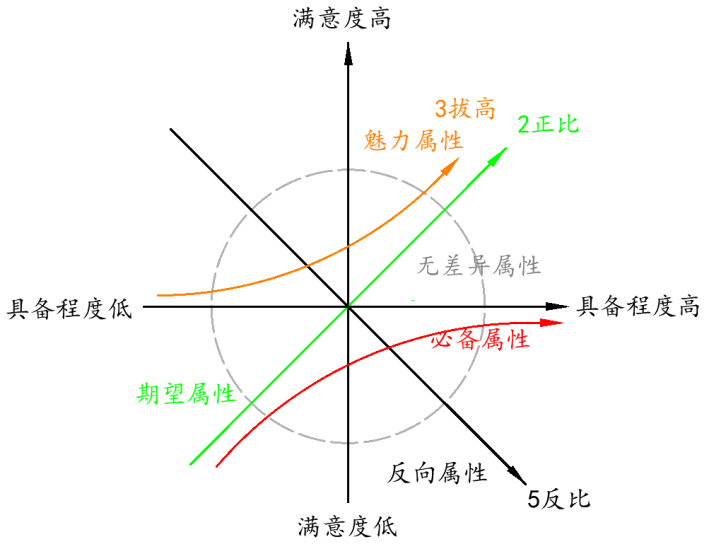


图 9　kano 检验曲线

根据 kano 模型一般可以讲顾客满意度进行如下分层：

满意度情况	衡量指标
满意	恰好满足供需相抵，完美符合魅力曲线（即如需求量为 5kg 土豆，则土豆的补货量至少有 5kg）
比较满意	未满足供需相抵，但是仍有同品类商品对需求量进行补充（即如需求量为 5kg 土豆，补货量仅有 4kg 土豆，可用 1kg 红薯对其进行补充。注意：此处应优先选择与土豆关联系数最高的单品，相关性系数越强补充后的满意度越高更匹配魅力曲线，稍稍偏向必备曲线）
不满意	物资短缺情况无同品类单品可以补充货物，低于必备曲线，记满意度为 0

表 10　基于 kano 模型的满意度分层表

为此我们制定如下的客户满意度步骤：

Step1: 选定某一品类蔬菜（如：花叶类、花菜类等），用上述的评价模型给该品类中各单品 (m) 进行打分，其后依据各自得分将单品进行降序排列。

Step2: 按照该单品的需求量售出商品，如若库存量（即补货量）充足，则售出相应数目的现有存货。并且根据是否满足消费者购物需求建立满意度评分函数，公式如下：

$$\text{满意度评分：}\ Score' = 1 \times \sum_{i=1}^{m} n \qquad \text{n 为销售量的总和}$$

Step3: 但是考虑到当顾客对某一蔬菜类单品的需求量大于现有的库存量（即补货量 x），则出售所有的现有存货并且遍历该单品同品类，找出与该单品相关性系数最大的且仍有剩余的单品进行补全剩余需求量。若一次遍历未能补全需求量则重复上述过程，直至需求量全部满足或者该品类所有单品都被遍历完毕后结束。由此将上述满意度得分函数改进为如下公式：

$$\text{满意度评分：}\ Score' = 1 \times \sum_{i=1}^{m} n + con_{m_i m_j} \times n_{m_i m_j}$$

上式中，n 为销售量的总和，$con_{m_i m_j}$ 为 m_i 单品与 m_j 单品的关联性系数，$n_{m_i m_j}$ 为在 m_i 单品缺货的情况下 m_j 商品对其的补充销售量。

以下附上食用菌蔬菜品类中各单品的相关系数热力图作为参考，其他 5 类蔬菜品类相关性热力图详见附件。

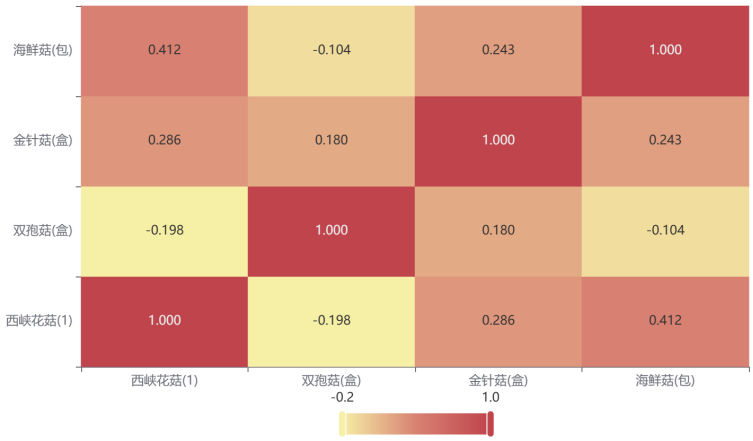


图 10　食用菌相关系数热力图

Step4: 若遍历完所有的同一品类的蔬菜类单品仍未满足需求量则，仍未满足的需求部分满意度评分归零。

以下附上顾客满意度分层对照表格。

5.4.5 条件函数

1. 由问题可知要求各单品的订购量达到最小陈列量为 2.5kg 则该单品的补货量中除去损耗与出售后的陈列量要至少为 2，5kg：

$$X_{cm}(1 - s_{cm}) - n_{cm} \geq 2.5 \tag{8}$$

上式中，X_{cm} 代表 c 蔬菜品类第 m 个编号的蔬菜单品补货量，s 为该单品的损耗率，n 为销售量。

2. 在超市正常运营的情况下，打折区间始终满足售价大于进货价要求。

$$w_{cm} \geq V_{cm} \tag{9}$$

上式中 w_{cm} 代表 c 蔬菜品类第 m 个编号的单品成本加成定价，V 代表蔬菜品类的成本。

3. 根据 w 和 h 的分布用 Matlab 进行一元非线性回归拟合，得到不同蔬菜品类的成本加成定价与品类销售总量的拟合函数如下：

$$n_{cm} = h_c(w_{cm}) \tag{10}$$

4. 成本加成定价处于一定的弹性区间且存在上下界。

$$\frac{V_{cm}\gamma_{cmin}}{(1-a_{cm}\varphi_{cm})(1-S_{cm})} \leq W_{cm} \leq \frac{V_{cm}\gamma_{cmax}}{(1-a_{cm}\varphi_{cm})(1-S_{cm})} \tag{11}$$

上式中的 φ_{cmin} 即为定价弹性系数的波动最小值，φ_{cmax} 即为定价弹性系数的波动最大值。

5. 为了将数据将无序的序列弱化并将其有序化用于规律分析，进行灰色预测模型的构建：

$$V_{cm}^{(0)} + aZ_{cm}^{(1)} = b \tag{12}$$

6.EWMA 指数加权移动平均模型建立

$$X_{cm} =< A, X > \tag{13}$$

上式中 A 为补货量数据确定的系数矩阵，即 $[\alpha_1, \alpha_2, \ldots, \alpha_n]$，且满足: $\sum_{i=1}^{n} \alpha_i = 1$；X 为补货量矩阵，即 $[X_{(i-n)c}, X_{(i-n-1)c}, \ldots, X_{(i-1)c}]$，在系数确定的情况下用 N_i 表示已知数据距离标本点的时间跨度，则以 $coef(i) = \frac{2}{N_i+1}$ 修正。

5.4.6 目标函数的构建

1. 为了使超市营销满意度最大化，建立满意度函数：

$$max \sum_{c=1}^{6} Score' \quad c \in [1,6]$$

2. 同时为了使超市收益最大化，建立收益函数：

$$max \quad f(X,W) = \sum_{c=1}^{6} \sum_{m=1}^{m_c} (n_{cm}W_{cm} - X_{cm}V_{cm})$$

上式中 $\sum_{m=1}^{m_c}$ 表示 c 品类所有编号的单品求和。

综合上述条件公式（8～13）得到条件限制函数:

$$s.t.\begin{cases} X_{cm}(1-s_{cm})-n_{cm}\geq 2.5 \\ w_{cm}\geq V_{cm} \\ n_{cm}=h_c(w_{cm}) \\ \frac{V_{cm}\gamma_{cmin}}{(1-a_{cm}\varphi_{cm})(1-S_{cm})}\leq W_{cm}\leq \frac{V_{cm}\gamma_{cmax}}{(1-a_{cm}\varphi_{cm})(1-S_{cm})} \\ V_{cm}^{(0)}+aZ_{cm}^{(1)}=b \\ X_{cm}=<A,X> \end{cases}$$

5.4.7 运用多目标粒子群算法对求解算法进行优化

这里运用多目标粒子群算法找到 parato 的最优解集 [5]。它与传统粒子群算法通过对信息的遍历，向着全局最佳位置更新信息，比之更优化的是应用不止单一变量来确定最佳位置。先储存粒子群的综合 Pareto 最优解其后对解进行更新: 若未储存最优解解，则将当前粒子群的 Pareto 最优解放入其中; 若已经储存最优解，则将当前最优解与档案中的解进行比对，若新解被储存解任意支配，则放弃该解; 反之则将新解储存起来; 若新解能够任意支配已储存的解，则将原始解删除。若档案中最优解不为一则随机选择一个确定为最优变更位置。依据此办法不断迭代最终得出答案。

表 11　蔬菜单品定价与补货量

分类	名称	定价	补货量	利润
水生根茎类	净藕 (1)	15.748823	10.51843075	293.55
	高瓜 (1)	12.969019	4.530557032	
	菱角	14.168374	3.683939787	
	洪湖藕带	20.133595	5.223114724	
食用菌	西峡花菇 (1)	6.3243272	4.521290823	121.82
	双孢菇 (盒)	4.5206305	15.67429315	
	金针菇 (盒)	2.2204222	18.7856615	
	海鲜菇 (包)	2.4674257	10.80129367	
茄类	紫茄子 (2)	5.6172377	17.53440397	161.28
	青茄子 (1)	5.1597989	4.263043234	
	长线茄	13.330291	5.640529348	
辣椒类	螺丝椒	12.129266	5.893900819	419.63
	芜湖青椒 (1)	4.6831025	22.35798045	
	小米椒 (份)	5.6737831	27.61772755	
	小皱皮 (份)	1.942506	10.84266232	
	螺丝椒 (份)	4.2204192	17.70160565	

Table 11 continued from previous page

分类	名称	定价	补货量	利润
	七彩椒 (2)	15.621525	1.585546042	
	姜蒜小米椒组合装 (小份)	3.3794271	9.016375837	
	红椒 (2)	20.12555	2.337956782	
花叶类	苋菜	3.9318525	10.39889446	612.33
	竹叶菜	3.6716511	16.13825908	
	上海青	7.1398956	5.106072208	
	木耳菜	4.9526125	9.789158079	
	菠菜	14.725768	1.451461962	
	娃娃菜	6.7339377	10.47216431	
	红薯尖	5.0197851	6.247602747	
	奶白菜	4.4761967	9.748876557	
	小青菜 (1)	5.8267905	6.306702375	
	云南生菜 (份)	4.7338275	48.35100995	
	云南油麦菜 (份)	4.8612566	24.92170994	
	菠菜 (份)	5.9555208	10.37975103	
花菜类	西兰花	12.335184	20.39961737	242.31
	枝江青梗散花	11.701756	5.702923158	

5.5 问题四的求解

5.5.1 模型的假设

1. 假设补货的时间间隔以天为单位，即一天补一次货。
2. 考虑以 t（天数）为单位的蔬菜品质亏损。
3. 补货同期内允许缺货现象发生，再次情况下可由其他同品类单品补充。
4. 在考虑超市运营实际情况下，保证供大于求，不会出现缺货断货等现象。
5. 在满足超市正常运营的情况下，打折区间始终满足售价大于进货价要求。
6. 成本加成定价处于一定的弹性区间且存在上下界。

5.5.2 相关数据采集

1. 单品存储费用 h_s: 在此前的问题中我们没有考虑在销售和补货的过程中滞留的货物有存储的成本，在第四问中我们可以市场调查和参考现有论文得到所给单品在单位时间单位质量下的存储费用。

2. 因销售尚未售出商品而未进新商品造成的单位时间单位质量损失 q: 我们在考虑实际销售的过程中，市场往往面临这样的决策，到底是继续卖现有的陈列（有可能因为质量下降，需求量和利润都降低）的商品，还是丢弃或者退回这些商品而补充新商品–在不同的情况下选择其中一种方式才能保证利润最大化。因此我们需要引入因售卖剩余商品而造成的这种隐形损失（也即沉没成本 q）来量化利润的增减。这个数据可以通过市场调查，观察多个商铺长期以来采取不同决策售卖同一单品后的利润偏差来得到。

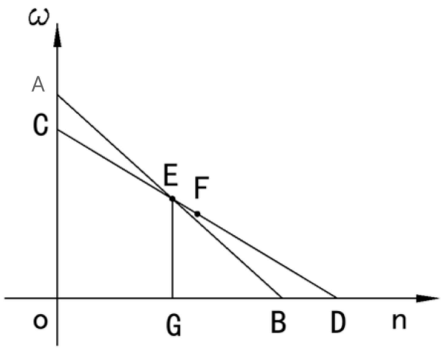
3. 各单品随着库存时间增长因氧化，风干，虫蛀等因素而造成品质下降直至不符合市场售卖标准的程度 $\theta(t)$: 此前的模型虽然考虑了降价打折和损耗率，但是受到已有数据的局限性较大，必须基于近期数据通过插值，拟合，预测的操作来判断在所规定周期内是否会打折是否有损耗。然而分析其本质，如果我们知道了单品在库存周期内品质下降的规律，就可以更加精准地判断打折力度，库存成本等因素。该数据可通过查找各类单品腐败变质规律的生物学实验数据得到。

4. 各单品在库存周期中开始变质时间的平均值 t_1 和各单品变质到无法达到市场售卖标准时间的平均值 t_2. 它们可以用来限制我们选取的变质模型的范围，这两样数据同样可以通过生物学实验数据得到。

5. 需求变量 Ω：缺货引起的信誉损失，缺货会引起一定程度的信誉损失，改变客户对单品乃至商家的购买偏好，也是我们需要考虑的因素，综合查找对信誉损失的论文我们可以为各个单品确定与缺货呈正相关的信誉损失量 Ω

5.5.3 模型建立

1. 定价和销售量的线性关系



经济学中，定价和售出量往往是相互制约的两个要素，在不同条件下，只有选取薄利多销和多利薄销中的一种策略才可以实现利润的最大化。较为典型的如香水这样的商品，商品的生命周期长，成本高，不易变质且售出频率低：这时就应该采取抬高利润降低销量的销售方法，因为陈列的香水生命周期长，纵然销售速度慢，在生命周期内总是可以慢慢卖完的，且商品的变质周期低。再者商品的售出群体少，售出频率低，只有抬高定价才能实现利润的最大化。而本题考虑的蔬菜类恰恰相反：生命周期短，容易变

质，售出频率高且成本低廉，这时就需要采取薄利多销的策略，通过降低利润抬高交易频率，快速在生命周期中售出单品才能实现利润的最大化。经济学中认为销量和定价有较好的线性拟合关系，图中的直线就是定价和销售量的线性回归线。如果 AB 是正常的定价直线，那么 CD 就是采取薄利多销策略以后的定价直线，E 为二者的交点，从 A 点到 C 点降低了定价，于是 B 点到 D 点就抬高了销售量。AC 的降价幅度越大，销售量 BD 的增长幅度也就越大，根据经济学原理，收入要达到最大必须使得实际销售量和定价落在 ED 线段上，以 F 为例，显然，当 E 向 F 移动时，定价降低而销量增加。而综合考量我们前文的边际成本效应，更证实了只有 F 向 D 点偏移时，才能实现我们薄利多销的销售战略。

2. 定价模型的确立

根据前文的回归直线，我们可以引入线性相关性参数 d, 衡量销售量的增长量和定价的减少量之间的关系:

$$\Delta n_t = d(w\varphi_{t-1} - w\varphi_t)$$

其后，我们来线性拟合销量的增量假定 T 为我们要制定补货量和定价策略的周期，那我们对第一天到最后一天 T 进行求和，将 Δn_t 累加，并考虑此前给出的销量 n 和成本加成定价 w 的三次拟合函数 $h(w)$ 对在 w 的改变下对销量的作用，得到下式:

$$\Delta n_{t-t_1} = \sum_{t=1}^{T}[\Delta h(w_t) + d(w\varphi_{t-1} - w\varphi_t)]$$

然后我们考虑利润衰减，它也是 t 从第一天到周期结束的累加，随着销量的增加，利润衰减就体现在我们得到的数据：单品存储费用 h_s 和因售出现有货物而导致无法购进新货的沉没成本 q 上，将它们对时间和质量累计，就得到了我们的利润衰减量。考虑到利润衰减量可能会受到时间因素的干扰，同时沉没成本 q 的统计量在时间预测的后期会受到波动。为了调整模型的真实性和稳定性。利用前文的加权移动平均模型为其做出权重修正，得到下式:

$$\sum_{t=1}^{T}[htStore(t) + qtStore(t)] \times (\frac{2}{1+N_t})^t$$

上式中引入加权移动平均参数 $(\frac{2}{1+N_t})^t$ 使模型的稳健性得到提高。

接下来考虑我们的模型中尚未确定的参数,除了后文中需要建立的库存函数 $Store(t)$ 外，在 $n = h(w) = aw^3 + bw^2 + cw + C_1$ 中我们也需要考虑三次项系数 a, 二次项系数 b 和一次项系数 c 的选取. 至于 $h(w)$ 最后的常数项 C_1，由于我们对 $\Delta h(w)$ 的计算方式即对时间的积分:

$$\Delta h(w) = \int_{t-1}^{t} h'(w)dt$$

所以在求导的过程中已经消去了 C_1，不会影响规划模型的最终结果，所以我们只考虑对参数 a,b,c,d 的求解。利用此前的销售数据，进行遗传算法和粒子群算法的综合求解，

设定: $pop-size=50$, $child-size=100$, 交叉率 $P_c=0.7$, 变异率 $P_m=0.3$, 迭代代数为 500 次粒子群算法参数设定: $c1=c2=0.2$, 最大迭代次数 500 得到以下参数结果:

表 12　参数运算结果

参数	a	b	c	d
食用菌类	6.34	-120.7	980.3	110.7
辣椒类	3.58	-100.6	976.1	101.2
茄类	2.45	-60.93	400.5	33.8
水生根茎类	1.7	-70.91	612.1	67.2
花菜类	2.39	-52.6	512.3	42.6
花叶类	81.55	-1135.76	4866.4	209.6

到此为止我们可以给出两个目标函数:

$$max \quad \Delta n_t = d(w\varphi_{t-1} - w\varphi_t)$$

$$min \quad \Delta n_{t-t_1} = \sum_{t=1}^{T}[\Delta h(w_t) + d(w\varphi_{t-1} - w\varphi_t)]$$

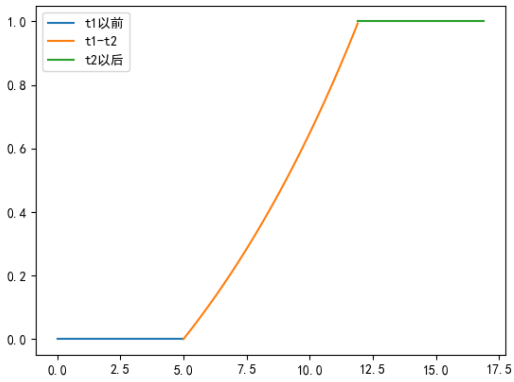
接下来我们确定补货模型：首先，根据我们收集到的单品品质下降时间过程，我们可以列出品质衰减的分段函数，在 $0-t_1$ 时间段内的单品品质没有发生变化，我们可以认为品质衰减程度是 0，在 t_1-t_2 时间段内，观察数据可以发现，单品的品质衰减呈现出先缓慢衰减而后急剧衰减的过程，符合良好的指数函数性质，于是我们采用:

$$\theta(t) = ae^{b(t-t_1)} + c \quad t_1 < t < t_2$$

拟合之，当时间超过 t_2。单品已经无法为市场所接受，此时认为单品的品质已经完全衰减，记衰减度为 1，最终我们得到衰减函数为:

$$\theta(t) = \begin{cases} 0 & t \le t_1 \\ ae^{b(t-t_1)} + c & t_1 < t < t_2 \\ 1 & t \ge t_2 \end{cases}$$

其绘图示意图如下所示。



然后考虑 $Store(t)$ 的分段函数，由于 $Store(t)$ 基本满足上一阶段的存货加上补货减去售出的形式，考虑商品质量下降对存货的影响，给出存货的函数如下：

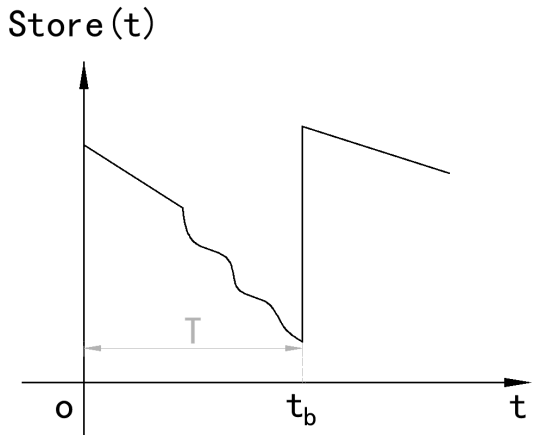
$$Store(t)=\begin{cases} X_t-\sum_{i=0}^{t}h(w_t)+Store(t-1) & t\le t_1 \\ Store(t_1)[1-ae^{b(t-t_1)}-c]+X_{t-t_1}-\sum_{i=1}^{t}h(w_t) & t_1<t<t_2 \\ Store(t_2)+X_{t-t_2}-\sum_{i=t_2}^{t}h(w_t) & t\ge t_2 \end{cases}$$

同时，我们需要利用附件数据计算出各单品较为稳定的需求率，即销售量和补货量之比，用 r 表示，r 的最大值记为最大需求率 M（一般为 1）。

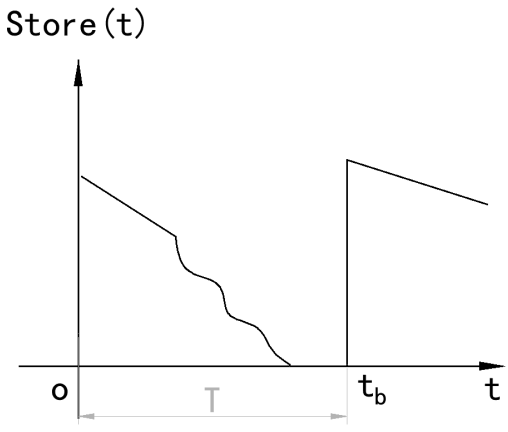
$$\text{需求率 r：}\ r=\frac{n}{x_t}=\frac{h(w)}{x_t}$$

$$\text{M 为最大需求率：}\ M=r_{max}$$

我们要了解补货的两种情况，一种是在货物终断前就完成了补货（如 a 图所示），一种是货物中断以后一段时间再完成补货（如 b 图所示）。



(a)

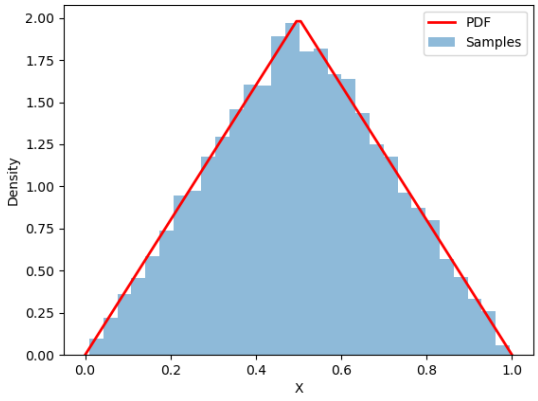


(b)

两种不同情形会对 $\theta(t)$ 的考量产生影响，同时间隔补货也会导致销售机会的缺失，这是需要我们考量的地方。不过，我们始终设定 t_b 为最近一次补货的时间点，将上一次补货的时间点记为原点 0, 把两次补货的间隔记作一个周期。

5.5.4 模型建立

考虑单品损耗满足一定的概率分布，引入概率密度 z，在 $[0, T_b]$ 上呈三角分布，分布函数为 $G(Z)$ 示意图如下：



于是，考虑两种不同的补货情况并分别积分，我们可以得到一个周期的库存总量的期望为：

$$Store(t) = \int_0^{t_2} \left\{ \frac{1}{2} r t_1^2 + r t_1 \int_{t_1}^{t_2} (a e^{b(t-t_1)} + c) dt + \int_{t_1}^{t_b} r\theta(t) \times [\int_t^{t_2} \theta(t) du] dt \right\} f(z) dz$$
$$+ \int_{t_2}^{t_b} \left\{ \frac{1}{2} r t_1^2 + r t_1 \int_{t_1}^{t_2} (a e^{b(t-t_1)} + c) dt + \int_{t_1}^{t_b} r\theta(t) \times [\int_t^{t_2} \theta(t) du] dt \right\} f(z) dz$$

丢失机会的期望函数为：

$$Loss(t) = \int_{t_2}^{t_b} (t - t_2) f(z) dz$$

于是，在每一时刻确定定价的情况下，忽略定价这一定量参数的影响，考虑信誉损失，丢失机会，以及存储费用，我们的收益期望为：

$$R(t) = \Omega n_{t_1 - t_2} - ht \times Store(t) - \int_0^{t_1} \theta(t) Store(t) dt - \Omega Loss(t)$$

收益期望最大化就是我们的第三个目标函数：

$$max R(t)$$

5.5.5 模型求解

$$max \quad \Delta n_t = d(w\varphi_{t-1} - w\varphi_t)$$
$$min \quad \Delta n_{t-t_1} = \sum_{t=1}^{T} [\Delta h(w_t) + d(w\varphi_{t-1} - w\varphi_t)]$$

$$
max \quad R(t)
$$

$$
s.t.\begin{cases}
\Delta n_t = d(w\varphi_{t-1} - w\varphi_t) \\
\Delta n_{t-t_1} = \sum_{t=1}^{T}[\Delta h(w_t) + d(w\varphi_{t-1} - w\varphi_t)] \\
\sum_{t=1}^{T}[htStore(t) + qtStore(t)] \times (\frac{2}{1+N_t})^t \\
\Delta h(w) = \int_{t-1}^{t} h'(w)dt \\
\theta(t) = \begin{cases} 0 & t \le t_1 \\ ae^{b(t-t_1)} + c & t_1 < t < t_2 \\ 1 & t \ge t_2 \end{cases} \\
Store(t) = \begin{cases} X_t - \sum_{i=0}^{t} h(w_t) + Store(t-1) & t \le t_1 \\ Store(t_1)[1 - ae^{b(t-t_1)} - c] + X_{t-t_1} - \sum_{i=1}^{t} h(w_t) & t_1 < t < t_2 \\ Store(t_2) + X_{t-t_2} - \sum_{i=t_2}^{t} h(w_t) & t \ge t_2 \end{cases} \\
r = \frac{n}{x_t} = \frac{h(w)}{x_t} \\
Store(t) = \int_0^{t_2} \left\{ \frac{1}{2}rt_1^2 + rt_1 \int_{t_1}^{t_2} (ae^{b(t-t_1)} + c)dt + \int_{t_1}^{t_b} r\theta(t) \times [\int_t^{t_2} \theta(t)du]dt \right\} f(z)dz \\
+ \int_{t_2}^{t_b} \left\{ \frac{1}{2}rt_1^2 + rt_1 \int_{t_1}^{t_2} (ae^{b(t-t_1)} + c)dt + \int_{t_1}^{t_b} r\theta(t) \times [\int_t^{t_2} \theta(t)du]dt \right\} f(z)dz \\
Loss(t) = \int_{t_2}^{t_b} (t - t_2) f(z)dz \\
R(t) = \Omega n_{t_1 - t_2} - ht \times Store(t) - \int_0^{t_1} \theta(t) Store(t)dt - \Omega Loss(t)
\end{cases}
$$

六、 模型评价与推广

优点:

1. 本文充分考虑了数据的特点和商品的特质，采用了季节性和周期性的时间序列分析模型观察商品销售特性。同时，在确定规划模型的过程中，我们也基于商品季节性，生命周期短等特质，对症下药地给出了边际成本的加成定价，薄利多销的销售策略，基于商品变质过程的补货周期等策略，贴近生产与销售实际。

2. 本文建模前广泛使用了检验过程，例如 SARIMA 模型，我们对每类样品做了 ADF 检验，通过了平稳性检验。同时，对灰色预测我们也做了级比检验，对 EWMA 做了偏差修正等等，确保了我们的模型真实稳健。

3. 本文在建模过程中有许多创新之处，比如巧妙地运用网格搜索和赤池信息量准则确定 SARIMA 模型的参数。针对大量数据用高斯混合聚类代替传统的聚类方式，引入价格弹性系数的成本加成定价法确定定价的范围。用熵权法-灰色关联法优化的 Topsis 模型选定单品，同时构造 Kano 模型量化满意度，打造双目标进行规划问题。以及引入信誉损失，品质下降函数等变量构建季节性商品的定价补货周期模型。

4. 本文广泛采用了优化算法确定参数并通过敏感性分析才确定了最优参数，模型的可接受度较高，可推广性强。

缺点:

1. 本文对数据的要求量较高，并且数据的连续性要求较强，对于离散的，时间跨度低的数据，模型无法达到较好的适用效果。

2. 本文对于弹性系数以及变质函数等变量的确定对现有科研成果和论文支撑的采用较多，缺少对生鲜商超这类特定地点的实际考察，可能与实际生产销售的情况有偏差。

参考文献

[1] 杨海民, 潘志松, 白玮. 时间序列预测方法综述 [J]. 计算机科学,2019,46(01):21-28.

[2] Fieller, E.C.; Hartley, H.O.; Pearson, E.S. (1957) Tests for rank correlation coefficients. I. Biometrika 44, pp. 470-481

[3] 武亚军, 价格理论与实践 [J]. 价格理论与实践,1997(01):1-3.

[4] 张于贤, 黄鑫, 韩文胜等. 基于熵权-灰色 TOPSIS 方法的农产品绿色物流发展评价研究及应用 [J]. 江苏农业科学,2018,46(16):319-322.DOI:10.15889/j.issn.1002-1302.2018.16.074.

[5] 胡旺,Gary G. YEN, 张鑫. 基于 Pareto 熵的多目标粒子群优化算法 [J]. 软件学报,2014,25(05):1025-1050.DOI:10.13328/j.cnki.jos.004496.

七、 附件

7.1 平稳性检验.py

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
plt.rcParams["font.sans-serif"]=["SimHei"] #设置字体
plt.rcParams["axes.unicode_minus"]=False #正常显示负号

original_data = pd.read_excel('时间测试.xlsx', index_col='销售日期')

from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.stattools import adfuller
from arch import arch_model

def test_constant_mean(data):
```

```
    # 计算移动平均
    rolmean = data.rolling(window=12).mean()

    # 进行ADF检验
    result = adfuller(data)

    print('ADF Statistic:', result[0])
    print('p-value:', result[1])
    print('Critical Values:')
    for key, value in result[4].items():
        print(f'{key}: {value}')

def test_stable_variance(data):
    # 计算条件方差
    model = arch_model(data)
    results = model.fit(disp='off')
    cond_variance = results.conditional_volatility

    # 进行ADF检验
    result = adfuller(data)

    print('ADF Statistic:', result[0])
    print('p-value:', result[1])
    print('Critical Values:')
    for key, value in result[4].items():
        print(f'{key}: {value}')

    # 绘制条件方差图形
    plt.figure(figsize=(16, 4))
    plt.plot(cond_variance, color='blue')
    plt.title('Conditional Variance')
    plt.show()

def test_time_dependent_covariance(data):
    # 计算差分后的数据
    diff_data = data.diff().dropna()

    # 进行ADF检验
    result = adfuller(diff_data)

    print('ADF Statistic:', result[0])
    print('p-value:', result[1])
    print('Critical Values:')
    for key, value in result[4].items():
        print(f'{key}: {value}')
```

```
def plot_data_properties(data, ts_plot_name="Time Series plot"):
    plt.figure(figsize=(16, 4))
    plt.plot(data)
    plt.title(ts_plot_name)
    plt.ylabel('Amount')
    plt.xlabel('Time')
    fig, axes = plt.subplots(1, 3, squeeze=False)
    fig.set_size_inches(16, 4)
    plot_acf(data, ax=axes[0, 0], lags=48);
    plot_pacf(data, ax=axes[0, 1], lags=48);
    sns.distplot(data, ax=axes[0, 2])
    axes[0, 2].set_title("Probability Distribution")
    plt.show()
plot_data_properties(original_data, '销售时间关系图')
```

7.2 SARIMA 分析.py

```
    import statsmodels.api as sm
    import pandas as pd
    lst=[]
    data=pd.read_excel('月度.xls')
    for i in range(1,252):
        y=data.iloc[:,i]
        mod = sm.tsa.statespace.SARIMAX(y,
                        order=(1, 1, 1),
                        seasonal_order=(1, 1, 1, 12),
                        enforce_stationarity=False,
                        enforce_invertibility=False)

        results = mod.fit()
        params = results.params
        weight = params[['ar.L1', 'ma.L1', 'ar.S.L12', 'ma.S.L12']]
        lst.append(weight)
    lst=pd.DataFrame(lst)
    lst.to_excel('coef.xlsx')
```

7.3 网格搜索确定模型参数.py

```
    import itertools
    import warnings
    from statsmodels.tsa.statespace.sarimax import SARIMAX
    import pandas as pd

    def find_best_sarima_model(data):
```

```
        best_aic = float("inf")
        best_order = None
        best_seasonal_order = None

        # 忽略警告信息
        warnings.filterwarnings("ignore")

        # 遍历参数组合
        for order in itertools.product(range(0, 1), range(0, 1), range(0, 1)):
            for seasonal_order in itertools.product(range(0, 1), range(0, 1), range(0, 1),
                range(0, 1)):
                try:

                    model = SARIMAX(data, order=order, seasonal_order=seasonal_order)
                    results = model.fit(disp=False)


                    aic = results.aic

                    # 更新最佳模型参数
                    if aic < best_aic:
                        best_aic = aic
                        best_order = order
                        best_seasonal_order = seasonal_order
                except:
                    continue

        print("Best SARIMA Model (AIC={}): SARIMA{}x{}".format(best_aic, best_order,
            best_seasonal_order))
    data=pd.read_excel('月度.xlsx')
    find_best_sarima_model(data)
```

7.4 分类器.py

```
    from sklearn.svm import SVC
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import VotingClassifier
import pandas as pd

# 创建分类器
svm = SVC(kernel='linear', probability=True)
lda = LinearDiscriminantAnalysis()
dt = DecisionTreeClassifier()
```

```
# 创建投票器模型
voting_model = VotingClassifier(estimators=[('svm', svm), ('lda', lda), ('dt', dt)],
                                weights=[0.5, 0.3, 0.2], voting='soft')

data=pd.read_excel('a.xlsx')

X_train=data.iloc[:200,0]
y_train=data.iloc[:200,1]
X_test=data.iloc[200:,0]
voting_model.fit(X_train, y_train)

# 使用模型预测新数据的标签
predictions = voting_model.predict(X_test)
```

7.5 kano 检验.py

```
import pandas as pd

data=pd.read_excel("花叶类.xlsx")
cof=pd.read_excel("coef花叶.xlsx")
score=0

#step1
for i in range(data.shape[0]):
    if data.iloc[i,1]>=data.iloc[i,2]:
        score+=data.iloc[i,2]
        data.iloc[i,1]-=data.iloc[i,2]
        data.iloc[i,2]=0
    elif data.iloc[i,1]<data.iloc[i,2]:
        score+=data.iloc[i,1]
        data.iloc[i,2]-=data.iloc[i,1]
        data.iloc[i, 1] = 0

#step2
for i in range(data.shape[0]):
    if data.iloc[i,2]!=0 :
        a=cof.iloc[1,:]
        for j in range(cof.shape[1]):
            max_index=cof.iloc[1,:].index(max(a))
            if data.iloc[max_index,1]>=data.iloc[i,2]:
                data.iloc[max_index,1]-=data.iloc[i,2]
                score+=cof.iloc[max_index,i]*data.iloc[i,2]
                data.iloc[i,2]=0
                break
            else :
```

```
            score += cof.iloc[max_index, i] * data.iloc[max_index, 1]
            data.iloc[i,2]-=data.iloc[max_index,i]
            data.iloc[max_index,i]=0
            a[max_index]=0
    if(data.iloc[i,2]==0):
        break
print(score)
```

7.6 基于熵权法和灰色关联分析优化的 TOPSIS 评价模型.m

```
%%基于熵权法和灰色关联分析优化的TOPSIS评价模型
clear;clc
%%第一步:读取数据
X=xlsread('指标.xls','B2:E50');
[n,m] = size(X);
disp(['共有' num2str(n) '个评价对象, ' num2str(m) '个评价指标'])

%%第二步:对正向化后的矩阵进行标准化
for i = 1:n
  for j = 1:m
    Z(i,j) = [X(i,j) - min(X(:,j))] / [max(X(:,j)) - min(X(:,j))];
  end
end
disp('标准化矩阵 Z = ')
disp(Z)

%%第三步:熵权法权重计算
Z=Z+0.0001;
weight=Entropy_Method(Z);
disp('熵权法确定的权重为: ')
disp(weight)

%%第四步:评价指标加权后的规范化结果计算
G=Z.*weight;
disp('评价指标加权后的规范化结果计算为: ')
disp(G)

%%第五步: 评价方案的理想解欧氏距离
l_P = sum([(weight.*(G - repmat(max(G),n,1))).^ 2 ],2) .^ 0.5; % l+ 与最大值的距离向量
l_M = sum([(weight.*(G - repmat(min(G),n,1))).^ 2 ],2) .^ 0.5; % l- 与最小值的距离向量

%%第六步: 评价方案与理想解之间灰色关联矩阵计算
t = abs(repmat(max(G),n,1)-G);
mmin = min(min(t));%计算最小差
mmax = max(max(t));%计算最大差
```

```
P = 0.5;%分辨系数
r_P0 = (mmin + P * mmax)./(t + P * mmax);%计算灰色关联分析

t =abs(repmat(min(G),n,1)-G+0.0001);
mmin = min(min(t));%计算最小差
mmax = max(max(t));%计算最大差
P = 0.5;%分辨系数
r_M0 = (mmin + P * mmax)./(t + P * mmax);%计算灰色关联分析

%%第七步：计算评价方案与理想解之间的关联程度
r_P=sum(r_P0,2)/m;
r_M=sum(r_M0,2)/m;

%%第八步：对计算结果进行无量纲化操处理
L_P=l_P/max(l_P);
L_M=l_M/max(l_M);
R_P=r_P/max(r_P);
R_M=r_M/max(r_M);


%%第九步：计算评价方案与理想方案的接近度
s_p=0.5*L_P+0.5*R_M;
s_m=0.5*L_M+0.5*R_P;
C=s_p./(s_p+s_m);
C = [max(C) - C] / [max(C) - min(C)];
disp('计算评价方案与理想方案的接近度为：')
disp(C)
```

7.7 人群搜索算法 (SOA).m

```
%人群搜索算法（SOA）优化的基于非线性规划的各蔬菜品类未来一周的日补货总量和定价策略决策模型
clear;clc
result1=zeros(7,6);
result2=zeros(7,6);
%%初始化种群
N = 1000;
d = 84;
ger = 20;
xlimit = [3,10;3.3,12;12.1,15;3,12;3.5,15;3.3,12;195,215;25,35;25,40;18,25;115,135;85,105];
umax=0.95;%最大隶属度值
umin=0.011;%最小隶属度值
wmax=0.9;%权重最大值
wmin=0.1;%权重最小值

%%初始化种群位置
```

```
for i = 1:N
    for j=1:d/2
    if mod(j,6)~=0
    x(i,j) = xlimit(mod(j,6), 1) + (xlimit(mod(j,6), 2)-xlimit(mod(j,6), 1))*unifrnd(0, 1);
    elseif mod(j,6)==0
    x(i,j) = xlimit(6, 1) + (xlimit(6, 2)-xlimit(6, 1))*unifrnd(0, 1);
    end
    end
end
for i = 1:N
    for j=d/2+1:d
    if mod(j,6)~=0
    x(i,j) = xlimit(mod(j,6)+6, 1) + (xlimit(mod(j,6)+6, 2)-xlimit(mod(j,6)+6,
        1))*unifrnd(0, 1);
    elseif mod(j,6)==0
    x(i,j) = xlimit(12, 1) + (xlimit(12, 2)-xlimit(12, 1))*unifrnd(0, 1);
    end
    end
end
xm = x; % 每个个体的历史最佳位置
ym = zeros(1, d); % 种群的历史最佳位置
fxm = zeros(N, 1); % 每个个体的历史最佳适应度
fym = -inf; % 种群历史最佳适应度（寻max则初始化为-inf，寻min则初始化为inf）

%%寻找最优个体
fx = zeros(N, 1);
for i = 1:N
fx(i) = object(x(i,:)); % 个体当前适应度
end
%更新个体历史最佳适应度和个体历史最佳位置
for i = 1:N
if fxm(i) < fx(i)
fxm(i) = fx(i); % 更新个体历史最佳适应度
xm(i,:) = x(i,:); % 更新个体历史最佳位置
end
end
%更新群体历史最佳适应度和群体历史最佳位置
if fym < max(fxm)
[fym, nmax] = max(fxm); % 更新群体历史最佳适应度
ym = xm(nmax, :); % 更新群体历史最佳位置
end


%%迭代寻优
record = zeros(ger, 1); % 记录器(记录每次迭代的群体最佳适应度)
Di=0*rand(N,d);
Di(1,:)=1;
```

```
L=0*rand(N,d);
Diego=0*rand(N,d);
Dialt=0*rand(N,d);
Dipro=0*rand(N,d);
for iter = 1:ger
for i=1:N
w=wmax-iter*(wmax-wmin)/ger;
Diego(i,:)=sign(xm(i,:)- x(i,:));%确定利己方向
Dialt(i,:)=sign(ym - x(i,:));%确定利他方向
if object(xm(i,:))>=object(x(i,:))
  Dipro(i,:)=-Di(i,:);
else
  Dipro(i,:)=Di(i,:);
end
Di(i,:)=sign(w* Dipro(i,:)+rand*Diego(i,:)+rand*Dialt(i,:));%确定经验梯度方向
[Orderfitnessgbest, Indexfitnessgbest]=sort(fxm,'descend');
u=umax-(N-Indexfitnessgbest(i))*(umax-umin)/(N-1);
U=u+(1-u)*rand;
H(iter)=(ger-iter)/ger;%选代过程中权重的变化
C(i,:)=H(iter)*abs(ym-5*rands(1,d));%确定高斯函数的参数
T=sqrt(-log(U));%确定搜索步长的大小
v(i,:)=C(i,:)*T;
v(1,find(v(1,:)>3*max(C(i,:))))=3*max(C(i,:));
v(i,find(v(i,:)<0))=0;
%更新位置
x(i,:) = x(i,:) + Di(i,:).*v(i,:);
% 边界位置处理
new_x = zeros(N, d);
for i = 1:d/2
    xi = x(:,i);
    if mod(i,6)~=0
    xi(xi > xlimit(mod(i,6),2)) = xlimit(mod(i,6),2);
    xi(xi < xlimit(mod(i,6),1)) = xlimit(mod(i,6),1);
    new_x(:,i) = xi;
    elseif mod(i,6)==0
    xi(xi > xlimit(6,2)) = xlimit(6,2);
    xi(xi < xlimit(6,1)) = xlimit(6,1);
    new_x(:,i) = xi;
    end
end
for i = d/2+1:d
    xi =x(:,i);
    if mod(i,6)~=0
    xi(xi > xlimit(mod(i,6)+6,2)) = xlimit(mod(i,6)+6,2);
    xi(xi < xlimit(mod(i,6)+6,1)) = xlimit(mod(i,6)+6,1);
    new_x(:,i) = xi;
    elseif mod(i,6)==0
```

```
    xi(xi > xlimit(12,2)) = xlimit(12,2);
    xi(xi < xlimit(12,1)) = xlimit(12,1);
    new_x(:,i) = xi;
    end
end
x = new_x;
end
record(iter) = fym; % 最佳适应度记录
end

%%导入结果
for i=1:7
  for j=1:6
  result1(i,j)=ym(1,6*(i-1)+j);
  end
end
 for i=1:7
  for j=1:6
  result2(i,j)=ym(1,42+6*(i-1)+j);
  end
end
```

7.8 object1.m

```
function y = object1(X)
v=[0.40 0.51 0.91 0.70 2.73 1.02 0.77 0.69 0.64 0.72 0.63 1.67 1.78 4.09 2.35 2.40 2.60
    3.85 0.81 0.88 1.48 1.35 0.64 0.49 0.32 0.60 3.39 0.56 2.85 3.34 0.73 0.31 0.42];
y=0;
y = y+(-28.286+2.909 * X(1,1)+3.168 * X(1,2)-2.21 * X(1,3)+1.24 * X(1,4)-0.043 *
    X(1,5)-0.548 * X(1,6)+2.835 * X(1,7)-0.43 * X(1,8)+1.049 * X(1,9)+0.328 *X(1,10)+3.9 *
    X(1,11)-1.261 * X(1,12))*X(1,1)-X(1,34)*v(1,1);
y = y+(-19.285+0.422 * X(1,1)+1.577 * X(1,2)+0.044 * X(1,3)+0.599 * X(1,4)-0.22 *
    X(1,5)+0.565 * X(1,6)+1.489 * X(1,7)+1.463 * X(1,8)-0.047 * X(1,9)+1.678 *X(1,10)+1.505
    * X(1,11)-1.474 * X(1,12))*X(1,2)-X(1,35)*v(1,2);
y = y+(6.276-2.985 * X(1,1)+2.885 * X(1,2)+1.053 * X(1,3)+2.433 * X(1,4)-0.621 *
    X(1,5)+0.255 * X(1,6)-1.37 * X(1,7)-0.285 * X(1,8)-0.663 * X(1,9)+1.005 *X(1,10)+0.139
    * X(1,11)-1.812 * X(1,12))*X(1,3)-X(1,36)*v(1,3);
y = y+(33.018-2.095 * X(1,1)-0.887 * X(1,2)+2.356 * X(1,3)-1.28 * X(1,4)-0.227 *
    X(1,5)-0.788 * X(1,6)-1.199 * X(1,7)+0.465 * X(1,8)-0.669 * X(1,9)-0.718 *X(1,10)-3.092
    * X(1,11)+0.648 * X(1,12))*X(1,4)-X(1,37)*v(1,4);
y = y+(-4.493-0.241 * X(1,1)+0.897 * X(1,2)-0.211 * X(1,3)+0.991 * X(1,4)-0.096 *
    X(1,5)+0.313 * X(1,6)+0.077 * X(1,7)+0.331 * X(1,8)-0.108 * X(1,9)+0.219 *X(1,10)+0.667
    * X(1,11)-1.132 * X(1,12))*X(1,5)-X(1,38)*v(1,5);
y = y+(44.172+2.014 * X(1,1)+4.203 * X(1,2)-4.008 * X(1,3)-2.448 * X(1,4)-0.127 *
    X(1,5)+5.332 * X(1,6)-10.043 * X(1,7)+6.731 * X(1,8)-1.623 * X(1,9)-2.427
```

```
    *X(1,10)+2.963 * X(1,11)-3.134 * X(1,12))*X(1,6)-X(1,39)*v(1,6);
y = y+(-7.289+1.381 * X(1,1)+0.912 * X(1,2)-0.677 * X(1,3)-1.079 * X(1,4)-0.04 *
    X(1,5)-0.426 * X(1,6)+0.529 * X(1,7)-0.725 * X(1,8)+0.584 * X(1,9)+0.763 *X(1,10)+1.24
    * X(1,11)+1.353 * X(1,12))*X(1,7)-X(1,40)*v(1,7);
y = y+(-1.418-1.118 * X(1,1)-0.083 * X(1,2)+1.247 * X(1,3)+0.041 * X(1,4)-0.177 *
    X(1,5)+0.267 * X(1,6)+0.182 * X(1,7)+0.386 * X(1,8)-0.356 * X(1,9)+1.027 *X(1,10)-0.795
    * X(1,11)+0.265 * X(1,12))*X(1,8)-X(1,41)*v(1,8);
y = y+(11.019-0.144 * X(1,1)+1.307 * X(1,2)-0.687 * X(1,3)+2.122 * X(1,4)-0.02 *
    X(1,5)-0.991 * X(1,6)+0.637 * X(1,7)-2.2 * X(1,8)+0.423 * X(1,9)-1.4 *X(1,10)+0.678 *
    X(1,11)-0.327 * X(1,12))*X(1,9)-X(1,42)*v(1,9);
y = y+(-60.537-5.597 * X(1,1)+16.676 * X(1,2)-6.698 * X(1,3)+21.064 * X(1,4)-1.468 *
    X(1,5)+9.832 * X(1,6)-4.713 * X(1,7)+5.102 * X(1,8)-2.783 * X(1,9)+3.015 *X(1,10)+13.67
    * X(1,11)-20.486 * X(1,10))*X(1,43)-X(1,43)*v(1,10);
y = y+(4.456-3.153 * X(1,1)+6.92 * X(1,2)-0.697 * X(1,3)+6.609 * X(1,4)-0.843 *
    X(1,5)+1.712 * X(1,6)-2.101 * X(1,7)-0.916 * X(1,8)-0.804 * X(1,9)+0.118 *X(1,10)+2.956
    * X(1,11)-3.785 * X(1,12))*X(1,11)-X(1,44)*v(1,11);
y = y+(21.825+1.73 * X(1,1)-1.793 * X(1,2)+0.24 * X(1,3)-5.431 * X(1,4)+0.151 *
    X(1,5)+0.075 * X(1,6)-2.066 * X(1,7)+0.7 * X(1,8)+0.164 * X(1,9)+0.428 *X(1,10)-1.261 *
    X(1,11)+4.283 * X(1,12))*X(1,12)-X(1,45)*v(1,12);
y = y+(47.729-1.97*X(1,13)-0.781*X(1,14))*X(1,13)-X(1,46)*v(1,13);
y = y+(61.389-0.657*X(1,13)-3.589*X(1,14))*X(1,14)-X(1,47)*v(1,14);
y =
    y+(253.809+8.137*X(1,15)+3.69*X(1,16)-24.346*X(1,17)-3.377*X(1,18))*X(1,15)-X(1,48)*v(1,15);
y =
    y+(252.081-2.224*X(1,15)-1.396*X(1,16)-16.005*X(1,17)+1.188*X(1,18))*X(1,16)-X(1,49)*v(1,16);
y =
    y+(32.364+4.053*X(1,15)+1.276*X(1,16)-5.346*X(1,17)-1.429*X(1,18))*X(1,17)-X(1,50)*v(1,17);
y =
    y+(227.835-2.662*X(1,15)-1.151*X(1,16)-13.842*X(1,17)+1.084*X(1,18))*X(1,18)-X(1,51)*v(1,18);
y = y+(21.614+2.215*X(1,19)-1.85*X(1,20)-0.744*X(1,21))*X(1,19)-X(1,52)*v(1,19);
y = y+(3.681+1.51*X(1,19)+1.965*X(1,20)-1.716*X(1,21))*X(1,20)-X(1,53)*v(1,20);
y = y+(-20.219+0.983*X(1,19)-3.001*X(1,20)+3.207*X(1,21))*X(1,21)-X(1,54)*v(1,21);
y = y+(576.048-5.131 * X(1,22)-58.75 * X(1,23)-36.989 * X(1,24)-1.82 * X(1,25)+1.078 *
    X(1,26)-0.596 * X(1,27)-2.661 * X(1,28)+1.568 * X(1,29))*X(1,22)-X(1,55)*v(1,22);
y = y+(-105.398-4.703 * X(1,22)+11.803 * X(1,23)+8.412 * X(1,24)-1.442 * X(1,25)+2.727 *
    X(1,26)-0.525 * X(1,27)+3.057 * X(1,28)+2.622 * X(1,29))*X(1,23)-X(1,56)*v(1,23);
y = y+(498.019-3.811 * X(1,22)-43.215 * X(1,23)-26.299 * X(1,24)-1.246 * X(1,25)+1.421 *
    X(1,26)-0.986 * X(1,27)-12.892 * X(1,28)+0.811 * X(1,29))*X(1,24)-X(1,57)*v(1,24);
y = y+(906.395-10.183 * X(1,22)-98.763 * X(1,23)-62.021 * X(1,24)-3.43 * X(1,25)+1.78 *
    X(1,26)+0.049 * X(1,27)-2.45 * X(1,28)+5.42 * X(1,29))*X(1,25)-X(1,25)*v(1,25);
y = y+(967.796-10.209 * X(1,22)-90.402 * X(1,23)-55.651 * X(1,24)-3.385 * X(1,25)+3.238 *
    X(1,26)-0.589 * X(1,27)-14.645 * X(1,28)+1.071 * X(1,29))*X(1,26)-X(1,59)*v(1,26);
y = y+(-56.777+0.281 * X(1,22)+5.59 * X(1,23)+3.571 * X(1,24)+0.057 * X(1,25)+0.124 *
    X(1,26)+0.028 * X(1,27)+0.414 * X(1,28)+0.108 * X(1,29))*X(1,27)-X(1,60)*v(1,27);
y = y+(-202.775+2.855 * X(1,22)+21.495 * X(1,23)+12.608 * X(1,24)+1.574 * X(1,25)-2.673 *
    X(1,26)-0.423 * X(1,27)+6.84 * X(1,28)-1.232 * X(1,29))*X(1,28)-X(1,61)*v(1,28);
y = y+(94.549-2.459 * X(1,22)-10.099 * X(1,23)-6.081 * X(1,24)-0.685 * X(1,25)+0.646 *
```

```
        X(1,26)-0.11 * X(1,27)-0.294 * X(1,28)+1.328 * X(1,29))*X(1,29)-X(1,62)*v(1,29);
    y =
        y+(2822.993-115.277*X(1,30)-6.288*X(1,31)-12.719*X(1,32)+1.742*X(1,33))*X(1,30)-X(1,63)*v(1,30);
    y =
        y+(2186.678-89.318*X(1,30)-1.723*X(1,31)-14.691*X(1,32)+1.168*X(1,33))*X(1,31)-X(1,64)*v(1,31);
    y =
        y+(10421.116-430.926*X(1,30)-8.675*X(1,31)-47.276*X(1,32)+27.157*X(1,33))*X(1,32)-X(1,65)*v(1,32);
    y =
        y+(2683.37-112.073*X(1,30)+11.615*X(1,31)-33.871*X(1,32)+7.699*X(1,33))*X(1,33)-X(1,66)*v(1,33);
    y=y/7;
    end
end
```

7.9 object2.m

```
    function y = object2(X)
y=0;
d=[0.91 0.85 0.87 0.86 0.94 0.95 0.86 0.88 0.84 0.89 1.00 0.88 0.94 0.91 0.92 0.88 0.93 0.97
     1.00 0.91 0.91 0.91 0.90 0.91 0.91 0.91 0.91 0.91 0.91 0.91 0.92 0.84 0.91];
Y=X.*d;
q=[10.16 15.52 4.21 6.66 1.88 11.49 4.98 7.38 5.46 37.71 24.51 10.11 14.67 5.14 6.79 3.34 1.76
     4.46 12.54 3.02 4.69 7.69 16.81 24.69 12.51 12.51 1.45 7.37 2.22 5.40 10.97 18.34 9.60];
for i=1:33
    y=y+Y(1,i)-q(1,i);
end
end
```

7.10 多目标粒子群优化算法 (MOPSO).m

```
    %%多目标粒子群优化算法(MOPSO)
    clear;clc
    result1=zeros(1,33);%定价确定
    result2=zeros(1,33);%补货量预测
    %%初始化种群
    N=1000;
    d=66;
    ger=20;
    xlow=[0.40 0.51 0.91 0.70 2.73 1.02 0.77 0.69 0.64 0.72 0.63 1.67 1.78 4.09 2.35 2.40 2.60
        3.85 0.81 0.88 1.48 1.35 0.64 0.49 0.32 0.60 3.39 0.56 2.85 3.34 0.73 0.31 0.42 8.92
        13.24 3.79 5.89 0.92 10.29 4.47 6.38 4.84 32.14 21.14 6.86 12.5 3.52 5.94 2.94 1.7 3.95
        10.86 2.61 4.19 6.78 14.17 21.29 11.14 11.14 1.02 6.86 2 4.59 9.86 16 8.71];
    xhigh=[4.77 4.71 10.00 6.76 17.50 8.22 6.61 5.98 6.50 5.58 5.20 6.88 15.43 16.04 17.87
        16.49 17.50 26.14 7.50 7.50 15.00 14.25 6.50 7.21 3.23 6.17 23.88 5.90 23.48 30.00 6.28
        2.35 3.43 17.85 26.49 7.59 11.77 1.85 20.57 8.94 12.76 9.69 64.29 42.29 13.71 25.01
```

```
    7.03 13.89 8.89 5.40 11.90 21.72 5.22 8.39 13.55 28.35 42.57 22.29 22.29 2.03 13.71 4
    9.17 19.7 32 17.4];
vlow=[-0.5,-5];
vhigh=[0.5,5];
w=0.5;
c1=0.6;
c2=0.6;
%%初始化种群位置
for i = 1:N
   for j=1:d
    x(i,j)=xlow(1, j)+(xhigh(1,j)-xlow(1,j))*unifrnd(0,1);
   end
end
%%初始化种群速度
for i = 1:N
   for j=1:d/2
    v(i,j)=vlow(1, 1)+(vhigh(1,1)-vlow(1,1))*unifrnd(0,1);
   end
end
for i = 1:N
   for j=d/2+1:d
    v(i,j)=vlow(1, 2)+(vhigh(1,2)-vlow(1,2))*unifrnd(0,1);
   end
end
xm=x;%每个个体的历史最佳位置
ym=zeros(1, d);%种群的历史最佳位置
fxm=zeros(N, 2);%每个个体的历史最佳适应度
fym=[-inf,-inf];%种群历史最佳适应度（寻max则初始化为-inf，寻min则初始化为inf）
%%群体更新
record=zeros(ger,2);%记录器(记录每次迭代的群体最佳适应度)
for iter=1:ger
fx=zeros(N,2);
%%个体当前适应度
for i=1:N
  fx(i,1)=object1(x(i,:));
  fx(i,2)=object2(x(i,[34:66]));
end
%%更新个体历史最佳适应度和个体历史最佳位置
for i=1:N
  if fxm(i,1)<fx(i,1)&&fxm(i,2)<fx(i,2)
  fxm=fx;%更新个体历史最佳适应度
  xm(i,:)=x(i,:);%更新个体历史最佳位置
  else
  r=unifrnd(0,1);
  if r<0.5
  xm(i,:)=x(i,:);%更新个体历史最佳位置
  else
```

```
  xm(i,:)=xm(i,:);%更新个体历史最佳位置
  end
  end
end
%%更新群体历史最佳适应度和群体历史最佳位置
for i=1:N
  if fym(1,1)<fxm(i,1)&&fym(1,2)<fxm(i,2)
  fym=fxm(i,:);% 更新群体历史最佳适应度
  ym=xm(i,:);% 更新群体历史最佳位置
  end
end
%%更新速度
v=v*w+c1*rand*(xm-x)+c2*rand*(repmat(ym, N, 1)-x);
%%边界速度处理
new_v=zeros(N,d);
for i=1:d/2
     vi=v(:,i);
     vi(vi>vhigh(1,1))=vhigh(1,1);
     vi(vi<vlow(1,1))=vlow(1,1);
     new_v(:,i)=vi;
end
for i=d/2+1:d
     vi=v(:,i);
     vi(vi>vhigh(1,2))=vhigh(1,2);
     vi(vi<vlow(1,2))=vlow(1,2);
     new_v(:,i)=vi;
end
v=new_v;
%更新位置
x=x+v;
%边界位置处理
new_x=zeros(N,d);
for i=1:d/2
    xi=x(:,i);
    xi(xi>xhigh(1,1))=xhigh(1,1);
    xi(xi<xlow(1,1))=xlow(1,1);
    new_x(:,i)=xi;
end
for i=d/2+1:d
    xi=x(:,i);
    xi(xi>xhigh(1,2))=xhigh(1,2);
    xi(xi<xlow(1,2))=xlow(1,2);
    new_x(:,i)=xi;
end
x=new_x;
record(iter,:)=fym;%最佳适应度记录
end
```

```
%导入结果
result1=ym(1,[1 :33]);
result2=ym(1,[34 :66]);
```