# 分布式鲁棒优化实践——Portfolio问题建模与实现

# 1问题介绍

投资组合是指我们拥有一定可供投资的资金和投资选项（基金、股票等），投资选项有着不同的收益率和风险，收益越高往往对应着风险越高。我们需要根据一定的历史数据决定我们在不同投资选项上的投资份额，以期获得最大收益。

本文对投资组合问题构建了分布式鲁棒优化模型并将其转化为混合整数规划模型（MIP）模型（2.1，2.2），使用Python通过 Tushare 获取投资选项收益数据同时使用Python 建立数学规划模型并调用求解器Gurobi 进行求解验证模型的正确性，最后对结果进行简要的分析（3.1，3.2，3.3）。

# 2数学模型

## 2.1分布式鲁棒优化模型

（1）参数

个可投资的投资选项（），年度收益为随机变量：，我们拥有一定数量的资本，在建模过程中设为“1”（资本分配比例）。

（2）

原问题：在损失不超过的概率大于等于的情况下，最大化期望收益

目标：

约束条件：

## 2.2构建MIP模型

转化基础：使用一定已知年份（年）的收益估计原问题中的收益（）。

设已知：为投资项目在年时的收益。设一个投资组合序列为，则投资组合序列数为：。

目标：

引入辅助变量：。

约束条件：

# 3实验

## 3.1数据获取

本文使用Python通过Tushare获取股票自2012年4月1日到2022年5月1日的每日涨跌信息（如表1示例，全部结果见附件），然后以4月的第一个交易日进行统计年度增长率（表2）。

表 1股票每日涨跌信息（600000.SH）

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| trade\_date | ts\_code | open | high | low | close | pre\_close | change | pct\_chg | vol | amount |
| 20220429 | 600000.SH | 7.98 | 8.06 | 7.86 | 8.03 | 7.99 | 0.04 | 0.5006 | 445128.46 | 354246.1 |
| 20220428 | 600000.SH | 7.86 | 7.99 | 7.86 | 7.99 | 7.83 | 0.16 | 2.0434 | 451751.69 | 358888.163 |
| 20220427 | 600000.SH | 7.91 | 7.95 | 7.81 | 7.83 | 7.87 | -0.04 | -0.5083 | 386886.14 | 304687.721 |

表 2年度增长率

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 股票  年份 | 600000  .SH | 600006  .SH | 600008  .SH | 600185  .SH | 600232  .SH | 600233  .SH | 600320  .SH | 600391  .SH | 600396  .SH | 600519  .SH |
| 2012 | 1.125419933 | 0.857566766 | 1.604373757 | 1.192439863 | 0.92706334 | 1.24195122 | 0.686956522 | 0.743948359 | 0.907514451 | 0.800728155 |
| 2013 | 0.968159204 | 1.055363322 | 0.82527881 | 1.347262248 | 1.039337474 | 0.700706991 | 1.07278481 | 0.879971077 | 1.318471338 | 0.98508639 |
| 2014 | 1.639260021 | 2.524590164 | 2.355855856 | 2.889839572 | 1.386454183 | 2.106502242 | 2.439528024 | 3.721446179 | 0.966183575 | 1.202166287 |
| 2015 | 1.139811912 | 0.857142857 | 0.548119822 | 0.641006662 | 1.420977011 | 1.47099521 | 0.567110036 | 0.928902628 | 0.66125 | 1.25468414 |
| 2016 | 0.888888889 | 1.046969697 | 0.545348837 | 0.335450346 | 1.119312437 | 0.846599132 | 1.277185501 | 0.857855954 | 0.943289225 | 1.589864948 |
| 2017 | 0.724628713 | 0.713458755 | 1.115138593 | 0.922547332 | 0.584462511 | 0.696153846 | 0.858096828 | 0.57550568 | 0.541082164 | 1.745265103 |
| 2018 | 0.976942784 | 1.032454361 | 0.812619503 | 1.061567164 | 1.227202473 | 0.847145488 | 0.838521401 | 0.849301878 | 0.866666667 | 1.263123842 |
| 2019 | 0.881993007 | 0.797642436 | 0.710588235 | 0.724077329 | 0.768261965 | 0.815942029 | 0.75638051 | 0.914399093 | 0.824786325 | 1.284051222 |
| 2020 | 1.079286422 | 1.566502463 | 1.046357616 | 1.381067961 | 0.839344262 | 0.991119005 | 1.042944785 | 1.127712337 | 1.404145078 | 1.853581142 |
| 2021 | 0.741046832 | 0.856918239 | 1.012658228 | 1.230228471 | 1.1484375 | 1.612903226 | 0.985294118 | 1.166575041 | 1.3099631 | 0.870633407 |

注：为了方便模型中的计算，此处模型增长率公式为：

## 3.2模型实现及求解结果

本模型使用Python建模调用求解器Gurobi进行求解，其中求解器初始化参数为：求解时间上限：3600s，MIPGAP：0。同时因为MIP模型复杂度随着投资选项的数量和使用的历史数据的数量呈现出指数级增长，本文仅使用三只和四只股票数据对**原问题**的MIP模型进行测试和分析。

（1）三只股票（'600519.SH', '600391.SH', '600232.SH'）

表 3三只股票求解结果

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 求解结果 | | | | 求解时间（s） |
| 目标 | 600519.SH | 600391.SH | 600232.SH |
|  | 1.21444 | 0.67491 | 0.05493 | 0.27016 | 24.87 |
|  | **-** | **-** | **-** | **-** | **infeasible** |
|  | **-** | **-** | **-** | **-** | infeasible |
|  | 1.28491 | 1.0 | 0.0 | 0.0 | 0.03 |
|  | **1.23089** | **0.73186** | **0.07673** | **0.19141** | **2.05** |
|  | - | - | - | - | infeasible |
|  | 1.28491 | 1.0 | 0.0 | 0.0 | 0.02 |
|  | **1.28491** | **1.0** | **0.0** | **0.0** | **0.02** |
|  | 1.28491 | 1.0 | 0.0 | 0.0 | 0.02 |

（2）四只股票（'600519.SH', '600391.SH', '600232.SH', '600396.SH'）

由于四只股票的复杂度较高，实验过程中并没有求解的最优解，只能输出到达求解时间的解。

表 4四只股票求解结果

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | 求解结果 | | | | | 求解时间（s） |
| 目标 | 600519.SH | 600391.SH | 600232.SH | 600396.SH |
|  | 1.20423 | 0.65283 | 0.06091 | 0.20658 | 0.07968 | 3600 |
|  | **Time limit reached** | | | | | **3600** |
|  | **-** | **-** | **-** | **-** | **-** | infeasible |
|  | 1.28491 | 1.0 | 0.0 | 0.0 | 0.0 | 0.18 |
|  | **1.22202** | **0.67351** | **0.14368** | **0.13178** | **0.05103** | **3600** |
|  | - | - | - | - | - | infeasible |
|  | 1.28491 | 1.0 | 0.0 | 0.0 | 0.0 | 0.25 |
|  | **1.28491** | **1.0** | **0.0** | **0.0** | **0.0** | **0.19** |
|  | 1.28491 | 1.0 | 0.0 | 0.0 | 0.0 | 0.20 |

## 3.3结果分析

从实验的结果中可以看出，随着的减小（加粗行），目标值逐渐减少，直至无可行解；随着的增加（下划线行），目标值也逐渐减少，直至无可行解。