SVM 多因子策略报告

金滢

1 SVM 算法原理

SVM 算法经常用于分类。n 维空间中每个点是一个样本，各个维度的坐标由各因子值确定（需要 standardize，在 python 中用 preprocessing 中的 scale 函数完成）。在使用线性核的情形下，相当于在 n 维空间中寻找 n-1 维线性超平面，使得各样本点到该超平面的距离和最小（容许一定的分类错误，惩罚程度由参数 C 确定）。在使用其他核函数（如 gauss 核、sigmoid 核等）的情形下，相当于将样本点投影由核函数变换到另一 n 维空间，在新的空间中寻找线性超平面（可以理解成训练处非线性的 n-1 维分类流形）。算法中能够被修改的参数主要是惩罚系数 C 和核函数类型，在一些特殊的核函数中似乎还可以修改函数中的某个系数 γ，但未做尝试。

2 本策略概述

利用现有的 40 个因子，采用 SVM 算法选择股票。设置参数 *back_days*，在第 t 天，使用从当天往前共 *back_days* 天的因子数据，将每天的股票按收益率排序，取前 10% 标记为 1，后 10% 标记为-1，这两部分数据拼合起来后作为训练数据，将所有 *back_days* 天的训练数据得到当天模型的训练数据。

第 t 天训练的模型用于第 t+5 天的预测，避免 5 天 forward return 的 forward bias。在第 t+5 天，可用该模型计算出当天样本点到分类超平面的（带符号）距离，将这个距离作为因子取值保存下来。

模型中主要修改的是惩罚系数 C 和核函数类型（kernel），其中 C=0.1,1,10，核函数取了线性核函数、高斯核函数和 sigmoid 核函数。

3 代码结构

为了方便检查代码，这里简要罗列一下代码结构。

- 读取因子数据，由于训练模型和预测之间相隔 5 天，所以 factorss 里存储 5 个 list，其中第 0 个 list 存储共 num_factos 个因子数据（DataFrame），这些因子数据是第 0,5,10,... 天的，第 1 个 list 存储第 1,6,11,... 天的；
- 读取 forward return 数据，将成为获取训练数据的依据；
- 计算 svm 策略给出的复合因子值，这个因子值为样本点到超平面距离：
  for j = 0,1,2,3,4：
  - 将天数序号 order 初始化为 j+5，忽略前面可能前溯几天会缺失数据的日子
  - while order < 总天数，循环：
    * 打印当天日期
    * 获取当天有所有因子数据的所有股票样本

    * 如果当天不是初始日,则利用 order-5 天训练出的模型计算因子得分并存储进全局变量 factor_results
    * 获得第 order,order-1,...,order-back_days+1 天的训练数据（为保证时间正确，获取数据时会打印时间信息）
    * 用训练数据训练模型，保存在变量 current_result 中

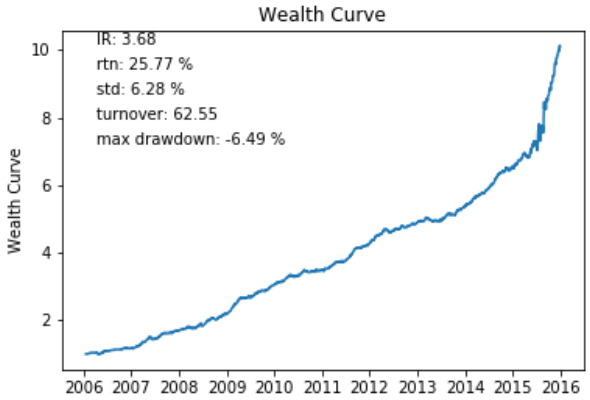
- 将所有计算所得因子按时间重新排序并输出

4 回测结果

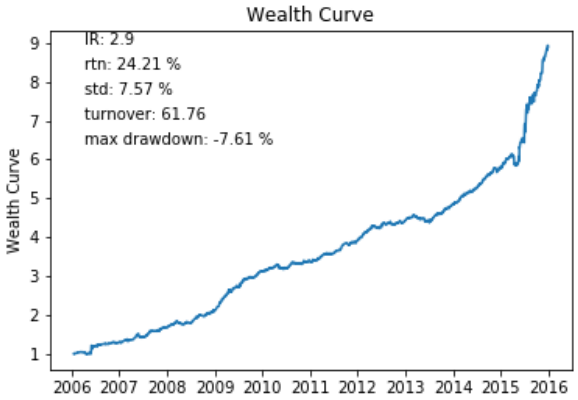
回测采用股指期货进行对冲，含交易费用。结果如下:

C	kernel	annualized return	IR	max drawdown	turnover
1	linear	15.67%	2.79	−6.79%	58.39
10	linear	14.84%	2.7	−6.11%	57.86
1	sigmoid	6.14%	1.32	−5.28%	51.93
10	sigmoid	0.37%	0.11	−9.71%	46.49
0.1	gauss	24.21%	2.9	−7.61%	61.76
1	gauss	25.77%	3.68	−6.49%	62.55
10	gauss	18.57%	3.36	−18.57%	61.03

比较好的结果的 wealth curve:



C=1，gauss 核



C=0.1，gauss 核