HAR模型预测股票收益波动率

项目描述

使用**HAR-type volatility models**来预测金融数据的波动率，并测试评估其预测性能。HAR-type volatility models指的是 *论文[1]* 中汇总的一系列线性模型。

采用金融数据（即预测对象）粗略地分为两类，一种是个股，另一种“综合性”标的物，如ETF基金、板块指数等。

- ETF基金：交易所交易型基金（Exchange Traded Funds），在场内像股票一样交易的基金。
- 指数：多只股票的加权平均，是一个股票组合。
- 板块：一些具有相同特征的股票的集合。

逻辑架构

1. 从原始数据集中提取出我们关心的特征，主要是参考 *论文[1]* 中的5种 **Volatility components** 。
2. 定义不同的HAR-type volatility model，生成相应的训练/测试数据集。
3. 使用 *论文[1]* 中的预测评估方法 **rolling window prediction method** ，采用回归算法训练，计算不同 HAR-type volatility model 的R2决定系数来评估预测表现。

大致可以看作，第1步是数据集处理和特征工程，第2步是算法模型，第3步是目标函数和评估指标。文档中不包括数据获取部分。

特征工程

价格序列、收益序列的计算

通过金融数据的价格序列，计算其价格序列和对数收益序列，是计算波动率序列的前置准备。

假设数据集按照相同的时间间隔Δ采样，日内价格序列的长度为 $M = 240/\Delta + 1$ ，收益序列的长度为 $N = M - 1$ 。

> 在下图中，若采样间隔为5分钟，则 $\Delta = 5, M = 240/5 + 1 = 49, N = 48$ 。
>
> - 第t日，日内价格序列，记为：$P_t = \{P_{t,0}, P_{t,1}, P_{t,2}, \cdots\}$，其中$P_{t,0}$为第t日开盘价，$P_{t,i}$为第i个时刻的价格
> - 日内对数价格序列，记为：$p_t = \{p_{t,0}, p_{t,1}, p_{t,2}, \cdots\}$，其中$p_{t,i} = \ln P_{t,i}$，即对价格求自然对数
> - 第i时刻的收益，记为：$r_{t,i} = p_{t,i} - p_{t,i-1}$
> - 日内收益序列，记为：$r_t = \{r_{t,1}, r_{t,2}, \cdots\}$
>
> 注意
>
> - 正常情况下，日交易时间为4小时（9:30-11:30 AM，1:00-3:00 PM）
> - 常见高频数据包括1分钟、5分钟数据
> - 若采样时间间隔为5分钟，则日内对数价格序列p_t包含49个数据（开盘价对应$p_{t,0}$，后续每5分钟获得一个价格数据），相应的日内收益序列r_t包含48个数据

参考 *论文[1]* 对隔夜收益和日内收益的计算公式做如下改动：

- **隔夜收益**（*overnight return variance*）

$$r_{(t,0)} = r_{(t,n)} = 100(lnP_{t,0} - lnP_{t-1,N})$$

- **日内收益**（i^{th} intraday return at day t）

$$r_{t,i} = 100(lnP_{t,i} - lnP_{t,i-1}), \quad i = 1, \ldots, N$$

Volatility components的计算

这一步是进一步提取出我们关心的特征，即特征工程，主要参考的是*论文[1]*中的5种 Volatility components。

待实现 解释变量的选择/增加，是否具有明确的金融学意义，例如交易量对波动率的影响。

1. **日波动率成分**（daily realized volatility）

$$RV_t^d = RV_t^{d0} + r_{(t,0)}^2$$

$$RV_t^{d0} = \sum_{i=1}^{N} r_{t,i}^2$$

2. **跳变成分**（daily discontinuous jump variation）和**连续成分**（daily continuous sample path variation）

$$J_t^d = I(Z_t > \phi(\alpha))(RV_t - RBV_t)$$
$$C_t^d = I(Z_t \leq \phi(\alpha))RV_t + I(Z_t > \phi(\alpha))RBV_t$$

$\phi(\alpha)$: the appropriate critical value from the standard normal distribution, $\alpha = 0.99$.

论文[1] [4] 对数学符号 Z_t，RBV_t 的计算公式解释存在错误，*论文[3]* 中的公式纠正了这一点。

$$Z_t = \frac{\frac{RV_t - RBV_{1,t}}{RV_t}}{\sqrt{\left(\left(\frac{\pi}{2}\right)^2 + \pi - 5\right)\frac{1}{M}\max\left(1, \frac{RTQ_t}{RBV_{1,t}^2}\right)}},$$

where

$$RTQ_{1,t} = M\mu_{4/3}^{-3}\left(\frac{M}{M-4}\right)\sum_{j=5}^{M}|r_{t,j-4}|^{4/3}|r_{t,j-2}|^{4/3}|r_{t,j}|^{4/3},$$

$$RBV_{1,t} = \mu_1^{-2}\left(\frac{M}{M-2}\right)\sum_{j=3}^{M}|r_{t,j-2}|\,|r_{t,j}|,$$

3. **半方差成分**（Realized semivariance）

$$RSV_t^{d+} = \sum_{j=1}^{M}\{r_{t,j} \geq 0\}r_{t,j}^2$$

$$RSV_t^{d-} = \sum_{j=1}^{M}\{r_{t,j} < 0\}r_{t,j}^2$$

问题 这一步是否加入了隔夜收益？因为累加的上限是M，而对数收益序列的长度是N，但 *论文[1]* 中写的比较模糊。$r_{t,M}$和$r_{t,n}$是否是等价的？如果是，在半方差成分的计算代码中还要加入隔夜收益。

2.1. Realized volatility

The daily realized volatility proposed by Andersen and Bollerslev (1998) can be computed as

$$RV_t^{d0} = \sum_{i=1}^{N} r_{t,i}^2 \tag{1}$$

where $r_{t,i}$ is the i^{th} intraday return ($i = 1, \cdots, N$) at day t, i.e., $r_{t,i} = 100(\ln P_{t,i} - \ln P_{t,i-1})$. N is the number of intervals in the trading day, and $P_{t,i}$ is the i^{th} intraday closing price at day t.

However, Eq. (1) does not consider the overnight return variance, (Andersen et al., 2011) and it does not involve the consistency estimation of integrated volatility. Therefore, according to Huang et al. (2013) and Gong et al. (2014), we obtain the new daily realized volatility

$$RV_t^d = RV_t^{d0} + r_{t,n}^2 = \sum_{j=1}^{M} r_{t,j}^2 \tag{2}$$

where $r_{t,n}$ is the overnight return, and $M = N + 1$.

4. **带符号的跳变成分**（Signed jump）

$$SJ_t^d = RSV_t^{d+} - RSV_t^{d-}$$

5. **带符号的半跳变成分**（Signed semi-jump）

$$SSJ_t^{d+} = I\{SJ_t^d \geq 0\} SJ_t^d$$
$$SSJ_t^{d-} = I\{SJ_t^d < 0\} SJ_t^d$$

算法模型

*论文[1]*中的16种HAR-type volatility models（代码已实现8种）

问题 关于波动率 RV_t^d 计算是否要开方？这一点还有待确认。 *论文[5]* 中给出有别与 *论文[1]*的模型：

$$\left(\sqrt{RV}\right)_{t+1-k:t} = \frac{1}{k}\sum_{j=1}^{k}\sqrt{RV_{t-j}}, \tag{17}$$

the *heterogeneous autoregressive* (HAR) model for realized volatility of Corsi (2004), including the daily, weekly and monthly realized volatility components, is given by

$$\sqrt{RV}_t = \alpha_0 + \alpha_d\sqrt{RV}_{t-1} + \alpha_w\left(\sqrt{RV}\right)_{t-5:t-1} + \alpha_m\left(\sqrt{RV}\right)_{t-22:t-1} + u_t. \tag{18}$$

现在版本的代码中，默认为开方的形式，加上

$$RV_t^d = \sqrt{RV_t^d}$$

1. HAR-RV model

$$RV_{t+1}^d = c + \alpha_1 RV_t^d + \alpha_2 RV_t^w + \alpha_3 RV_t^m + \epsilon_{t+1}$$

2. HAR-RV-J model

$$RV_{t+1}^d = c + \alpha_1 RV_t^d + \alpha_2 RV_t^w + \alpha_3 RV_t^m + \beta_1 J_t^d + \epsilon_{t+1}$$

3. HAR-CJ model

$$RV_{t+1}^d = c + \alpha_1 RV_t^d + \alpha_2 RV_t^w + \alpha_3 RV_t^m + \beta_1 J_t^d + \beta_2 J_t^w + \beta_3 J_t^m + \epsilon_{t+1}$$

4. HAR-RSV model

$$RV_{t+1}^d = c + \alpha_1 RSV_t^{d+} + \alpha_2 RSV_t^{w+} + \alpha_3 RSV_t^{m+} + \beta_1 RSV_t^{d-} + \beta_2 RSV_t^{w-} + \beta_3 RSV_t^{m-} + \epsilon_{t+1}$$

5. HAR-RSV-J model

$$RV_{t+1}^{d} = c + \alpha_1 RSV_t^{d+} + \alpha_2 RSV_t^{w+} + \alpha_3 RSV_t^{m+} + \beta_1 RSV_t^{d-} + \beta_2 RSV_t^{w-} + \beta_3 RSV_t^{m-} + \phi_1 J_t^{d} + \epsilon_{t+1}$$

6. HAR-RV-SJ model

$$RV_{t+1}^{d} = c + \alpha_1 RV_t^{d} + \alpha_2 RV_t^{w} + \alpha_3 RV_t^{m} + \beta_1 SJ_t^{d} + \epsilon_{t+1}$$

7. HAR-RV-SSJ(1) model

$$RV_{t+1}^{d} = c + \alpha_1 RV_t^{d} + \alpha_2 RV_t^{w} + \alpha_3 RV_t^{m} + \beta_1 SSJ_t^{d+} + \phi_1 SSJ_t^{d-} + \epsilon_{t+1}$$

8. HAR-RV-SSJ(2) model

$$RV_{t+1}^{d} = c + \alpha_1 RV_t^{d} + \alpha_2 RV_t^{w} + \alpha_3 RV_t^{m} + \beta_1 SSJ_t^{d+} + \beta_2 SSJ_t^{w+} + \beta_3 SSJ_t^{m+} + \phi_1 SSJ_t^{d-} + \phi_2 SSJ_t^{w-} + \phi_3 SSJ_t^{m-} + \epsilon_{t+1}$$

库函数scikit-learn中的3种回归算法

线性模型的学习算法使用的是scikit-learn.linear_model中的三种，分别是线性回归、Lasso回归、岭回归。

1. 线性回归目标函数

$$J(\beta) = \sum (y - X\beta)^2$$

2. Lasso回归目标函数，惩罚项为L1范数。其中 $ESS(\beta)$ 表示误差平方和，$\lambda l_1(\beta)$表示惩罚项

$$\begin{aligned} J(\beta) &= \sum (y - X\beta)^2 + \lambda \|\beta\|_1 \\ &= \sum (y - X\beta)^2 + \sum \lambda \left|\beta\right| \\ &= ESS(\beta) + \lambda l_1(\beta) \end{aligned}$$

3. 岭回归目标函数，惩罚项为L2范数

$$J(\beta) = \sum (y - X\beta)^2 + \lambda \|\beta\|_2^2$$
$$= \sum (y - X\beta)^2 + \sum \lambda \beta^2$$

预测评估

预测评估方法：滚动窗口 （rolling window prediction method）

用长度为L的滑动窗口作为训练集，会得到滑动窗口后一天RV_t^d的预测值。

具体来说，假设滑动窗口内的**波动率分量序列**为$\{RV_{t-L+1}, \ldots, RV_t\}$，用该序列来训练HAR模型，然后将$RV_t$输入训练后的模型来预测$\widehat{RV}_{t+1}^d$。

其中t的取值范围是 $\{L \leq t \leq N-1, t \in N^+\}$，N是波动率序列的长度，获得了波动率预测值序列 $\{\widehat{RV}_{L+1}^d, \ldots, \widehat{RV}_N^d\}$。

那么假设数据集的大小为N，我们就会获得N-L个预测样本。

然后将**波动率预测值序列**和**真实的波动率序列**带入**R2决定系数的计算公式**，用于模型性能评估。

论文[2] 中的实验，数据集的大小为4766，2500用于训练模型，滑动窗口的大小约为其数据集大小的一半。

待实现 考虑选择更多滑动窗口大小 window_size ，观察不同的 window_size 对预测性能的影响。

性能评估指标：R2决定系数

1. 代码里采用的R2决定系数，是 *sklearn.linear_model.LinearRegression.score* 函数计算实现的，满足标准的定义。

> scikit-learn官方文档对R2决定系数的定义如下图，更多见sklearn.metrics.r2_score。

3.3.4.8. R² score, the coefficient of determination

The `r2_score` function computes the coefficient of determination, usually denoted as R².

It represents the proportion of variance (of y) that has been explained by the independent variables in the model. It provides an indication of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model, through the proportion of explained variance.

As such variance is dataset dependent, R² may not be meaningfully comparable across different datasets. Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get a R² score of 0.0.

If $\hat{y}_i$ is the predicted value of the i-th sample and y_i is the corresponding true value for total n samples, the estimated R² is defined as:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \epsilon_i^2$.

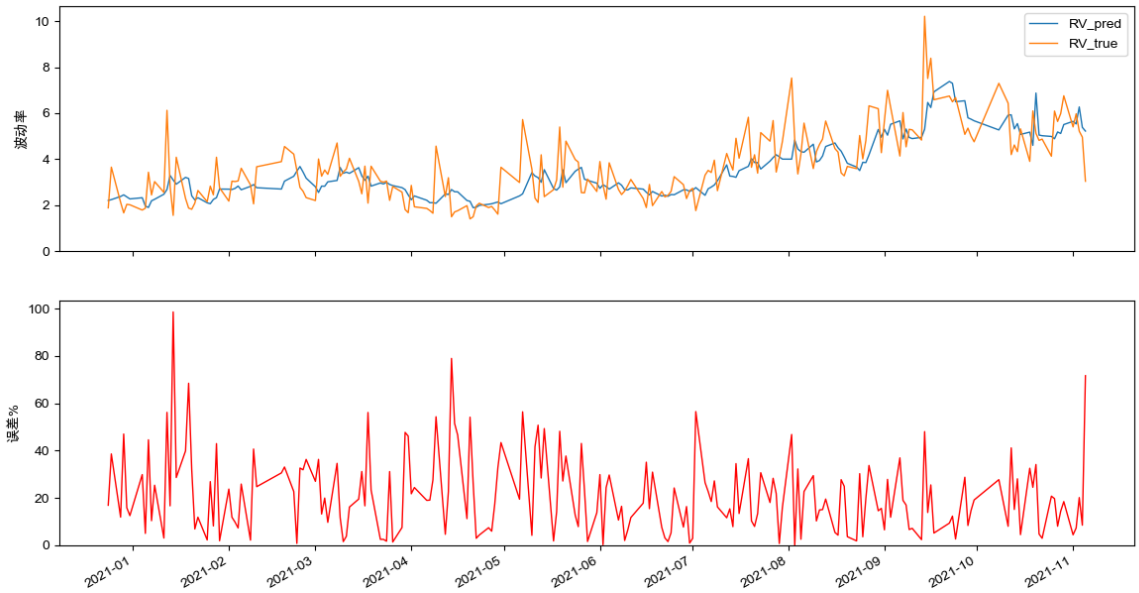
Note that `r2_score` calculates unadjusted R² without correcting for bias in sample variance of y.

2. 待实现 对于预测结果的分析，考虑对趋势判断的正确性（分类问题：涨/跌）

结果可视化

1. 测试集数据（含日期）的：真实波动率、预测波动率、误差（百分比），纵坐标从0开始不缩放，两张子图共享同一横坐标。

SH#600141-岭回归.png



更早版本文档和代码的问题

评分估计的95%置信区间（只适用于方案c）

> 训练集与测试集如何划分?
>
> **方案c.** 随机划分（测试集可能在样本中段），保留一定的时间粒度，如以周为单位（连续5日）。
>
> **方案c**，选取周为单位时，用评分估计的平均得分和95%置信区间可发现，R2决定系数的**方差非常大**，模型不稳定；

```
================================================================================
文件名:SH#880534-5min.xls 天数:434 标题:880534 锂电池 5分钟线 前复权
Model: 线性回归 , Accuracy: 0.040 (+/- 1.467)
Model: 岭回归 , Accuracy: 0.095 (+/- 1.259)
Model: Lasso回归 , Accuracy: 0.048 (+/- 1.464)
最优的R2决定系数:0.137 最优的模型: 岭回归
================================================================================
文件名:SZ#300014-5min.xls 天数:434 标题:300014 亿纬锂能 5分钟线 前复权
Model: 线性回归 , Accuracy: -0.592 (+/- 3.811)
Model: 岭回归 , Accuracy: -0.602 (+/- 3.828)
Model: Lasso回归 , Accuracy: -0.554 (+/- 3.593)
最优的R2决定系数:0.151 最优的模型: Lasso回归
================================================================================
文件名:SH#880715-5min.xls 天数:431 标题:880715 磷概念 5分钟线 前复权
Model: 线性回归 , Accuracy: 0.686 (+/- 0.702)
Model: 岭回归 , Accuracy: 0.686 (+/- 0.696)
Model: Lasso回归 , Accuracy: 0.687 (+/- 0.684)
最优的R2决定系数:0.903 最优的模型: Lasso回归
================================================================================
文件名:SH#600141-5min.xls 天数:431 标题:600141 兴发集团 5分钟线 前复权
Model: 线性回归 , Accuracy: 0.455 (+/- 0.768)
Model: 岭回归 , Accuracy: 0.455 (+/- 0.762)
Model: Lasso回归 , Accuracy: 0.455 (+/- 0.764)
最优的R2决定系数:0.232 最优的模型: Lasso回归
```

在测试方案c中，k-fold交叉验证会得到一个score序列，采用的指标同样是R2决定系数。

```python
# k-fold + 95%置信区间
def k_fold(models, X, y):
    cv = ShuffleSplit(n_splits=10, test_size=5, random_state=0)
    for type, model in models:
        scores = cross_val_score(model, X, y, cv=cv, scoring='r2')
        print("Model:", type, ", Accuracy: %0.3f (+/- %0.3f)" % (scores.mean(), scores.std() *
2))
```

置信区间的计算公式：

$$[\mu - 1.96 * \frac{\sigma}{\sqrt{n}}, \mu + 1.96 * \frac{\sigma}{\sqrt{n}}]$$

σ是观测值的标准差，μ是score序列的均值，n为一次实验样本数量，标准误差 $SE = \frac{\sigma}{\sqrt{n}}$。

评分估计的95%置信区间是因为k-fold交叉验证等价于进行了k次实验，每次实验的样本数量n在之前版本的代码中取了$n = 1$。

想法 我的代码是参考sklearn中文文档使用的交叉验证评分估计的95%置信区间。但有一点存在疑惑，根据中心极限定律和大数定律，样本均值服从正态分布，但是r2_score序列不一定服从正态分布？如果把r2_score序列视作只有一个样本的均值所组成的序列，这样就说得通了，这样的话 $n = 1$ 反而是对的，并且在实验中表现出置信区间非常大的情况，虽然保证了置信度，但这种方法对于预测评估就没什么用了。

3.1.1. 计算交叉验证的指标

使用交叉验证最简单的方法是在估计器和数据集上调用 `cross_val_score` 辅助函数。

下面的例子展示了如何通过分割数据，拟合模型和计算连续 5 次的分数（每次不同分割）来估计 linear kernel 支持向量机在 iris 数据集上的精度:

```
>>> from sklearn.model_selection import cross_val_score
>>> clf = svm.SVC(kernel='linear', C=1)
>>> scores = cross_val_score(clf, iris.data, iris.target, cv=5)
>>> scores
array([ 0.96...,  1\.  ...,  0.96...,  0.96...,  1\.        ])
```

评分估计的平均得分和 95% 置信区间由此给出:

```
>>> print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
Accuracy: 0.98 (+/- 0.03)
```

论文引用

[1] Forecasting the volatility of crude oil futures using HAR-type models with structural breaks

[2] 2012 _HAR modeling for RV forecasting Chap15 Handbook

[3] A reduced form framework for modeling volatility of speculative prices based on realized variation measures

[4] Forecasting Return Volatility of the CSI 300 Index Using the Stochastic Volatility Model with Continuous Volatility and Jumps

[5] The volatility of realized volatility