机器学习的量化投资应用

2018 年 11 月 19 日

目录

课程目的

1. 学习Python语言基础和基本数据分析技巧
2. 了解机器学习原理和应用
3. 学习使用机器学习开发量化投资策略的基础
4. 学习策略构建的基本思想和评价方式

为了达到更好的教学结果，本课程将使用上海商品期货交易所2017年1月-2018年10月的5分钟频率橡胶主力连续合约的行情数据作为教学和实验数据。

课程要求

1. 本课程不要求任何编程、机器学习和量化投资基础，如果有则更好
2. 能够接入互联网的电脑
3. 对编程、机器学习、量化投资的热情

预期结果

1. 熟练使用Python进行数据分析
2. 初步理解掌握机器学习的机制
3. 能够使用机器学习方法快速完成量化策略的开发

目录

开发环境

- 本课程所有教学案例以 iPython notebook 形式储存在https://github.com/shgefu/Use-AI-Tech-to-build-Quant-Demo，不需要任何本地计算机环境配置
- 再本地使用 python 和机器学习进行策略开发需要进行开发环境的安装配置，具体流程如下：
  1. 安装Anacondahttps://www.anaconda.com/download
  2. 安装所需包
  3. 安装集成开发环境PyCharm社区版https://www.jetbrains.com/pycharm/download
  4. 新建项目和文件

Python简介

- 1990年出现，1991年1.0版本发行
- 多用途面向对象高级语言，动态类型，语法简洁高效，结构灵活
- "胶水语言"，能够结合其他很多低级(c/c++等)或高级语言(Matlab, Julia, Scala等)，实现跨语言编程，同时保证开发效率和运行效率
- 丰富的各类第三方库，方便直接开发，科学计算和数据分析能力强大

Python语言的特点示例：

- 原生数据类型
  - 基础类型：`int`、`float`、`str`等
  - 容器类型：`list`、`dict`等
- 基本语法
- 流程控制
- 函数和方法

Numpy简介

- 什么是Numpy
  - 以c语言为内核的高效数值运算库
  - 各种科学计算库的基础库
- Numpy能用来干什么
  - 普通运算
  - 大量数据、大型矩阵计算

Numpy运算示例：

- 普通四则运算
- 矩阵运算

Pandas简介

- 什么是Pandas
  - 类似Excel的二维数据运算和处理库
  - 以Numpy为基础，包含更直观易用的方法
- Pandas能用来干什么：
  - 记录数据
  - 以更直观的方式分析和变换数据

Pandas数据处理示例：

- 读取和保存数据
- 数据运算和变换
- 数据索引和查找
- 数据异常值处理
- 统计模型应用

Scikit-learn

- 什么是Scikit-learn
  - 数据处理、机器学习和统计学习库
  - 包含完备的数据分析框架
- Scikit-learn能用来干什么
  - 数据分析处理流程
  - 监督学习（分类和回归），无监督学习（聚类）
  - 统计分析

目录

数据结构介绍

本课程使用上海商品期货交易所2017年1月-2018年10月的5分钟频率橡胶主力连续合约的行情数据作为策略开发数据，行情数据的基本结构如下：

- 交易时段：日盘由上午9点至11点30结束，盘中10点15至10点30休市，下午1点30至3点结束，夜盘由晚上9点至11点结束
- 数据包括：时间戳、每5分钟K线的开盘价，最高价，最低价，收盘价，交易量，持仓量

基本统计

在处理交易数据之前，我们需要通过一些统计信息对数据有一个直观认识：

- 基本统计信息：最小值min()，最大值max()，平均值mean()，标准差std()，数量coutn()
- 分布情况：hist()

缺失和异常值分析

由于数据来源等原因，用于构建策略的数据可能存在数据缺失情况，我们需要了解缺失情况并做出相应的处理：

- 缺失占比
- 缺失分布

异常值是指由于数据生成或获取过程中造成的极个别数据与其他数据存在较大差异的情况，异常值分析方法包括：

- 分位数法
- 标准差法

缺失值和异常值处理

根据前面已经获得的数据统计基本信息，我们将进一步对数据进行对应情形的处理。对于缺失数据，我们可以有如下几种处理方式

- 去掉样本：drop
- 后填样本：backfill
- 前填样本：forwardfill

具体采用哪一类处理方式需要根据具体情况而定。对于采用分位数法或者标准差法获得的异常值信息，我们一般将样本数据直接去掉，因为其存在将会影响整个数据集合的统计特征。

数据标签

通过之前的数据分析和处理过程，我们已经获得了干净并且具有市场本身规律和特征的数据集合。这些数据是机器学习的基础，在将上述数据送入机器学习模型之前，我们必须将每个样本打上标签，即告诉机器学习算法，通过什么数据应该学习到什么数据特征。对于机器学习和机器学习模型的介绍我们将在"模型"部分涉及。

- 什么是数据标签：将处理过的数据分类（在本课程的应用场景下），机器学习算法通过标签学习其对应样本特征的规律
- 怎么给行情数据打标签：
  - 交易逻辑：我们的策略希望在低点（高点）多头（空头）进场，在高点（低点）多头（空头）出场，以获得正收益
  - 采用收益的正负对样本打上标签：对于在时间点t上的样本X_t，我们根据h时长之后的收益$r_{t,t+h}$的正负将样本X_t打上标签$y_t \in \{-1, 1\}$
  - 数据标签将数据分为了两种类型，监督类型的机器学习算法就可以理解需要学习X_t和y_t之间的关系，这些关系集成在一起就是我们的机器学习模型

数据平衡性检验

数据的平衡性指不同标签的样本占总样本的比例是否相等或者近似相等。例如，在根据价格走势所计算出的收益序列对样本进行二分类的过程中，如果价格序列来自于带有上升趋势的行情，上述打标签方法有可能导致带1标签，即正收益的样本数大于带−1标签，即负收益的样本数。数据如果不平衡可能导致以下一些问题：

- 机器学习算法输出模型可能只适合于某种形态的市场
- 较高的预测准确率来自于模型直接将所有样本预测为其中一类数据，即机器学习没有真正学习样本数据的特征，无法在实盘预测过程中给出正确的预测结果

数据平衡的检验可以通过分析样本标签的占比体现。

数据平衡性调整

如果带标签数据出现不平衡的情况，我们可以通过上采样和下采样的数据处理方式解决数据不平衡问题：

- 上采样：对样本中占比较少的一类通过scikit-learn中的上采样方法进行重采样
- 下采样：对样本中占比较多的一类通过scikit-learn中的下采样方法进行冲采样
- 抽样去除样本中占比较多的一类中的部分样本，使两类样本占比接近

目录

特征生成

特征工程是机器学习框架的核心之一，如何构建生成特征将直接影响机器学习模型预测效率的高低。

- 什么是特征：在前面章节中我们介绍了如何对样本X_i进行打标签，样本X_i一般情况下是一个一维向量，
  即$X_i = (x_{i1}, x_{i2}, \cdots, x_{ik})$，其中等式右边的每一个$x_i$都是一个特征，所以$X_i$是一个包含$k$个特征的样本
- 怎么可以生成特征：
  - 基础数据的变换，如我们可以对沪胶主连数据的6个基本字段进行运算生成特征
  - 已有指标规则，如Ta-lib指标库，即技术分析所使用的各类指标
  - 文献

特征缩放和变换

特征缩放是指按照一定规则对生成的特征进行不改变其统计性质的运算，使其能够更好地适应机器学习模型。

- 为什么要进行特征缩放：机器学习模型的核心是统计方法，而统计方法的有效性有许多假设前提，某些数据特别是金融时序数据往往不满足这些假设前提，从而导致机器学习方法效率降低等问题
- 有哪些特征缩放方法：标准化、归一化等
- 如何进行特征缩放：使用scikit-learn中数据处理方法对特征进行缩放

特征变换是指改变特征统计性质的运算，特征变换能够在已有的特征基础上衍生更多特征，并使得衍生特征与已有特征之间的不存在线性关系，为更好的训练模型提供信息。

相关性分析

在前一小节中我们学习了如何生成特征，但生成的不同特征并没有评价标准告诉我们特征的优劣，所以对生成的特征进行分析也是机器学习方法的重要部分。相关性分析指通过两两分析特征之间的相关性，检验特征之间是否存在线性关系。

- 搞相关性的特征会减弱模型的预测效率，降低模型的稳定性
- 相关性分析是分析特征之间的线性相关关系

特征重要性分析

特征重要性分析相对建立机器学习模型而言是事后分析，即通过机器学习模型对不同特征发挥预测能力的贡献来评价那些特征重要性比较高。特征重要性分析的流程可以分为以下步骤：

- 准备特征
- 样本打标签和样本数据处理
- 建立机器学习模型(将在本课程的"模型"部分详细介绍)
- 根据机器学习模型对特征评价，从特征全集中挑选部分特征重要性较高的特征
- 再次挑选出来的重要性较高的特征进行机器学习建模，输出模型作为策略使用模型

其他分析

非监督机器学习方法可以用作特征分析。

目录

模型原理

机器学习模型由三部分组成：

1. 模型结构：封装统计方法的一整套流程，用作预测
2. 模型参数：控制模型结构的输入
3. 学习机制：调整或优化模型参数以得到模型的机制

机器学习模型种类繁多，从简单的逻辑回归到复杂的神经网络，选择适合的模型类型能够帮助我们高效的完成我们交给模型的任务。本课程将采用随机森林机器学习方法讲解建模过程。

- 什么是随机森林：决策树模型的组合，决策树模型是机器学习方法中较为简单的模型，其优点是直观容易解释（比一般线性模型更容易解释），缺点是容易数据过拟合，而将决策树模型集成组合起来就形成了随机森林。
- 什么是决策树：决策树是一个预测模型；他代表的是对象属性与对象值之间的一种映射关系。树中每个节点表示某个对象，而每个分叉路径则代表的某个可能的属性值，而每个叶结点则对应从根节点到该叶节点所经历的路径所表示的对象的值。

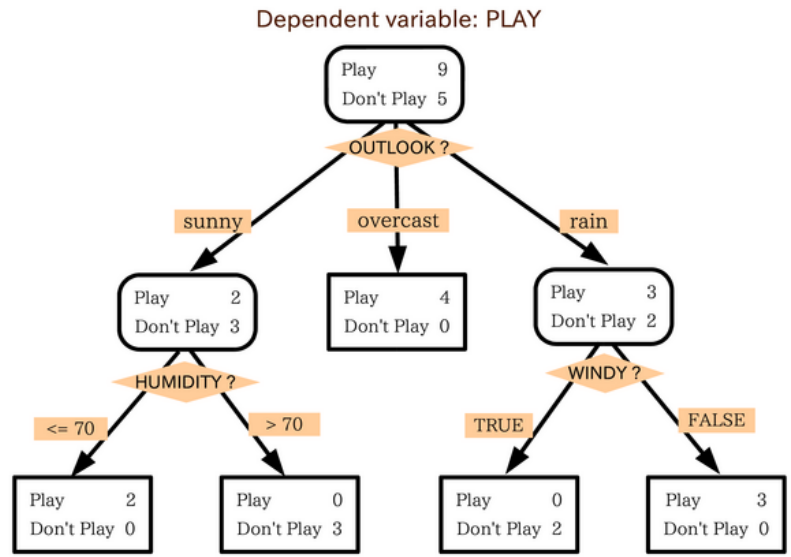
决策树原理

高尔夫俱乐部会员是否打球和天气、温度、湿度、是否有风之间的关系

Play golf dataset

Independent variables				Dep. var
OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
sunny	85	85	FALSE	Don't Play
sunny	80	90	TRUE	Don't Play
overcast	83	78	FALSE	Play
rain	70	96	FALSE	Play
rain	68	80	FALSE	Play
rain	65	70	TRUE	Don't Play
overcast	64	65	TRUE	Play
sunny	72	95	FALSE	Don't Play
sunny	69	70	FALSE	Play
rain	75	80	FALSE	Play
sunny	75	70	TRUE	Play
overcast	72	90	TRUE	Play
overcast	81	75	FALSE	Play
rain	71	80	TRUE	Don't Play

(a) 样本数据



(b) 决策树示例

随机森林原理

随机森林是将多个决策树模型包装在一起，随机森林模型的预测输出来自于每一个决策树模型预测结果的平均，例如我们对沪胶主连数据以及相关特征建模以后，随机森林以及其中所包含的决策树的图示：

模型超参和寻参

构建随机森立的参数将在较大程度上影响模型预测能力的好坏，对于scikit-learn中的随机森林模型，其参数和对应含义如下：
决策树相关参数（决策树是随机森林的基本构成）：

- criterion: "gini"或者"entropy"，是计算属性的gini(基尼不纯度)还是entropy(信息增益)，来选择最合适的节点
- splitter: "best" or "random" 随机选择属性还是选择不纯度最大的属性
- max_features: 选择最适属性时划分的特征不能超过此值
- max_depth: 设置树的最大深度，默认为None，这样建树时，会使每一个叶节点只有一个类别，或是达到min_samples_split。
- min_samples_split:根据属性划分节点时，每个划分最少的样本数。
- min_samples_leaf:叶子节点最少的样本数。
- max_leaf_nodes: 叶子树的最大样本数。
- min_weight_fraction_leaf: 叶子节点所需要的最小权值
- verbose:(default=0) 是否显示任务进程

随机森林自身参数：

- n_estimators=10：决策树的个数，越多越好，但是性能就会越差，至少100左右，可以达到可接受的性能和误差率
- bootstrap=True：是否有放回的采样
- oob_score=False：oob（out of band，带外）数据，即：在某次决策树训练中没有被bootstrap选中的数据
- n_jobs=1：并行job个数
- warm_start=False：热启动，决定是否使用上次调用该类的结果然后增加新的。
- class_weight=None：各个label的权重。

模型保存和载入

对于训练好以后的模型我们可以用python的pickle库保存和载入

交叉验证

交叉验证（Cross validation），有时亦称循环估计，是一种统计学上将数据样本切割成较小子集的实用方法。于是可以先在一个子集上做分析，而其它子集则用来做后续对此分析的确认及验证。一开始的子集被称为训练集。而其它的子集则被称为验证集或测试集。交叉验证是一种评估统计分析、机器学习算法对独立于训练数据的数据集的泛化能力。交叉验证一般要尽量满足：

- 训练集的比例要足够多，一般大于一半
- 训练集和测试集要均匀抽样

交叉验证可以分为以下3类：

1. k-folder cross-validation： k个子集，每个子集均做一次测试集，其余的作为训练集。交叉验证重复k次，每次选择一个子集作为测试集，并将k次的平均交叉验证识别正确率作为结果。
2. K * 2 folder cross-validation：是k-folder cross-validation的一个变体，对每一个folder，都平均分成两个集合s0,s1，我们先在集合s0训练用s1测试，然后用s1训练s0测试。
3. least-one-out cross-validation：假设dataset中有n个样本，每个样本单独作为一次测试集，剩余n-1个样本则做为训练集

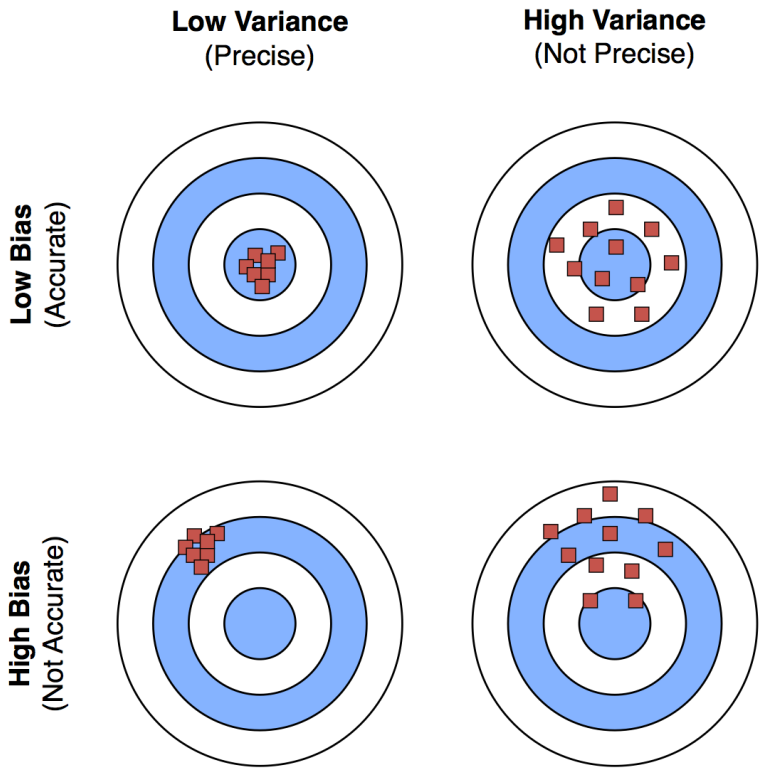
模型评价方法

本课程使用的是随机森林分类器（classifier）模型，对于分类器类型模型其评价指标主要包括：accuracy， precision，recall，F-score，ROC-AUC，gini系数等。对于2分类问题，可以使用混淆矩阵获得模型预测能力的初步评价。

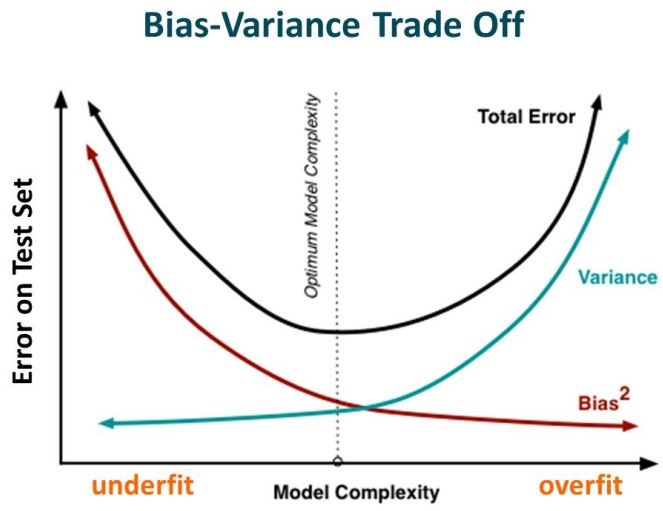
混淆矩阵		预测		
		正类 1	负类 0	合计
实际	正类 1	TP	FN	TP+FN
	负类 0	FP	TN	FP+TN
	合计	TP+FP	FN+TN	TP+FN+FP+TN

- 查准率(precision)：$p = \frac{TP}{TP+FP}$
- 召回率(recall)：$r = \frac{TP}{TP+FN}$
- 准确率(accuracy)：$a = \frac{TP+TN}{TP+FN+TN+FP}$
- F-1值：$f1 = \frac{2*p*r}{p+r}$

模型欠拟合和过拟合

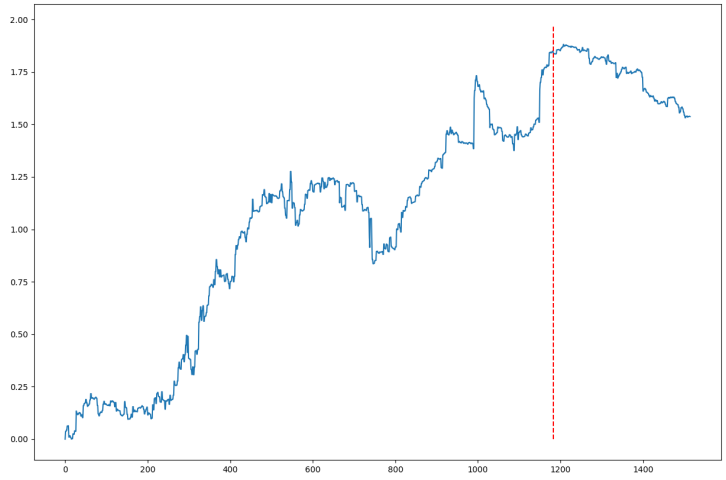


(c) 偏误和方差

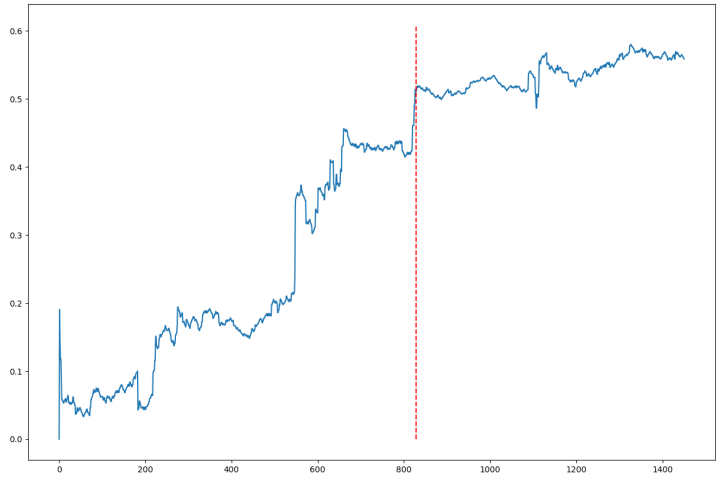


(d) 欠拟合和过拟合

模型欠拟合和过拟合的回测表现



(e) 欠拟合或过拟合



(f) 最优拟合

目录

信号生成

我们将数据分为两个部分，第一部分用作模型训练和检验，第二部分用作使用生成模型作为策略的回测表现检验。生成回测阶段的信号方法步骤如下：

1. 载入模型($model$)对回测部分数据的模型信号进行预测：
   - 持仓：持仓根据模型预测的概率值大小的线性变换决定，即 $position = f(model.predict\_proba(X_{backtest}))$
   - 方向：持仓方向根据模型预测的标签值决定，即 $direction = model.predict(X_{backtest})$
2. 根据得到的仓位大小（多少手） $position$和持仓方向（多或空） $direction$ 两个序列，我们可以得到策略信号 $signal = position \times direction$

回测方法

- 本课程基于便利性考虑，使用回测数据循环的方式进行简化回测
- 将已经获得的*signal*序列和回测数据序列中的收盘价传入本课程为同学们准备的回测函数即可得到回测结果

回测评价

我们主要采用以下指标对回测结果（收益率序列）进行评价：

- 收益率大小
- 收益率波动性
- 夏普比率
- 最大回撤

目录

课程小结

- 数据处理
- 特征生成
- 模型训练和评价
- 策略回测评价