# 基于支持向量机的机器学习策略

## 机器学习策略简介

### 什么是机器学习策略？

从广义上来说,机器学习是一种能够赋予机器学习的能力以此让它完成直接编程无法完成的功能的方法.但从实践的意义上来说,机器学习是一种通过利用数据,训练出模型,然后使用模型预测的一种方法.而机器学习策略,则是通过机器学习方法,通过输入训练数据与模型来获得对未来数据的价格或者涨跌的预期判断,进而进行投资的量化策略.

## 支持向量机简介

### 什么是支持向量机？

支持向量机(Support Vector Machines, SVM) 被Vapnik与他的合作者提出于1995年,基础为统计学习理论和结构风险最小化原则.支持向量机具有完备的理论基础和出色的学习能力,是借助于最优化方法解决有限样本机器学习问题的数据挖掘出色方法之一.

### 支持向量机的原理是？

假设给定一个特征空间上的训练集:

其中为第个特征向量,为的类标记,当时,称为正例,当时,称为负例.

#### 线性可分SVM

假设训练数据集线性可分,通过间隔最大化或等价求解相应凸二次规划问题得到分离超平面和相应的分类决策函数为线性可分SVM.

为了量化分类的正确性和确信度,引入函数间隔的概念.

对于给定的训练数据集和超平面,定义超平面关于样本点的函数间隔为:

定义超平面关于训练数据集的函数间隔为关于中所有样本点的函数间隔的最小值:

为了取消成比例改变导致函数间隔变化但超平面不变的问题,引入规范化,此时函数间隔变为几何间隔.

对于给定的训练数据集和超平面,定义超平面关于样本点的几何间隔为:

定义超平面关于训练数据集的函数间隔为关于中所有样本点的几何间隔的最小值:

可知函数间隔和几何间隔的关系为:

,

##### 硬间隔最大化

通过寻找最大化几何间隔的分离超平面可以以充分大的确信度对训练数据进行分类,最大化几何间隔又称为硬间隔最大化,此时的约束最优化问题为:

即最大化超平面关于训练数据集的几何间隔

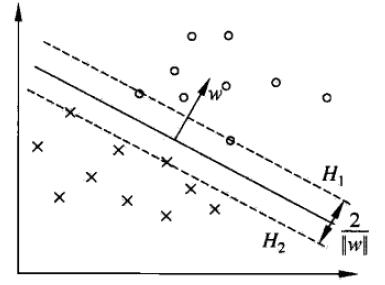
考虑,问题可改写为:

基于与引入规范化同样的原因,的取值对结果没有影响,故取,考虑

有以下线性可分SVM学习的最优化问题:

该问题是一个凸二次规划问题.可以证明,若训练数据集线性可分,那么可将训练数据集中的样本点完全正确分开的最大间隔超平面存在且唯一.

在线性可分的情况下,训练数据集的样本点与分离超平面距离最近的样本点的示例称为支持向量,如下图的和上的点:



和之间的距离称为间隔,为,和称为间隔边界.

##### 线性可分SVM对偶学习算法

通过运用拉格朗日对偶性,可以得到原始问题的对偶问题,对偶问题往往更容易求解,也便于引入核函数推广到非线性分类问题.拉格朗日对偶性在此不赘述,但有个重要定理需要特别指出:对于原始问题和对偶问题,在满足特定条件时,则分别为原始问题和对偶问题的解的充要条件为满足KKT条件.KKT条件表述如下:

(KKT的对偶互补条件)

向引入拉格朗日乘子定义拉格朗日函数

其中,为拉格朗日乘子向量.

根据拉格朗日对偶性可以得到原始问题的对偶问题即极大极小问题:

先求,分别让对的偏导等于0,有:

把代入整理可得:

接下来再求,其对偶问题为:

等价于下面的最优化问题:

可以证明以下定理:设为对偶优化问题的解,则且原始最优化问题可按下式求解:

此时有分离超平面:

分类决策函数:

考虑原始最优化问题和对偶最优化问题,将数据集中对应的样本点的实例称为支持向量.可证明支持向量一定在间隔边界上.

##### 线性可分SVM学习算法

构造求解约束最优化问题

求得最优解

计算并选取计算.

求得分离超平面:

分类决策函数:

#### 软间隔最大化线性SVM

假设训练数据集线性不可分且存在特异点,除特异点以外的数据线性可分.

为了使得线性不可分而不满足中约束条件的的训练数据集可被训练,对每个样本点引入松弛变量,并支付相应的距离代价,使得变为:

其中,称为惩罚系数,取值由实际问题决定.这个思路称为软间隔最大化.

此时线性不可分SVM的学习问题变为如下的凸二次规划原始问题:

可以证明为的解,且唯一,不唯一且存在于一个区间中.

的对偶问题是:

##### 线性SVM学习算法

选择惩罚参数,构造并求解凸二次规划问题

求得最优解

计算并选取计算.

求得分离超平面:

分类决策函数:

#### 非线性SVM

假设训练数据集能用中的一个超曲面将正负例分开,这种分类问题被称为非线性可分问题.

对于这种问题,需要应用SVM的核技巧如下:

通过非线性变换将输入空间(欧式空间或离散集合)对应于一个特征空间(希尔伯特空间)

输入空间中的超曲面模型对应于特征空间中的超平面模型(SVM)

通过求解特征空间中的线性SVM完成非线性可分分类问题的学习任务.

设为输入空间(欧式空间的子集或离散集合),又称为特征空间(希尔伯特空间),若使得,则称为核函数,称为映射函数.称为和的内积.

用核函数代替对偶问题中的目标函数和分类决策函数的内积,有新的目标函数:

和新的分类决策函数:

若使用多项式核函数,对应的SVM为次多项式分类器,分类决策函数成为:

若使用高斯核函数,对应的SVM为高斯径向基函数(RBF)分类器,分类决策函数成为:

##### 非线性SVM学习算法

选取适当核函数和适当的惩罚参数,构造并求解最优化问题

求得最优解

并选取计算.

构造分类决策函数:

可以证明,当为正定核函数时,为凸二次规划问题并有解.

## 附录

### B. Gram矩阵

定义:维欧式空间任意个向量的内积组成的矩阵

称为个向量的Gram矩阵,行列式称为Gram行列式.

定理:欧式空间中向量的Gram矩阵必为半正定矩阵,为正定矩阵线性无关.

证明:对向量组应用Schmidt正交化过程,得到一组正交向量组,把用表示如下:

由的正交性可得

因为的阶主子阵也是Gram矩阵,所以行列式的值大于等于零,因此也是半正定矩阵.

,对于线性相关向量组,必有一向量是其余向量的线性组合,反之亦真.设是的线性组合,有:

把的第列减去第列的倍,由内积的线性性质有,故有:

综上可得,为正定矩阵线性无关.

证毕.

## 机器学习策略实现（基于掘金量化平台）

### 策略思想

* 选取7个特征变量:1.收盘价/均值2.现量/均量3.最高价/均价4.最低价/均价5.现量6.区间收益率7.区间标准差
* 用时间窗口为15个交易日的历史数据训练出预测未来5个交易日收益率为上涨或下跌的二分类SVM模型
* 在对行情数据预测为上涨的时候买入,并采取止盈止损操作

### 策略主要步骤实现

#### 订阅数据

subscribe(symbols=context.symbol, frequency='60s')

订阅数据需要调用subscribe函数.

* symbols设置订阅的标的代码.
* frequency设置周期级别,这里设置60s 表示以六十秒为周期.

#### 获取历史数据

recent\_data = history(symbol=context.symbol, frequency='1d',  
 start\_time=start\_date, end\_time=end\_date,  
 fill\_missing='Last',df=True)

获取历史数据需要调用history函数.

* symbols需要设置获取数据的标的代码.
* frequency设置周期级别,这里设置1d 表示以一天为周期.
* start\_time设置获取数据的开始时间.
* end\_time设置获取数据的结束时间.
* fill\_missing设置补全数据的方式,这里设置Last 表示用上一个收盘价填充缺失值.
* df设置是否返回DataFrame类型数据,这里设置True 表示返回该类型的数据.

#### 设置SVM的参数并训练

context.clf = svm.SVC(C=1.0, kernel='rbf', gamma='auto', shrinking=True, tol=0.001, cache\_size=200, verbose=False, max\_iter=-1, random\_state=None)  
context.clf.fit(x\_train, y\_train)

* C 设置误差项的惩罚参数.
* kernel 设置算法中使用的核函数,这里设置rbf 表示用RBF核函数.
* g1amma 设置核的函数系数,这里设置auto 表示设置系数为1/特征数.
* shrinking 设置是否使用启发式搜索算法,这里设置True 表示为使用.
* tol 设置停止训练的精度.
* cache\_size 设置训练需要的内存.
* verbose 设置是否输出详细信息,此处设置False表示不输出.
* max\_iter 设置最大的迭代次数,此处设置-1表示不限制.
* random\_state 设置随机数种子使得结果可复现,此处设置None表示随机生成.

#### 获取当前年月日

today = bar.bob.strftime('%Y-%m-%d')

#### 返回指定日期的上一个交易日

last\_day = get\_previous\_trading\_date(exchange='SHSE', date=today)

* exchange 设置交易所,此处设置SHSE为上交所.
* date 设置作为参考的时间点.

#### 获取持有的仓位

position = context.account().position(symbol=context.symbol, side=PositionSide\_Long)

* symbol 设置标的.
* side 设置仓位方向,此处设置为PositionSide\_Long 为多仓.

#### 获取预测的结果

prediction = context.clf.predict(features)[0]

#### 获取持有的仓位

position = context.account().position(symbol=context.symbol, side=PositionSide\_Long)

* symbol 设置标的.
* side 设置仓位方向,此处设置为PositionSide\_Long 为多仓.

#### 调仓到目标持仓比例

order\_target\_percent(symbol=context.symbol, percent=0.95, order\_type=OrderType\_Market, position\_side=PositionSide\_Long)

* symbol 设置标的.
* percent 设置目标比例.
* order\_type 设置订单类型,此处设置为OrderType\_Market 为市价单.
* side 设置仓位方向,此处设置为PositionSide\_Long 为多仓.

#### 平当前所有可平持仓

order\_close\_all()

### 策略回测分析

#### 回测报告



#### 分析

我们选取了SHESE.60009上海机场2016-04-01到2017-07-30的数据作为训练集,2017年8月1日至2017年9月5日作为回测周期,应用上述的SVM机器学习模型作为买卖的基础.可以看出：

* **胜率**（具有盈利的平仓次数与总平仓次数之比）达到了100%,也即策略在回测期间的每次开仓都是盈利的.
* **卡玛比率**（年化收益率与历史最大回撤之比）是使用**最大回撤率**来衡量风险.**采用最大回撤率来衡量风险,关注的是最极端的情况.**卡玛比率越高表示策略承受每单位最大损失获得的报酬越高.在这里卡玛比率达到了19.57.
* **夏普比率**（年化收益率减无风险收益率的差与收益波动率之比）也即每承受一单位的总风险,会产生3.27单位的超额回报.
* **策略收益曲线**总体曲线向上,体现了机器学习在量化投资中的巨大潜力.