# 基于支持向量机的投资策略

分工：

吴思婷：PPT、研究报告 、模型测试

何友鑫：代码、SVM思路

李希孟：数据收集、清洗

### 策略背景、发展与创新

#### 1.1背景

数据挖掘（Data Mining）是从大量的、不完全的、有噪声的、模糊的、随机的数据中提取隐含在其中的、人们事先不知道的，但又是潜在有用的信息和知识的过程。

与数据挖掘相近的同义词有数据融合、数据分析和决策支持等。在量化投资中，数据挖掘的主要技术包括关联分析、分类/预测、聚类分析等。

关联分析是研究两个或两个以上变量的取值之间存在某种规律性。例如，研究股票的某些因子发生变化后，对未来一段时间股价之间的关联关系。关联分为简单关联、时序关联和因果关联。关联分析的目的是找出数据库中隐藏的关联网。一般用支持度和可信度两个阈值来度量关联规则的相关性，还不断引入兴趣度、相关性等参数，使得所挖掘的规则更符合需求。

分类就是找出一个类别的概念描述，它代表了这类数据的整体信息，即该类的内涵描述，并用这种描述来构造模型，一般用规则或决策树模式表示。分类是利用训练数据集通过一定的算法而求得分类规则。分类可被用于规则描述和预测。

预测是利用历史数据找出变化规律，建立模型，并由此模型对未来数据的种类及特征进行预测。预测关心的是精度和不确定性，通常用预测方差来度量。

聚类就是利用数据的相似性判断出数据的聚合程度，使得同一个类别中的数据尽可能相似，不同类别的数据尽可能相异。

#### 1.2优缺点

支持向量机（Support Vector Machine，SVM）方法是数据挖掘的一种方法，是通过一个非线性映射，把样本空间映射到一个高维乃至无穷维的特征空间中（Hilbert空间），使得在原来的样本空间中非线性可分的问题转化为在特征空间中的线性可分的问题，简单地说，就是升维和线性化。升维就是把样本向高维空间做映射，一般情况下这会增加计算的复杂性，甚至会引起维数灾难，因而人们很少问津。但是作为分类、回归等问题来说，很可能在低维样本空间无法线性处理的样本集，在高维特征空间中却可以通过一个线性超平面实现线性划分（或回归）。

一般的升维都会带来计算的复杂化，SVM方法巧妙地解决了这个难题：应用核函数的展开定理，就不需要知道非线性映射的显式表达式；由于是在高维特征空间中建立线性学习机，所以与线性模型相比，不但几乎不增加计算的复杂性，而且在某种程度上避免了维数灾难。这一切要归功于核函数的展开和计算理论。

正因为有这个优势，使得SVM特别适合于进行有关分类和预测问题的处理，这就使得它在量化投资中有了很大的用武之地。

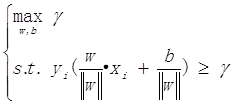
这种方法的缺点也很明显，例如缺乏必要的经济学解释，只是单纯地从数据中找规律。很多时候都带有随机性，导致策略不够稳定。

#### 1.3支持向量机的相关理论基础

**间隔最大化**

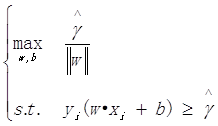
支持向量机学习的基本思想是求解能够正确划分训练数据集并且几何间隔最大的分离超平面。对线性可分得训练数据集而言，线性可分分离超平面有无穷多个，但是几个间隔最大的分离超平面是唯一的。（这里的间隔最大化也称为硬间隔最大，与训练数据集近似可分时软间隔最大化相对应）。间隔最大化意味着以充分大的确信度对训练数据进行分类，也就是说不仅将正负实例点进行分开，而且对离超平面较近的实例点也有足够大的确信度将它们分开，这样的超平面对未知的新实例也有较好的分类预测能力。

求解几何间隔最大化的分离超平面，等价于以下约束条件下的最优化问题：

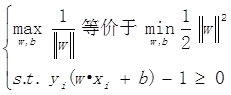


即我们希望最大化超平面(w,b)关于训练数据集的几何间隔γ，并且超平面(w,b)关于每个训练样本点的几何间隔至少为γ。

[考虑到函数间隔和几何间隔的关系，上述约束条件下的最](http://www.dearopinion.com/data/" \t "_blank) 优化问题也可以改写成



此外，上式中函数间隔取值并不影响最优化问题的解，因此我们可以令函数间隔等于1，带入上式得到下面的线性可分支持向量机学习的最优化问题：



如果求出了上述约束最优化问题的解w \* ,b \* ，那么就可以得到最大间隔分离超平面 [w \* x+b \*](http://www.dearopinion.com/data/" \t "_blank)=0及分类决策函数f(x)=sign(w \* x+b \* )，即线性可分支持向量机模型。

支持向量和间隔边界

在线性可分得情况下，训练数据集中的样本点与分离超平面距离最近的样本点的实例称为 支持向量（ Support Vector ） ，它是使约束条件等号成立的点，也就是

clip_image028

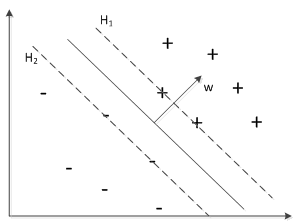
对于yi=+1的正例，支持向量在超平面H 1 上

clip_image030

对于yi=-1的负例，支持向量在超平面H 2 上

clip_image032

下图中，在H 1 和H 2 上的点就是支持向量，H 1 和H 2 相互平行，并且没有实例点落在它们中间，分离超平面位于H1 和H 2 形成一条长带的中间。长带的宽带（H 1 和H 2 之间的距离）称为间隔，H 1 和H 2 称为 间隔边界 ，间隔的大小依赖于超平面的法向量w，等于2/||w||。



在决定分离超平面时只有支持向量起作用，而不依赖于其他实例点。如果移动支持向量将改变所求的解，但是如果在间隔边界以外移动其它实例点，则不影响所求的解。由于支持向量在确定分离超平面中起着决定性的作用，所以将这种分离模型称为支持向量机。支持向量的个数一般很少，所以支持向量机由很少的“重要”训练样本确定。

### 二、策略收益来源

策略收益来源主要来自对指数走势的分类和预测，在较大的概率上预测准指数的趋势，则可以从中获得对应收益。

### 三、方法说明

该策略利用HS300 2005年至2017的样本数据，把数据分为两个部分，第一部分为训练数据，即2005年到2016年，第二部分为测试数据2017年1-12月。

具体步骤如下：

1. 对所有样本数据进行标记，针对下一个交易日的涨跌幅：涨跌幅低于-1%记为-2; 涨跌幅高于-1%，低于0%，记为 -1;涨跌幅高于 1%记为 2; 涨跌幅低于-1%，高于0%，记为 1;
2. 对下一期的收益率进行线性回归找出显著的解释因子。
3. 利用SVM对训练数据进行建模，并用这个模型来预测测试样本。

### 四、策略应用

读文件，lm回归判断哪些解释变量是有用的，减少嗓音。确定自变量和因变量

data\_sample <- read.zoo("./data/HS300\_5.csv",sep=",",header=T,format = "%Y-%m-%d")  
data\_sample <- na.omit(data\_sample)  
fit <- lm(ret~open+close+high+low+volume+mv10+mv20+vol10+vol20+rsi5+rsi14+macd.macd1+signal.macd1+macd.macd2+signal.macd2+dn+mavg+up+pctB,data = data\_sample)  
summary(fit)

##   
## Call:  
## lm(formula = ret ~ open + close + high + low + volume + mv10 +   
## mv20 + vol10 + vol20 + rsi5 + rsi14 + macd.macd1 + signal.macd1 +   
## macd.macd2 + signal.macd2 + dn + mavg + up + pctB, data = data\_sample)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.092163 -0.007645 0.000413 0.008760 0.090727   
##   
## Coefficients: (3 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.606e-03 2.297e-03 2.005 0.045037 \*   
## open -6.408e-05 1.751e-05 -3.659 0.000258 \*\*\*  
## close -3.926e-05 1.579e-05 -2.487 0.012936 \*   
## high 8.806e-05 2.023e-05 4.352 1.4e-05 \*\*\*  
## low 1.283e-05 1.653e-05 0.776 0.437755   
## volume -7.027e-14 4.726e-12 -0.015 0.988138   
## mv10 1.195e-05 1.981e-05 0.603 0.546551   
## mv20 -1.151e-05 1.473e-05 -0.781 0.434633   
## vol10 5.009e-06 1.221e-05 0.410 0.681690   
## vol20 -6.301e-06 8.962e-06 -0.703 0.482050   
## rsi5 -2.334e-05 2.446e-05 -0.954 0.340036   
## rsi14 -4.370e-05 5.207e-05 -0.839 0.401356   
## macd.macd1 1.609e-03 1.266e-03 1.271 0.203903   
## signal.macd1 -4.636e-04 5.239e-04 -0.885 0.376194   
## macd.macd2 -3.872e-04 8.115e-04 -0.477 0.633314   
## signal.macd2 -8.485e-04 1.548e-03 -0.548 0.583625   
## dn NA NA NA NA   
## mavg NA NA NA NA   
## up NA NA NA NA   
## pctB 6.473e-03 2.955e-03 2.190 0.028582 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.01806 on 2849 degrees of freedom  
## Multiple R-squared: 0.01937, Adjusted R-squared: 0.01386   
## F-statistic: 3.517 on 16 and 2849 DF, p-value: 2.566e-06

确定训练样本和测试样本

x <- data\_sample[,-c(4:21)]  
y <- data\_sample[,20]  
insams<- "2005-01-01"  
insame<- "2016-12-31"  
osams<- "2017-01-01"  
osame<- "2017-12-31"  
inrow <- which(index(data\_sample) >= insams & index(data\_sample) <= insame)  
outrow <- which(index(data\_sample) >= osams & index(data\_sample) <= osame)

计算SVM在2种分类机，4种核函数下模型的错误次数

type <- c("C-classification","nu-classification")  
kernel <- c("linear","polynomial","radial","sigmoid")  
accuracy <- matrix(0,2,4)  
for (i in 1:2)  
{  
 for ( j in 1:4)   
 {  
 model <- svm(x[inrow,],y[inrow],type=type[i],kernel = kernel[j])  
 pred\_temp <- predict(model,x[outrow])  
 accuracy[i,j] <- sum(pred\_temp!=as.vector(y[outrow]))  
 }  
}  
dimnames(accuracy) <- list(type,kernel)  
accuracy

## linear polynomial radial sigmoid  
## C-classification 113 112 145 213  
## nu-classification 117 218 166 145

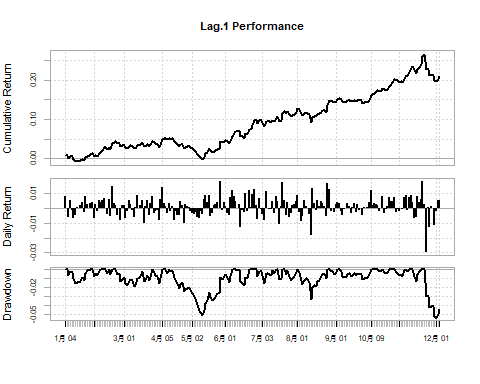
由以上结果可知，使用SVM进行实验，type="C-classification",kernel = "polynomial"的模型最优。

model1 <- svm(x[inrow,],y[inrow],type="C-classification",kernel = "polynomial")  
pred1 <- predict(model1,x[outrow,])  
#table(pred1,y[outrow])  
outresult\_out<- confusionMatrix(pred1,y[outrow])  
outresult\_out

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction -2 -1 1 2  
## -2 0 0 0 0  
## -1 0 0 0 0  
## 1 8 91 114 13  
## 2 0 0 0 0  
##   
## Overall Statistics  
##   
## Accuracy : 0.5044   
## 95% CI : (0.4373, 0.5714)  
## No Information Rate : 0.5044   
## P-Value [Acc > NIR] : 0.5266   
##   
## Kappa : 0   
## Mcnemar's Test P-Value : NA   
##   
## Statistics by Class:  
##   
## Class: -2 Class: -1 Class: 1 Class: 2  
## Sensitivity 0.0000 0.0000 1.0000 0.00000  
## Specificity 1.0000 1.0000 0.0000 1.00000  
## Pos Pred Value NaN NaN 0.5044 NaN  
## Neg Pred Value 0.9646 0.5973 NaN 0.94248  
## Prevalence 0.0354 0.4027 0.5044 0.05752  
## Detection Rate 0.0000 0.0000 0.5044 0.00000  
## Detection Prevalence 0.0000 0.0000 1.0000 0.00000  
## Balanced Accuracy 0.5000 0.5000 0.5000 0.50000

简单回测展示2017年收益情况 预测为1或2开多仓，预测为-2开空仓

signal <- ifelse( pred1==1 | pred1==2,1,ifelse(pred1==-2 ,-1,0))  
simreturn <- data\_sample$ret[outrow]  
cost <- 0  
strategy\_return <- Lag(simreturn)\*Lag(signal)-cost  
cumm\_return<- Return.cumulative(strategy\_return)  
annual\_return <- Return.annualized(strategy\_return)  
charts.PerformanceSummary(strategy\_return)



### 结果分析

2017年HS300指数的收益率为18.8%，该策略为21%，略高于指数收益，表现还不错。但在11月有一个很大的回撤区间导致收益大幅损失。