《金融风险管理》课程论文：
基于企业风险的 SVM 选股策略

141292018　桑梓洲

摘要

企业财务风险与投资者利益密切相关，其预警研究也一直是国内外学术界关注的重要课题。受课堂上 Altman[1] 等学者研究启发并跟随机器学习时代浪潮，我们试图结合针对小样本、高维数据分类效果较好的 SVM 技术对企业财务风险识别做出优化。但受制于数据来源最终我们又引入盈利、估值等指标形成了基于企业风险的批量选股策略，经过 14-16 年三年沪市数据实证检验表明在合适的参数下尽管市场环境不同仍可取得稳健的超额收益。

关键词： 企业风险　SVM　选股策略　Z-Score

1　引言

1.1　选题动机

风险管理是现代金融框架中的重要组成部分，一学期丰富的课程下来我们得以大概领略下其全貌。但由于学识和理解尚浅论文选题并不容易，我总觉得要么十分成熟偏向被动消极防守，无论是 VaR，GARCH 等还是其变体；要么对金融、计量和现实的理解要求颇高，如后面的一系列高深的论文。不过最后我还是幸运地在刘老师课堂上看到了 Altman 的 Z 值模型，相比我国大而无当的银行我想我还是更愿关心股票市场，不仅样本量比较大而且上市公司的财务数据也可以很方便地获得。

选题的另一方面原因在于核心技术，决策之际恰好刚刚完成瞿慧老师《金融系统仿真》的大作业汇报。那是几个星期前出于听起来比较好玩选了一篇关于 SVR (Support Vector Machines for Regression) 的论文，虽然也属于支持向量机但实际用于做非线性回归，而在本次实验中恰好可用 SVM 来进行分类。我们最初的想法是用它来在 Altman 的 Z-Score 基础上做下改

进，毕竟那是 60 年代的论文而支持向量机是机器学习理论中较为年轻且效果不错的算法。

但最终我们还是受制于数据，学生版 Wind 提供的近三年中被标为 ST 的公司少之又少，根本无法用于训练。我们只好将研究从被动的风险识别结合一些其他指标转为主动的选股策略，虽然从未上过财务等相关课程但好在参加了一个花钱就行的野鸡实习并从中收获良多，在 10-11 月的投资组合构建中我把刘老师前前后后教给我的宏观和风管知识全用上啦，那边评价说有史最好哈哈哈，非常感谢您！同时在当时的股票选取策略中虽然我想到了机器学习的思想但是用 Excel 人工操作的，本次论文的 SVM 选股也算是完成一件当时的心愿吧。

1.2 文献综述

我们主要从指标和模型两方面展开企业财务危机预警的研究发展过程：

1.2.1 预警指标研究

1. 传统财务指标传统财务指标是指可直接从企业财务报表中获取的指标。Fitzpatrick (1932)、Altman (1968) 等采用常规的财务指标，如负债比率、流动比率、资产周转速度等作为预警指标。该方法因方便易得而被很多研究者采用。

2. 现金流量指标现金流量指标基于公司理财的基本原理：企业的价值应等于预期现金流量的净现值，当现金流出现问题时企业可能会陷入财务危机。在 Gentry et al.(1985) 研究的基础上 Aziz et al.(1988) 比较了 Z 模型、Zeta 模型和现金流量模型预测企业发生财务危机的准确率，发现现金流量模型的预测效果较好。

3. 市场收益指标市场收益指标的提出者发现股票收益率也如同财务指标一样可以预测破产，Altman & Brenner (1981) 的研究表明破产企业股票在破产前至少一年内在资本市场上表现欠佳。Aharony et al.(1980) 提出了一个基于市场收益率方差的破产预测模型，此外还有 Marais et al. (1984) 提出的基于市场信息及股价变动有关的预警模型。

1.2.2 预警方法研究

预警方法可分为定性方法和定性方法两大类，这里我们只介绍定性方法。定性方法又可以分为两大类，传统统计方法和人工智能专家系统方法：

传统统计方法

1. 单变量分析法单变量分析法核心是通过训练样本和测试样本寻找具有最高判别能力的单个指标。Fitzpatrick (1932) 以 19 家企业作为样本并划分为破产和非破产两组发现判别能力最高的是” 净利润/股东权益” 和“股东权益/负债”两个比率。

2. 多元线性判别分析 (MDA) 由于单个指标不能充分反映企业财务状况，学者们开始考虑结合多个比率进行分析。基于统计分析的 MDA 方法可从多个指标中筛选出提供较多信息的变量并建立判别函数，使该判别函数在对观测样本分类时错判率最小。美国学者 Altman (1968) 最早将 MDA 应用到财务危机预警研究中，通过对破产和非破产制造业企业的观察他对 22 个财务比率经过数理统计筛选得到五个变量建立了著名的 Z-Score 模型以及在此基础上改进的”Zeta” 模型 [2]。

3. 逻辑回归方法 (LR) 相比于前面提到的线性模型，基于极大似然估计的 LR 模型将财务危机转化为一个概率，且无需 MDA 中严格的假设。代表的又 Martin (1977) 和 Ohlson(1980) 的研究，此方法在 70 年代末依赖财务危机预警中应用较多。

人工智能专家系统方法 90 年代开始，随着计算机技术的发展人们陆续将人工智能领域的技术加入到企业财务风险识别中：

1. 人工神经网络 (ANN) 人工神经网络是一种平行分散处理模式，目前研究应用最为广泛的是前馈三层 BP 神经网络，它由输入层、隐藏层和输出层三层神经元组成。输入层用以接受网络的输入变量，与隐藏层的个神经元相互连接，中间层从输入层或其他隐藏层接收输入，并为输出层提供输入。当我们认为企业经营有正常或危机两种状态时财务危机预警相当于一个二类判别问题，就可以将 ANN 中输出神经元的输出值设正常企业为 0 或危机企业为 1。采用 ANN 方法构建的模型一般利用系统输入与输出所组成的数据建立系统模型，当神经网络接收到一组输入信息就会产生反应，然后与预期反应相比较。如果错

误率超过可接受范围就需要对权重 W_{ij} 做出修改或者增加隐藏层数目并开始新的学习过程，经过这样的反复循环直到错误率降低至可以接受的水平才会停止。训练阶段结束后 ANN 模型就可发挥预测功能了。Odom & Sharda (1990) 是将 ANN 方法应用于企业财务危机语境中最具代表性的学者之一。他们以 1975 到 1982 年间的 65 家失败企业于 64 家正常企业配对建立 ANN 预警模型并将预测效果和 Altman Z 值模型对比，发现 ANN 模型具有较佳的预测能力。

2. 遗传算法 (GA) 该方法是通过模仿生物遗传进化规律，在大量复杂概念空间内水机搜索的技术，适合用于服从大量软或硬约束的多参数优化问题，可用于企业破产预测，并居于财务比率值和定型变量进行 if then 判别规则提取，结构清楚容易理解。进行该方面研究的学者有：Varetto & Franco(1998) 等，其中 Franco 采用 GA 分别提取线性函数和判别规则，结果表明可以获得不受统计约束的最优线性方差，与 MDA 相比省时且受主观影响小，但预测结果不如 MDA 方法。

3. 支持向量机 (SVM) SVM 方法是一种基于统计学习理论的机器学习方法，该算法最早由 Vapnik 等人在 95 年提出。该算法将平面不可分的数据通过非线性变换映射到高维特征空间并通过核函数 (kernel trick) 简化计算，适合解决小样本、高维问题。同时和 ANN 不同，基于凸优化理论的 SVM 可以保证求导全局最优解而不会陷入局部最优。根据 CMU 计算机系教授 Eric Xing 所述 SVM 是现成的监督学习算法中效果最好的。[1] Fan & Palaniswami (2000) 首次建立基于 SVM 的财务危机预警模型，通过和其他方法的对比发现具有较好的分类效果。

2 SVM 准备

因为 SVM 理论对数学要求较高且不易理解，我们这里只从简单应用的角度谈谈 SVM 的直观理解和软件包的使用。

2.1 直观理解 SVM

在介绍 SVM 理论之前先简单说一下分类问题。分类 (Classification) 是机器学习和数据挖掘中最重要、最频繁使用的算法，其基本作用就是：从

[1] "Believed by many to be the best "off-the-shelf" supervised learning algorithm"

一组已经带有分类标记的训练样本数据集来预测一个测试样本的分类结果。生活中的分类问题有很多，比如给定身高和体重区分男生女生；给出一些葡萄酒的化学成分区分酒店品种；再比如情感分析中判断一句话的情感到底是积极消极还是中性，事实上给定足够的训练样本以上问题均可通过 SVM 实现。

了解了 SVM 要解决的分类问题是什么之后，我们从最简单的线性可分问题说起。图 1 中有两组待分类数据，直观来看想取得最好的分类效果就意味着我们需要找到一条直线让两组样本的间隔 (margin) 最大，也即分得最开。么我们假设这条直线的表达式是 $w^T x + b = 0$, 两类样本的标签分别是 $+1$ 和 -1，我们想让分类为 $+1$ 的样本都在直线 $w^T x + b = 1$ 之上，同理分类为 -1 的样本都在直线 $w^T x + b = -1$ 之下，处在直线上的点就是支持向量，支持向量机的核心平面只由支持向量决定，这也是其得名的原因。经过解析几何知识我们不难得到两条直线之间的距离是 $\frac{2}{||w||}$, 最大化间隔也即变成了以下约束：

$$\min \frac{1}{2}||w||^2, s.t.\ \ y_i(w^T x + b) \geqslant 1$$

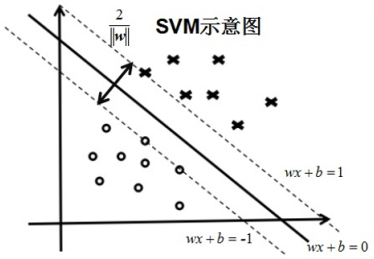

图 1: SVM 示意图

其最优求解可用凸优化理论中的拉格朗日对偶问题结合 KKT 条件或 SMO 算法得到，因为对数学要求比较高，所以我干脆也不费力去打自己都看不懂的数学公式了。值得注意的是因为是转化为二次优化求解，原问题维数不会影响算法复杂度，算法复杂度由支持向量个数决定，因此适合小样本、高维数据。

对于线性不可分的数据，支持向量机通过把它们映射到高维线性可分的空间，然后依然寻找超平面让两组样本之间间隔最大，但这样做的一大问题是高维会使计算复杂度升高导致维数灾难，但神奇的是可以通过核函数简化计算 (kernel trick)，从而大大降低计算复杂度。此外上述公式最简单的版

本，实际应用中可通过引入松弛变量允许存在适当的误差防止过度拟合，罚函数的存在使得控制误差大小成为可能。

为了保持内容完整性我们也顺带简要介绍下 SVR。SVR 本质上我们可以理解为非线性回归，依旧从最简单的线性情形开始，SVR 希望找到一条直线让样本点尽可能多地落在直线 $w^T x + b = 0$ 为中心宽度为 ε 的管道内。相比于最小二乘法点到直线的距离平方和最小化，SVR 试图最小化点到 ε 宽度管道的距离。对于非线性回归依然通过映射到高维并用核函数简化计算得到。对于简单情形的 ε–SVR 其数学表达式如下：

$$\min \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*), s.t. \begin{cases} y_i - w^T x_i - b \leqslant \varepsilon + \xi_i \\ w^T x_i + b - y \leqslant \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geqslant 0 \end{cases}$$

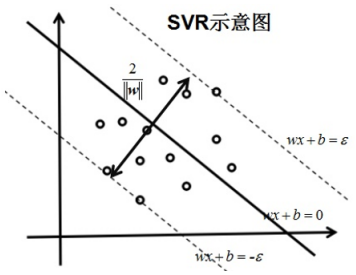

图 2: SVR 示意图

但在后面的实证研究中因为问题需要我们将收益率分为高于大盘和低于大盘两类所以只会使用到分类问题的 SVM。关于 SVM 更多具体证明请参见机器学习相关书籍 [3] 及课件。

2.2 SVM 软件包使用

总体说明 尽管 SVM 理论对于初学的我们十分艰深，看懂都费劲更别说编程实现了，幸运的是已有好心高手帮我们做好了方便使用的软件包。国立台湾大学的林智仁教授制作的 LibSVM 是目前最好的软件包之一，用 C++ 编写并支持 Matlab, Java, Python 等多个平台，相比 Matlab 自带的 SVM 程序支持的功能更 (完) 多 (爆)。在 Matlab 中的使用也十分简单，只需将软件包解压后在 Matlab 的 Set Path 中添加其目录下的 windows 文件夹即

可 (带有 64 位编译好的程序，若 32 位需自己编译)，要使用也很简单，基本只会用到以下三个主要函数：

```
[y,x]=libsvmread(filePath); …
    %读取数据，数据格式后面会有具体说明
model=svmtrain(y,x,'Parameters'); % …
    训练模型，参数后面会有具体说明
[py,acc,¬]=svmpredict(y,x,model); % …
    测试集检验，得到结果和精度，py为predicted …
    y,acc代表accuracy
```

注意第三步中的 svmpredict 函数中的 y 值并非用于预测，而是用于和预测值对比检验模型效果，当然此检验只对训练集起作用，对于未知分类的测试集可以让 y 全为某一数值，反正返回的精度也是无意义的。

LibSVM 数据格式 LibSVM 对数据输入格式有严格要求，其格式必须为：

$$\begin{array}{cccccc} y_1 & 1:x_{11} & 2:x_{12} & \dots & n:x_{1n} \\ y_2 & 1:x_{21} & 2:x_{22} & \dots & n:x_{2n} \end{array}$$

数据在 Matlab 中存储的格式也为稀疏矩阵格式，个人猜想这与 LibSVM 擅长解决高维问题有关，如维数可达成百上千维的文本情感分类，一个词对应一个维度，一句话中只占据稀疏的几个维度，这时 LibSVM 数据格式的优势就立刻显现出来。

LibSVM 参数说明 在 LibSVM 使用三部曲中最有技术含量的一步就是参数调优了，在参数调优之前首先要理解各参数含义：

- -s: SVM 类型，其中 0 为分类常用的 C-SVC，3 为回归常用的 ε-SVR
- -t: 核函数类型，0 为线性，1 为多项式，2 为较常用的径向基函数 (RBF)
- -c: 惩罚函数，越大则对误差要求越严格
- -g: 核函数系数，类似 $y = kx + b$ 中的 k, 核函数具体形式在自带文档中给出

- -p: ε-SVR 中的 ε 值，越大则允许的回归管道宽度越大
- -v: 交叉验证次数，随机分成 n 份交互检验分类/回归结果

其中后两项在本次实验中并未用上，更多详细说明请参见自带的文档。此外，LibSVM 设计十分科学，以上各项参数填错或不填均按默认值处理，因此不必担心崩溃。预测效果衡量上，回归返回均方误差 MSE，分类返回分类精度 Accuracy。

3 实证研究

3.1 研究概述

在这一部分，我们拟根据上市公司包括风险因素在内的诸多指标结合 SVM 技术选出具有正的超额收益的股票构建投资组合。我们暂且无根据粗浅地认为低于大盘收益率的股票可能存在两大方面原因，一方面是自身经营状况恶化陷入财务危机，这与风险指标相关；另一方面可能公司本身未出现严重的问题但市场存在错误估计了公司的价值，这些与估值指标相关。

因为高维数据不会增加 SVM 的计算复杂度或降低分类精度[2]，在风险指标选取上，我们除了选取了经典的 Altman Z-Score 中的五项指标外还纳入了文献综述中涉及到的关于偿债能力、现金流量、资产结构、波动率等指标；同时又考虑到股市涨跌幅和估值和盈利也密切相关，我们又引入了诸如 PE、PB、ROE 等诸多指标。

注意到我国股市不同市场间存在的股票类型及其带来的诸如估值等一系列差异，我们选取了沪市大盘股票并将其按照和沪市大盘收益率的关系分为 $+1$ 和 -1 两类样本，因为学生版 Wind 只有三年数据，我们初步计划采用 14 年数据作为训练集，15 和 16 年数据作为测试集。但实验中途发现未经过参数调优的效果极差，动辄 -20% 的收益率让人无法直视。后来我们考虑在 14 年数据做训练集基础上用 15 年数据进行参数调优，再用得到的模型去对 16 年的数据进行检测。结论发现，尽管 15 年和 16 年从至今收益率的视角看市场结构很不相同，但在 $c = 1, g = 7$ 的参数下均可取得显著稳健的超额收益，参数和收益率似乎存在着极强的相关性。

[2]我们注意到许多用到 SVM 的研究都引入许多个自变量，但实际上并不知道引入无关变量会不会降低 SVM 的分类精度

3.2 具体操作

数据获取及预处理

我们于 2017 年 12 月 22 日从 Wind 上获取了全体 A 股的 14、15 及 16 年的年报数据，对于估值指标对应为距当日一年、两年和三年的数据，包含估值、盈利、资本结构、偿债、营运、成长、Z 值等共 17 个指标。通过股票号选取沪市股票为研究对象，在 Excel 中根据到当日收益率于大盘到当然收益率关系，正的超额收益标为 $+1$，负的超额收益标为 -1。

考虑到指标种类繁多且范围混杂，市盈率的范围可以从负的几万到正的几万，而资本结构又都在 0-100 之间，直接放入 SVM 势必会影响分类效果。我们首先写了一个归一化函数，对于每个序列，选取原始序列的 5% 和 95% 分位数作为上下界，这样就很大程度减小了因为异常值的存在（比如一个负几万的 PE 导致所有的 PE 都是小数点后三位）的影响；同时我们对于最小值为负的指标将其线性映射到 $(-1, 1)$，对于恒为正的指标映射到 $(0, 1)$, 具体的 Matlab 代码如下：

```
1   %standardize.m
2   function …
        [std_seq]=standardize(ori_seq,lbound,ubound)
3   %该函数将原始序列ori_seq映射到区间(lbound,rbound)
4   %并返回标准化后的序列std_seq
5   std_seq=ori_seq;
6   left=quantile(ori_seq,0.05); …
        %选取5%分位数作为左边界
7   right=quantile(ori_seq,0.95); …
        %选取95%分位数作为左边界
8   for i=1:length(ori_seq);
9       k=ori_seq(i);
10      if(k<left)
11          std_seq(i)=lbound;
12      elseif (k>right)
13          std_seq(i)=ubound;
14      else
```

```
15              std_seq(i)=(k-left)*(ubound-lbound)/...
16              (right-left)+lbound;
17          end;
18      end;
```

读取数据并进行归一化处理代码如下：

```
1       A=xlsread('sh2014.xlsx');
2       B=zeros(size(A,1),size(A,2));
3       B(:,1)=A(:,1);
4       for i=2:size(A,2);  % …
            第一列是标签+1或-1无需标准化，从第二列开始
5           lbound=0; %默认左边界为0
6           if min(A(:,i))<0  …
                %如果最小值小于0则要求左边界为-1
7               lbound=-1;
8           end
9           B(:,i)=standardize(A(:,i),lbound,1); % …
                右边界恒为1
10      end
```

在用 Matlab 标准化并将结果保存为 txt 之后，我们用以下 Java 程序将其转为 LibSVM 要求的输入格式：

```
1       FileReader fr=new FileReader("f:/sh2014.txt");
2       BufferedReader br = new BufferedReader(fr);
3       FileWriter fw=new FileWriter("f:/sh14svm.txt");
4       BufferedWriter bw =new BufferedWriter(fw);
5       String s;
6       String[] str;
7       int i=0;
8       while ((s=br.readLine())!=null){
9           str=s.split(" ");
10          bw.flush();
```

```
11          StringBuilder newStr=new …
               StringBuilder(str[0]);
12          for(int j=1;j≤17;++j){
13               str[j]=" "+j+":"+str[j];
14               newStr.append(str[j]);
15          }
16          bw.write(newStr.toString());
17          bw.newLine();
18      }
19      fr.close();
20      fw.close();
```

参数调优

正如之前所说，参数调优是 SVM 初级使用中唯一有点技术含量的工作。我们最开始的思路是根据 14 年的训练数据用 Matlab Sky 论坛上 SVM 达人 Faruto 写好的包直接进行基于遗传算法 (GA) 的参数调优，但问题在于可以被调优的参数只有分类精度，而非收益率；另一个问题时遗传算法调优半天训练集的分类精度从 70% 升至 72%，在测试集的组合收益率上也没有明显的提升，保持在 −20% 左右，所以我们只能另想办法找出最优参数。

我们改进后的思路是在用 14 年数据做训练集的基础上，根据 15 年的数据反馈找到使组合收益率和选出股票数量都比较适中的 c, g 值，然后再用这个训练出来的模型检验最终效果。因为训练和调优来自两个数据集，所以这时那些编写好的包都派不上用场了。我们先手动输入 c 值和 g 值找到一个合适的范围，然后再编写程序进行指定范围内的参数寻优。选取的衡量指标是分类精度 (选出股票的数量) 和组合收益率，第一次纯天然手工操作得到的数据列表如表 1。

粗调确定大体范围之后，我们编写了一段 Matlab 程序将初步确定的大范围的 c, g 进行网格式搜索寻优，因为时间水平有限程序比较初级尚停留在半自动化阶段：

```
1       resultMatrix=zeros(100,4);
2       k=1;
```

表 1: 参数粗调

c	g	Accuracy(%)	Return(%)
10	0.8	23.55	-7.64
100	0.8	26.80	-8.51
1000	0.8	27.29	-9.47
10	10	4.73	10.92
100	10	3.35	5.46
1000	10	3.44	5.46

```
3
4       for c=1:100:1000; % c的网格
5           for g=10:10:100; % g的网格
6               cmd=['-c ',num2str(c),' -g …
                    ',num2str(g)]; % 注意空格
7               model=svmtrain(oy,ox,cmd);
8               [result,acc,¬]=svmpredict(ty,tx,model);
9               result(length(chg))=-1; % …
                    因为转格式程序有缺陷少一行
10              num=length(find(result==1));
11              result(find(result==-1))=0; % …
                    矩阵乘法将分类-1的股票标记为0
12              ret=result'*chg/num; % chg 是股价涨跌幅
13              resultMatrix(k,:)=[c g acc(1) ret];
14              k=k+1;
15          end
16      end
17      xlswrite('resultCG1.xlsx',resultMatrix); …
            %写入到Excel保存
```

利用以上网格更改参数范围后，在 $c:1-7, g:5-15$ 间寻优程序得到结果如表 2 所示。

如何定义最优呢？我们发现如果片面要求组合收益率最高便会导致选出的股票较少，反之要求选出股票数量充足就会导致等权重组合收益率的

表 2: 14 数据预测 15：参数半精调

c	g	Accuracy(%)	Return(%)	Acc×Ret
1	5	7.00	8.09	56.60
1	7	3.74	19.07	71.41
1	9	2.46	16.27	40.07
1	11	1.87	16.84	31.53
1	13	1.38	12.51	17.25
1	15	0.89	23.52	20.85
4	5	15.57	4.44	69.11
4	7	9.46	9.32	88.11
4	9	6.11	12.09	73.85
4	11	4.14	11.82	48.91
4	13	2.56	11.90	30.47
4	15	1.97	12.62	24.87
7	5	15.36	3.60	57.14
7	7	9.46	9.32	88.11
7	9	6.01	12.21	73.40
7	11	4.14	11.82	48.91
7	13	2.56	11.90	30.47
7	15	1.97	12.62	24.87

下降。折衷考虑我们采用选出股票数量占比 × 收益率作为衡量指标，发现较大的 c 可以选出更多股票但会降低组合收益率，g 相对较大时选出的股票数量较少但但具有较高的收益率。我们还发现参数 g 比 c 对组合收益率的影响更大，且最优 g 值在 5 至 9 之间，同时想选出最高收益的股票 c 值在 1 以内比较合适。表 3 是利用 14 年数据作为训练集、15 和 16 年数据分别作为测试集并按最终组合收益率降序排列的结果。

尽管从 15.12.22 至 17.12.22 上证综指收益率为 -8.93%，而从 16.12.22 到 17.12.22 上证综指收益率为 $+6.25\%$，两个时期的市场结构不尽相同，但从要求最高收益率组合的条件去看我们发现两组样本 c, g 参数存在高度的相关性，最优 g 似乎在 7 附近而最优 c 大概在 0.7 左右，同时我们发现小范围内的参数从中心到两端 (比如说 $g = 3$ 或 $g = 5$ 相对于 $g = 7$) 其收益

率呈现明显递减，这也从另一个角度表明了我们结论的可靠性。我想等 17 年报数据出来了一定要用以上模型和参数，选出股票之后再到模拟炒股软件中买一下股票持有一段看看效果。

表 3: 用 14 数据训练得到的最高收益率组合与 c, g 的关系

(a) 15 年结果，市场收益率-8.93%

c	g	Accuracy(%)	Return(%)
0.8	5	3.94	25.32
0.6	7	1.48	22.78
0.6	9	1.48	22.78
1	7	3.74	19.07
0.8	9	1.67	18.10
0.6	11	1.18	17.42
1	11	1.87	16.84
1	9	2.46	16.27
0.8	11	1.38	16.04
1.2	11	2.66	15.38

(b) 16 年结果，市场收益率 +6.25%

c	g	Accuracy(%)	Return(%)
0.6	9	0.37	53.32
0.6	11	0.18	47.17
0.6	7	1.01	40.01
0.8	11	0.64	33.50
0.8	9	0.82	33.29
1	11	0.82	26.62
0.6	5	2.10	20.39
0.8	7	1.92	18.44
0.8	5	3.66	15.27
0.6	3	4.48	13.94

4 结论

我们主要受 Altman (1968) 的启发并在其基础上试图结合所学的机器学习方法进行优化。但因为受数据所困中途不得不加以变通形成了选股策

略。时间精力原因最终只完成了上证市场 2014-2016 三年的基于 SVM 的选股策略，并发现以 14 数据做训练集 15、16 做训练集时所得最高收益率组合其最优参数间存在高度相关性。

我们的不足主要有以下几点：1) 数据方面，无论时间还是选取范围都过于小，三年上证数据具有偶然性；2) 研究方法方面，缺乏足够的对比，比如引入和未引入风险因素效果的对比；3) 编程方面，程序编写比较笨拙，对大量数据时很不方便；4) 核心理论方面，SVM 理论比较复杂，对其理解也只停留在表面应用层次。未来可能会在编程方面做些改进能让数据处理更自动化一些，届时将可能更方便地对各板块各行业进行类似的操作，使得研究结果具有较高的可信性。虽然存在诸多不足，我个人认为本次报告还是一次十分有趣的探索。

参考文献

[1] Edward I. Altman. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *Journal of Finance*, 23(4):589–609, 1968.

[2] Edward I. Altman, Robert G. Haldeman, and P. Narayanan. Zeta tm analysis a new model to identify bankruptcy risk of corporations. *Journal of Banking and Finance*, 1(1):29–54, 1977.

[3] Simon S Haykin. *Neural networks and learning machines*. China Machine Press, 2009.

[4] 彭静. 网络环境中企业财务危机预警研究. PhD thesis, 上海交通大学, 2008.

[5] 邹新月施锡铨. 典型判别分析在企业信用风险评估中的应用. 财经研究, 27(10):53–57, 2001.