华 东 师 范 大 学 软 件 学 院

2013 年软件工程学士学位论文

基于情感分析的金融走势选择性预测

Selective Prediction of Financial Trends Based on Sentiment Analysis

姓　　名：黄一夫

学　　号：10092510437

班　　级：2009 级 4 班

指导教师姓名：钱卫宁

指导教师职称：教授

2013 年 5 月

目　录

摘 要

一直以来，金融走势预测在学术界和工业界备受关注。金融走势预测的主要目的是，通过建立预测模型，分析金融数据，来预测金融涨跌的宏观走向。目前的研究主要集中在两个方面：数据的选取与模型的选取。数据的选取方面，有仅使用历史金融数据，还有结合历史金融数据和其他指标数据；模型的选取方面，有线性模型，非线性模型和随机游走模型等。

本文基于情感分析选择性地预测金融走势。首先使用经 WordNet 扩展后的 POMS Biploar 情感词列表对 Twitter 数据进行情感分析，提取出六维情感序列，分别为冷静-焦虑，同意-敌对，欢乐-失望，自信-怀疑，活力-疲劳和清醒-迷惑。然后分别使用六种情感序列与 DJIA 走势序列做格兰因果关系测试，得出延后期为 3 天的同意-敌对情感序列对 DJIA 走势序列具有最强的预测能力。接着实现多流选择性隐马尔可夫模型，结合同意-敌对情感序列和 DJIA 走势序列，对 DJIA 的未来走势作出预测，发现同意-敌对情感序列的确能提升预测精度，选择性隐马尔可夫模型的预测准确性和可控性优于传统的线性回归模型。最后使用模型的预测结果作出模拟投资，并对投资回报进行对比分析。

关键词：情感分析，走势预测，多流，选择性预测，隐马尔可夫模型

Abstract

Financial trends prediction attracts much attention from both academe and industry all the time. The goal of financial trends prediction is building prediction model and analyzing financial data to predict the ups and downs of financial trends. Currently researchers focus on two aspects: data selection and model selection. From the aspect of data selection, some use only historical financial data and some combine historical financial data with other indictors. From the aspect of model selection, some use linear model, some use nonlinear model and some use random walk model, etc.

This paper selectively predicts financial trends based on sentiment analysis. First I use the POMS Biploar List extended by WordNet to analyze the sentiment of tweets and get six dimension sentiments including composed-anxious, agreeable-hostile, elated-depressed, confident-unsure, energetic-tired and clearheaded-confused. Second I perform Granger Causality Analysis between each of six sentiments and DJIA trend, and find lagged 3 days agreeable-hostile sentiment has most predictive power to DJIA trend. Then I implement multi-stream selective hidden markov model to predict DJIA trend based on both historical DJIA trend and agreeable-hostile sentiment, and find agreeable-hostile sentiment can indeed improve prediction accuracy and selective hidden markov model works better than linear regression model. Finally I use the models to simulate the stock investment and analyze the returns on investment.

Keywords: sentiment analysis, trends prediction, multi-stream, selective prediction, hidden markov model

一、 绪论

(一)研究背景

随着互联网和金融领域的飞速发展，每时每刻都会有大量的金融数据产生。一般来说，金融数据是指金融时间序列，即股票价格数据，期货交易数据等等，它们是资产价值随着时间演变产生的随机变量，时间间隔可以是分秒，可以是日、周、月、季度、年、甚至更大的时间单位。基于金融数据，有许多分析与挖掘的应用，其中金融走势预测，例如股票走势预测，汇率走势预测等，一直以来备受关注，因为其直接影响着投资者的利益。对金融走势进行预测分析，可以探索金融走势背后的原因，使我们对金融市场有更加深入的理解。当达到一定的性能指标时，可推出作为商用，在宏观上提升投资者的效益。

金融走势预测的主要目的是，通过建立预测模型，分析金融数据，来预测金融涨跌的宏观走向。目前的研究主要集中在两个方面：数据的选取与模型的选取。数据的选取方面，有仅使用历史金融数据，还有结合历史金融数据和其他指标数据，例如调查问卷，搜索引擎数据，Twitter 情感数据等。模型的选取方面，有线性的回归模型，非线性的神经网络模型，随机游走的隐马尔可夫模型等。

早期的研究主要基于效率市场假说[1]。根据效率市场假说，资产价值在很大程度上取决于新的信息，而新的信息又是无法预测的，所以金融走势无法得到很好的预测。而目前行为金融学[2]迅速发展起来。行为金融学表明，微观上个人情感能显著地影响个人行为及决策，宏观上社会群体的情感状态会影响着集体的决策，进而发现公众情绪与经济指标相关并对其拥有一定的预测能力。度量公众情绪的方法很多：采用调查问卷，随机取样个人情绪来代表公众情绪；采用搜索引擎搜索日志，汇总相关的搜索量来代表公众情绪；采用在线社交媒体，通过情感分析的方法来获得公众情绪。研究表明[3]，基于在线社交媒体 Twitter 的情感分析能够更好地对公众情绪进行建模，进而预测金融市场。

线性回归模型是最简单且最常用的预测模型，其原理为求解一次多元方程组来估计线性方程里的各个参数，然后再用得到的线性方程来预测。经典的有自回归模型，滑动平均模型，以及两者相结合的自回归滑动平均模型等[4]。但普遍认为，金融走势是非线性的，因此越来越多较为复杂的方法如神经网络，支持向量机，随机游走等被运用到其中。其中基于随机游走的隐马尔可夫模型备受关注，因为其不仅拥有良好的概率统计基础，而且结构容易改造以达到特定的目的[31][32][33][34]。

(二)相关工作

目前金融走势预测的研究主要集中在两个方面：数据的选取与模型的选取。

数据的选取方面，一部分研究仅仅借助历史金融数据来预测金融走势。例如 Dmitry 等人[5]使用历史 S&P500 走势来预测未来 S&P500 走势，Manuele 等人[8]尽力基于历史股票数据对单支股票走势进行预测，龚健等人[6]利用历史上证指数来建立上证指数的预测模型。另一部分研究综合考虑了历史金融数据和其他指标数据。例如 Bollen 等人[7]结合历史 DJIA 走势和 Twitter 情感序列来共同预测未来 DJIA 走势，Yang 等人[9]加入博客情感数据来预测电影票房，Eric 等人[10]从博客中提取焦虑情感索引来预测股票市场，更多利用情感数据的研究还有[23][24][25][26]。另外，除了情感数据之外，stockcharts 提供很多与股票相关的其他指标数据。

模型的选取方面，一部分研究基于简单的线性回归模型。Ruey[4]介绍了经典的自回归模型，滑动平均模型，以及两者的结合，Yang 等人[9]基于话题模型提出了情感相关的自回归模型，Hyunyoung[12]介绍了 Google Trends 预测模型，为对数项的线性回归。另一部分研究集中于较复杂的非线性的模型。例如 Gang 等人[11]提出了一种建立自组织模糊神经网络的在线算法，Dmitry 等人[5]引入了一种选择性的隐马尔可夫预测模型，Satish 等人[13]介绍了隐马尔可夫模型和支持向量机在金融预测中的应用。

同时，业界也不断涌出金融预测相关的公司和产品。Derwent Capital Markets 是一家以社交媒体情感分析为基础的金融衍生品交易公司。公司声称在 2012 年 3 月建立起了世界上第一个基于社交媒体情感分析的交易平台，其研究基础正是 Bollen 等人的工作。股票雷达是一款移动端的专业股票行情分析软件。通过实时扫描各大股吧、财经微博、名家博客抓取业内专家、民间高手、普通股民的投资观点，结合券商机构的研究报告、行业权威新闻进行智能整合准确分析出股票的涨跌。中科精诚发布投资舆情指数暨趋势预测。该指数运用中科院计算所的语义分析技术，实时扫描计算网络海量投资相关信息，模拟及预测金融市场投资人的交易行为，为投资者提供参考指标。另外，国外的 Stocktwits 和 Piqqem 是著名的股票讨论平台，为投资者提供群体智慧来支持决策。

(三)论文工作

通过对现有方法的分析比较，本文在数据选取方面，结合历史金融走势和大众情感数据，在模型选择方面，采用具有选择性的隐马尔可夫预测模型。针对同时要

处理金融序列和情感序列，本文提出一种多流选择性隐马尔可夫模型，并配合模型修改相应的参数训练，状态标记和预测，最后实现了基于情感分析的金融走势选择性预测系统。本文以 DJIA 数据和 Twitter 数据为例，通过对比试验，说明了利用新模型进行 DJIA 走势预测，有更高的准确性和可控性，最后使用预测结果模拟了股票投资，并对收益情况作出了分析。

本文的主要内容如下：

数据获取模块。金融数据来源于 Yahoo!Finance 上的 DJIA 数据，情感数据来源于 Twitter 数据的情感分析。实现了金融数据获取接口，通过调用 Yahoo!Finance 提供的 API 获得 DJIA 数据，然后对获取到的 DJIA 数据进行预处理，得到 DJIA 走势序列；实现了 Twitter 数据获取接口，通过调用 Twitter 提供的 API 获得 Twitter 数据，然后对获取到的 Twitter 数据进行情感分析，得到 Twitter 情感序列。由于时间和硬件等限制，本文中使用的 Twitter 数据为 SNAP 的 Twitter 数据集[14]。经试验得出延后期为 3 天的 Twitter 同意-敌对情感序列对 DJIA 走势序列具有最强的预测能力。

预测模块。本文实现了多流选择性隐马尔可夫模型，主要在于修改后的隐马尔可夫模型的参数训练，状态标注和预测的设计和实现。本文在选择性隐马尔可夫模型上引入多流的概念，降低了模型预测的风险率，并给出了在数据充足情况下的其他实现方式。经试验得出选择性隐马尔可夫模型的预测效果优于传统线性回归模型，多流选择性隐马尔可夫模型的预测效果优于经典的选择性隐马尔可夫模型。在得到预测结果之后，本文模拟了在相同的起始资金下，使用不同的预测模型进行股票投资，得出各种预测模型的收益，并对投资回报作出了分析。

(四)论文组织

本文主要研究了基于情感分析的金融走势选择性预测，在对数据进行预处理和情感分析之后，对经典的隐马尔可夫模型也进行了相应的改进，并实现了相关的算法，然后用新的模型进行预测，得到了更低的风险率，最后根据预测结果，模拟了股票投资，并对投资回报作出了分析。

本文的结构组织如下：

第一章，绪论。首先介绍了金融走势预测的研究背景和意义，然后简要介绍了金融走势预测的相关工作，接着介绍了本文所做的工作，最后概述了本文的结构组织，在整体上搭建了全文框架。

第二章，金融走势预测方法概述。首先介绍了金融走势预测方法在数据选择上的分类，接着详细介绍了基于历史金融数据的走势预测和结合其他指标数据的走势预测，最后详细介绍了基于隐马尔可夫模型的走势预测，为下文的模型改进进行了铺垫。

第三章，多流选择性隐马尔可夫模型。首先给出多流选择性隐马尔可夫模型的形式化定义，然后对参数训练，状态标注和预测等算法进行详细描述和实现，本章节是全文重点。

第四章，系统实现和实验结果。首先介绍金融走势选择性预测系统的框架设计，接着分别介绍数据获取模块和预测模块的设计与实现，最后对实验结果进行了详细地描述和对比分析，进行讨论并得出结论。

第五章，总结和展望。总结全文，分析讨论并得出结论，并对论文中不足的地方进行叙述，由此作出未来展望。

二、 金融走势预测方法概述

本章节对金融走势预测方法进行概述，主要介绍数据选取方面金融走势预测方法的分类。在介绍基于历史金融数据的走势预测时，着重介绍选择性预测的概念以及相关的评价指标。在介绍结合其他指标数据的走势预测时，着重介绍情感分析的概念以及相关的评价指标。最后介绍了隐马尔可夫模型的概念，以及隐马尔可夫模型在走势预测上的应用，为下文的模型改进提供足够的先验知识。

(一)金融预测方法的分类

金融预测方法分为两种：走势预测和价格预测。走势预测指的是对未来的涨跌进行预测，价格预测指的是对未来的价格进行预测。

在机器学习领域里，偏向于对金融数据做走势预测，这样可以把预测问题转化为目前已经研究得较为深入的分类问题上来，可以利用许多基础的和改进的分类模型进行走势预测。在金融分析领域，偏向于对金融数据做价格预测，他们认为连续的价格才能完整地刻画金融数据的本质特性，离散的走势丢弃掉了一些有用的信息。

本文采用第一种预测方法，因为风险投资是由走势而非价格决定，并且其可以引入选择性预测的概念，这样可以在预测的覆盖率和风险率的折衷上有很好的控制，而这种可控性在风险投资中是很重要的。

金融走势预测方法按数据的选取可以分为两类：仅历史金融数据，结合历史金融数据和其他指标数据。

一部分研究仅仅借助历史金融数据来预测金融走势。例如 Dmitry 等人[5]使用历史 S&P500 走势来预测未来走势，他们通过引入选择性预测的概念，得到可控性高的预测结果。Manuele 等人[8]尽力基于历史数据对单支股票走势进行预测，他们尝试了歧义准则，获得了较好的预测结果。龚健等人[6]利用历史上证指数来建立上证指数的预测模型，通过 BIC 准则的使用，优化了模型参数，也得到了较好的预测结果。

另一部分研究综合考虑了历史金融数据和其他指标数据。例如 Bollen 等人[7]结合历史 DJIA 走势和 Twitter 情感序列来共同预测未来 DJIA 走势，他们基于 Twitter 冷静情感数据，采用自组织模糊神经网络模型，获得了很低的预测风险率。Yang 等人[9]在普通的自回归模型中加入了微博情感语义，将普通的词袋模型降维到预定义的情感词空间上，然后对电影票房进行预测。Eric 等人[10]从博客中提取焦虑情感索引来预测股票市场，通过一系列的格兰杰因果关系测试，发现了焦虑情感索引确实能提高股票预测的效果。

(二)基于历史金融数据的走势预测

1. 简介

基于历史金融数据的走势预测，指的是对一条金融时间序列$P = \{p_1, p_2, \dots, p_n\}$，在$t$时刻，利用目前已知的一个或者多个变量，求出该金融时间序列$t+1$时刻相对于t时刻的涨跌Y_{t+1}：

$$Y_{t+1} = sign(p_{t+1} - p_t)$$

由于金融数据的波动性，上式为 0 的情况极少，故其实质上近似等于机器学习中类标个数为 2 的监督学习，可以形式化地定义如下：

$$Y_{t+1} = F(X_t)$$

其中X_t为输入向量，代表t时刻下已知的一个或者多个变量；Y_{t+1}为输出结果，值域为$\{1, -1\}$，1 代表涨，-1 代表跌；F为预测模型，可以通过使用历史金融数据，训练得到预测模型的参数，进而确定F。

对于标准的走势预测，评价指标为准确率P：

$$P = \frac{C}{U}$$

C为预测结果集中正确的个数，U为预测结果集的大小。

2. 选择性预测

选择性预测[15]，指的是一种走势预测模型的框架。它可以评估自己预测结果，当预测结果可信度没有达到一定阈值的时候，拒绝输出预测结果。可以形式化地定义如下：

$$Y_{t+1} = \begin{cases} F(X_t), & if\ G(X_t) = 1 \\ reject, & if\ G(X_t) = 0 \end{cases}$$

其中G为预测自信度，值域为$\{0,1\}$，值为 1 时表示$F(X_t)$的预测结果是可接受的，值为 0 时表示$F(X_t)$的预测结果是不可接受的。由此，可导出两种特殊情况：当$G(X_t) \equiv 1$时，没有预测结果不可接受，是标准的走势预测方法；当$G(X_t) \equiv 0$时，所有的预测结果不可接受，此时预测模型没有输出。

由于引进了选择性预测的概念，评价指标变为覆盖率C和风险率R。

覆盖率C：

$$C = \frac{A}{U}$$

A是预测结果集中可接受的个数，U是预测结果集的大小。

风险率R：

$$R = \frac{F}{A}$$

F是预测结果集中可接受中错误的个数，A是预测结果集中可接受的个数。

由此，可以绘制风险率-覆盖率曲线，目的是获得拥有风险率足够低并且覆盖率足够高的曲线。

在评估多个选择性预测模型时，可以固定其中一个评价指标，以获得更加优化的另外一个评价指标。Pietraszek[16]将上述选择性预测的评估模型进行了规范的定义：有界弃权模型，给定覆盖率，学习器应该输出在该覆盖率下风险率最小的选择性预测模型；有界提升模型，给定风险率，学习器应该输出在该风险率下覆盖率最大的选择性预测模型。

如图 2－1 展示了风险率-覆盖率曲线，我们的目的是获得更偏于下方，即更光滑的曲线，因为其满足上述有界弃权模型和有界提升模型。

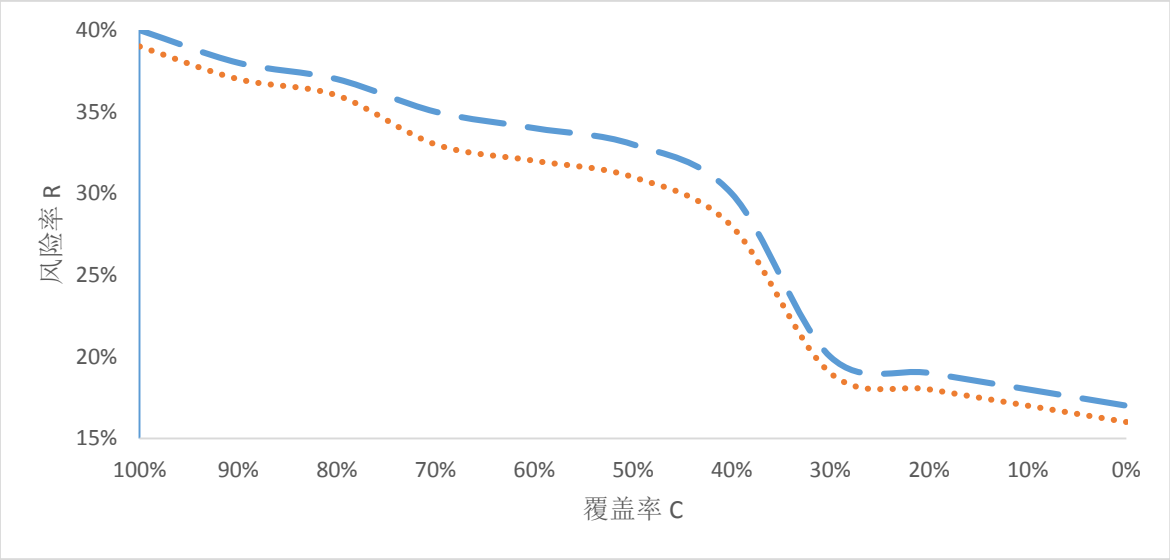

图 2－1 风险率-覆盖率曲线

Figure 2-1 Risk-Coverage Curve

(三)结合其他指标数据的走势预测

1. 简介

结合其他指标数据的走势预测，指的是不仅基于历史金融数据，而且结合其他相关的指标数据对金融走势进行预测。行为金融学表明，微观上个人情感能显著地影响个人行为及决策，宏观上社会群体的情感状态会影响着集体的决策，进而发现公众情绪与经济指标相关并对其拥有一定的预测能力。度量公众情绪的方法很多：采用调查问卷，随机取样个人情绪来代表公众情绪；采用搜索引擎搜索日志，汇总相

关的搜索量来代表公众情绪；采用在线社交媒体，通过情感分析的方法来获得公众情绪。研究表明[3]，基于在线社交媒体 Twitter 的情感分析能够更好地对公众情绪进行建模，进而预测金融市场。

2. 情感分析

情感分析，又称意见挖掘，是一种自然语言处理应用，旨在从文本中提取主观信息[22]，即作者的态度。这些态度可能表示了作者的判断或评估，情感状态，或者是作者希望读者产生的情感。情感分析的一个基本工作是对文本的极性进行分析，极性可以是单维的，也可以是多维的。一般来说，每个极性都有一个预先定义好的极性词表，极性则是统计文本中极性词的词频，同时也可能考虑权重，再按所需的粒度进行聚集，得到的情感序列。目前，已有许多基于博客进行情感分析的研究工作[20][21][27][29]。

按输出结果可以将情感分析分为两种：单维情感分析和多维情感分析。

单维情感分析指的把每个文档按积极和消极两个极性来分，因为这两个极性是互补的，所以其实质上是单维情感分析。OpinionFinder[17]提供了单维情感分析的 API，Alex Davies 也提供一个专门针对 Twitter 的单维情感分析词表。

多维情感分析认为人的情感应该是丰富的，所以把每个文档按多种极性来分，比如冷静，同意，生气等等。POMS[18][30]是心理学家设计来度量心情的调查问卷，共包含三个版本，分别为 POMS Standard, POMS Brief 和 POMS Bipolar。其中 POMS Bipolar 版本包含属于 6 种极性的一共 72 个形容词，6 种极性包括冷静-焦虑，同意-敌对，欢乐-失望，自信-怀疑，活力-疲劳和清醒-迷惑，每种极性包括 12 个形容词，这些形容词有的增强正极性，有的增强负极性。

鉴于 POMS Bipolar 情感词的数量较小，在实际运用中可能无法充分地捕获文档中的情感，因此需要通过一定的方法对 POMS Bipolar 情感词表进行扩充。本文通过使用 WordNet 进行同义词扩充。WordNet 是一个英语字典，其根据词条的意义将它们分组，每一个具有相同意义的词条组称为一个 synset，即同义词集合。在扩充同时将原 POMS Bipolar 情感词的正负性和极性继承到扩充后的同义词集合上，扩充后的情感词表一共有 638 个词。经 WordNet 扩充后的 POMS Bipolar 情感词请参照附录，第一个值代表情感词，第二个值代表该情感词极性的正负，1 为正，0 为负，第三个值代表情感词极性的类型，A 代表冷静-焦虑，B 代表同意-敌对，C 代表欢乐-失望，D 代表自信-怀疑，E 代表活力-疲劳，F 代表清醒-迷惑。另外，情感词典的扩充也可以利用 Google N-gram 等其他方法进行实现。

通过情感分析，可以得到单维和多维的情感序列，需要对这些情感序列进行评价，以确定其是否适用于预测金融走势。一般地，可以使用大事记来检验情感序列，观察相应情感序列在大事记时间点是否有正确的起伏。本文从统计的角度，对情感序列和金融走势序列做格兰杰因果关系测试，用以确定情感序列是否能用于预测金融走势序列。该测试通过寻找小于 0.1 或小于 0.05 的p值，来获得具有显著预测能力的情感序列，以及具体的延后项。另外也可以通过计算两条序列的皮尔逊χ^2统计量等方法来验证两者的相关性。

(四)基于隐马尔可夫模型的走势预测

隐马尔可夫模型是一个统计学习模型，描述了由隐藏的马尔可夫链随机转移生成可见的观测序列的过程，属于生成模型。它在语音识别，信息抽取，走势预测等领域有着广泛的应用。对于走势预测，隐马尔可夫模型体现出独有的实用性和易改造性。高度的表达性和良好的效率，使得隐马尔可夫模型在走势预测领域备受关注。

1. 隐马尔可夫模型概念

（1）隐马尔可夫模型定义

隐马尔可夫模型可以形式化定义为一个五元组$\lambda = \{N, M, \pi, A, B\}$：

N为状态的个数，状态集合为$I = \{i_1, i_2, \dots, i_N\}$；

M为观察值的个数，观察值集合为$V = \{v_1, v_2, \dots, v_M\}$；

$\pi = \{\pi_1, \pi_2, \dots, \pi_N\}$，状态起始概率的集合，$\sum_{i=1}^{N} \pi_i = 1$；

$A = \{a_{ij} | i, j = 1,2, \dots, N\}$，状态转移概率，$\sum_{i=1}^{N} a_{ij} = 1$，$\sum_{j=1}^{N} a_{ij} = 1$；

$B = \{b_{ij} | i = 1,2, \dots, N, j = 1,2, \dots, M\}$，观察值概率分布，$\sum_{j=1}^{M} b_{ij} = 1$。

图 2－2 是一个隐马尔可夫模型的例子，其$N = 2, M = 2, \pi = \{0.94,0.06\}, A = \{0.95,0.05,0.11,0.89\}, B = \{0.948,0.052,0.192,0.808\}$。

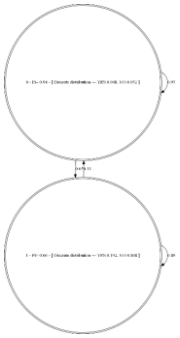

图 2－2 隐马尔可夫模型示例

Figure 2-2 Hidden Markov Model Example

（2）隐马尔可夫模型过程

隐马尔可夫模型生成长度大于 1 的观察序列的过程是：

①按模型中状态起始概率π随机选择一个状态，

②按该状态下的观察值概率分布B随机输出一个观察值，

③按从该状态的状态转移概率A随机转移到下一个状态，

④按该状态下的观察值概率分布B随机输出一个观察值，

⑤如果还需要输出观察值，则返回③，否则终止。

其中，每个状态下的观察值概率分布可以是离散的，也可以是连续的。

图 2－3 为隐马尔可夫模型生成观察序列的过程，模型内部进行状态的转移，即序列x，这是不可见的，模型外部输出观察值的序列，即序列y，这是可见的。

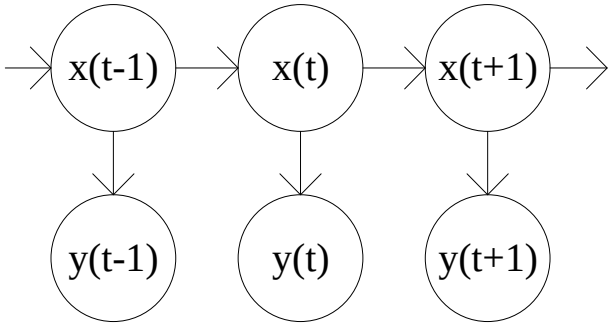

图 2－3 隐马尔可夫模型过程

Figure 2-3 Hidden Markov Model Process

（3）隐马尔可夫模型问题

隐马尔可夫模型有三个经典问题：

问题一，给定一个观察序列$O=\{o_1,o_2,\dots,o_T\}$和隐马尔可夫模型λ，如何计算观察序列O由模型λ产生的概率$P(O|\lambda)$。

问题二，给定一个可见的观察序列$O=\{o_1,o_2,\dots,o_T\}$和隐马尔可夫模型λ，如何求出一个不可见的状态序列$I=\{i_1,i_2,\dots,i_T\}$，使得I能最合理地解释O。

问题三，给定一个观察序列$O=\{o_1,o_2,\dots,o_T\}$，如何训练模型λ的参数，即π,A,B，使得$P(O|\lambda)$最大。

隐马尔可夫模型的三个经典问题都有相应的解决方案：

1）问题一，概率计算问题

朴素地，可以列举出所有的起始状态，观察值输出，和状态转移的情况，然后将所有的情况进行汇总。令状态序列为$I=\{i_1,i_2,\dots,i_T\}$，则其在模型λ下出现的概率为：

$$P(I|\lambda) = \pi_{i_1} a_{i_1 i_2} a_{i_2 i_3} \dots a_{i_{T-1} i_T}$$

此时观察序列为$O = \{o_1, o_2, \dots, o_T\}$的概率为：

$$P(O|I, \lambda) = b_{i_1 o_1} b_{i_2 o_2} \dots b_{i_T o_T}$$

则O和I同时出现的概率为：

$$P(O, I|\lambda) = \pi_{i_1} b_{i_1 o_1} a_{i_1 i_2} b_{i_2 o_2} \dots a_{i_{T-1} i_T} b_{i_T o_T}$$

然后求和所有可能的状态序列I，就能得到：

$$P(O|\lambda) = \sum_{I} P(O, I|\lambda) = \sum_{i_1, i_2, \dots, i_T} \pi_{i_1} b_{i_1 o_1} a_{i_1 i_2} b_{i_2 o_2} \dots a_{i_{T-1} i_T} b_{i_T o_T}$$

此时的算法复杂度为$O(TN^T)$，是指数级的复杂度，因此需要更有效的算法来解决这个问题。

前向后向算法[19]是公认的高效率算法，其通过保存中间计算结果来提高效率，是典型的使用空间换取时间的策略。

定义前向概率为：

$$\alpha_{ti} = P(o_1, o_2, \dots, o_t, i_t = i_i|\lambda)$$

可以递推地求得α_{ti}和$P(O|\lambda)$。

初值，

$$\alpha_{1i} = \pi_i b_{io_1}, i = 1,2, \dots, N$$

递推，

$$\alpha_{t+1,i} = \left[\sum_{j=1}^{N} \alpha_{tj} a_{ji}\right] b_{io_{t+1}}, i = 1,2, \dots, N, t = 1,2, \dots, T-1$$

终止，

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_{Ti}$$

前向算法递推中的计算过程如图 2—4 所示，

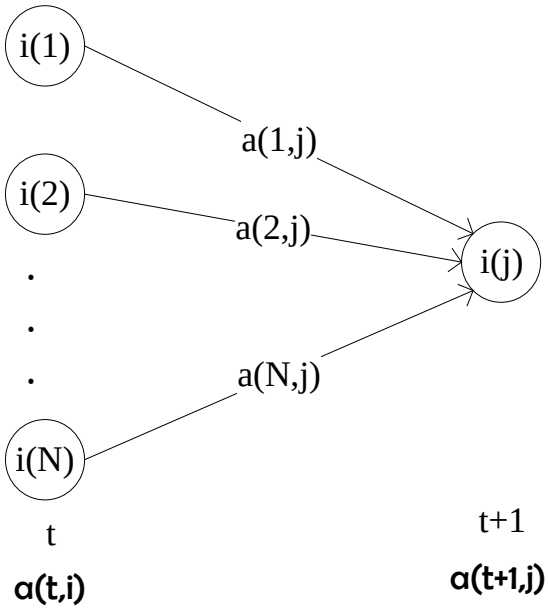

图 2－4 前向算法递推关系

Figure 2-4 Forward Algorithm Recurrence Relation

定义后向概率为：

$$\beta_{ti} = P(o_{t+1}, o_{t+2}, \dots, o_T | i_t = i_i, \lambda)$$

可以递推地求得β_{ti}和$P(O|\lambda)$。

初值，

$$\beta_{Ti} = 1, i = 1,2, \dots, N$$

递推，

$$\beta_{ti} = \sum_{j=1}^{N} a_{ij} b_{jo_{t+1}} \beta_{t+1,j}, i = 1,2, \dots, N, t = T-1, T-2, \dots, 1$$

终止，

$$P(O|\lambda) = \sum_{i=1}^{N} \pi_i b_{io_1} \beta_{1i}$$

后向算法递推中的计算过程如图 2－5 所示，

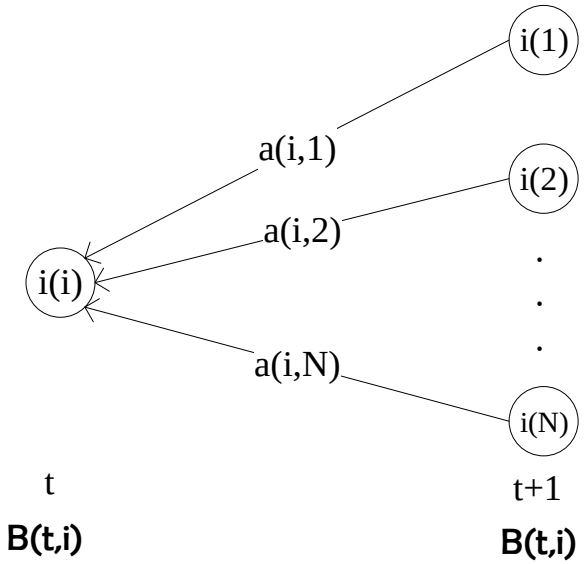

图 2－5 后向算法递推关系

Figure 2-5 Backward Algorithm Recurrence Relation

利用前向概率和后向概率可得

$$P(O|\lambda) = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{ti} a_{ij}\, b_{jo_{t+1}} \beta_{t+1,j}\,, t = 1,2,,\dots,T-1$$

此时的算法复杂度为$O(N^2T)$。

2）问题二，标注问题

近似地，有如下贪婪算法。定义变量γ_{ti}，表示隐马尔可夫模型λ在t时刻，处于状态i_i的概率。γ_{ti}可以通过问题一中的前向后向算法有效地计算出：

$$\gamma_{ti} = \frac{\alpha_{ti}\beta_{ti}}{P(O|\lambda)} = \frac{\alpha_{ti}\beta_{ti}}{\sum_{j=1}^{N}\alpha_{tj}\beta_{tj}}$$

然后在时刻t，对所有的状态计算出γ_{ti}，选择其中最大者作为模型λ在t时刻下最可能的状态，即

$$i_t^* = argmax(\gamma_{ti}), i = 1,2,\dots,N$$

从而得到观察序列O在模型λ下最可能的状态序列$I^* = \{i_1^*, i_2^*, \dots, i_T^*\}$。但这样的近似计算有一定的问题，因为预测的状态序列可能存在实际不可能发生的情况。

一般采用维特比算法进行计算。维比特算法实质上是通过动态规划算法来求概率最大路径，需要引入两个变量δ_{ti}和ψ_{ti}：

$$\delta_{ti} = max_{i_1,i_2,\dots,i_{t-1}}\, P(i_1,\dots,i_t = i, o_1,\dots,o_t|\lambda), i = 1,2,\dots,N$$

其代表时刻t状态为i_i所有单个路径$(i_1, i_2, \dots, i_t)$中概率最大值

$$\psi_{ti} = argmax_{j=1,2,\dots,N}\delta_{t-1,i}a_{ji}, i = 1,2,\dots,N$$

其代表时刻t状态为i_i的所有单个路径$(i_1, i_2, \dots, i_{t-1}, i)$中概率最大的路径的第$t-1$个节点。

然后递推并且回溯求出概率最大路径。由于基于隐马尔可夫模型的走势预测中没有使用该算法，此处省略。

3）问题三，模型训练问题

这个问题一直是其中难度最大的一个，目前还没有算法可以求得该问题的最优解。一般地，Baum-Welch 迭代算法使用得较多，其是一种梯度下降的优化技术，是 EM 算法的具体实现。引入变量$\xi_{t,i,j}$，其代表模型λ在t时刻，从状态i_i转移到状态i_j的概率

$$\xi_{t,i,j} = \frac{\alpha_{ti}a_{ij}b_{jO_{t+1}}\beta_{t+1,j}}{P(O|\lambda)} = \frac{\alpha_{ti}a_{ij}b_{jO_{t+1}}\beta_{t+1,j}}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_{ti}a_{ij}b_{jO_{t+1}}\beta_{t+1,j}}$$

初始化，对$n = 0$，随机初始化模型的参数，即$a_{ij}^{(0)}, b_{jk}^{(0)}, \pi_i^{(0)}$，然后计算$\gamma_{ti}$和$\xi_{t,i,j}$，

递推，对$n = 1,2,\dots$，首先估计模型参数，

$$a_{ij}^{(n)} = \frac{\sum_{t=1}^{T-1}\xi_{t,i,j}}{\sum_{t=1}^{T-1}\gamma_{ti}}$$

$$b_{jk}^{(n)} = \frac{\sum_{t=1,o_t=v_k}^{T}\gamma_{tj}}{\sum_{t=1}^{T}\gamma_{tj}}$$

$$\pi_i^{(n)} = \gamma_{1i}$$

然后计算γ_{ti}和$\xi_{t,i,j}$，

终止，再进行一次参数估计，得到最后的模型λ的参数$a_{ij}^{(n+1)}, b_{jk}^{(n+1)}, \pi_i^{(n+1)}$。

2. 隐马尔可夫模型在走势预测上的应用

以上三个问题实质上是结合在一起的，这里讨论一下隐马尔可夫模型在走势预测上的应用。

首先将走势序列进行预编码，得到观察序列，然后对观察序列求得对应的标签序列。例如，令走势序列为$\{-1,-1,1,1,1,-1,1,1,1,1,-1\}$，其中-1 代表跌，1 代表涨，按窗口长度 3 进行编码，窗口长度与预测延后项有关，可以得编码后的观察序列为$\{1,3,7,6,5,3,7,7\}$，对应的标签序列为$\{1,1,-1,1,1,1,1,-1\}$，标签序列中的值代表在对应

的观察序列的值下第二天的涨跌情况。

然后随机初始化隐马尔可夫模型，使用观察序列对模型的参数进行训练，同时使用标签序列对模型的状态进行标注。例如，随机初始化一个隐马尔可夫模型λ，状态数为 5，观察值的个数为 8，基于观察序列{1,3,7,6,5,3,7,7}，使用 Baum-Welch 迭代算法进行训练，同时使用对应的标签序列$\{1,1,-1,1,1,1,1,-1\}$对隐马尔可夫模型λ中状态的标签进行标记。特别地，每个状态的标记使用访问该状态时最可能的标签进行标记：

$$l_i = argmax_{l=1,-1} \sum_{t=1,l_t=l}^{T} \gamma_{ti}, i = 1,2,\dots,N$$

最后再使用另一个观察序列，此观察序列是待预测的观察序列，以同样的编码方式进行编码，借用问题二中的解法，求出其在观察序列结束时最可能的状态，然后输出该状态的标签作为预测结果。例如待预测的观察序列是{3,7,6,5,3,7,7,6}，计算该观察序列在模型λ下的γ_{Ti}，对所有的状态，选择其中拥有最大γ_{Ti}的状态作预测状态，并输出该状态的标签作为预测结果。

因为隐马尔可夫模型的初始化是随机的，而训练算法 Baum-Welch 迭代算法对初始值是敏感的，所以，不同的模型初始化将可能训练出不同的隐马尔可夫模型，进而在使用相同的训练序列和测试序列，可能会得到不同的预测结果。这样的话，隐马尔可夫模型的预测风险率不能直接使用单次训练和预测的结果，需要使用多次下的期望预测结果。

三、 多流选择性隐马尔可夫模型

本章对多流选择性隐马尔可夫模型进行介绍。首先向经典的隐马尔可夫模型引入多流和选择性的概念，并添加放缩机制，然后对模型参数的训练，状态标注与预测算法进行相应的改进与实现。

(一)模型

经典的隐马尔可夫模型虽然已经可以用做走势预测，但是预测效果一般且其不能提供风险投资中重要的可控性。为了提升其预测效果和可控性，我们引进了多流和选择性的概念。多流指的是使用多个观察序列同时训练隐马尔可夫模型，这些观察序列来源不同并且有一定的相关性，目的在于降低预测风险率；选择性的概念是指当隐马尔可夫预测模型的预测可信度不够高时拒绝进行预测，以得到可控性较高的结果。

（1）多流

对隐马尔可夫模型引入多流的概念已经有了一些研究[19][28]，本文根据 Lawrence 的工作[19]，对经典的隐马尔可夫模型增添如下多流的定义。令多流观察序列$O = \{O^{(1)}, O^{(2)}, \dots, O^{(K)}\}$，其中第$k$条观察序列为$O^{(k)} = \{O_1^{(k)}, O_2^{(k)}, \dots, O_{T_k}^{(k)}\}$。使用多流观察序列进行参数估计时，需要使得$P(O|\lambda)$最大化：

$$P(O|\lambda) = \prod_{k=1}^{K} P\left(O^{(k)}|\lambda\right) = \prod_{k=1}^{K} P_k$$

另外，还可以将多条观察序列编码成一条观察序列，该方法要求较大的数据量，否则编码之后得到的观察序列会过于稀疏以至于无法使用。

（2）选择性

对隐马尔可夫模型引入选择性的概念已经有了一些研究[15]，本文根据 Dmitry[15] 的工作，对经典的隐马尔可夫模型增添如下选择性的定义。

访问率v_i，给定一个观察序列O和隐马尔可夫模型λ，其代表模型λ输出O的整个过程中访问状态i_i的概率：

$$v_i = \frac{1}{T}\sum_{t=1}^{T} \gamma_{ti}$$

风险率r_i，给定一个观察序列O和隐马尔可夫模型λ，其代表模型λ输出O的整个过程中访问状态i_i并且状态i_i的标签和标签序列里的标签不同的概率：

$$r_i = \frac{\frac{1}{T}\sum_{t=1,l_t \neq l_i}^{T} \gamma_{ti}}{v_i}$$

风险状态集合RS，该集合下的所有状态的预测输出为拒绝：

$$RS = \{i_1, i_2, \dots, i_i\}, i_1, i_2, \dots, i_i \in I$$

风险率与覆盖率的控制方法如下：给定一个覆盖率C_B，则拒绝率为$1 - C_B$。将所有的状态按风险率r_i从大到小进行排序，然后从前往后选取风险状态并同时累加风险状态的访问率v_i，直到加上下一个状态的访问率超过当前设定的拒绝率$1 - C_B$为止，此时得到的风险状态集合即为：

$$RS = \{i_1, \dots, i_K | \sum_{j=1}^{K} v_{i_j} \leq 1 - C_B, \sum_{j=1}^{K+1} v_{i_j} > 1 - C_B\}$$

但是这样可能会出现一定的问题，假设状态数太少或者有状态包含较高的访问率，此时可控性会降低，因为风险状态集合是以状态为最小单位的，反应到风险率覆盖率折中曲线，其将成为粗糙的阶跃函数，从而可控性和可用性都将降低。因此需要提出算法将风险率覆盖率折中曲线进行平滑，即精化计算RS时第$K+1$个状态的访问率$v_{i_{K+1}}$。

（3）放缩

除了多流和选择性，根据前向后向概率的定义，α_{ti}和β_{ti}是由多项状态转移概率a_{ij}和观察值输出概率b_{jk}相乘得出，而a_{ij}和b_{jk}范围均在 0 和 1 之间，因此，当观察序列足够长时，上述概率相乘将产生浮点数下溢。故还需要对前向后向算法引入放缩机制[19]。定义放缩参数C_t，

$$C_t = \frac{1}{\sum_{i=1}^{N} \alpha_{ti}}, 1 \leq t \leq T$$

放缩之后，

$$\alpha_{ti}^{s} = C_t \alpha_{ti}, 1 \leq i \leq N, 1 \leq t \leq T$$

$$\beta_{ti}^{s} = C_t \beta_{ti}, 1 \leq i \leq N, 1 \leq t \leq T$$

在使用放缩之后的前向后向概率α_{ti}^{s}和β_{ti}^{s}重新计算得出γ_{ti}^{s}和$\xi_{t,i,j}^{s}$

(二)算法实现

1. 模型参数的训练

根据修正定义的隐马尔可夫模型，在训练模型参数的时候，需要同时考虑多流，选择性和放缩。

关于多流，在进行模型参数估计的时候，需要将多个观察序列的情况都考虑进去，即将各个观察序列，按其在模型下出现的概率成分比例，相加起来。

关于选择性，可采用递归精化的方法来光滑风险率-覆盖率曲线。递归精化指的是，初始隐马尔可夫模型中可能存在访问率较高的风险状态，可以新生成一个隐马尔可夫模型代替此状态，因为新生成的隐马尔可夫模型中各个状态的访问率的总和等于被精化状态的访问率。如果新的隐马尔可夫模型中还存在访问率较高的风险状态，那么就再生成一个隐马尔可夫模型代替此状态，这样递归地进行精化，直到消除访问率较高的风险状态为止，一般地，可以人为设置一个的访问率上界v_B，当精化后低于此值时便终止递归精化。图 3－1 是一个简单的递归精化示例，初始的隐马尔可夫模型包括状态 1 和状态 2，其中状态 1 为访问率较高的风险状态，因此状态 1 被包含状态 3 和状态 4 的新隐马尔可夫模型替代，精化之后的隐马尔可夫模型包括状态 2，3 和 4，其中状态 4 也为访问率较高的风险状态，被包含状态 5 和状态 6 新隐马尔可夫模型替代，精化之后的隐马尔可夫模型包括状态 2，3，5 和 6，此时没有访问率较高的风险状态，终止精化。图 3－2 是状态 1 被包含状态 3 和状态 4 的新隐马尔可夫模型替代的过程，状态 1 的转入替换为状态 2 和状态 3 的转入，状态 1 的转出替换为状态 2 和状态 3 的转出，状态 1 的自转替换为状态 3 和状态 4 的自转和相互转移。

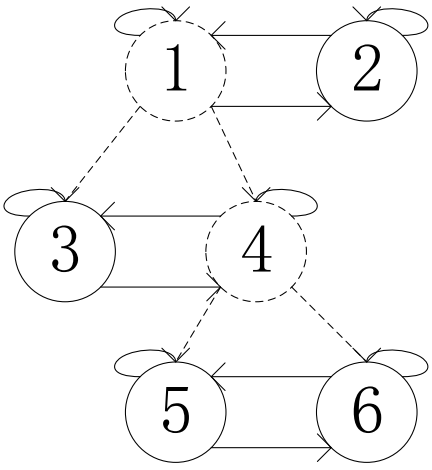

图 3－1 递归精化示例

Figure 3-1 Recursive Refinement Example

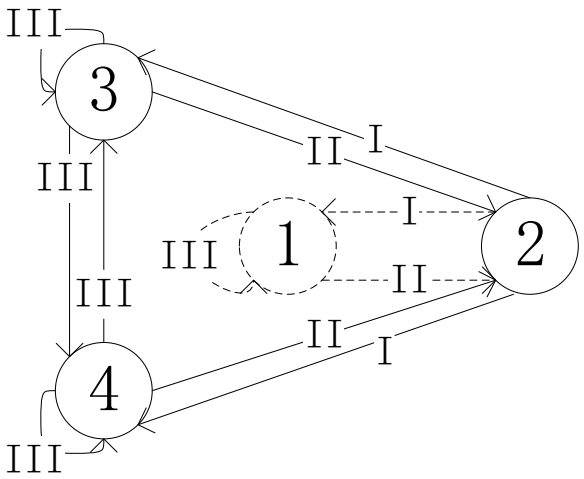

图 3－2 递归精化替换过程

Figure 3-1 Recursive Refinement Replacing Process

关于放缩，需要在对每一个观察序列计算前向后向概率的时进行相应的放缩。

整个训练过程如图 3－3 所示：

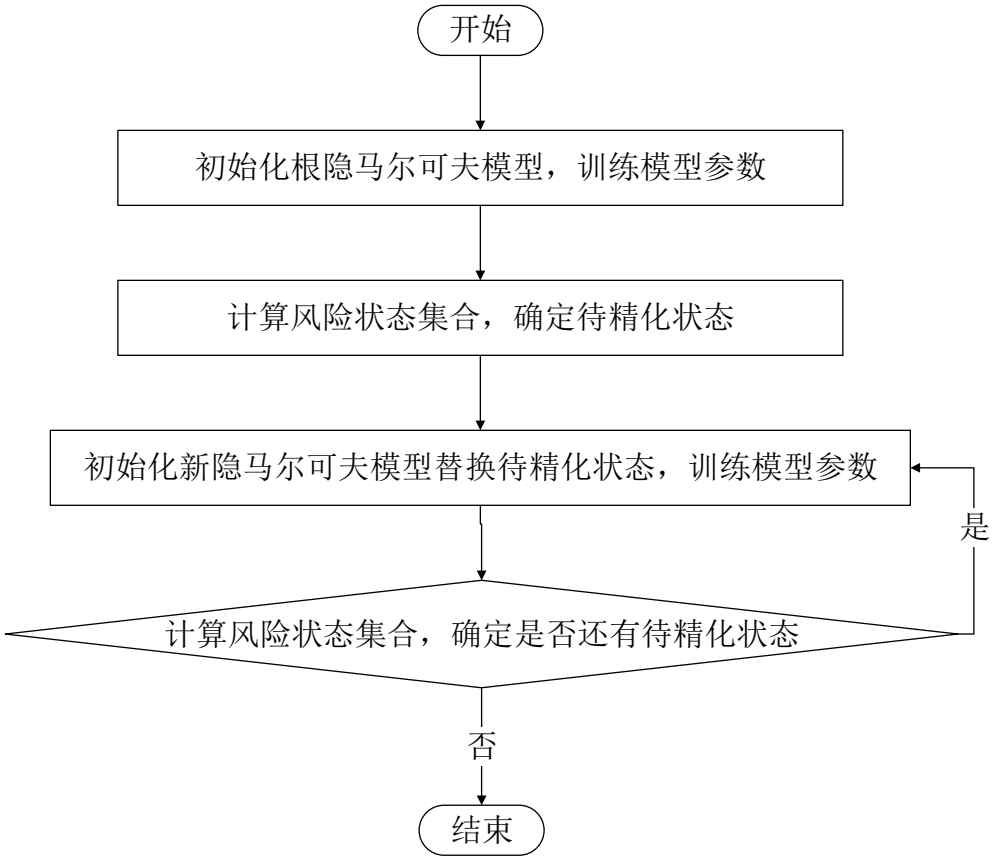

图 3－3 模型参数训练流程

Figure 3-3 Model Training Process

①首先随机初始化一个根节点的隐马尔可夫模型λ_0，使用编码后的历史金融走势序列$O^{(1)}$和编码后的情感序列$\{O^{(2)}, O^{(3)}, \ldots, O^{(K)}\}$对模型进行训练，训练算法使用经典

的 Baum-Welch 算法，同时按多流和放缩进行相应改进。

②然后确定模型应当满足的覆盖率C_B，以此找到需要精化的那个状态，即计算风险状态集合RS时的第$K+1$个状态，称之为高访问率风险状态i_h。

③接着随机生成一个新的隐马尔可夫模型λ_r，新模型λ_r中所有状态的标签l继承于原状态i_h的标签l_{i_h}，新模型λ_r中每个状态的起始概率π_i分割原状态i_h的起始概率π_{i_h}，新模型λ_r中每个状态的转入继承于原状态i_h的所有转入，新模型λ_r每个状态的转出继承于原状态i_h的所有转出，新模型λ_r内部状态的自转和互相转移继承于原状态i_h的自转，并使用编码后的历史金融走势序列和情感序列对新模型进行训练，直到达到算法的收敛条件。

④最后用新生成的隐马尔可夫模型λ_r替换掉原高访问率风险状态i_h，并在新模型上再次计算风险状态集合RS，如果还存在高访问率风险状态i_h，则返回③，否则终止。

第③步中的递归精化训练算法如下：

输入：一个N个状态的隐马尔可夫模型λ，高访问率风险状态i_h，多流观察序列$O=\{O^{(1)},O^{(2)},\dots,O^{(k)}\}$

随机生成一个n个状态隐马尔可夫模型λ_r

对每个$j=1,2,\dots,N,j\neq h$，将转移i_ji_h替换为$i_ji_{N+1},i_ji_{N+2},\dots,i_ji_{N+n}$，将转移$i_hi_j$替换为$i_{N+1}i_j,i_{N+2}i_j,\dots,i_{N+n}i_j$

将高访问率风险状态i_h在λ中记录为已精化，去除其观察值概率分布，对于所有的$j=N+1,N+2,\dots,N+n$，设置$l_{i_j}=l_{i_h}$

当不收敛时，重做如下过程：

对于每个$j=1,2,\dots,N,j\neq h,k=1,2,\dots,n$，更新

$$a_{j(N+k)}=a_{jh}\pi_{N+k}$$

$$a_{(N+k)j}=a_{hj}$$

对于每个$j=N+1,N+2,\dots,N+n$，更新

$$\pi_j=\pi_h\pi_j$$

对于每个$j,k=N+1,N+2,\dots,N+n$，更新

$$a_{jk}=a_{hh}a_{jk}$$

重估

$$\pi_j = \frac{\sum_{i=1}^{K}\frac{1}{P_i}(\gamma_{1j}^{(i)s} + \sum_{t=1}^{T_k-1}\sum_{k=1,k\neq h}^{N}\xi_{t,k,j}^{(i)s})}{Z}$$

$$a_{jk} = \frac{\sum_{i=1}^{K}\frac{1}{P_i}\sum_{t=1}^{T_k-1}\xi_{t,j,k}^{(i)s}}{\sum_{l=N+1}^{N+n}\sum_{i=1}^{K}\frac{1}{P_i}\sum_{t=1}^{T_k-1}\xi_{t,j,l}^{(i)s}}$$

$$b_{jm} = \frac{\sum_{i=1}^{K}\frac{1}{P_i}\sum_{t=1,o_t^{(i)}=m}^{T_k}\gamma_{tj}^{(i)s}}{\sum_{i=1}^{K}\frac{1}{P_i}\sum_{t=1}^{T_k}\gamma_{tj}^{(i)s}}$$

收敛后，再更新一次所有的$a_{j(N+k)}, a_{(N+k)j}, \pi_j, a_{jk}$

输出：一个$N-1+n$个状态隐马尔可夫模型λ

该递归精化训练算法的算法复杂度为$O(KTn^2)$。

2. 状态标注与预测

由于最后预测的是金融走势序列，因此在状态标注与预测环节，仅使用金融走势序列进行处理，整个流程如图 3—4：

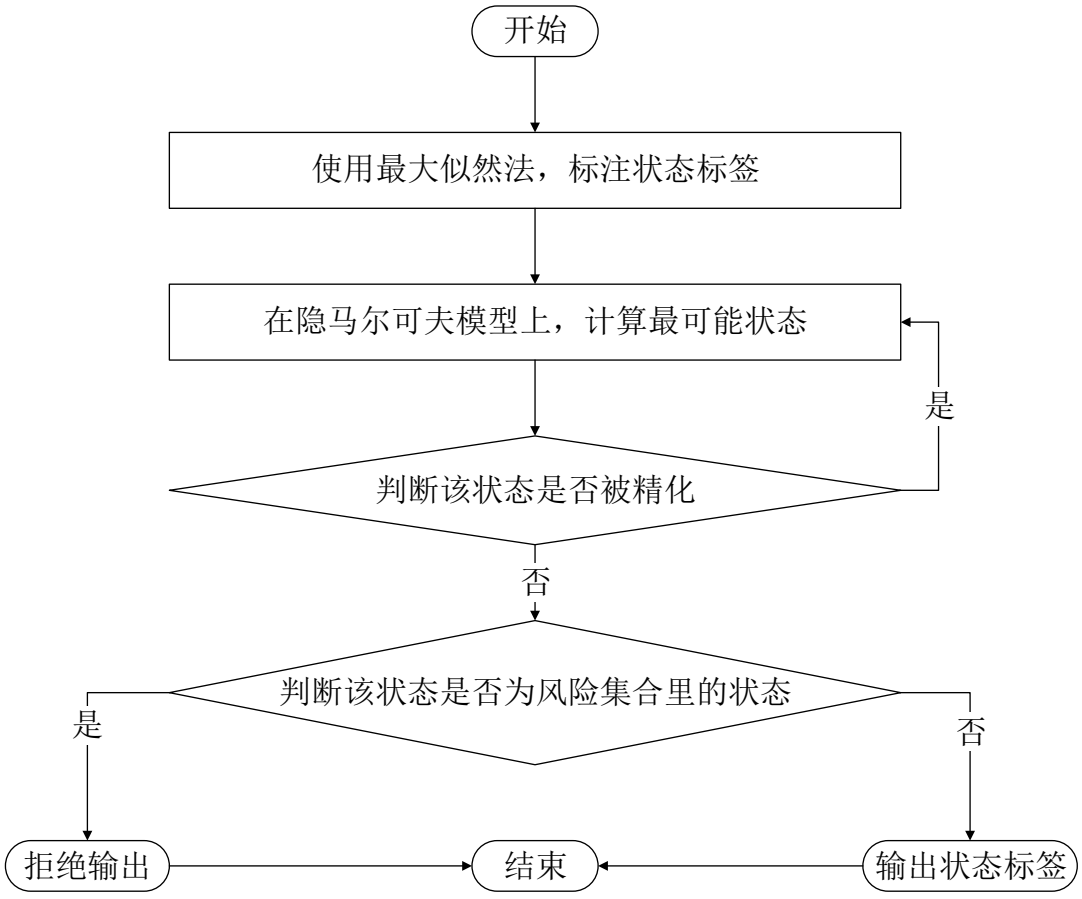

图 3—4 模型预测流程

Figure 3-4 Model Prediction Process

该模型是利用状态的标签进行预测的，因此在预测前需要先对模型中所有状态进行标注。同样是使用γ_{ti}^{s}，对每个状态，分别计算访问该状态时标签序列中对应标签为涨和为跌的期望概率，然后从两者中选出概率大者作为该状态的标签，即

$$l_i = argmax_{l=1,-1} \sum_{t=1,lt=l}^{T} \gamma_{ti}^{s}$$

在预测阶段，仅输入一个观察序列，即金融走势序列，找出序列末尾T时刻最可能的状态，此时因为涉及到模型引入了多流和选择性，所以有一些改动。

①首先从最初的隐马尔可夫模型开始，计算γ_{Ti}^{s}找出T时刻最可能的状态。

②如果该状态没有被精化，则跳至③，否则，对精化之后的隐马尔可夫模型，重新计算γ_{Ti}^{s}找出T时刻最可能的状态，且该状态应该属于精化之后新加入的状态，然后返回②。

③如果该状态是风险状态集合RS中的状态，则拒绝输出预测结果，否则输出该状态的标签作出预测结果。

四、 系统实现与实验结果

本章对系统实现与实验结果进行介绍。首先概述系统整体架构设计，然后详细介绍系统中的数据获取模块和预测模块的设计，实现和中间结果，最后对比分析金融走势预测的结果并得出相应的结论。

(一)系统框架设计

基于情感分析的金融走势选择性预测系统主要由两部分组成：数据获取模块和预测模块，如图 4－1 所示。数据获取模块的职能是预处理金融数据和情感数据，得到历史金融走势序列和有预测价值的情感序列；预测模块的职能是基于输入的历史数据进行建模，当有新的数据到来时，输出预测，然后基于预测模块输出的结果，进行模拟投资，最后对投资结果进行分析。由此，本文介绍了基于情感分析的金融走势选择性预测系统的一个框架。

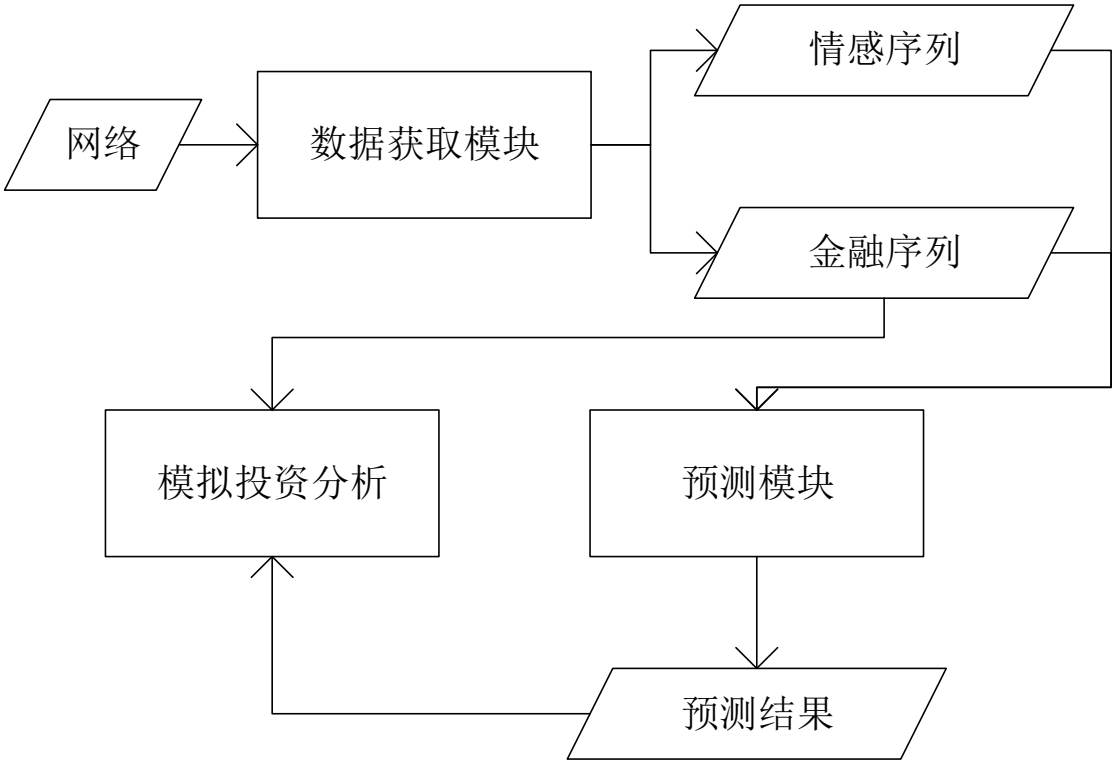

图 4－1 系统框架

Figure 4-1 System Framework

(二)数据获取模块

数据获取模块主要分为金融数据获取模块和情感数据获取模块。

金融数据获取模块的流程如图 4－2 所示，按需求从 Yahoo!Finance 上采集相关的金融数据，然后计算出金融走势，以及一些其他预处理。

形式化地，令第t天的 DJIA 收盘价为C_t，则第t天的 DJIA 的变化为：

$$D_t = C_t - C_{t-1}$$

第t天的 DJIA 的增长率为：

$$R_t = \frac{C_t - C_{t-1}}{C_{t-1}}$$

第t天的 DJIA 的对数增长率为：

$$r_t = ln\frac{C_t}{C_{t-1}}$$

第t天的 DJIA 的走势为：

$$T_t = sign(C_t - C_{t-1})$$

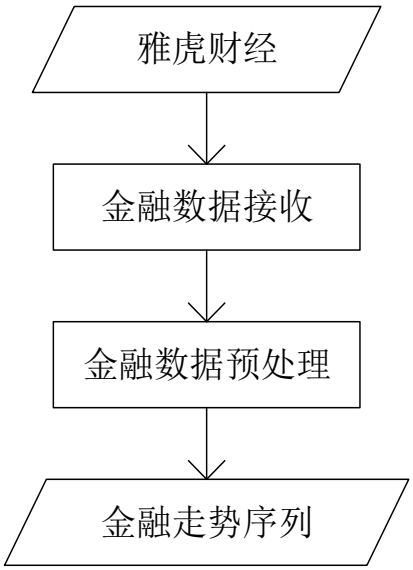

图 4－2 金融数据获取模块流程

Figure 4-2 Financial Data Acquisition Process

下表是金融数据获取模块的类说明：

表 4－1 金融数据获取模块的类说明

Table 4-1 Financial Data Acquisition Process Class Specification

类名	描述
YahooFinanceRecord	金融数据记录类，包含日期，收盘价等属性
YahooFinanceReceiver	金融数据接收类，通过 URL 获取金融数据记录
YahooFinanceProcesser	金融数据处理类，计算金融走势等属性

图 4－3 为 DJIA 从 2009 年 6 月 1 日到 2009 年 12 月 31 日期间的每日收盘价格曲线。

图 4－4 为 DJIA 从 2009 年 6 月 2 日到 2009 年 12 月 31 日期间的每日收盘价格走势曲线。

图 4－3 DJIA 收盘价曲线

Figure 4-3 DJIA Closing Price Curve

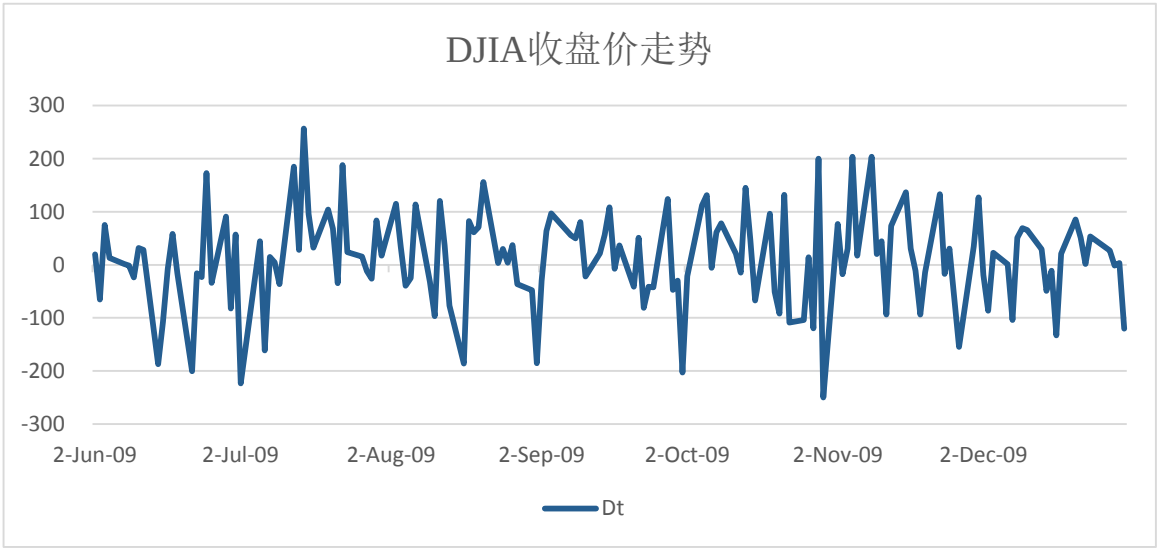

图 4－4 DJIA 收盘价走势曲线

Figure 4-4 DJIA Closing Price Trends Curve

情感数据获取模块的流程如图 4－5 所示，按需求从 Twitter 获取数据，然后对 Twitter 数据进行过滤，词干化，使用经 WordNet 扩展后 POMS Biploar 六维情感词表进行情感分析，最后再按一定的粒度进行聚集。

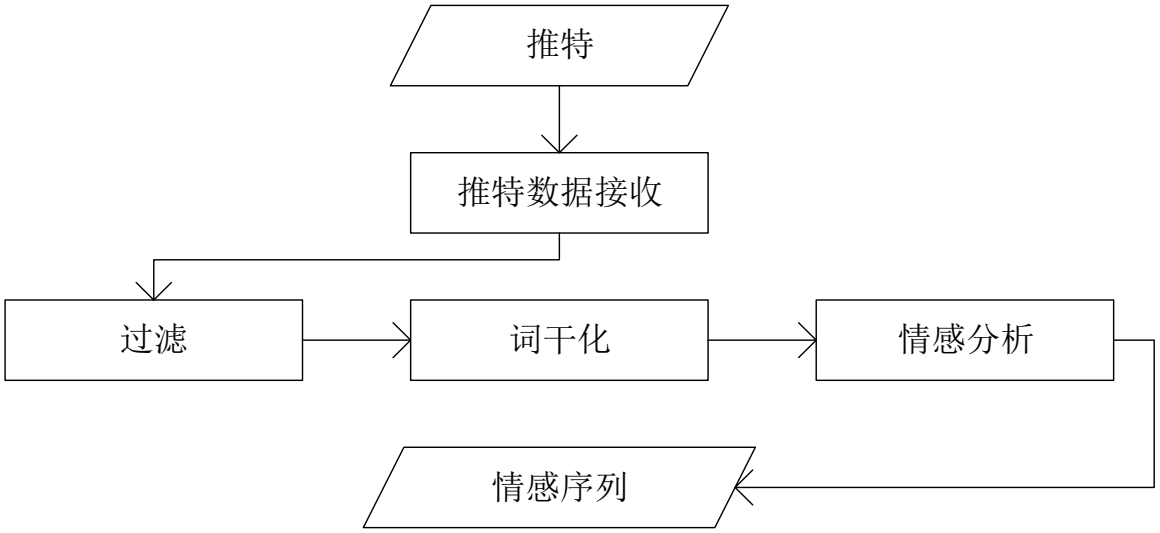

图 4－5 情感数据获取模块流程

Figure 4-5 Sentiment Data Acquisition Process

下表是金融数据获取模块的类说明：

表 4－2 情感数据获取模块的类说明

Table 4-2 Sentiment Data Acquisition Process Class Specification

类名	描述
TwitterRecord	Twitter 数据记录类，包含 Tweet 等属性
TwitterReceiver	Twitter 数据接收类，通过 URL 获取 Twitter 数据记录
Filter	过滤类，保留主观 Tweet
Stemmer	词干化类，词干化 Tweet
SentimentAnalyzer	情感分析类，对 Tweet 进行情感分析

在进行过滤的时候，定义了三类标签：

①包含 http:, www.；

②包含 i feel, I am feeling, i'm feeling, i dont feel, i'm, im, i am, makes me；

③包含 stock, finance, financial, djia, dow, dji, bearish, bear market, bullish, bull market, wall street, $aa, $axp, $ba, $bac, $cat, $csco, $cvx, $dd, $dis, $ge, $hd, $hpq, $ibm, $intc, $jnj, $jpm, $ko, $mcd, $mmm, $mrk, $msft, $pfe, $pg, $t, $trv, $unh, $utx, $vz, $wmt, $xom。

规定，当 Tweet 包含第一类标签时抛弃，当 Tweet 包含第二类或者第三类标签时保留。

实验所用的 Twitter 数据为 2009 年 6 月到 2009 年 12 月共 4 亿 7 千 6 百万条，使用上述过滤规则后剩余 3 千 7 百万条，如表 4－2 所示，保留的 Twitter 数据占总 Twitter 数据的 8%，其中包含第二类标签的 3 千 7 百万条，包含第三类标签的仅二

十万条，即包含第二类标签的 Twitter 数据在过滤后的 Twitter 数据里占超过 99.5%，由此发现第三类标签的数目过小而不利于分析，故之后的分析仅考虑包含第二类标签的 Twitter 数据。

表 4－3Twitter 数据过滤结果

Table 4-3 Twitter Data Filtered Result

month	subject	relative	stay	count	subject/stay	relative/stay	stay/count
200906	591707	5254	596659	18572084	99.17005%	0.88057%	3.21267%
200907	2011892	14499	2025252	46203172	99.34033%	0.71591%	4.38336%
200908	11733007	59541	11786089	132210436	99.54962%	0.50518%	8.91464%
200909	7556427	42982	7595256	94176126	99.48877%	0.56591%	8.06495%
200910	6189737	34473	6220921	75520065	99.49872%	0.55415%	8.23744%
200911	4762231	23175	4783024	56838024	99.56528%	0.48453%	8.41518%
200912	4228752	21113	4247997	53033653	99.54696%	0.49701%	8.01000%
all	37073753	201037	37255198	476553560	99.51297%	0.53962%	7.81763%

在进行词干化的时候，不仅对过滤之后的 Tweet 进行词干化，还要对情感词表里的所有词进行词干化，以提高情感分析阶段的匹配度。

在进行情感分析的时候，首先对每一条 Tweet，按空格进行分词，对每一个词，扫描整个情感词表，如果在情感词表里出现，则按其正负特性在相应的情感极性里加 1 或减 1，这样消耗完 Tweet 得到一个六维的情感向量，再将这个向量进行单位化，所得即为此条 Tweet 的情感向量。然后以天为单位，求当天所有情感向量的平均向量，用以代表该天的情感向量。由于在处理过程中同时记录了每天的 Tweet 数，发现在 2008/10/25, 2009/6/8, 2009/6/11, 2009/7/3, 2009/7/4, 2009/7/5, 2009/7/9, 2009/7/15 八天 Tweet 数目明显少于其他日期的 Tweet 数目，因此去除前六天的数据，并对后两天的数据做线性插值。

接着对得到六维情感序列计算 z-score 放缩，得到的六维情感序列如图 4－6 到图 4－11 所示。

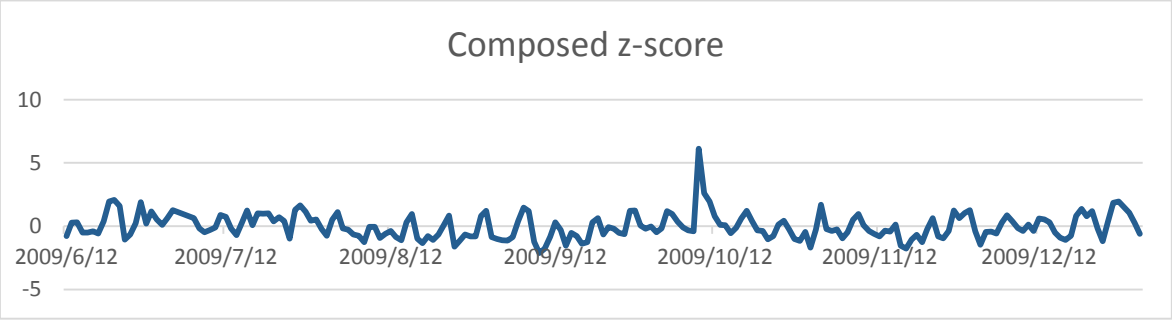

图 4－6 冷静-焦虑情感序列

Figure 4-6 Composed/Anxious Index

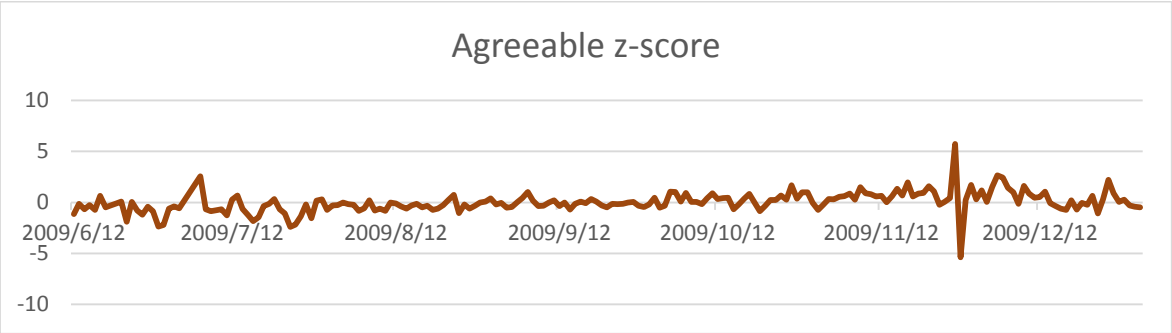

图 4－7 同意-敌对情感序列

Figure 4-7 Agreeable/Hostile Index

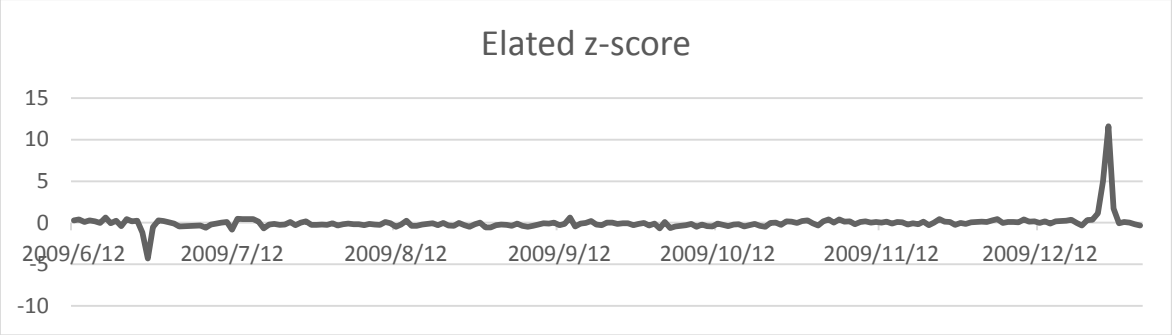

图 4－8 活力-疲劳情感序列

Figure 4-8 Elated/Depressed Index

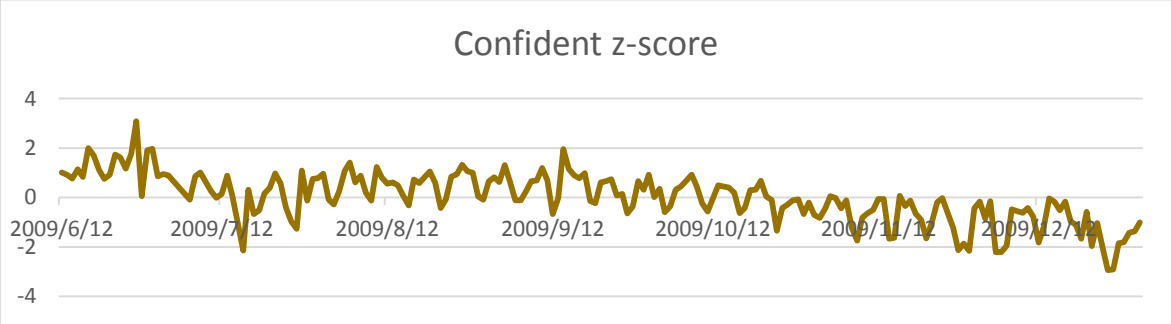

图 4－9 自信-怀疑情感序列

Figure 4-9 Confident/Unsure Index

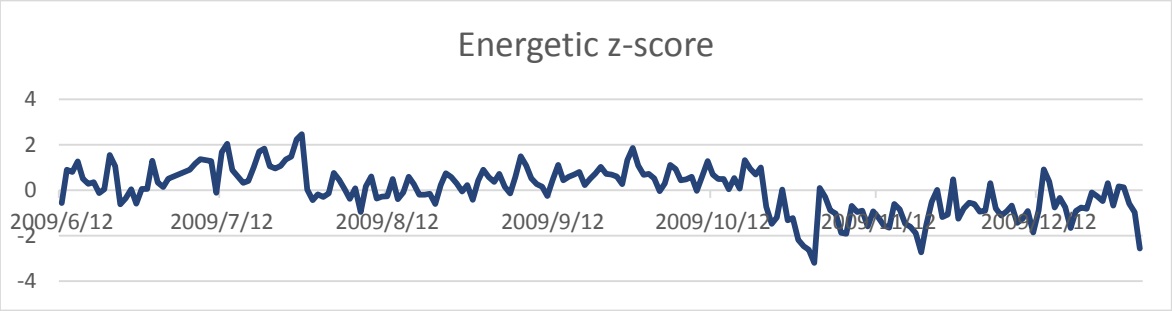

图 4－10 活力-疲劳情感序列

Figure 4-10 Energetic/Tired Index

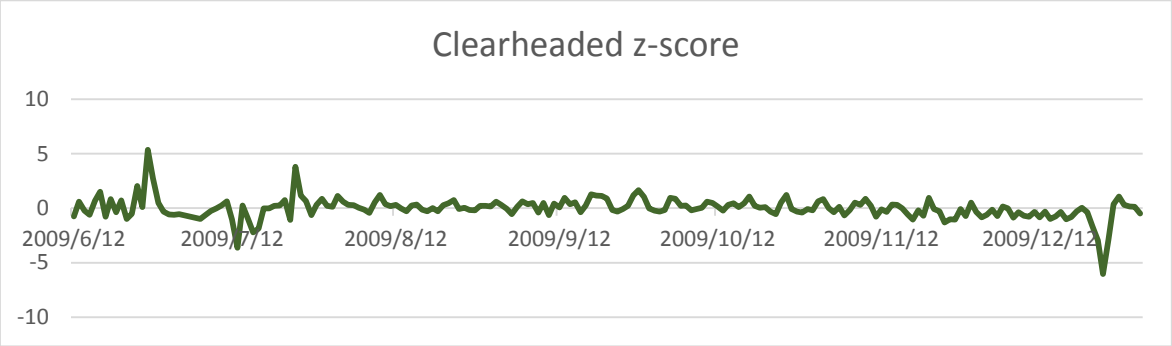

图 4－11 清醒-迷惑情感序列

Figure 4-11 Clearheaded/Confused Index

当生成金融走势序列和 Twitter 情感序列之后，需要分别使用每条 Twitter 情感序列和金融走势序列做格兰因果关系分析，如表 4－4 所示，以明确最具预测能力的 Twitter 情感序列及相应的延后项。在实验中同时采用了基于 Alex Davies 的情感词表提取的单维情感序列的格兰因果关系分析，发现其效果确实没有多维情感分析好。

表 4－4 格兰因果关系分析结果

Table 4-4 Granger Causality Analysis Result

Lagged Days	Composed /Anxious	**Agreeable /Hostile**	Elated /Depressed	Confident /Unsure	Energetic /Tired	Clearheaded /Confused
1	0.723009776	0.512862214	0.9399375	0.880644906	0.857355253	0.342346356
2	0.86129301	0.166551184	0.8289756	0.576292251	0.933422157	0.310755746
3	0.434470424	**0.062817907**	0.9608715	0.455076866	0.993825935	0.377955186
4	0.435631775	0.127495831	0.9903607	0.637129619	0.803028135	0.514455259
5	0.593896982	0.212591485	0.9854185	0.534574688	0.755306207	0.708745583
6	0.630440149	0.206866576	0.9689204	0.656838808	0.557477213	0.738674666
7	0.694607494	0.107745913	0.9858471	0.688712317	0.577784406	0.851840215

从格兰因果关系分析结果中可以发现延后项为 3 天的 Twitter 同意-敌对情感序列对 DJIA 走势序列具有最大的预测能力，因为该情况下的$p-value$为小于 0.1 的显著水平。图 4－12 是 DJIA 走势序列和 Twitter 同意-敌对情感序列的 z-score 比较，可以看出 Twitter 同意-敌对情感序列前三天的走势与 DJIA 走势序列有一定的相似。

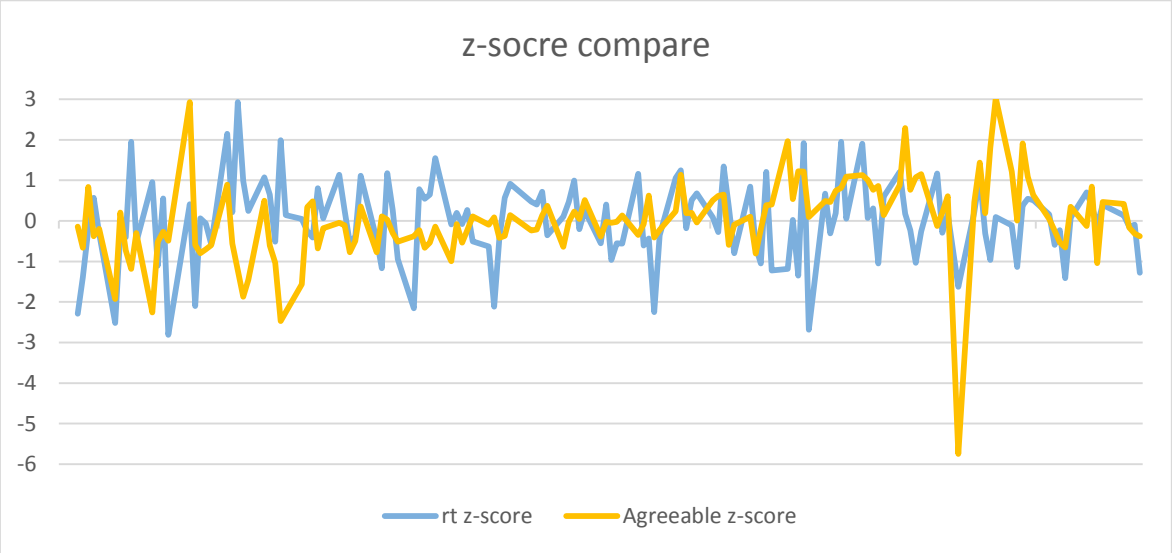

图 4－12 DJIA 走势序列与 Twitter 同意-敌对情感序列

Figuare 4-12 DJIA trend vs Twitter Agreeable/Hostile Index

(三)预测模块

预测模块的具体流程如图 4－13 所示。首先使用金融走势序列和情感序列训练预测模型的参数，同时使用金融走势序列对模型中的状态进行标注，然后使用金融走势序列基于模型进行预测得到预测结果，接着基于预测模块输出的结果进行模拟投资，最后对投资结果进行分析。

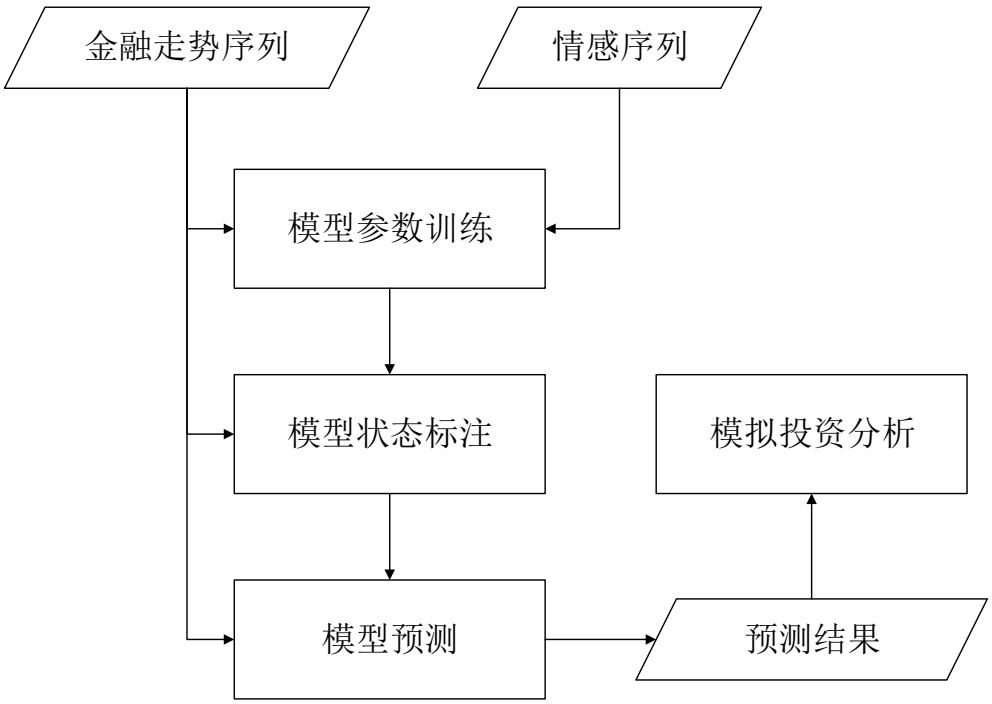

图 4－13 预测模块流程

Figure 4-13 Predicion Process

修正定义后，引入多流，选择性，放缩机制的隐马尔可夫预测模型，下表是预测模块的类说明：

表 4－6 预测模块类说明

Table 4-6 Prediction Process Class Specification

类名	描述	类型
HMM	隐马尔可夫模型类	模型
MultistreamsHMM	多流选择性隐马尔可夫模型类	模型
BaumWelchLearner	BaumWelch 训练类	训练算法
BaumWelchScaledLearner	放缩 BaumWelch 训练类	训练算法
MultistreamRecursiveRefine-BaumWelchScaledLearner	多流递归精化放缩 BaumWelch 训练类	训练算法
ForwardBackwardCalculator	前向后向算法类	评估算法
ForwardBackwardScaledCalculator	放缩前向后向算法类	评估算法
Observation	观察值类	抽象类
ObservationDiscrete	离散型观察值类	观察值
ObservationInteger	整数型观察值类	观察值
Opdf	观察值概率分布类	抽象类
OpdfDiscrete	离散型观察值概率分布类	观察值分布
OpdfInteger	整数型观察值概率分布类	观察值分布
OpdfFactory	观察值概率分布抽象工厂类	抽象工厂
OpdfDiscreteFactory	离散型观察值概率分布抽象工厂类	具体工厂
OpdfIntegerFactory	整数型观察值概率分布抽象工厂类	具体工厂

下图是预测模块的类图，在面向对象设计时，对观察值和概率分布的实现采用抽象工厂设计模式，并同时使用了继承，多态和组合等特性：

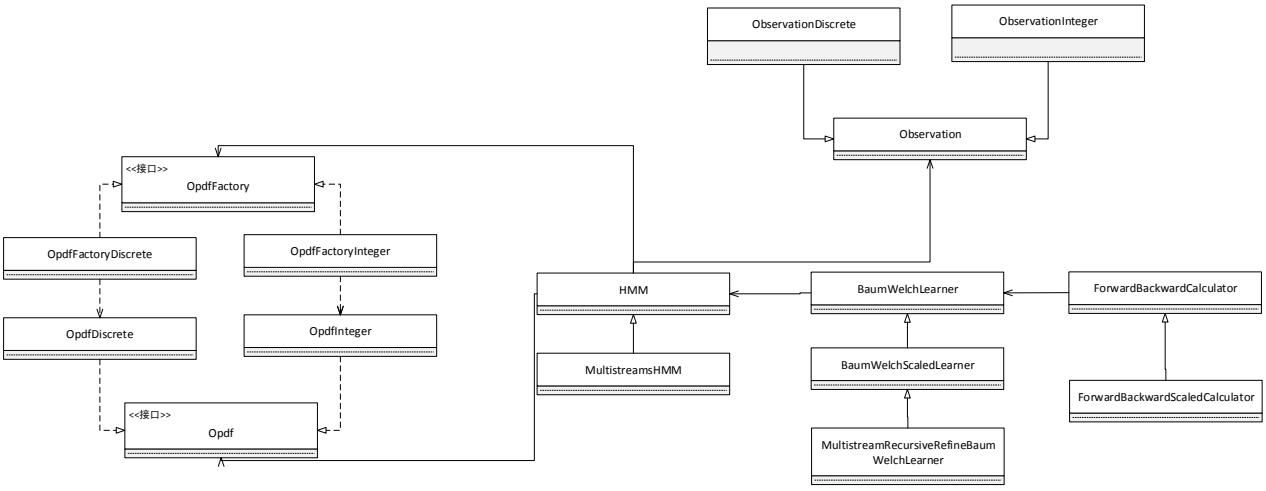

图 4－14 预测模块类图

Figure 4-14 Prediction Process Class Diagram

模拟投资分析的职能是使用预测模型预测的结果进行模拟投资。

一般地，有两条基线：

①买进后持有的投资曲线 Buy and Hold

②能预知未来的投资曲线 Hindsight

使用这两条曲线分别为投资收益曲线的下界和上界。一般地，预测模型的投资收益曲线应当位于这两条曲线之间。

(四)实验结果

针对上述修正定义的多流选择性隐马尔可夫模型，本文基于 java 语言进行实现，并使用 DJIA 走势序列和 Twitter 同意-敌对情感序列对模型进行测试。

实验一共实现了四种模型：

①基于 DJIA 走势的线性回归模型$lRegress_D$；

②结合 DJIA 走势和 Twitter 同意-敌对情感序列的线性回归模型$lRegress_{DS}$；

③基于 DJIA 走势的选择性隐马尔可夫模型$sHMM_D$；

④结合 DJIA 走势和 Twitter 同意-敌对情感序列的多流选择性隐马尔可夫模型$sHMM_{DS}$。

经预处理之后的数据，时间范围为 2009 年 6 月 15 日到 2009 年 12 月 31 日，其中不包含休市日期，共 140 天数据。使用前 90 天作为训练数据，使用后 50 天作为测试数据，得到的实验结果如图 4－15：

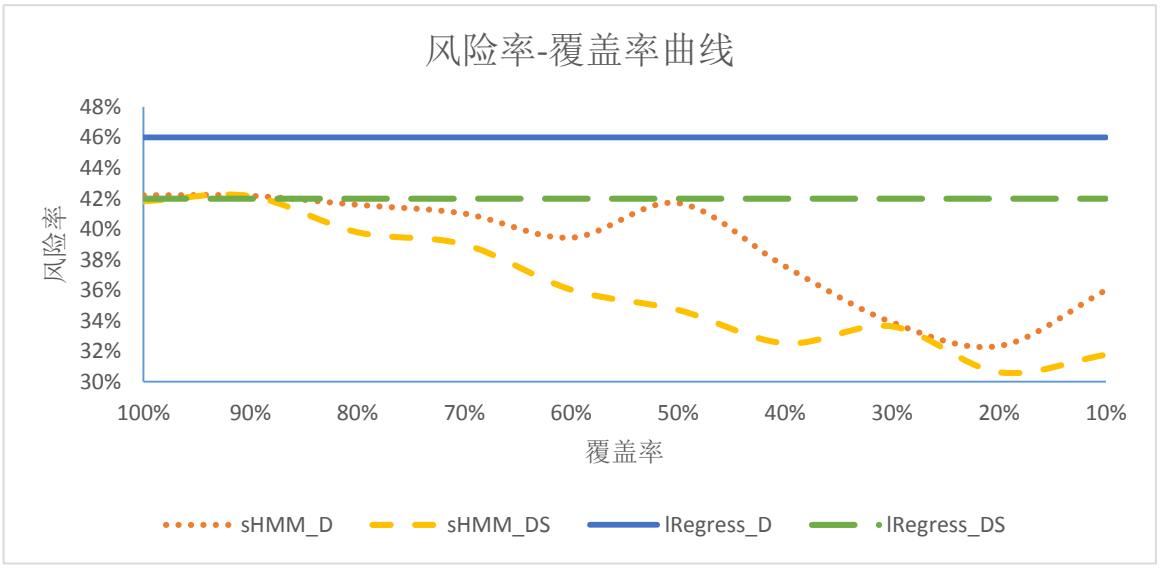

图 4－15 风险率-覆盖率曲线

Figure 4-15 Risk-Coverage Curve

从图中可以明显地看出：

对于线性回归模型$lRegress_D$和$lRegress_{DS}$，没有可控性而言，可看作其风险率在所有的覆盖率下保持不变，因此体现在风险率-覆盖率曲线上是一条水平线。通过对

比基于 DJIA 走势的线性回归模型$lRegress_D$，和结合 DJIA 走势和 Twitter 同意-敌对情感序列的线性回归模型$lRegress_{DS}$，可以发现结合 DJIA 走势和 Twitter 同意-敌对情感序列的线性回归模型$lRegress_{DS}$的风险率低于基于 DJIA 走势的线性回归模型$lRegress_D$。

对于选择性隐马尔可夫模型$sHMM_D$和$sHMM_{DS}$，由于引入了选择性，需要同时考虑有界弃权模型和有界提升模型。通过对比基于 DJIA 走势的选择性隐马尔可夫模型$sHMM_D$，和结合 DJIA 走势和 Twitter 同意-敌对情感序列的多流选择性隐马尔可夫模型$sHMM_{DS}$，可以发现结合 DJIA 走势和 Twitter 同意-敌对情感序列的多流选择性隐马尔可夫模型$sHMM_{DS}$的风险率更低，覆盖率更高。

通过对比线性回归模型$lRegress$和选择性隐马尔可夫模型$sHMM$，可以发现选择性隐马尔可夫模型$sHMM$的风险率低于线性回归模型$lRegress$，并且拥有很大的可控性。通过对比仅基于 DJIA 走势，和结合 DJIA 走势和 Twitter 同意-敌对情感序列，可以发现 Twitter 同意-敌对情感序列确实能降低模型的预测风险率。

在模拟投资中，使用如下两种预测模型：

①仅使用历史 DJIA 走势的线性回归模型 Regress

②使用历史 DJIA 走势和 Twitter 同意-敌对情感的线性回归模型 Regress Combine

对上述两种模型进行相应的投资模拟分析，采用如下投资策略：如果预测明天涨，则持有股票；如果预测明天跌，则不持有股票。发现基于该种投资策略的结果如图 4－16 所示，此时模拟投资曲线的确设定的上下界之间，但走势预测风险率低的预测模型，模拟投资曲线不一定越高。

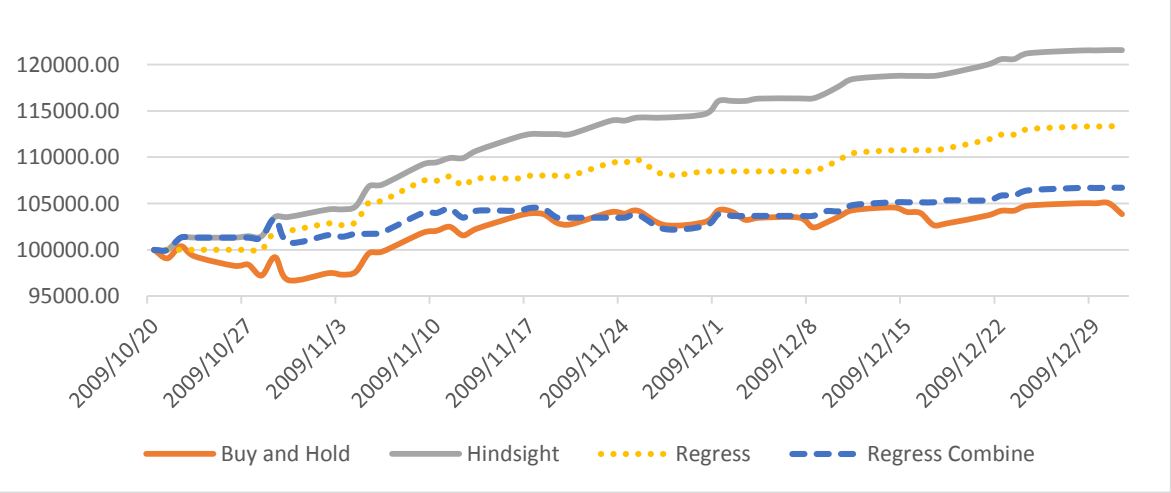

图 4－16 模拟投资曲线

Figure 4-16 Investment Simulation Curve

五、 总结和展望

(一)总结

本文提出了基于情感分析的金融走势选择性预测系统框架，并以 DJIA 数据和 Twitter 数据为例，对框架进行了相应的实现：主要地，实现了 Twitter 数据多维情感分析的方法，使用经 WordNet 扩展后的 POMS Bipolar 六维情感词表，提取出冷静-焦虑，同意-敌对，欢乐-失望，自信-怀疑，活力-疲劳和清醒-迷惑六种情感序列，并通过进一步的统计分析，发现延后 3 天的 Twitter 同意-敌对情感序列对 DJIA 走势拥有最强的预测能力；实现了多流选择性隐马尔可夫模型并配合模型修改相应的参数训练，状态标记和预测，再结合 DJIA 走势和 Twitter 同意-敌对情感序列，对未来的 DJIA 走势进行预测，发现该模型的准确性和可控性高于其他对比的模型；使用模型预测的结果基于一定的投资策略进行模拟投资，发现在该投资策略下，走势预测风险率越低的预测不能保证获得越高的回报。

本文得到如下结论：多流的预测效果高于单流，即适当地加入相关情感序列有助于金融走势的预测；隐马尔可夫预测效果高于线性，即隐马尔可夫模型预测风险率低于线性回归模型；选择性预测的引入，使得预测的可控性增强，这样可以为投资者提供丰富的决策空间。

(二)未来展望

本文最重要的工作是将金融数据和情感数据在选择性隐马尔可夫模型上进行了结合，目前采用的结合方式是多流，即将金融序列和情感序列看作同时产生的两个序列，从而对隐马尔可夫模型进行训练。本文同时也尝试过其他方法，比如将金融序列和情感序列编码成一条观察序列，但是由于 Twitter 数据量的限制，此种编码方式将产生稀疏的观察值，从而不能获得较好的结果，未来希望能在数据量充分的情况下采用更好的组合方式。本文是基于 Twitter 的情感分析来预测 DJIA 走势，下一步基于微博的情感分析来预测上证指数走势，期货走势等等，也会相当有趣。本文只考虑了离散的隐马尔可夫模型，未来可以进一步探索概率分布是由多个高斯函数加权混合而成的连续隐马尔可夫模型。本文尝试使用模型预测的结果基于一定的投资策略进行模拟投资，但发现在该投资策略下，走势预测风险率越低的预测不能保证获得越高的回报，未来可进一步探索更加合理的投资策略，使得走势预测风险率越低获得的投资回报越高。

参考文献

[1] E.F. Fama. The behavior of stock-market prices [J]. The Journal of Business. 1965, 38 (1): 34–105.

[2] J.R. Nofsinger. Social mood and financial economics [J]. Journal of Behaviour Finance. 2005, 6 (3): 144–160.

[3] Huina Mao, Scott Counts, Johan Bollen. Predicting Financial Markets: Comparing Survey, News, Twitter and Search Engine Data [J]. CoRR. 2011, abs/1112.1051.

[4] 蔡瑞胸. 金融时间序列分析 [M]. 北京：人民邮电出版社，2009，49-56.

[5] Ran El-Yaniv and Dmitry Pidan. Selective Prediction of Financial Trends with Hidden Markov Models [C]. NIPS'11, Granada, 2011.

[6] 龚健，马成虎. 基于隐马尔可夫链的上证股指建模 [J]. Finance. 2012, 2 (1): 45-49.

[7] Johan Bollen, Huina Mao and Xiaojun Zeng. Twitter mood predicts the stock market [J]. Journal of Computational Science. 2011, 2(1): 1-8.

[8] M. Bicego, E. Grosso and E. Otranto. A Hidden Markov Model approach to classify and predict the sign of financial local trends [C]. SSPR'08, Orlando, 2008.

[9] Yang Liu, Xiangji Huang, Aijun An and Xiaohui Yu. ARSA: A Sentiment-Aware Model for Predicting Sales Performance Using Blogs [C]. SIGIR'07, Amsterdam, 2007.

[10] Eric Gilbert and Karrie Karahalios. Widespread Worry and the Stock Market [C]. ICWSM, Washington, 2010.

[11] Leng G, Prasad G, McGinnity TM. An on-line algorithm for creating self-organizing fuzzy neural networks [J]. Neural Netw. 2004, 17(10): 1477-93.

[12] HYUNYOUNG CHOI, HAL VARIAN. Predicting the Present with Google Trends [J]. Economic Record. 2012, 88(s1): 2–9.

[13] Satish Rao, Jerry Hong. Analysis of Hidden Markov Models and Support Vector Machines in Financial Applications [R]. 2010, 5, 12.

[14] J. Yang, J. Leskovec. Temporal Variation in Online Media [C]. WSDM '11, Hong Kong, 2011.

[15] R. El-Yaniv and Y. Wiener. On the foundations of noise-free selective classification [J]. JMLR. 2010, 11(May): 1605–1641.

[16] T. Pietraszek. Optimizing abstaining classifiers using ROC analysis [C]. ICML, Bonn, 2005.

[17] Wilson, Theresa and Hoffmann, Paul and Somasundaran, Swapna and Kessler, Jason and Wiebe, Janyce and Choi, Yejin and Cardie, Claire and Riloff, Ellen and Patwardhan, Siddharth. OpinionFinder: A system for subjectivity analysis [C]. EMNLP, Vancouver, 2005.

[18] McNair, Douglas; Lorr, Maurice; Droppleman, Leo. Profile of Mood States (POMS) [M]. 1989.

[19] Lawrence R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition [J]. Proceedings of the IEEE. 1989. 77 (2): 257–286.

[20] Gilad Mishne and Maarten de Rijke. Capturing Global Mood Levels using Blog Posts [C]. AAAI, Palo Alto, 2006.

[21] Peter Sheridan Dodds, Christopher M. Danforth. Measuring the Happiness of Large-Scale Written Expression Songs, Blogs, and Presidents [J]. Journal of Happiness Studies. 2010. 11 (4): 441-456.

[22] Ellen Riloff and Janyce Wiebe. Learning Extraction Patterns for Subjective Expressions [C]. EMNLP, Stroudsburg, 2003.

[23] Robert P. Schumaker and Hsinchun Chen. Textual Analysis of Stock Market Prediction Using Breaking Financial News: The AZFinText System [J]. ACM Transactions on Information Systems. 2009, 27(2): 1-19.

[24] Daniel Gruhl, R. Guha, Ravi Kumar, Jasmine Novak, and Andrew Tomkins. The Predictive Power of Online Chatter [C]. KDD, New York, 2005.

[25] Brendan O'Connor, Ramnath Balasubramanyan, Bryan R. Routledge, Noah A. Smith. From Tweets to Polls Linking Text Sentiment to Public Opinion Time Series [C]. AAAI, Atlanta, 2010.

[26] Xue Zhang, Hauke Fuehres, Peter A. Gloor. Predicting Stock Market Indicators Through Twitter "I hope it is not as bad as I fear" [J]. Procedia - Social and Behavioral Sciences. 2011. 26: 55-62.

[27] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic Sentiment Mixture Modeling Facets and Opinions in Weblogs [C]. WWW, New York, 2007.

[28] Oualid Missaoui. GENERALIZED MULTI-STREAM HIDDEN MARKOV MODELS [M]. Louisville: Proquest, 2011, 45-74.

[29] Alberto Pepe, Johan Bollen. Between conjecture and memento: shaping a collective emotional perception of the future [C]. AAAI, Chicago, 2008.

[30] Norcross JC, Guadagnoli E, Prochaska JO. Factor structure of the Profile of Mood States (POMS): two partial replications [J]. J Clin Psychol. 1984. 40(5):1270-1277.

[31] Matthew Brand. Coupled hidden Markov models for modeling interacting processes [R].1997, 6, 3.

[32] Zoubin Ghahramani, Michael I. Jordan. Factorial Hidden Markov Models [J]. Machine Learning. 1997. 29(2-3): 245-273.

[33] Lawrence K. Saul and Michael I. Jordan. Mixed Memory Markov Models: Decomposing Complex Stochastic Processes as Mixtures of Simpler Ones [J]. Machine Learning. 1999. 37(1): 75-87.

[34] Shai Fine, Yoram Singer. The Hierarchical Hidden Markov Model [J]. Machine Learning. 1998. 32: 41-62.

附录

1．数据获取模块

（1）金融数据接收代码片段

```
url = "http://ichart.finance.yahoo.com/table.csv?s=" + symbol + "&a="
        + (startmonth - 1) + "&b=" + startday + "&c=" + startyear
        + "&d=" + (endmonth - 1) + "&e=" + endday + "&f=" + endyear
        + "&g=" + level + "&ignore=.cvs";
```

（2）Twitter 数据接收代码片段

```
return Status.constructStatuses(get(baseURL +
        "statuses/public_timeline.xml", false), this);
```

（3）经 WordNet 扩充后的 POMS Bipolar 极性词表

seren	1	A	anxiou	0	A	smolder	0	B
clear	1	A	queasi	0	A	smoulder	0	B
shaki	0	A	unquiet	0	A	wrath	0	B
unsteadi	0	A	troubl	0	A	wroth	0	B
compos	1	A	relax	1	A	wroth	0	B
unagit	1	A	degag	1	A	friendli	1	B
tranquil	1	A	laid-back	1	A	affabl	1	B
imperturb	1	A	mellow	1	A	amiabl	1	B
unflapp	1	A	unstrain	1	A	cordial	1	B
collect	1	A	uneasi	0	A	genial	1	B
equanim	1	A	untroubl	1	A	chummi	1	B
pois	1	A	secur	1	A	matei	1	B
self-collect	1	A	unafraid	1	A	palli	1	B
self-contain	1	A	nervou	0	A	palsi-walsi	1	B
self-possess	1	A	aflutt	0	A	companion	1	B
cool	1	A	excit	0	A	comrad	1	B
coolhead	1	A	peac	1	A	hail-fellow	1	B
nerveless	1	A	halcyon	1	A	hail-fellow-well-met	1	B
unflurri	1	A	iren	1	A	couthi	1	B
unflust	1	A	pacif	1	A	couthi	1	B
unperturb	1	A	peaceabl	1	A	cozi	1	B
unruffl	1	A	pacifist	1	A	intim	1	B
tens	0	A	pacifist	1	A	inform	1	B
arous	0	A	dovish	1	A	neighborli	1	B
wound up	0	A	jitteri	0	A	neighbourli	1	B
suspens	0	A	angri	0	B	social	1	B
suspens	0	A	rage	0	B	well-dispos	1	B
cliff-hang	0	A	tempestu	0	B	well dispos	1	B
nail-bite	0	A	wild	0	B	amic	1	B
taut	0	A	stormi	0	B	grouchi	0	B

antsi	0	A	aggrav	0	B	crab	0	B
fidgeti	0	A	provok	0	B	crabbi	0	B
fret	0	A	anger	0	B	cross	0	B
itchi	0	A	enrag	0	B	fussi	0	B
edgi	0	A	infuri	0	B	grumpi	0	B
high-strung	0	A	madden	0	B	bad-temper	0	B
highli strung	0	A	black	0	B	ill-temper	0	B
jumpi	0	A	choler	0	B	ill-natur	0	B
nervi	0	A	irasc	0	B	kindli	1	B
overstrung	0	A	hot under the collar	0	B	charit	1	B
restiv	0	A	huffi	0	B	benevol	1	B
uptight	0	A	mad	0	B	good-heart	1	B
electr	0	A	sore	0	B	openheart	1	B
strain	0	A	indign	0	B	larg-heart	1	B
pump	0	A	incens	0	B	kind	1	B
pump-up	0	A	outrag	0	B	benign	1	B
pump up	0	A	umbrag	0	B	benign	1	B
wire	0	A	irat	0	B	furiou	0	B
unrelax	0	A	ir	0	B	feroci	0	B
calm	1	A	livid	0	B	fierc	0	B
savag	0	B	debonair	1	C	grim	0	C
violent	0	B	jaunti	1	C	blue	0	C
sympathet	1	B	pollyannaish	1	C	depress	0	C
commis	1	B	upbeat	1	C	dispirit	0	C
condol	1	B	optimist	1	C	down	0	C
empath	1	B	sad	0	C	downcast	0	C
empathet	1	B	bittersweet	0	C	down in the mouth	0	C
bad temper	0	B	dole	0	C	low	0	C
agreeabl	1	B	mourn	0	C	low-spirit	0	C
mad	0	B	heavyheart	0	C	glum	0	C
huffi	0	B	melancholi	0	C	lonesom	0	C
sore	0	B	melanchol	0	C	lightheart	1	C
good-natur	1	B	pensiv	0	C	lone	0	C
good-humor	1	B	wist	0	C	alon	0	C
good-humour	1	B	tragic	0	C	lone	0	C
annoi	0	B	tragic	0	C	lone	0	C
irrit	0	B	tragicom	0	C	solitari	0	C
mif	0	B	tragicom	0	C	unaccompani	0	C
nettl	0	B	play	1	C	joy	1	C
peev	0	B	coltish	1	C	gleeful	1	C
piss	0	B	frolicsom	1	C	jubil	1	C
piss off	0	B	frolicki	1	C	joyou	1	C
rile	0	B	rollick	1	C	downheart	0	C
roil	0	B	sportiv	1	C	jolli	1	C

steam	0	B	devilish	1	C	jocund	1	C
stung	0	B	rascal	1	C	jovial	1	C
displeas	0	B	roguish	1	C	merri	1	C
affection	1	B	elfin	1	C	mirth	1	C
fond	1	B	elfish	1	C	discourag	0	C
lovesom	1	B	elvish	1	C	demor	0	C
tender	1	B	arch	1	C	demoralis	0	C
love	1	B	impish	1	C	dishearten	0	C
cheer	1	C	implik	1	C	pessimist	0	C
beam	1	C	mischevi	1	C	elat	1	C
glad	1	C	pixil	1	C	exult	1	C
beamish	1	C	prankish	1	C	exult	1	C
smile	1	C	puckish	1	C	pride	1	C
twinkli	1	C	wick	1	C	rejoic	1	C
blith	1	C	kittenish	1	C	triumphal	1	C
blithesom	1	C	friski	1	C	triumphant	1	C
lightsom	1	C	ludic	1	C	gladden	1	C
light-heart	1	C	mock	1	C	exhilar	1	C
buoyant	1	C	teas	1	C	high	1	C
chirpi	1	C	quizzic	1	C	in high-spirit	1	C
perki	1	C	deject	0	C	sublim	1	C
cheeri	1	C	amort	0	C	uplift	1	C
gai	1	C	chapfallen	0	C	gloomi	0	C
sunni	1	C	chopfallen	0	C	weak	0	D
chipper	1	C	crestfallen	0	C	fallibl	0	D
debonair	1	C	deflat	0	C	frail	0	D
imperfect	0	D	bold	1	D	impetu	1	D
weak	0	D	fearless	1	D	sharp	1	D
decrepit	0	D	dare	1	D	assert	1	D
debil	0	D	audaci	1	D	self-assert	1	D
feebl	0	D	brave	1	D	cocki	1	D
infirm	0	D	dauntless	1	D	uncertain	0	D
ricketi	0	D	hardi	1	D	confid	1	D
sapless	0	D	intrepid	1	D	assur	1	D
weakli	0	D	unfear	1	D	cocksur	1	D
strong	1	D	daredevil	1	D	overconfid	1	D
firm	1	D	temerari	1	D	posit	1	D
forc	1	D	embolden	1	D	reassur	1	D
stiff	1	D	foolhardi	1	D	self-confid	1	D
vehement	1	D	headi	1	D	surefoot	1	D
beardown	1	D	rash	1	D	sure-foot	1	D
beef-up	1	D	reckless	1	D	capabl	1	D
brawni	1	D	heroic	1	D	inadequ	0	D
hefti	1	D	heroic	1	D	incap	0	D

muscular	1	D	nervi	1	D	incompet	0	D
power	1	D	overreach	1	D	unequ to	0	D
sinewi	1	D	vault	1	D	defici	0	D
bullneck	1	D	overvali	1	D	lack	0	D
bullocki	1	D	unsur	0	D	want	0	D
fortifi	1	D	incertain	0	D	self-assur	1	D
hard	1	D	ambival	0	D	energet	0	E
knockout	1	D	doubt	0	D	physic	0	E
sever	1	D	dubiou	0	D	brisk	0	E
ironlik	1	D	grope	0	D	merri	0	E
robust	1	D	power	1	D	rattl	0	E
secur	1	D	almighti	1	D	snappi	0	E
unassail	1	D	all-power	1	D	spank	0	E
unattack	1	D	omnipot	1	D	zippi	0	E
invulner	1	D	mighti	1	D	canti	0	E
sure	1	D	muscular	1	D	drive	0	E
potent	1	D	puissant	1	D	high-energi	0	E
timid	0	D	rule	1	D	indefatig	0	E
bash	0	D	reign	1	D	tireless	0	E
coi	0	D	regnant	1	D	unflag	0	E
timor	0	D	regent	1	D	unweari	0	E
trepid	0	D	influenti	1	D	exhaust	1	E
intimid	0	D	herculean	1	D	dog-tire	1	E
mousi	0	D	superhuman	1	D	fag	1	E
mousei	0	D	self-doubt	0	D	plai out	1	E
diffid	0	D	forc	1	D	spent	1	E
shy	0	D	bruis	1	D	wash-out	1	E
faint	0	D	emphat	1	D	worn-out	1	E
faintheart	0	D	exclamatori	1	D	worn out	1	E
faint-heart	0	D	forcibl	1	D	gone	1	E
cowardli	0	D	physic	1	D	spent	1	E
fear	0	D	impel	1	D	drain	1	E
activ	0	E	bounci	0	E	vigor	0	E
combat-readi	0	E	peppi	0	E	fatigu	1	E
fight	0	E	spirit	0	E	attent	0	F
activist	0	E	zippi	0	E	captiv	0	F
activist	0	E	breezi	0	E	absorb	0	F
hand-on	0	E	bubbl	0	E	engross	0	F
proactiv	0	E	bubbl	0	E	enwrap	0	F
sporti	0	E	effervesc	0	E	intent	0	F
on the go	0	E	frothi	0	E	wrap	0	F
hyperact	0	E	scintil	0	E	advert	0	F
overact	0	E	sparkli	0	E	heed	0	F
hot	0	E	live	0	E	observ	0	F

agil	0	E	warm	0	E	oversolicit	0	F
nimbl	0	E	raci	0	E	solicit	0	F
quick	0	E	alert	0	E	thought	0	F
spry	0	E	resili	0	E	pai attent	0	F
acrobat	0	E	springi	0	E	perplex	1	F
athlet	0	E	elast	0	E	at a loss	1	F
gymnast	0	E	vital	0	E	nonplus	1	F
about	0	E	anim	0	E	nonpluss	1	F
astir	0	E	tire	1	E	puzzl	1	F
particip	0	E	all in	1	E	baffl	1	F
involv	0	E	beat	1	E	befuddl	1	F
bustl	0	E	bush	1	E	bemus	1	F
busi	0	E	dead	1	E	confound	1	F
go	0	E	bleari	1	E	lost	1	F
open	0	E	blear	1	E	maze	1	F
springi	0	E	bleari-ei	1	E	at sea	1	F
sluggish	1	E	blear-ei	1	E	metagrobol	1	F
sulki	1	E	bore	1	E	metagrobolis	1	F
slow	1	E	world-weari	1	E	metagrabol	1	F
dull	1	E	burn-out	1	E	metagrabolis	1	F
inact	1	E	burnt-out	1	E	mystifi	1	F
inert	1	E	careworn	1	E	stuck	1	F
soggi	1	E	drawn	1	E	abl to concentr	0	F
torpid	1	E	haggard	1	E	muddl	1	F
readi-to-go	0	E	raddl	1	E	addl	1	F
weari	1	E	worn	1	E	muzzi	1	F
aweari	1	E	droop	1	E	woolli	1	F
full of pep	0	E	flag	1	E	wooli	1	F
drowsi	1	E	footsor	1	E	woolli-head	1	F
drows	1	E	jade	1	E	wooli-mind	1	F
dozi	1	E	weari	1	E	businesslik	0	F
asleep	1	E	knacker	1	E	effici	0	F
inattent	1	E	drain	1	E	earnest	0	F
oscit	1	E	rag	1	E	purpos	0	F
yawn	1	E	travel-worn	1	E	daze	1	F
live	0	E	unrefresh	1	E	stun	1	F
aliv	0	E	unrest	1	E	stupefi	1	F
bounc	0	E	whack	1	E	stupid	1	F
foggi	1	F	befog	1	F	effici	0	F
groggi	1	F	cloud	1	F	businesslik	0	F
logi	1	F	dazzl	1	F	cost-effici	0	F
stupor	1	F	trancelik	1	F	cost-effect	0	F
letharg	1	F	punch-drunk	1	F	econom	0	F
unenerget	1	F	silli	1	F	econom	0	F

mental alert	0	F	slaphappi	1	F	expediti	0	F
confus	1	F	space-out	1	F	high-octan	0	F
disori	1	F	clearhead	0	F	streamlin	0	F
addlebrain	1	F	clear-think	0	F	effect	0	F
addlep	1	F	clear	0	F	compet	0	F
puddinghead	1	F	uncloud	0	F	bewild	1	F
muddlehead	1	F	mix-up	1	F			

2．预测模块

（1）训练代码片段

```
public void train(String trainPath, String labelPath)
{
    BaumWelchLearner bwl = new BaumWelchScaledLearner();
    MultistreamsHMM.trainSequences = generateSequences(trainPath);
    for (int i = 0; i < MultistreamsHMM.iterate; i++)
    {
        this.hmm = bwl.iterate(this.hmm,
            MultistreamsHMM.trainSequences);
    }
    ForwardBackwardCalculator fbc = new
        ForwardBackwardScaledCalculator(
            MultistreamsHMM.trainSequences.get(0), this.hmm,
            EnumSet.allOf(ForwardBackwardCalculator.
                Computation.class));
    double xi[][][] =
        bwl.estimateXi(MultistreamsHMM.trainSequences.get(0),
            fbc, this.hmm);
    double gamma[][] = bwl.estimateGamma(xi, fbc);
    MultistreamsHMM.trainLabel = generateLabelSeq(labelPath);
    this.label = generateLabel(this.hmm.nbStates(),
        MultistreamsHMM.trainLabel, gamma);
    this.visit = generateVisit(this.hmm.nbStates(),
        MultistreamsHMM.trainSequences.get(0), gamma);
    this.risk = generateRisk(this.hmm.nbStates(),
        MultistreamsHMM.trainLabel,
        MultistreamsHMM.trainSequences.get(0), gamma, this.label,
        this.visit);
    this.heavyState = generateHeavyState(this.risk, this.visit,
        MultistreamsHMM.rejectBound, MultistreamsHMM.visitBound);
    this.riskSet = generaterejectsubset(this.risk, this.visit,
        MultistreamsHMM.rejectBound, MultistreamsHMM.visitBound);
    // reference this
    MultistreamsHMM nowmultihmm = this;
    while (nowmultihmm.heavyState != -1)
    {
```

```
            nowmultihmm.nextmultihmm = RRtrain(nowmultihmm,
                MultistreamsHMM.trainSequences,
                MultistreamsHMM.refineStateNum);
            nowmultihmm = nowmultihmm.nextmultihmm;
        }
    }
```

（2）状态标注代码片段

```
    private static int[] generateLabel(int stateNum, int[] labelSeq,
            double gamma[][])
    {
        if (stateNum <= 0 || labelSeq.length <= 0)
        {
            System.out.println(stateNum);
            System.out.println(labelSeq.length);
            throw new IllegalArgumentException(
                "stateNum and must labelSeq.length be strictly
                positive");
        }
        if (labelSeq.length != gamma.length || stateNum !=
                gamma[0].length)
        {
            System.out.println(labelSeq.length + ":" + gamma.length);
            System.out.println(stateNum + ":" + gamma[0].length);
            throw new IllegalArgumentException("gamma is wrong");
        }
        int[] Li = new int[stateNum];
        for (int i = 0; i < stateNum; i++)
        {
            double up = 0.;
            double down = 0.;
            for (int t = 0; t < labelSeq.length; t++)
            {
                if (labelSeq[t] == 1)
                {
                    up += gamma[t][i];
                }
                else if (labelSeq[t] == -1)
                {
                    down += gamma[t][i];
                }
                else
                {
                    throw new IllegalArgumentException("label must be 1
                        or -1");
                }
```

```
            }
            if (up >= down)
            {
                Li[i] = 1;
            }
            else if (up < down)
            {
                Li[i] = -1;
            }
        }
        return Li;
    }
    private static int[] generateLabelRefined(int oldstate, int
        heavyState, int[] oldli, int stateNum)
    {
        if (oldstate <= 0 || heavyState < 0 || stateNum <= 0
                || oldli.length <= 0)
        {
            System.out.println(oldstate);
            System.out.println(heavyState);
            System.out.println(stateNum);
            System.out.println(oldli.length);
            throw new IllegalArgumentException(
                "Number of oldstate/heavyState/stateNum/oldli must be
                positive");
        }
        if (oldstate != oldli.length)
        {
            System.out.println(oldstate + ":" + oldli.length);
            throw new IllegalArgumentException("oldstate !=
                oldli.length");
        }
        int[] rr_Li = new int[oldstate - 1 + stateNum];
        int counter_i = 0;
        for (int i = 0; i < oldstate; i++)
        {
            if (i == heavyState)
                continue;
            rr_Li[counter_i] = oldli[i];
            counter_i++;
        }
        for (int i = oldstate - 1; i < oldstate - 1 + stateNum; i++)
        {
            rr_Li[i] = oldli[heavyState];
        }
```

```
        return rr_Li;
    }
```

（3）预测代码片段

```
    public void predict(String testPath)
    {
        MultistreamsHMM.testSequences = generateSequences(testPath);
        MultistreamsHMM.predict = new String[MultistreamsHMM.pass][];
        for (int p = 0; p < MultistreamsHMM.pass; p++)
        {
            MultistreamsHMM.predict[p] = new
            String[MultistreamsHMM.testSequences.size()];
            for (int i = 0; i < MultistreamsHMM.testSequences.size();
                i++)
            {
                // reference
                MultistreamsHMM nowmultihmm = this;
                BaumWelchLearner nowbwl = new BaumWelchScaledLearner();
                ForwardBackwardCalculator nowfbc = new
                    ForwardBackwardScaledCalculator(
                        MultistreamsHMM.testSequences.get(i),
                        nowmultihmm.hmm,
                        EnumSet.allOf(ForwardBackwardCalculator.
                            Computation.class));
                double[][][] nowXi = nowbwl.estimateXi(
                    MultistreamsHMM.testSequences.get(i), nowfbc,
                    nowmultihmm.hmm);
                double[][] nowGamma = nowbwl.estimateGamma(nowXi,
                    nowfbc);
                int nowmaxstate =
                    generateMaxState(nowGamma[MultistreamsHMM.
                        testSequences.size() - 1]);
                while (nowmaxstate == nowmultihmm.heavyState)
                {
                    nowmultihmm = nowmultihmm.nextmultihmm;
                    BaumWelchLearner nextbwl = new
                        BaumWelchScaledLearner();
                    ForwardBackwardCalculator nextfbc = new
                        ForwardBackwardScaledCalculator(
                            MultistreamsHMM.testSequences.get(i),
                            nowmultihmm.hmm,
                            EnumSet.allOf(ForwardBackwardCalculator.
                                Computation.class));
                    double[][][] nextXi = nextbwl.estimateXi(
                        MultistreamsHMM.testSequences.get(i), nextfbc,
                        nowmultihmm.hmm);
```

```
                double[][] nextGamma = nextbwl.estimateGamma(nextXi,
                    nextfbc);
                nowmaxstate = generatemaxrefinedstate(nextGamma[
                    MultistreamsHMM.testSequences.size() - 1],
                    this.nextmultihmm.hmm.nbStates()
                    - this.hmm.nbStates() + 1);
            }
            if (isrejectState(nowmultihmm.riskSet, nowmaxstate))
            {
                MultistreamsHMM.predict[p][i] = "reject";
            }
            else
            {
                MultistreamsHMM.predict[p][i] =
                    nowmultihmm.label[nowmaxstate]
                    + "";
            }
        }
    }
}
```

3．论文中涉及到的网站

Gallup Indexes. http://www.gallup.com/poll/122840/Gallup-Daily-Economic-Indexes.asp

Google Trends. https://www.google.com/trends

Twitter. http://www.twitter.com

Sentiment Analysis Wiki. http://en.wikipedia.org/wiki/Sentiment_analysis

WordNet. http://wordnet.princeton.edu/

Granger Causality. http://en.wikipedia.org/wiki/Granger_causality

Derwent Capital Markets Wiki. http://en.wikipedia.org/wiki/Derwent_Capital_Markets

Stockradar. http://stockradar.net/

Stocktwits. http://stocktwits.com/

Piqqem. http://piqqem.com/

Alex Davies Twitter Sentiment Analysis. http://alexdavies.net/twitter-sentiment-analysis/

Sentiment Symposium Tutorial. http://sentiment.christopherpotts.net/index.html

Google N-Gram. http://storage.googleapis.com/books/ngrams/books/datasetsv2.html

Stockcharts. http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators%5D

Xinqings search. http://xinqings.nlsde.buaa.edu.cn/

Pearson Wiki. http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient

致谢

衷心感谢论文指导老师周水庚教授和钱卫宁教授对我的悉心指导。周老师和钱老师的治学态度科学严谨，这篇论文得以最终完成，离不开周老师和钱老师的支持和帮助。他们求实创新的科研作风，深深感染了我，是我学习的榜样。

衷心感谢家人一直以来的关心和支持，您们推动着我前进的步伐。

衷心感谢软件学院的所有老师和同学，尤其是项目指导老师罗远哉副教授，蒋林华副教授和赵慧教授，以及项目成员同学庄涵和陈豪彦。您们让我的编程知识和科研能力得到了很大的提升，我将永远受用。

衷心感谢华东师范大学的培养，我希望将来能用学校学到的本领和知识回报社会，回报学校。