

毕业设计说明书

|  |  |  |  |
| --- | --- | --- | --- |
| **作 者:** | 王 凡 | **学 号：** | 913106840624 |
| **学 院:** | 计算机科学与工程 | | |
| **专业(方向):** | 软件工程（卓工） | | |
| **题 目:** | 云计算环境下具有可变方 | | |
|  | 差的虚拟机价格预测方法 | | |

**指导者：**

(姓 名) (专业技术职务)

(姓 名) (专业技术职务)

**评阅者：**

(姓 名) (专业技术职务)

年 月

（本页作为封二打印装订）

声 明

我声明，本毕业设计说明书及其研究工作和所取得的成果是本人在导师的指导下独立完成的。研究过程中利用的所有资料均已在参考文献中列出，其他人员或机构对本毕业设计工作做出的贡献也已在致谢部分说明。

本毕业设计说明书不涉及任何秘密，南京理工大学有权保存其电子和纸质文档，可以借阅或网上公布其部分或全部内容，可以向有关部门或机构送交并授权保存、借阅或网上公布其部分或全部内容。

学生签名：

年 月 日

指导教师签名：

年 月 日

毕业设计说明书中文摘要

|  |
| --- |
| （空1行）  竞价型虚拟机实例的价格作为一种随时间变化的随机变量，在统计学科中我们将其称为时间序列。在实际问题中，很多的序列方差并非是一个常数，而是一个变量，相应的，方差值的时间序列就构成了异方差时间序列。  异方差时间序列为竞价型虚拟机实例价格的风险预测提供了很有价值的参考。本课题对Spot价格序列进行了特征分析，序列特征表明其符合广义自回归条件异方差模型（GARCH）的建模条件。基于该模型的数学描述及其推导的原理，我们提出了算法的总体思路，并在数学模型的基础上展开，梳理了该模型的建立过程。随后利用统计分析的数学工具推导并论证了模型的预测方法，得出递推公式。在递推公式的基础之上，我们设计了预测算法，然后给出了算法的多种描述。利用推导出的算法，即可设计相应的预测软件。  为了更好的分析竞价型虚拟机实例的价格的波动率，我们选取了R语言和Java语言作为实验研究的手段。本课题利用这两种语言开发了可视化程序对我们推导的算法进行了数据验证。模型的相关参数选取需由实验过程来确定，我们设计了多种校验方法，根据模型校验的结果来调整软件的参数。  最后，本课题对大量的测试实例进行了算法验证。对于得到的实验结果，利用R语言的统计工具对数据进行了理论分析。  （空2行）  关键词 竞价实例 异方差 自回归模型 时间序列 向前预测 |

毕业设计说明书外文摘要

|  |
| --- |
| **Title**  Virtual Machine Price Forecasting Method  with Variable Variance in Cloud Computing Environment  **Abstract**  The price of Spot instances is a random variable that changes over time, which can be modeled as a time series. In practical problems, a lot of sequence variance of the time series is not a constant, but a variable.  Heteroscedastic time series provides a valuable reference for the price risk prediction of Spot instances. This paper analyzes the price of Spot instances series, which is in line with the conditions of GARCH. Based on the principle of the model, we propose the idea of the algorithm. And we simulate the model building process. We use the mathematical tools of statistical analysis to derive the prediction method. On the basis of the formula, we design a prediction algorithm. Using this algorithm, we can develop the corresponding software.  In order to analyze the variance of the price of Spot instances, we used R and JAVA to write the software. We verify the algorithm by using this software. We design a variety of calibration methods which adjusting the parameters of the software according to the results of model calibration.  Finally, a large number of test examples are verified. For the results, we used R to analyze them.  （空2行）  Keywords Heteroscedasticity AutoRegressive Time series Forward forecast |

（空2行）

目 次

1. 引言 1

1.1 竞价型虚拟机实例背景描述 1

1.2 已有技术及相关工作 1

1.3 论文章节安排 3

2. 课题目标 4

2.1 待解决问题描述 4

2.2 Spot价格特征分析 4

3. 提出的预测算法 6

3.1 预备知识 6

3.1.1 广义自回归条件异方差模型 6

3.2 提出基于异方差模型的预测方法 10

3.2.1 算法总体思想 10

3.2.2 预处理 10

3.2.3 模型建立 10

3.2.4 基于模型预测 12

3.2.5 算法描述 14

4. 实验分析 17

4.1 测试实例说明 17

4.2 参数设置说明 17

4.3 实验环境（JAVA+R） 18

4.4 实验结果分析 24

注：1. 目次中的内容一般列出“章”、“节”二级标题即可；

2．X、Y表示具体的阿拉伯数字；

3．**页眉**中的页码用罗马数字（Ⅰ、Ⅱ、Ⅲ、Ⅳ、Ⅴ……）表示。

（为保证双面打印效果，目次部分页数应是偶数。如果不够，加一个空页。）

# 1. 引言

## 1.1 竞价型虚拟机实例背景描述

近十年来，伴随着移动互联网的兴起，云计算资源的需求激增，越来越多的云计算资源服务提供商开始提供各种类型的云计算服务。国际上较知名的有微软的Azure、亚马逊的AWS等[4]。而在国内，阿里巴巴的ECS、腾讯公司的CVM以及新浪的SAE等虚拟机服务则占据了大片市场，成为许多个人开发者不可或缺的工具，为数以万计的人们提供云计算服务[3]。

虚拟机实例作为一种特殊商品，其价格并不是一尘不变的。随着时间变化，虚拟机实例的价格也在一定范围内产生波动。而消费者需要权衡它的价格、性能和稳定性之间的各种关系，选择适合的解决方案[4]。通过对实例历史价格的统计分析，我们期望找出其变化规律，以便我们可以为解决方案提供技术参考。随着越来越多的虚拟机商业化，对实例价格走势的分析显得愈加重要和有意义。通过对大量数据的建模分析，虚拟机实例提供商可以更加合理的调控价格，实现利益最大化。而消费者也迫切需要衡量价格、性能和可靠性之间的各种关系，通过统计分析，我们也希望能找到其平衡点[5]。

## 1.2 已有技术及相关工作

虚拟机实例的价格作为一种随时间变化的随机变量，在统计学科中我们将其称为时间序列。时间序列是统计学科的一个重要分支。其主要研究随着时间变化的事物发生、发展的过程，寻找事物发展变化的规律，并预测未来的走势[8]。

当我们得到一个虚拟机实例的历史价格即观察值序列之后，首先需要对观察值序列进行预处理操作。预处理操作即对平稳性和纯随机性进行一定的检验。根据检验结果，可以将序列分为不同的类型，对不同的类型我们采取不同的分析方法[8]。其中，平稳性是时间序列所具有的一种统计特征，一般情况下有两种定义方式[25]。分别是“严平稳”和“宽平稳”[13]。所谓严平稳，就是一种条件较为苛刻的平稳性定义，其定义了只有当序列的所有统计性质都不会随着时间的推移而发生变化时，该序列才能被认为是平稳的。但是，因为在实践中获得随机序列的联合分布是非常困难的，而且即使知道随机序列的联合分布，计算和应用也相当的不便。所以严平稳时间序列通常只具有理论意义，在实践中用的更多的是条件较为宽松的宽平稳时间序列。宽平稳是使用序列的特征统计量来定义的一种平稳性。它认为序列的统计性质主要由它的低阶矩决定，所以只要保证序列低阶矩（二阶）平稳，就能保证序列的主要性质近似稳定[13]。

综上所述，当我们得到虚拟机实例的价格观察值序列之后，首先判断其平稳性。通过平稳性检验，可以把价格观察值序列分为平稳序列和非平稳序列两大类。

对于平稳序列，在统计上，我们通常建立一个线性模型来拟合该时间序列的发展情况，借此来提取其中有用的信息[2]。目前最常用的平稳序列拟合模型是ARMA模型。ARMA模型的全称是自回归移动平均模型，它又可以细分为AR模型、MA模型和ARMA模型三大类[13]。

对于非平稳序列，又可以分为确定性时序分析和随机时序分析两大类[13]。确定性时序分析虽然是非平稳的，但是这些时间序列具有非常显著的趋势，我们可以分解并找到这种趋势，并利用这种趋势对序列的未来发展做出合理的预测[6]。常用的时间序列分解方法有Wold分解定理和Cramer分解定理，而趋势分析方法则有趋势拟合法和平滑法等[13]。但是，就虚拟机实例价格分析这一课题而言，我们分析之后发现它并不适合这种情况。确定性分解方式尽管简单，但它仅适用于非常确定的序列信息，对具有随机性的信息则不适用。而我们知道，虚拟机实例价格的时间序列多是由随机性信息组成的，因而并不满足假设。此外，确定性因素分解方法把所有序列的变化都归结为四因素的综合影响，却始终无法提供明确的方法判断各大因素之间确切的作用关系[6]。这些问题显示出确定性因素分解方法在分析虚拟机实例价格方面的不足。而对于非平稳序列的随机分析就是为了弥补上述不足，提供更为精确的时序分析。

非平稳序列的随机分析又涵盖了多种分析行为，本课题所研究的便是其中的关于异方差的建模预测。针对模型预测的研究已经有一套较为成熟的理论。但对波动率预测效果的研究较少[24]。查阅相关文献可以知道此项研究已有的成果，对于波动率的探究虽然少，但涵盖了多个学科。

如在数量经济学和金融学方面，异方差模型预测多用于股市价格的实证分析。吉林大学分析了比较了异方差类模型的模型特征[15]。在对模型特征分析后，选取GARCH模型对模拟的股市价格进行了异方差预测的数据验证。山东大学则对上证指数收益率效应做了分析和检验[18]。在研究上证指数的异方差行为模式后，观察到了序列尖峰厚尾和聚类特征。并成功对股市价格实例建模，通过真实数据的训练和拟合，对未来的价格波动率做出了合理的预测。在波动率的相关研究上，自回归条件异方差模型是有强力的数学工具，可以帮助我们对非平稳序列进行有理有据的分析建模。

绝大多数的时间序列波动率应用都是对股市的研究。然而异方差的独特属性并不只能用来对股市价格进行建模。由于序列的波动率往往代表着序列的平缓度，所以异方差模型也可以用来度量风险，成为风险指标。浙江工商大学利用异方差的特性对原油价格市场进行了风险分析。将时间序列的异方差相关的模型指标引入原油价格市场后，他们对风险行为进行了量化。可以直观的了解到当前石油价格的风险[16]。从而对价格危机做出预警，有效的减少由于价格冲击带来的损失。

对于异方差的研究预测是确定投资组合、风险分析及防范、资产定价等很多金融问题的基础[22]。本课题对竞价型虚拟机实例所进行的研究，也是为了更好的节省成本，规避风险。竞价型实例具有廉价且价格多变的特点，而目前对竞价实例的波动率研究几乎没有，这就让本课题的研究显得更为迫切。结合计算机编程技术，本课题将对竞价型实例的异方差进行深入的分析，为云资源价格的风险防范提供有效的参考。

## 1.3 论文章节安排

本文正文部分除引言外，共分为三个部分。

第二章介绍了课题需要解决的问题，并对Spot价格序列进行了特征分析。本课题主要针对波动率即异方差序列进行研究，设计基于广义自回归条件异方差模型（GARCH）的预测方法，对Spot价格序列的波动率进行预测和数据验证。

第三章又分为两个模块。首先3.1节罗列了研究预测算法所需要的基础知识，并详细介绍了广义自回归条件异方差模型的数学模型及其推导的原理，为后续的预测分析提供了坚实的理论基础。其次，3.2节提出了算法的总体思路，并在数学模型的基础上展开，梳理了该模型的建立过程。随后，我们利用统计分析的数学工具推导论证了模型的预测方法，并得出递推公式。在递推公式的基础之上，我们设计了预测算法，然后给出了算法的多种描述。利用推导出的算法，即可设计相应的预测软件。

作为实验分析，第四章着重说明了软件开发相关的工作。首先该章节对测试实例的预处理进行了详细的解析，随后说明了模型参数选取需由实验过程来确定，根据模型校验的结果来调整软件的参数。4.3节是实验环境的介绍，这一节主要展示了本课题开发的可视化软件及其使用方法。最后，我们对大规模的测试实例进行了算法验证。针对实验结果，利用R的统计工具对数据进行了多种分析。

# 2. 课题目标

## 2.1 待解决问题描述

对于普通的统计分析，我们有一套非常成熟的建模方法。如AR模型、MA模型和ARMA模型等[8]。然而多数的分析方法对历史数据有较高的要求，它们普遍假设历史数据具有较强的规律性，通过寻找这些规律，利用相关数学模型得到预测数据。而我们知道，虚拟机实例价格受到非常多的因素的影响，并不能简单地认为它是有较强规律性的历史数据[5]。相反的，虚拟机实例价格存在非常强的随机性，这就使得常规的预测方式不再适用。为此，本课题希望通过利用非平稳时间序列的随机分析来对虚拟机实例价格进行预测。

本课题拟利用广义自回归条件异方差模型（GARCH）对虚拟机实例的价格序列进行建模，设计基于合适的的竞价型实例价格预测算法，开发软件界面，进行大规模数据验证。

## 2.2 Spot价格特征分析



图2.2.1

图2.2.1为c4.large-spotprice型虚拟机在美国东1c区windows实例在某一段时间的价格序列。此图可看出序列具有一定规律性，自然想到考虑使用趋势分析法进行合理预测。然而，同样是这个竞价实例，它在前一段时间区间的价格序列却如图2.2.2所示



图2.2.2

可看到在第80步附近，价格出现了一次猛烈的增长，以致其原本的数据在图中都看不清变化趋势。通过分析大量真实数据，发现这种随机的骤变在竞价实例的序列中较为常见。因此认为竞价实例价格存在非常强的随机性，这就使得常规的预测方式不再适用。而广义自回归条件异方差模型适用于此种随机序列，所以我们希望能利用广义自回归条件异方差模型对样本进行数学建模，以实现预测分析。

# 3. 提出的预测算法

## 3.1 预备知识

### 3.1.1 广义自回归条件异方差模型

据亚马逊的文档说明，竞价型实例是描述可据其本身需要启动任意数量的虚拟服务器、配置安全和网络以及管理存储。竞价型实例指定用户愿意为运行特定实例类型支付的最高小时价格。现货的价格随供需关系而上下浮动，但用户支付的价格不会超过用户所指定的最高价。

按照时间的先后顺序把随机事件变化发展的过程记录下来就构成了时间序列。在统计研究中，常用按时间排序的一局随机变量来表示一个随机事件的时间序列，简记为或。用表示该随机序列的n个有序观察值，称为序列长度为的观察值序列。

1982年经济学家恩格尔提出了自回归条件异方差模型（ARCH模型）。该模型假设在历史数据已知的情况下，零均值、纯随机残差序列具有异方差性，即

(式1)

在正态分布的假定之下，有

(式2)

而异方差则等价于残差的平方的均值

(式3)

使用残差平方序列的自相关系数，可以考察异方差函数的自相关性。

考察的结果有如下两种：

a).自相关系数恒为零，即

这说明异方差函数是纯随机的。此时历史数据对未来异方差的估计一点作用都没有，至今也没有有效方法提取其中的异方差信息，这种情况本课题不做讨论。

b).存在某个自相关系数不为零，即

这说明在残差平方序列中蕴藏着某种相关信息，可以通过构造适当模型提取这些相关信息，以获得到序列的异方差的波动的特征。

具有结构的模型称为q阶自回归条件异方差模型，即ARCH(q)。

ARCH模型则是利用了历史序列的信息，得到条件方差的信息

(式4)

得到一个观察值序列之后，通常首先提取序列的水平相关信息，然后分析残差序列中所蕴含的波动相关信息。将这二者综合起来才能得到较为完整和精确的分析结果。ARCH模型提取异方差中所蕴含的相关信息完整结构为

(式5)

式中，为的确定性信息拟合模型，。

信息提取后需要对序列进行拟合，而要拟合ARCH模型，需要进行ARCH检验。ARCH检验是一种特殊的异方差检验，它不仅要求序列具有异方差性，而且要求这种异方差性是由于某种自相关关系造成的，这种自相关关系可以用残差序列的自回归模型进行拟合。常用的两种ARCH检验统计方法是Portmanteau Q检验和LM检验。

a).Portmanteau Q检验

Portmanteau Q检验的基本推导思路是：如果残差序列方差非齐而且具有集群效应，那么残差平方序列一般来说就有自相关性。所以方差非齐检验可以转化成残差平方序列的自相关检验。

Portmanteau Q检验假设条件如下式所示

(式6)

用表示残差平方序列的延迟k阶自相关系数，则该假设条件可等价表达为

(式7)

Portmanteau Q检验统计量其实就是的LB统计量

(式8)

式中，n为观察序列长度，为残差序列延迟i阶自相关系数，有

(式9)

原假设成立时，Portmanteau Q统计量近似服从自由度为q-1的分布

(式10)

当检验统计量的P值小于显著水平α时，拒绝原假设，认为该序列方差非齐且具有自相关关系。

b).LM检验

LM检验全称为拉格朗日乘子检验。其构造思想为：如果残差序列方差非齐且具有集群效应，那么残差平方序列通常具有自相关性。就可以尝试通过自回归模型拟合残差平方序列

(式11)

于是方差齐性检验就可以转化为这个方程是否显著成立的检验。

拉格朗日乘子检验的假设条件为

对残差平方序列构造q阶自回归方程，假设条件等价为

(式12)

记总误差平方和为，自由度为。回归平方和为，自由度为q。其中，SSE为回归方程残差平方和，，自由度为。则LM检验统计量为

(式13)

原假设成立时，LM(q)近似服从自由度为的分布

(式14)

当LM(q)检验统计量的P值小于显著水平α时，拒绝原假设，认为该序列方差非齐，并且可以用q阶自回归模型拟合残差平方序列中的自相关关系。

通过相关的检验，我们可以分析虚拟机实例价格序列方差是否非齐。如果检验显示出序列方差非齐且具有显著自相关关系，则可以利用ARCH模型提取残差平方序列中所蕴含的相关信息。如果检验显示出高阶模型显著成立，即ARCH(q)模型的阶数q较大，则说明残差平方序列具有长期相关性，可采用GARCH(广义自回归条件异方差)模型进行拟合。

GARCH模型可以修正ARCH模型在异方差函数具有长期自相关性下产生的拟合误差。它的结构如下：

(式15)

式中,为的确定性信息拟合模型，。这个模型简记为GARCH(p,q)。

## 3.2 提出基于异方差模型的预测方法

### 3.2.1 算法总体思想

本小节主要内容是根据已知的数学知识，得出广义自回归条件异方差模型如何应用于竞价型实例的预测。算法的总体思想是利用上述的预备知识推导异方差预测的递推公式，根据得到的公式来预测未来的异方差。

### 3.2.2 预处理

建模所用的原始数据为json格式，记录了某分类的所有虚拟机的价格历史序列。对于单一区域，根据虚拟机安装的操作系统，又相应分为windows和linux/unix。这些原始数据，在模型拟合时需要分开处理，所以本课题编写了一些java方法对其进行了预处理，以c4.2xlarge-spotprice\_us-east为例，将其分为8个不同的json文件，对应的正好是4个区域2种操作系统。这些经过预处理的数据即为测试实例。预处理后的文件其包含了两个属性Timestamp和SpotPrice，分别存储时间戳和该时间戳对应的历史价格。对于模型所用的数据的详细说明见第4章。

在模型拟合时，原始数据的时间间隔为1分钟，而实际建模预测的过程中，由于竞价实例的价格并不会在短时间内出现大幅度的波动，训练数据的冗余度较高，且考虑到我们仅希望预测未来的异方差走势而不是精确值，所以在建模时为了提升预测的时间长度以更好的利用历史数据，每30分钟只取了最高的价格作为模型训练数据。这样的话，模型训练实际使用的是历史数据中每30分钟的最高价格，预测的也是未来每30分钟的最高价格。此举牺牲了一定的时间精确性，但更好的使用了历史数据，并可以预测未来较长时间的异方差，从而协助我们更好的分析。

### 3.2.3 模型建立

异方差模型的建立同绝大多数的时间序列分析一样，首先我们要检验的是原始数据的平稳性。根据平稳性的统计指标，我们再决定是否需要差分操作。随后我们建立一个均值方程。这个方程主要考察数据的相关性。如果此步骤显示相关性不显著，则要额外考虑消除线形依赖。这可以通过对异方差的序列建立一个ARMA（一种常用的计量经济模型）来实现。随后我们需要校验这个均值方程的残差是否满足模型的特性。如不满足，则要对参数进行调整。如果残差序列显示均值方程符合模型的对应效应，那么模型建立完成。



图3.2.3.1

最后，通过诸多指标（ACF等，详见第4章）对模型进行参数检验，以求得到最佳的模型参数。对于模型繁琐的建立过程，R语言提供了相应的库（TSA）供我们调用。这使得模型的建立在代码上显得较为简洁。

### 3.2.4 基于模型预测

在上述数学描述中，收益率的时间序列由下式给出

(式16)

(式17)

式中的𝜔、α均为待定参数，是零均值且有单位方差的新息，与是独立的，。具有单位方差，所以的条件方差为。

针对上述公示，经济学家恩格尔做了如下扩展

(式18)

式中的q就是指的ARCH模型的阶数。此公示经过一定的迭代既可用于ARCH模型的预测。1986年另有两位经济学家（Bollerslev和Taylor）对上式进行了进一步的完善，得到了新的混合模型预测方法GARCH(p,q)（广义自回归条件异方差模型）。其数学公示如下

(式19)

可以看到，这个新的公示中包含了条件方差的滞后项p，添加的p阶滞后项就是GARCH模型的阶数。在有些文献中GARCH(p,q)也写作GARCH(q,p)，相应的软件参数也会不同，为了统一描述，本课题中给出GARCH参数时均为GARCH(p,q)。

根据公式描述，进行换元操作可推导出向前预测h步的条件方差的递推公式，当h>p时，有

(式20)

一般的，对于任意的h，递推公式会显得较为复杂，因为

(式21)

其中

(式22)

并且

(式23)

可以看出，基于GARCH模型的数学描述，我们可以很方便的理解条件方差的计算过程。为简便公式的写法，以GARCH(1,1)为例，可以假设有n个观察值，并且

(式24)

由于条件方差的初始值为，如果模型满足平稳性，那么可令其等于，抑或用来替代。考察

(式25)

可以看出，条件波动率一步向前预测值表示收益率平方和期望条件波动率估计值的加权平均值，也就是长期方差。通过此方法，我们可以拟合出预期收益率的长期方差值。利用递推公式，可以通过代入模型拟合后的待定参数值𝜔、α和β的值对条件方差进行向前预测。理论上，如果模型满足平稳性假设，预测的条件方差伴随着预测步长的增加，会逐步趋近于长期方差。

基于递推预测算法，很自然的想到递归求解，可以看出，针对后一步向前预测，需要用到前一步的预测值。所以我们考虑向前预测步长h=1和h>1两种情况。当h>1时，作为普遍情况，利用

(式26)

进行预测，为便于软件建模计算，考虑

(式27)

预测步骤可写成如下形式

(式28)

上式为本课题软件建模的重要依据。其中𝜔、α和β参数由数据训练之后的模型给出。

同样的，当h=1时，即向前第1步预测时，我们用来替代其前一步的预测值，得到

(式29)

式中由模型的学习数据最后一步的值的平方来近似，则是模型学习数据的方差值。

至此，我们得到了向条件方差前一步预测、向前多步预测的理论方法。同时，我们也能利用式

(式30)

计算长期方差，进而对多步预测值进行一定的检验。

### 3.2.5 算法描述

基于前面的分析，我们设计了一套算法用来实现模型的向前预测。按照递推的思想，首先利用式29计算初始的向前预测值，随后根据式28，我们可以算出下一步的向前预测值。

显然，根据这个步骤，我们有递归和动态规划两种递推方式。分别对应Algorithm 1和Algorithm 2。按此算法，我们可以得到H步向前预测的异方差值。在编写软件进行算法验证时，我们给定一个数据框（R语言的一种数据结构），不断对其中的数据进行这样的操作，随着数据框的向前移动，我们就能得到未来的异方差时间序列。

值得特殊说明的是，下列算法描述中，入参部分模型参数数组a[3]，代表模型的三个待定参数𝜔、α和β。

Algorithm 1 : GARCH-PREDICT //递归描述

Input: 模型参数数组a[3], 预测步长H, 原始数据数组d[learnstep]

Begin:

初始化 P ← 0, RS ← 0;//P是临时变量，用来存储中间值

If(H>1)

H ← H-1;

P ← GARCH-PREDICT(a,H,d); //递归

RS ← a[1] + a[2] \* P + a[3] \* P;//利用式28计算后一步的异方差

Else // 第一步

RS ← a[1] + a[2] \* d[learnstep]^2 + a[3] \* sd(d)^2;//利用式29计算初始异方差近似值

If end;

return RS;

End

Algorithm 2 : GARCH-PREDICT-DYNAMIC //动态规划描述

Input: 模型参数数组a[3], 预测步长H, 原始数据数组d[learnstep]

Begin:

初始化 n ← 2, rs[H];

rs[1] ← a[1] + a[2] \* d[learnstep]^2 + a[3] \* sd(d)^2;//利用式29计算初始异方差近似值

while n<H do

rs[n] ← a[1] + a[2] \* rs[n-1] + a[3] \* rs[n-1];//利用式28计算后一步的异方差

n ← n+1;

return rs[H];

End

Algorithm 1介绍了递归思想的算法描述。求H步的异方差值需要用到H-1步的异方差值，利用式28逐步递归，直到H=1，这时利用式29计算模型的初始异方差值即可。Algorithm 2则和我们的推导过程十分类似，首先用式29和模型参数去计算异方差的初始值，然后利用初始的异方差求后一步的异方差，逐步向后，直到求得H步的异方差值。这两种描述在数学本质上是一致的。不过Algorithm 1的时间复杂度比Algorithm 2高很多，所以在实验分析时我们实际使用的是Algorithm 2的动态规划算法。

# 4. 实验分析

## 4.1 测试实例说明

本课题使用的测试实例是Amazon各类型EC2实例真实的价格数据。原始数据为json格式，记录了某分类的所有虚拟机的价格历史序列。如c4.2xlarge-spotprice\_us-east表示美国东部地区所有c4.2xlarge型虚拟机的历史价格，这其中包含了美国东部b区、东部c区、东部d区和东部e区的所有信息。对于单一区域，根据虚拟机安装的操作系统，又相应分为windows和linux/unix。

针对这些原始数据，明显需要分开处理，本课题编写了一些java方法对其进行了预处理，以c4.2xlarge-spotprice\_us-east为例，将其分为8个不同的json文件，对应的正好是4个区域2种操作系统。这些经过预处理的数据即为测试实例。

预处理后的json文件均命名为SpotPriceHistory，以便于R进行批量处理。其包含了两个属性Timestamp和SpotPrice，分别存储时间戳和该时间戳对应的历史价格。

时间戳间隔一般为1分钟，在实际建模预测的过程中，由于竞价实例的价格并不会在短时间内出现大幅度的波动，训练数据的冗余度较高，且考虑到我们仅希望预测未来的异方差走势而不是精确值，所以在R建模过程中，为了提升预测的时间长度以更好的利用历史数据，每30分钟只取了最高价格作为模型训练数据。这样的话，模型训练实际使用的是历史数据中每30分钟的最高价格，预测的也是未来每30分钟的最高价格。此举牺牲了一定的时间精确性，但更好的使用了历史数据，并可以预测未来较长时间的异方差。

## 4.2 参数设置说明

GARCH建模最重要的就是定阶，即确定GARCH模型中p和q的参数。通过以上分析，我们发现1小节中的每一组测试实例都有各自的特点，参数取值就显得尤为重要。本课题中参数的取值主要依据模型样本呈现出的相关性，参考引言中的文献说明我们知道，建模预测行为就是寻找并利用相关性的过程。我们有一系列成熟的数学工具用以分析GARCH过程相关性是否显著，不过其推导和计算相当繁琐，好在现代编程语言提供了诸多的便利。R中有许多函数可用以分析GARCH过程相关性。此课题通过调用R的一些分析函数作为定阶参考，为模型的选取提供了数据支持。

在相关性上，通过ACF、PACF及其扩展函数可以看到模型样本的自相关模式的分布图。函数ACF计算（并且通过默认绘图）对（可能是多变量）时间序列的自相关函数的估计。函数PACF计算（并通过默认绘图）一个（可能是多变量）时间序列的部分自相关函数的估计[14]。

利用上述函数，我们可以看到模型样本的相关性是否显著，对于相关性显著度不高的样本，建模过程可能需要较高的阶数，如果模型相关性不显著，则需要考虑利用显著的数据进行训练。换言之，如果ACF和PACF分析说明样本学习区间缺乏显著性，那么此时建模预测是及其困难的，应当考虑使用更多的数据进行建模。

预测之前需要对模型进行参数检验。如果模型是被正确识别的，那么它的残差应当是近似独立同分布的，因此模型残差图是模型参数确定的一个重要依据。如果残差图呈现出相关性，说明模型拟合不符。其中标准残差的定义为。此外新息的正态性也可以通过QQ图来检验。

另一个较为直观的参数是GARCH模型的AIC值，该值用以度量模型被正确识别的程度。理论上模型越准确，AIC值就越低，所以在保证条件方差非负的基本条件下p、q的取值应尽量使AIC值越低越好。

此外，参考文献中大量的GARCH建模经验可以发现，在GARCH模型建模时，如果阶数过高，会致使模型太过复杂，偏离实际问题。因此在将GARCH模型应用于真实数据时，应当考虑不能让p、q值过大。通常小于等于3即可。如果大于3的阶数仍不能正确识别模型，那么可认为拟合失败，样本参数不符合GARCH建模的基本假设。

## 4.3 实验环境（JAVA+R）

R语言内置多种统计学及数字分析功能，本课题开发了一个可视化的Java程序界面，通过Java加载测试数据，后台调用R语言库函数，进行数据建模和预测，返回计算结果，并在程序界面进行展示。



图4.3.1

R的功能可以通过安装包（Packages，用户撰写的功能）增强。因为S的一些缘由，R相较于其他的数学用编程语言或统计学软件工具有更强的面向对象功能。

Java调用R语言进行计算通常有两种方式：JRI 和 Rserve。本课题采用 Rserve作为数据同步工具。这是一个基于TCP/IP的服务器，通过二进制协议传输数据，可以提供远程连接，使得客户端语言能够调用R。目前Rserve作为一个package发布在CRAN上，可以直接使用install.packages(“Rserve”)进行安装。需要使用时在R控制台下加载该包，然后输入命令Rserve()，开启服务器，就可以供客户端调用。相较于JRI，Rserve的一大优势是跨平台。这里的跨平台不只是软件平台（操作系统），Rserve调用可以在多个硬件机器上实现。只要有网络通信，Java软件就能调用到R进行计算。

本课题开发的软件界面如图所示。

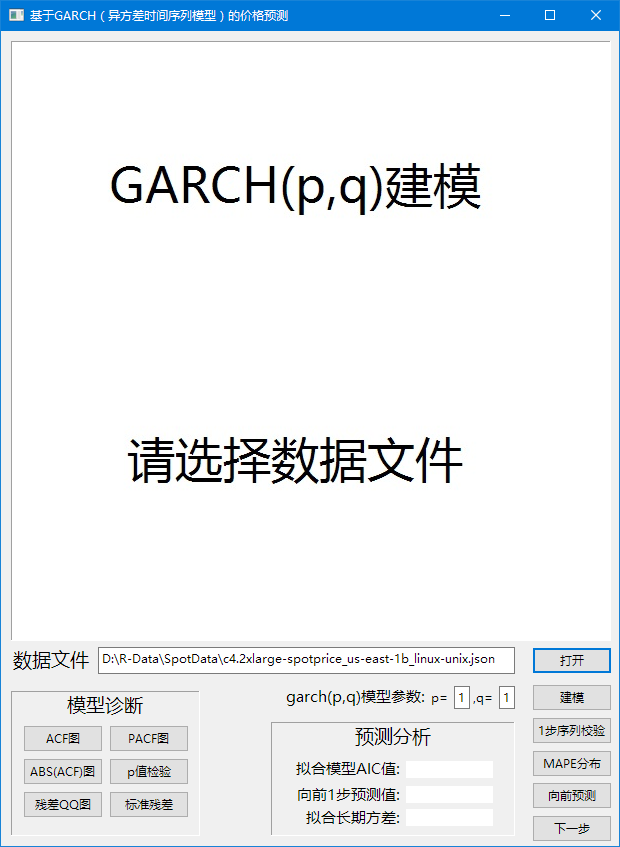


图4.3.2

选择测试用例后，设置p和q的参数并点击建模按钮进行软件建模。左下方的模型诊断区可查看模型是否正确识别，从而为数据参数（p值和q值）提供理论依据。

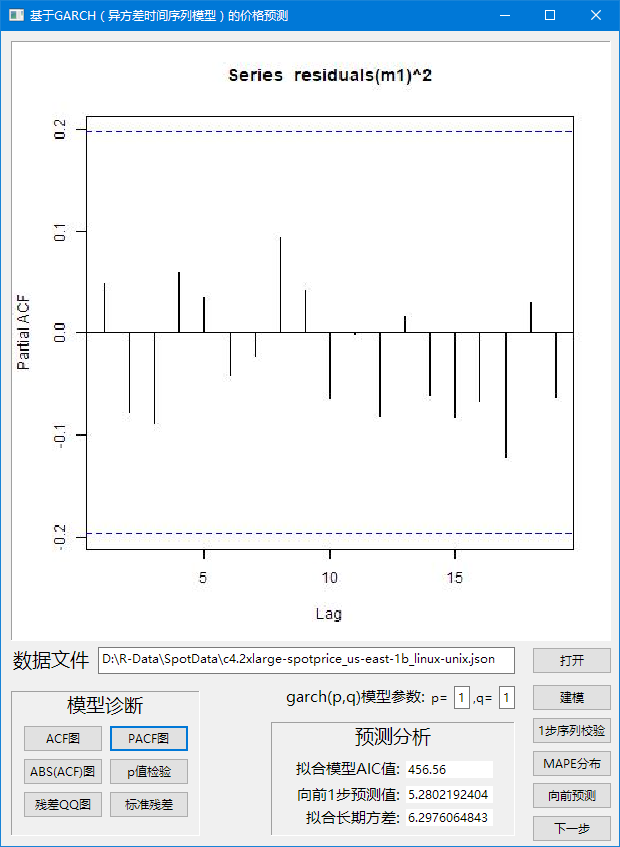


图4.3.3

例如上图的PACF图可以看出残差不相关性非常显著，模型拟合较好。

右下角的“向前预测”和“下一步”按钮可以模拟递推的过程（演示的这个过程是按步长进行的）。“1步序列校验”按钮可以利用模型对步长内的异方差序列进行预测，并和实际值比较。

值得注意的是，这里的实际值并不是真正意义上的“实际观测值”，它只是一个近似数据。我们知道，方差是一个过程值，即需要一段数据才能计算出来。而本课题所拟合预测的异方差走势，是异方差的时间序列，尽管异方差时间序列是离散的，但每个时间戳都有与之对应的异方差的值。而我们的真实数据不可能给出每个点的方差值，因为方差是根据一段数据所平均出来的。

为了解决这个问题，这里采用了两种方式表示“实际观测值”。其一是对观测数据取小段，分别计算它们的方差，得出“平均变化趋势”，与预测曲线进行对比。另一方法则是用收益率的绝对值来替代，其数学描述为

(式31)

统计分析中我们一般认为，上式可用于计算瞬时方差的近似值。

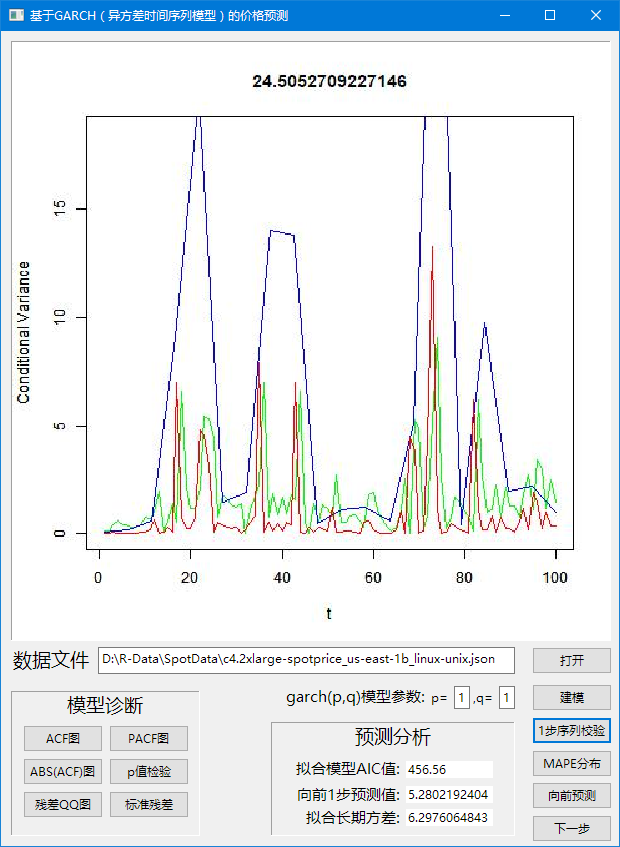


图4.3.4

图中红色曲线是通过模型预测出的异方差走势图，蓝色曲线是通过计算平均方差绘制的实际异方差大致走势图，绿色曲线则是用实际观测值通过上述公示计算出的收益率曲线。

“MAPE分布”按钮可以看到预测的准确度，值越小则越准确：

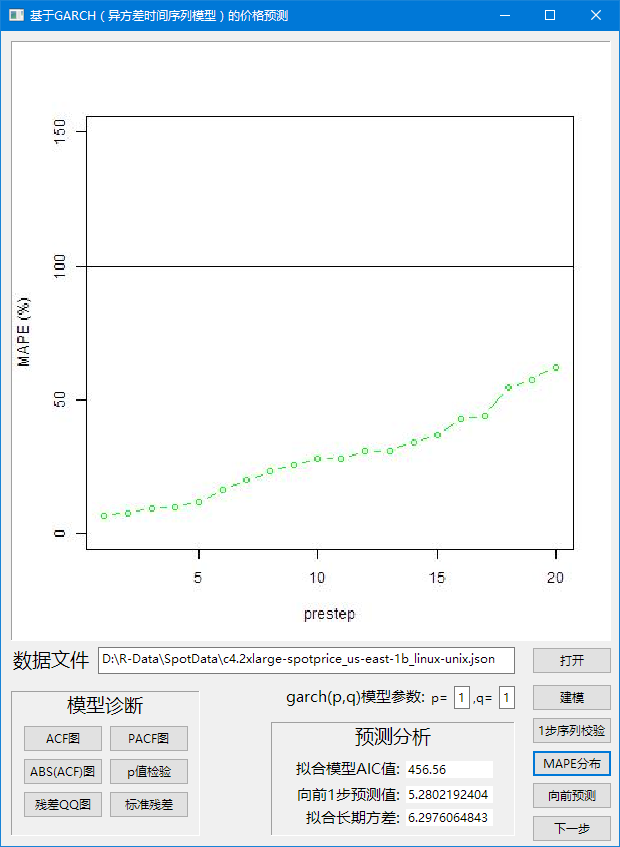


图4.3.5

可以看到预测（红线）和实际值（绿线）偏差伴随着时间呈现出更大的差距。

点击“向前预测”和“下一步”按钮，可以看到模拟的递推过程

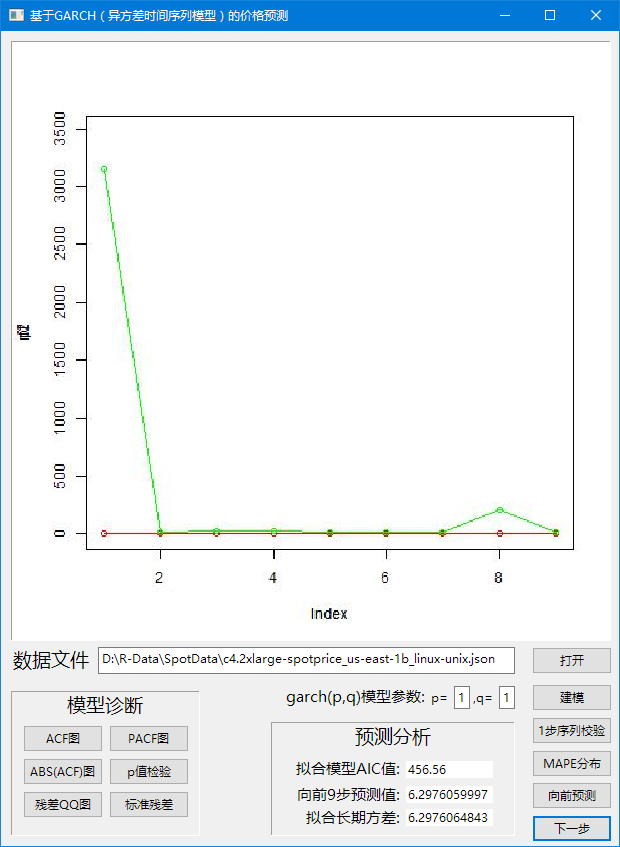


图4.3.6

正如前面我们所推断的，向前预测值会随着预测步数的增加而趋近于长期方差值。

此Java程序界面实现了数据的加载和预测结果的展示功能，并将计算生成的中间数据（分布json数据和图片文件）全部保存在主机上，以供后续分析。

## 4.4 实验结果分析

针对各型号虚拟机在不同地区的价格序列，本课题进行了大规模的数据分析和算法验证。

对于每个测试实例，我们在若干个不同的时间段取样，分别用广义自回归条件异方差模型进行建模，并用该时间段的序列值进行模型训练，然后利用得到的模型对未来时间段的异方差走势进行了预测。

预测准确度的衡量指标是MAPE值。对两个序列的相似程度有很多的衡量方式，我们采用了由accuracy函数计算的MAPE值作为指标。这个函数由R语言的forecast库提供，参数为两个不同的序列，返回值是一个数据框，其中的MAPE值就是我们所需要的。返回的MAPE值越小，说明两组序列的相似度越高。假设两组序列完全相同，那么返回的MAPE值等于零。一般情况下，当返回的MAPE值小于100时，就认为两组序列具有较高的相似性——这里的MAPE量纲是百分数，也就是说返回的MAPE等于1时，代表百分之一。值得我们特别注意的是，在很多的R语言库中都有名为accuracy的函数，它们所返回的MAPE值的量纲可能不同，需要格外关注。有些文献中提到，当MAPE小于1时可以认为两组序列相似，其实它的计算方式和我们所使用的accuracy函数差了一个百分号，并不代表MAPE值扩大了100倍。

综上所述MAPE可以体现某两组序列的相似程度，那么它就能成为我们分析算法准确度的衡量指标。对于预测的序列和真实的序列，我们二者按等比例进行分段，对每一段的两种序列，通过accuracy函数计算其MAPE值，就能得到MAPE序列。这是此节分析算法准确度的基本原理。

以c4.2xlarge-spotprice型虚拟机在美国东部1b区的linux/unix测试实例为例，我们分别在五个时间段对其进行了建模，并利用训练模型对建模点的未来50小时的异方差序列进行了预测。对于预测的未来序列，本课题所开发的软件将其存储到了json文件中。我们对预测序列进行了准确度分析。通过对预测序列的准确度分析，可以得到准确度随预测步数的变化，也就是预测准确度序列。

由于我们对c4.2xlarge-spotprice型虚拟机在美国东部1b区的linux/unix测试实例进行了五次建模和预测，所以可以得到五组预测准确度序列。针对这五组预测准确度序列，为了便于直观的分析，我们把它绘制成了对应的曲线。



图4.4.1

上图是c4.2xlarge-spotprice型虚拟机在美国东部1b区的linux/unix测试实例的五次建模预测准确度分析曲线。横坐标是预测长度——并非预测步长。事实上，本课题所涉及的实验结果分析均为一步向前预测的序列值，超过一步的向前预测由于缺少训练数据，会导致预测准确度有失偏颇。这里的prestep仅仅是过程为了方便MAPE的计算而分割的。此图中20步对应的是50小时。纵坐标代表MAPE值。

曲线的主干部分是五组预测准确度序列（MAPE序列）的平均值曲线。而每个点都有与之对应的上限和下限，这是通过五组数据所计算出的置信区间，本课题中计算所用的置信度均为0.9。我们观察曲线可以发现，随着预测时间的增加，模型所预测的异方差走势和测试实例的真实数据的偏差越来越大。在前半部分的MAPE值甚至不到20%，可见预测非常准确，而到了后半部分，预测偏差越来越大。在曲线末端MAPE值已经超过了60%，如果继续预测下去，后续的数据将不再可信。

事实上，c4.2xlarge-spotprice型虚拟机在美国东部1b区的linux/unix测试实例是所有实验数据中建模识别度最高的测试实例之一。



图4.4.2

上图是c4.2xlarge-spotprice型虚拟机在同一地区的windows测试实例。尽管此组测试实例建模识别度也很高，但是相较于linux/unix而言，预测偏差明显要高很多。在半步长（prestep=10）前，预测准确度还较高，MAPE偏差值保持在50%以内，但明显组的模型并没有组那么耐用，prestep>15的数据已经超过100%且还在极速上升，说明后续的预测值已经不再具有参考价值，故我没有在此图中绘制出。

通过比较c4.2xlarge-spotprice型虚拟机在美国东部1b区的linux/unix测试实例和windows测试实例，我们看出，linux/unix测试实例的广义自回归条件异方差模型能够的预测长度比windows测试实例的数据模型更长远。从初期来看，两个测试实例的模型都识别得很好，能够就向前一步的异方差数据进行比较合理的模拟预测。

然而windows测试实例所能坚持的时间并不长，我们接下来看看造成这种差异的原因。分析两个测试实例原始数据我们发现，由于虚拟机的价格变动一般较小，不利于我们观察分析，所以对两个实例的原始数据进行了差分化处理。原始价格序列经过差分后的某一段时间的曲线图如下



图4.4.3



图4.4.4

从图中看，linux/unix测试实例和windows测试实例有两个较为明显的差异，其一是linux/unix测试实例的纵坐标（经过差分运算的价格）比windows测试实例要大，差不多是后者的两倍；此外，windows测试实例在time值在介于60至90的区间时纵坐标的变化很小，近乎平缓。通过查看二者的原始数据我们发现，linux/unix测试实例比windows测试实例变化幅度更大、变化更为频繁。从两张图的直观感受中也可以看出，linux/unix测试实例比windows测试实例携带了更多的变化信息。这就是造成预测准确度差异的根本原因。

在相同数量的训练数据的前提下，linux/unix测试实例比windows测试实例更为复杂，体现出更多的变化特性。因此在我们使用广义自回归条件异方差模型对两组实例进行建模时，前者所得到的数据训练更为有效。换句话说，虽然linux/unix测试实例比windows测试实例的用以训练模型的数据是一样多的，但是模型在windows测试实例中训练不足，所以在用以预测时，没有达到前者那么好的效果。

通过分析c4.2xlarge-spotprice型虚拟机在美国东部1b区的linux/unix测试实例和windows测试实例的实验结果，我们可以明确，模型拟合程度与原始数据的变化率有一定的关系，而拟合更好的模型，在预测时可信的时间就越久。

在对大量测试实例进行上述分析后，我们将所有的MAPE序列绘制成了同样的曲线，但是这些曲线中有的并不满足我们的分析。除了因模型拟合失败而明显错误的数据曲线外，也有一些反常的曲线。



图4.4.5

上图的曲线如果仅从主干走势分析的话，可以看到模型拟合是比较准确的，符合我们的预期。但是这条曲线的问题在于MAPE取值的置信区间太大，MAPE走势非常不确定。为了探明造成这种现象的原因，我们查看了其MPAE的原始json数据。数据显示，前四组数据MAPE值趋势相似，且均保持在100以内，而第五组数据的MAPE数据如下表所示

表4.4.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 1 | 16.51316 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 2 | 20.19673 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 3 | 22.79461 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 4 | 23.49618 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 5 | 27.75276 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 6 | 29.4487 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 7 | 36.08702 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 8 | 43.29258 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 9 | 54.19224 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 10 | 55.60032 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 11 | 60.70397 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 12 | 65.92489 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 13 | 66.8607 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 14 | 67.05261 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 15 | 68.23473 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 16 | 84.35155 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 17 | 97.26842 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 18 | 99.77564 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 19 | 289.6269 |
| GARCH | c4.large-spotprice\_us-east-1e\_linux-unix | 4 | 20 | 317.0069 |

此组数据与前四组差距很大，但因为只占总数据的五分之一，对整体均值影响较小，但却导致了整体置信区间增大。也就是说，对于这组测试实例，我们取了五段不同的时间进行建模，前四段区间的建模识别程度都较好，而最后一组效果较差，从而影响了整体分析。查看第五组数据后发现其分布较为平缓，拟合模型训练不足，所以MAPE值增长较快。

通过分析实验结果我们发现，模型拟合识别度与实验参数的取值有很大的关系，而在向前预测操作时，影响我们预测准确度的往往是用来训练模型的原始数据。原始数据的数据复杂程度对模型训练的程度有着明显的差异。好的原始数据能够被广义自回归条件异方差模型高效地利用，从而提高预测的精度。而差的原始数据特别是那些“一尘不变”的数据，对异方差的预测训练作用欠佳。透过这些分析我们明确了模型的优势和劣势，从而能够对不同的数据采用不同的训练方式。

（空2行）

结 论（小三号黑体，居中）

×××××××××（小四号宋体，1.5倍行距，西文字符使用Times New Roman字体）×××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××……

（空2行）

致 谢（小三号黑体，居中）

×××××××××（小四号宋体，1.5倍行距，西文字符使用Times New Roman字体）×××××××××××××××××××××……

（空2行）

参 考 文 献（小三号黑体，居中）

[1] Galante G, Bona L C E D, Mury A R, et al. An Analysis of Public Clouds Elasticity in the Execution of Scientific Applications: a Survey[J]. Journal of Grid Computing, 2016, 14(2):193-216.

[2] Juve G, Chervenak A, Deelman E, et al. Characterizing and profiling scientific workflows[J]. Future Generation Computer Systems, 2013, 29(3):682-692.

[3] Ben-Yehuda O A, Ben-Yehuda M, Schuster A, et al. Deconstructing Amazon EC2 Spot Instance Pricing[C]// IEEE International Conference on Cloud Computing Technology and Science Proceedings. IEEE, 2011:304-311.

[4] Singh V K, Dutta K. Dynamic Price Prediction for Amazon Spot Instances[J]. 2015:1513-1520.

[5] Zhang Q, Zhu Q, Boutaba R. Dynamic Resource Allocation for Spot Markets in Cloud Computing Environments[C]. IEEE Computer Society, 2011:178-185.

[6] Dadashov E, Cetintemel U, Kraska T. Putting Analytics on the Spot: Or How to Lower the Cost for Analytics[J]. Internet Computing IEEE, 2014, 18(5):70-73.

[7] Lorido-Botran T, Miguel-Alonso J, Lozano J A. A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments[J]. Journal of Grid Computing, 2014, 12(4):559-592.

[8] Javadi B, Thulasiram R K, Buyya R. Statistical Modeling of Spot Instance Prices in Public Cloud Environments[C]// 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing. IEEE, 2011:219-228.

[9] Li X, Cai Z. Elastic Resource Provisioning for Cloud Workflow Applications[J]. IEEE Transactions on Automation Science & Engineering, 2015:1-16.

[10] 沈虹, 李小平. 带准备时间和截止期约束的云服务工作流调度算法[J]. 通信学报, 2015, 36(6):183-192.

[11] 刘少伟, 孔令梅, 任开军等. 云环境下优化科学工作流执行性能的两阶段数据放置与任务调度策略[J]. 计算机学报, 2011, 34(11):2121-2130.

[12] 陈冬林, 陈玲, 马明明 等. 云计算IaaS现货实例定价方法研究[J]. 计算机工程与设计, 2013, 34(10):3366-3370.

[13] 王燕. 时间序列分析——基于R[M]. 基于R应用的统计学丛书,2015,3:44-173.

[14] Hadley Wickham, Dirk Eddelbuettel, Yihui Xie. Rdocumentation[K] 2017 form https://www.rdocumentation.org/.

[15] 于洋. 非线性GARCH类模型及其在股票收益率波动性上的应用[D]. 吉林大学, 2007.

[16] 杨波. 基于GARCH类\_EVT模型的原油价格市场风险的度量[D]. 浙江工商大学, 2010.

[17] 张保. 基于GARCH模型的利率与股价指数收益率波动相关性研究[D]. 东北大学, 2012.

[18] 成城. 基于GARCH模型的上证指数波动率特征分析[D]. 山东大学, 2014.

[19] 葛敏霞. 基于GARCH族模型的VaR方法计算在证券市场的实证分析[D]. 湖北工业大学, 2016.

[20] 孙建武. 基于GARCH族模型的我国股指期货对股市波动影响分析[D]. 南京航空航天大学, 2013.

[21] 梁恒. 基于GARCH族模型的我国沪深股市波动非对称性研究[D]. 安徽大学, 2014.

[22] 徐斌. 基于改进的GARCH模型预测风险的研究[D]. 湖北工业大学, 2011.

[23] 焦鹏. 基于模糊GARCH模型的中国股票市场波动性研究[D]. 西南财经大学, 2011.

[24] 黄海南, 钟伟. GARCH类模型波动率预测评价[J]. 北京师范大学金融研究中心, 1003-207(2007)06-0013-07.

[25] 郭诗朦. 基于时间序列方法的网络时延预测与改进型广义预测控制算法在网络控制系统中的应用[D]. 北京交通大学, 2013.