软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(12):3208−3222 [doi: 10.13328/j.cnki.jos.005014]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于隐马尔可夫模型的软件状态评估预测方法*

吴 佳, 曾惟如, 陈瀚霖, 唐雪飞

(电子科技大学 信息与软件工程学院,四川 成都 610054)

通讯作者: 吴佳, E-mail: jiawu@uestc.edu.cn

摘 要: 随着软件系统功能和性能的强化和提高,企业的管理效率在不断提升,运营模式也越来越丰富.与此同时,软件系统变得越来越复杂,这向软件系统管理和维护提出了严峻的挑战.如何通过采集系统外部特征参数,对系统内部状态进行客观、准确地评估和预测,成为亟待解决的问题.为此,提出了一种基于隐马尔可夫模型的软件系统状态评估预测方法.该方法基于软件系统外在特征参数,通过 *K*-means 方法构建系统的观测状态,并以此建立隐马尔可夫模型,建立起系统外在状态(观测状态)和内部状态(隐藏状态)之间的联系;再利用三次指数平滑法对具有周期性变化的系统特征参数进行预测,即可预测系统未来状态.针对基于 B/S 软件架构的信息管理系统的实验,其结果表明该方法对系统状态评估和预测具有较高的准确性.

关键词: 状态预测;隐马尔可夫模型;*K*-meas;B/S 架构;三次指数平滑法

中图法分类号: TP316

中文引用格式: 吴佳,曾惟如,陈瀚霖,唐雪飞.基于隐马尔可夫模型的软件状态评估预测方法.软件学报,2016,27(12):3208−3222. http://www.jos.org.cn/1000-9825/5014.htm

英文引用格式: Wu J, Zeng WR, Chen HL, Tang XF. Approach of measuring and predicting software system state based on hidden Markov model. Ruan Jian Xue Bao/Journal of Software, 2016,27(12):3208−3222 (in Chinese). http://www.jos.org.cn/1000-9825/5014.htm

Approach of Measuring and Predicting Software System State Based on Hidden Markov Model

WU Jia, ZENG Wei-Ru, CHEN Han-Lin, TANG Xue-Fei

(School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China)

Abstract: With increased improvement on the capability and performance of software systems, enterprises have improved the management efficiency and enhanced the business model. Meanwhile, as software systems become more and more complex, severe challenges for the management of software systems are encountered. This paper presents a method for measuring and predicting software system state based on hidden Markov model. It establishes the linkage between the exterior state (the observation state) and the interior state (the hidden state) of the software system. *K*-means method is applied to construct the observation state of system. Triple order exponential smoothing is used to predict the future state of the system exterior state which changes cyclically. The experimental analysis on B/S information management system shows that the proposed method has high accuracy for measuring and predicting the software system state.

Key words: state predicting, hidden Markov model, *K*-means, browser/server, triple order exponential smoothing

* 基金项目: 国家自然科学基金(61503059); 四川省科技计划项目-科技支撑计划(2014GZ0019); 国家科技支撑计划(2012BAH44F02)

Foundation item: National Natural Science Foundation of China (61503059); Science and Technology Support Program of Sichuan Province-Key Technology Support Program (2014GZ0019); National Key Technology Research and Development Program of the Ministry of Science and Technology of China (2012BAH44F02)

收稿时间: 2015-08-24; 修改时间: 2015-09-23; 采用时间: 2015-12-22

随着 IT 行业的发展,大型软件管理系统发生了巨大的变化,其具有更好的开放性、更小的管理难度、更低的维护成本和更简易的操作性.在大型软件管理系统为企业自动化办公带来便利的同时,其固有的复杂结构也给管理者的监控工作带来了困扰;同时,系统一旦出现宕机等情况,其代价又是无法估量的.因此,准确评估并成功预测软件系统的状态就显得异常重要,这将有助于及早发现问题,及时规避风险,降低维护成本.

由于软件系统的复杂结构,管理者很难对系统内部实际状态进行准确地评估和预测.因为系统表露出来的是一些外在特征,所以我们只能通过这些系统外在特征来推测系统内部状态.当前最常见的软件状态监测的方法是阈值法,阈值法将系统部分的外在特征数据与预先设定阈值相比较,可以大致了解系统状态.该方法计算量小,但是阈值常常需要人为预先设定,因而自适应性差,主观性强.另外,如今大型软件系统大多由多个子系统构成,每个子系统又可由多个特征参数表征,仅凭部分参数无法准确地推断系统实际状态,因此,传统的办法有着很大的局限性.人工神经网络作为近年来的研究热点,已在多个领域表现出了良好的智能特性.神经网络采用一系列互连的神经元模拟了人脑的结构,具有较强的学习能力.它利用一组已知类别的样本调整分类器的参数,使其达到最佳的分类性能.但采用神经网络对软件系统状态进行评估缺点也很明显:首先,很难获得已知类别的样本,也就是说,采集得到的系统外在特征很难通过人工或者经验判断其所反映的系统真实内部状态;其次,神经网络无法建立起系统过去、现在和未来状态之间的联系.

针对当前方法的不足,本文提出了一种基于隐马尔可夫模型(Hidden Markov model,简称 HMM)的软件系统状态预测方法.该方法通过收集系统外在特征参数,利用隐马尔可夫模型建立系统内部状态与外部特征之间联系,实时了解并预测系统状态.首先收集系统运行时的特征参数,如数据库后台进程状态参数、各个服务器状态参数等.由于软件系统的复杂性,采集到的状态数据是高维的,而隐马尔可夫模型的输入数据要求是一维数据,因此需要对采集数据进行预处理.我们利用 *K*-means 聚类算法对历史数据进行聚类,以此作为划分系统观测状态的依据.接下来,通过三次指数平滑法预测系统特征参数,输入模型就可预测系统状态.

本方法主要优点在于:

(1) 提出了一种基于隐马尔可夫模型的软件系统状态评估和预测的方法,可通过系统外部特征参数准确地预测系统隐藏状态;

(2) 提出了一套合理的系统观测状态划分的方案;

(3) 使用三次指数平滑法实现了对随着时间序列呈周期性变化的系统参数的精确预测.

本文第 1 节讲述基于 B/S 信息管理系统的软件架构及系统外部特征相关问题.第 2 节描述如何对软件系统建立隐马尔可夫模型以及训练模型参数的过程.第 3 节简述系统状态预测的方法.第 4 节给出实验验证设计和结果.第 5 节总结全文.

1 B/S(Browser/Server)架构信息管理系统

由于数据库技术的日趋成熟,信息管理系统不断地被应用于多个领域,使我们的工作更加高效.信息管理系统采用的典型架构是 B/S 架构[1,2].该架构是在沿用动态网页技术的基础上,将信息管理系统软件的理念应用其中.B/S 架构的系统有着开放性好、管理难度小、维护简单和操作简易的优势,它能够更好地适应网络管理的需求,这也使得 B/S 架构的信息管理系统成为信息管理系统的首选系统.因此,本文将以 B/S 架构的信息管理系统为例,研究软件系统状态评估和预测的方法.

1.1 B/S信息管理系统的软件架构

B/S 架构即为浏览器和服务器架构,它是对传统 C/S(client/server)架构的一种改进.B/S 架构下将系统的核心部分放到了服务器上,客户端只需要安装一个浏览器.典型的 B/S 信息管理系统由 3 层结构组成:客户层、中间层和数据层(如图 1 所示).客户层即为浏览器或桌面窗口程序;中间层则是 Web 服务器即应用服务器;数据层包含两部分:数据库及数据库宿主服务器.

整个 B/S 架构的信息管理系统的状态由客户端(浏览器)的状态、Web 服务器(应用服务器)的状态、数据库及其宿主服务器的状态共同决定(如图 2 所示).其中,客户端即浏览器不包含在本文所探讨的系统中,所以不对

其健康状况进行讨论.以下将对 Web 服务器健康状况和数据库及其宿主服务器的健康状况进行探讨.

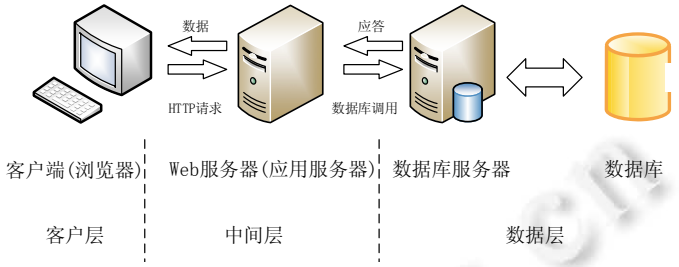


Fig.1 Three layers in B/S architecture

图 1 B/S 架构的 3 层结构

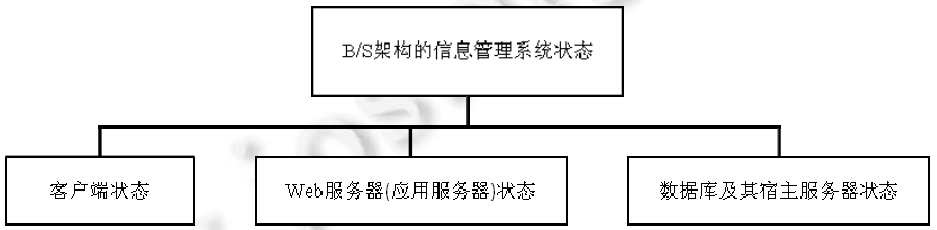


Fig.2 Modules of information management system based on B/S architecture

图 2 B/S 架构信息管理系统状态的组成

1.2 B/S架构系统的数据采集

在实现对系统状态评估预测之前,必须采集能够反映系统状态的相关特征参数.由于 B/S 架构软件系统状态主要由 Web 服务器和数据库服务器决定,因此我们只从这两大系统中采集相关数据.

1.2.1 Web 服务器数据采集

WebLogic[3,4]是一款用于开发、集成、部署和管理的大型分布式 Web、网络和数据库应用的 Java 应用服务器,它同时也将 Java 动态功能和 Java Enterprise 标准引入其中.WebLogic 服务器具有支持业内标准全面、可扩展性高和可靠性强等特点,使其被许多企业所选择.本文将以 WebLogic 服务器为例,对 Web 服务器健康指标进行评价,其他服务器可以类似评价.

WebLogic 服务器的特征分为硬件特征和软件特征.

- 硬件特征即为 WebLogic 服务器的硬件的状态,主要包括 CPU 状况、磁盘状况、内存状况、交换区状况和网络质量(见表 1),其中,网络质量又包含服务器接收和发送的错误率和丢包率;
- 软件特征则是反映 WebLogic 系统软件层面参数,主要包括 WebLogic Server 线程状态、Server 状态、WebLogic 内存状态和 JVM(Java virtual machine)堆的状态(见表 2).

Table 1 Hardware features of WebLogic server

表 1 WebLogic 服务器硬件特征数据采集

系统特征采集项目	具体采集指标
CPU	CPU 利用率
硬盘	磁盘利用率
内存	内存利用率
交换区	交换区利用率
网络质量 (包括 4 个状态参数)	WebLogic 数据接收错误率 WebLogic 数据接收丢包率 WebLogic 数据发送错误率 WebLogic 数据发送丢包率

Table 2 Features of WebLogic

表 2 WebLogic 服务器软件特征数据采集

系统特征采集项目	具体采集指标
WebLogic Server 线程 (包括 2 个状态参数)	WebLogic 线程总量 WebLogic 单位时间线程请求
Server 队列	等待请求队列
WebLogic 内存	内存利用率
JVM 堆	内存利用率

1.2.2 数据层的数据采集

数据层主要是对企业数据和用户数据进行组织、存储和管理,并且为应用服务器提供相应的接口.Oracle 数据库[5,6]具有功能强大、性能稳定的特点,因而被广泛地应用于企业中.本文将以 Oracle 数据库为例,对数据库及其宿主服务器健康指标进行说明,其他数据库也可以类似评价.

数据层数据采集分为 Oracle 服务器硬件特征和 Oracle 数据库软件特征这两类.

- Oracle 服务器外在状态即为数据库宿主服务器硬件方面的状态,主要包括 CPU 状况、磁盘状况、内存状况、交换区状况和网络质量,其中,网络质量又包含服务器接收和发送的错误率和丢包率,见表 3;
- Oracle 数据库状态即为数据库自身的一系列状态,主要包括数据库后台进程、数据库表空间、数据库命中率和数据库响应时间,其中,每个状态中都包含若干个指标(见表 4).

Table 3 Features of Oracle database host server

表 3 Oracle 数据库宿主服务器数据采集

系统特征采集项目	具体采集指标
CPU	CPU 利用率
硬盘	磁盘利用率
交换区	交换区利用率
内存	内存利用率
网络质量 (包括 4 个状态参数)	Oracle 数据接收错误率 Oracle 数据接收丢包率 Oracle 数据发送错误率 Oracle 数据发送丢包率

Table 4 Features of Oracle database

表 4 Oracle 数据库数据采集

系统特征采集项目	具体采集指标	
数据库后台进程 (包括 12 个状态参数)	PMON 进程 SMON 进程 CKPT 进程 CJQ0 进程 MMAN 进程 MMON 进程	DBW0 进程 LGWR 进程 PSP0 进程 QMNC 进程 RECO 进程 MMNL 进程
数据库表空间 (包括 4 个状态参数)	ANJIAN 表空间 SYSAUX 表空间	SYSTEM 表空间 USERS 表空间
数据库命中率 (包括 6 个状态参数)	Oracle 缓存命中率 Oracle 软解析率 Oracle 内存排序率	Oracle 解析执行率 Oracle 解析调用率 Oracle 锁竞争比率
数据库响应时间(包括 2 个状态参数)	Oracle 等待时间比	Oracle 事务响应时间

在完成数据采集后,将数据输入预测模型,实现对系统未来状态的预测.

2 B/S 架构软件系统下的状态建模

在上一节中,本方法采集了大量的关系复杂的数据.本节将对复杂数据进行预处理,为 B/S 架构软件系统状态建立隐马尔可夫模型,并对模型参数进行训练以建立系统内部状态与外部特征之间的联系.本节将重点介绍

系统模型的建立,观测状态的构造以及模型参数的训练等.

2.1 系统模型建立

为了推测系统内部状态,首先对系统建立模型,找到系统外部状态与内部状态的对应关系.如前所述,大型软件系统具有系统真实状态不可观测的特点,即:系统内部状态不能被直接的观测到,只能根据由系统产生的一系列系统特征参数来推测系统内部状态.这一特点与隐马尔可夫模型相适应.

隐马尔可夫模型[7]是一种被广泛应用于语言信号处理、图像处理和生物信号处理等[8−10]领域的模式识别和预测的模型,该模型包含两种状态:隐含状态和观测状态(如图 3 所示).隐含状态反映了系统内部实际状态,通常无法通过直接观测而得到,但是隐含状态表现出的系统特征参数(称为观测状态)可被直接观测到并且与隐含状态相关联.因此,为了推测系统状态,可通过建立观测状态和隐含状态之间的联系,由观测状态来计算得到隐含状态.

由此可以看出,隐马尔可夫模型和软件系统状态体系的相似程度极高.因而,对软件系统建立隐马尔可夫模型具有很强的合理性.

隐马尔可夫模型的隐含状态集合通常用 S 表示,$S=\{s_1,s_2,\ldots,s_N\}$,其中,N 为隐马尔可夫模型的隐含状态个数.模型的可观测状态集合通常用 V 表示,$V=\{v_1,v_2,\ldots,v_M\}$,其中,M 为隐马尔可夫模型中可观测状态个数.其结构如图 3 所示.

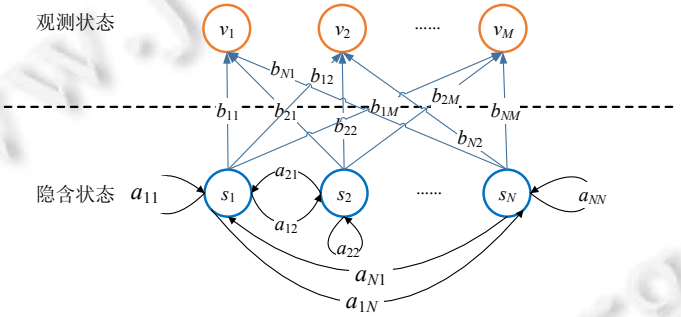


Fig.3 Hidden Markov model

图 3 隐马尔可夫状态模型

除此以外,隐马尔可夫模型还包括如下参数:

(1) $A=\{a_{ij}\}$,$a_{ij}=P\{q_{t+1}=s_j|q_t=s_i\}$.$A$ 为隐含状态转移概率矩阵,大小为 $N\times N$.其中,元素 a_{ij} 为系统在 t 时隐含状态 s_i,$t+1$ 时刻转移到状态 s_j 的概率,如图 3 所示;

(2) $B=\{b_j(k)\}$,$b_j(k)=P\{o_t=v_k|q_t=s_j\}$.$B$ 为观测状态转移概率矩阵,大小为 $M\times N$.其中,元素 $b_j(k)$表示 t 时刻系统在隐含状态 s_j 产生观测状态 v_k 的概率,如图 3 所示;

(3) $\pi=\{\pi_i\}$,$\pi_i=P\{q_1=s_i\}$.π称为初始状态概率矩阵,π_i 表示系统在初始时刻处于隐含状态 s_i 的概率,且有:

$$\sum_{i=1}^{N}\pi_i=1.$$

通常用$\lambda=\{A,B,\pi\}$表示隐马尔可夫系统模型参数.

2.2 状态空间的构造

根据隐马尔可夫模型的定义,我们首先需要确定模型的观测状态集和隐含状态集,即,状态空间的构造.本文将隐含状态分为正常、注意、异常和危险这 4 个状态,即,隐含状态个数 $N=4$.由上文对隐含状态概率转移矩阵的定义可知:

$$A=\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix},$$

其中,$\sum_{j=1}^{N} a_{ij}=1$,N=4,$i\in[1,4]$.后续将给出计算该矩阵的方法.

系统的观测状态是由隐含状态产生的可被观测到的特征参数构成,要准确构建观测状态,首先需要采集表征系统状态的参数.由于系统的复杂性,为了准确表征系统观测状态,我们采集了包括 Web 服务器、数据库及其宿主服务器的健康状况在内的 45 个系统后台参数.将这 45 个特征参数构成一个向量组 v,表示系统的一个可观测状态,$v\in R^{45}$.由于每个特征可有多个值,可观测状态集相当大.为了减少计算复杂度,需要对数据进行降维处理.通常的处理方法将每个特征划分为 k 个状态,对每个特征进行离散化处理.当特征数较少时,该方法是可行的.但是由于我们采集的特征参数个数众多,即便是采用离散化处理方法,可观测状态集也是巨大的,即 k^{45}.这样,同样会陷入维数灾难(curse of dimensionality)中.为了避免该问题,我们采用数据聚类算法对数据进行预处理,聚类结果即为我们所需要的观测状态空间.常用的聚类分析方法[29]有 *K*-means 算法[13−15]、层级聚类算法、SOM 聚类算法、FCM 聚类算法等.这些算法各有特点,其中:层级聚类算法采用的是“合并”或“分裂”形式,这一过程有不可修改的性质,使得该算法效果不是很理想;SOM 算法是基于模糊理论建立的聚类算法,该算法对初始聚类中心敏感,需要人为确定聚类数,容易陷入局部最优解;SOM 与实际大脑处理有很强的理论联系,但是聚类过程的时间开销较大.*K*-means 算法在聚类中心的选取上虽然存在一定随机性,但是通过多次聚类取最优的方式,这一问题可以得到有效解决.相关文献也做出了聚类算法效果的比较[29],综合聚类效果和聚类算法的时间开销考虑,本文将采用 *K*-means 算法对数据进行预处理.

将数据划分为 K 类,每一个聚类对应一类观测状态,观测状态个数 M=K.由此可得转移概率矩阵:

$$B=\begin{pmatrix} b_{11} & b_{12} & b_{13} & \cdots & b_{1M} \\ b_{21} & b_{22} & b_{23} & \cdots & b_{2M} \\ b_{31} & b_{32} & b_{33} & \cdots & b_{3M} \\ b_{41} & b_{42} & b_{43} & \cdots & b_{4M} \end{pmatrix}.$$

横行表示可观测状态,纵列表示 4 个隐藏状态,其中,$\sum_{j=1}^{N} b_j(k)=1$,$k\in[1,M]$.后续将给出计算观测概率矩阵的方法以及聚类方法.

2.3 模型参数的训练

根据上文构造的模型的状态空间和系统历史运行数据,本节将对模型参数$\lambda=\{A,B,\pi\}$进行训练.模型参数的训练实际是根据观测状态序列推导出模型参数的最大似然估计.对于模型参数训练,没有准确的解析解,通常采用 Baum-Welch 算法[16,17]求得近似解.该算法采用 EM(expectation-maximization)算法不断进行迭代,更新模型参数λ,直到达到预设条件.

2.3.1 模型参数的重估

首先对模型参数$\lambda=\{A,B,\pi\}$进行初始化,随机的对 A,B,π这 3 个参数赋值并使之满足:

$$\sum_{j=1}^{N} b_j(k)=1, \sum_{j=1}^{N} a_{ij}=1, \sum_{i=1}^{N} \pi_i=1.$$

然后定义前向变量$\alpha_t(i)=P[o_1,o_2,\ldots,o_t,q_t=s_i|\lambda]$.$\alpha_t(i)$表示系统在 t 时刻处于状态 s_i 且产生观测状态序列$O=\{o_1,o_2,\ldots,o_t\}$的概率,由图 4 可见其递推公式:

$$\alpha_{t+1}(j)=\left[\sum_{i=1}^{N}\alpha_t(i)a_{ij}\right]b_j(o_{t+1}),$$

其中,t=1,2,…,T−1;i,j=1,2,…,N.

另外,

$$\alpha_1(i)=\pi_1 b_i(o_1).$$

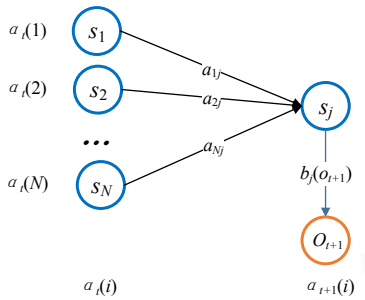


Fig.4 Derivation of $\alpha_{t+1}(j)$ from $\alpha_t(i)$

图 4 由$\alpha_t(i)$推导$\alpha_{t+1}(j)$

定义后向变量$\beta_t(i)=P[o_{t+1},o_{t+2},\ldots,o_T|q_t=s_i,\lambda]$.$\beta_t(i)$表示系统在时刻 t 处于状态 s_i 的情况下,产生观测状态序列 $O=\{o_{t+1},o_{t+2},\ldots,o_T\}$的概率,由图 5 可见其递推公式:

$$\beta_t(i)=\sum_{j=1}^{N}\beta_{t+1}(j)a_{ij}b_j(o_{t+1}),$$

其中,$t=1,2,\ldots,T-1$;$i,j=1,2,\ldots,N$.

另外,

$$\beta_T(i)=1.$$

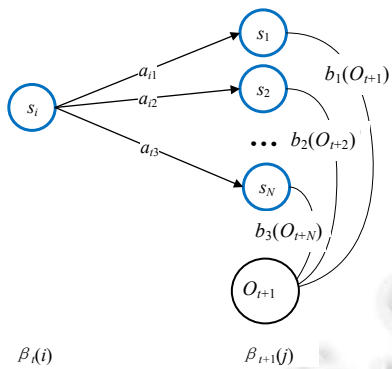


Fig.5 Derivation of $\beta_t(i)$ from $\beta_{t+1}(j)$

图 5 由$\beta_{t+1}(j)$推导$\beta_t(i)$

我们可以得到给定模型参数λ和观测状态序列 O 的情况下,系统在 t 时刻处于隐含状态 s_i,在 $t+1$ 时刻转移到隐含状态 s_j 的概率:

$$\xi_t(i,j)=P(q_t=s_i,q_{t+1}=s_j\mid O,\lambda)=\frac{\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_t(i)a_{ij}b_j(o_{t+1})\beta_{t+1}(j)}.$$

据此计算出系统在时刻 t 处于状态 s_i 的概率:

$$\gamma_t(i)=P(q_t=s_i\mid O,\lambda)=\sum_{j=1}^{N}\xi_t(i,j).$$

由此推出初始状态概率:

$$\overline{\pi}_i=\gamma_1(i)=\sum_{j=1}^{N}\xi_1(i,j).$$

状态转移概率:

$$\overline{a}_{ij}=\frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1}^{T-1}\gamma_t(i)}.$$

观测概率:

$$\overline{b}_j(k)=\frac{\sum_{t=1}^{T}\gamma_t(j)\delta(o_t,k)}{\sum_{t=1}^{T}\gamma_t(j)}(1\leqslant k\leqslant M,1\leqslant j\leqslant N),$$

其中,$\begin{cases}\delta(o_t,k)=1, \text{当}o_t=v_k\text{时}\\ \delta(o_t,k)\neq 1, \text{当}o_t\neq v_k\text{时}\end{cases}$,$v_k$ 为观测状态取值.

2.3.2 训练结束的判断

在获得新的模型参数 $\overline{\lambda}=\{\overline{A},\overline{B},\overline{\pi}\}$ 后,我们可以通过判断 $|\log(P(O|\overline{\lambda}))-\log(P(O|\lambda))|<\varepsilon$ 是否成立来决定迭代是否结束:若成立,则结束迭代,获得模型参数 $\overline{\lambda}$;反之,令 $\lambda=\overline{\lambda}$,再次进行迭代计算,获取新的 $\overline{\lambda}$,重复判断;直到获得符合要求的模型参数 $\overline{\lambda}$,结束迭代.其中,$P(O|\lambda)$和 $P(O|\overline{\lambda})$ 分别是在模型参数为λ和 $\overline{\lambda}$,时间 T范围内产生观测状态序列 $O=\{o_1,o_2,\ldots,o_T\}$的概率,通过向前算法(forward algorithm)[7]计算得到.

1) 初始化

$$\alpha_1(i)=\pi_1 b_i(o_1),1\leqslant i\leqslant N;$$

2) 迭代计算

$$\alpha_{t+1}(j)=\left[\sum_{i=1}^{N}\alpha_t(i)a_{ij}\right]b_j(o_{t+1}),1\leqslant t\leqslant T-1,1\leqslant j\leqslant N;$$

3) 终止

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

其中,T 为数据采样时间.

Baum-Welch 算法保障算法最后收敛于局部最优解[26],Baum 也给出了相应证明[27],本文将不再详细描述.

3 系统状态预测

上一节软件系统状态建立了隐马尔可夫模型,并且通过迭代计算得到隐马尔可夫模型参数.本节将三次指数平滑法(triple/three order exponential smoothing)[18−20]与隐马尔可夫模型相结合,构建出一种基于系统特征参数对系统真实状态进行预测的方法:首先,根据已经收集得到的数据,利用三次指数平滑法对系统在下一时段的各个特征参数进行预测;然后,利用构建好的隐马尔可夫模型推测出系统的真实状态.

3.1 系统特征参数预测

软件系统特征参数往往具有一定的规律性,如整体具有一定变化趋势(trend);长期范围内具有一定的周期性波动,或称为季节性(seasonality),因而可采用指数平滑法(exponential smoothing)对参数变换作预测.指数平滑法有 3 种不同形式:

- 一次指数平滑法针对没有趋势和季节性的序列;
- 二次指数平滑法用于处理有趋势但没有季节性的序列;
- 三次指数平滑法可以对同时含有趋势和季节性的时间序列进行预测.

本文根据数据特点,采用三次指数平滑法对系统特征参数进行预测.

利用指数平滑法进行系统特征参数的预测在很多文献中已经有详细的介绍[28],本文仅作简要介绍.假设需要对某项参数 x 进行预测.已采集了参数 x 在过去 i 个时间内的数据,那么其在 $i+h$ 时刻的数据可通过下式计算得到:

$$s_i = \alpha(x_i - p_{i-k}) + (1-\alpha)(s_{i-1} + t_{i-1}),$$
$$t_i = \beta(s_i - s_{i-1}) + (1-\beta)t_{i-1},$$
$$p_i = \gamma(x_i - s_i) + (1-\gamma)p_{i-k},$$
$$x_{i+h} = s_i + ht_i + p_{i-k+(h \bmod k)},$$

其中,

- s_i 是之前 i 个数据的平滑值,α是平滑参数,$\alpha\in[0,1]$.α越接近 1,平滑后的值越接近当前时刻的数据值;反之,α越接近 0,平滑后的值越接近前 i 个数据的平均值;
- t_i 表示平滑后的趋势,β是平滑参数,$\beta\in[0,1]$;
- p_i 表示平滑后的季节性变化趋势,γ是平滑参数,$\gamma\in[0,1]$.其中,k 为周期;
- 初始值选择,通常:$s_0=x_0,t_0=x_1-x_0,p_0=0$.

预测得到系统未来一段时间内特征参数后,本文将使用状态空间的构造过程中的 *K*-means 算法,构造出未来一段时间内系统的观测状态.

3.2 系统状态预测

在获得系统参数的预测值 $o_1,o_2,\ldots,o_t$ 基础上,我们使用逐步前进搜索的 Viterbi 算法[18]推测出对应的系统内部状态 $q_1,q_2,\ldots,q_t$.

定义韦特比变量$\delta_t(i)$表示给定模型λ,系统在时刻 t 处于状态 i,并且观察到 $o_1,o_2,\ldots,o_t$ 的最佳状态转换序列为 $q_1,q_2,\ldots,q_t$ 的概率.另外设定 T 个数组 $\psi_1(N),\psi_2(N),\ldots,\psi_T(N)$,其中,$\psi_t(i)$记录在时刻 t 系统处于状态 i 的最佳状态转换序列前时刻 t–1 的状态:

$$\delta_t(i) = \max_{q_1,q_2,\ldots,q_{t-1}} P(q_1,q_2,\ldots,q_{t-1},q_t=i,o_1,o_2,\ldots,o_t \mid \lambda).$$

$\delta_t(j)$和$\psi_t(j)$的推导过程如下:

$$\delta_t(j) = [\max_{1\leqslant i\leqslant N} \delta_{t-1}(i)a_{ij}]b_j(o_t),$$
$$\psi_t(j) = \arg\max_{1\leqslant i\leqslant N} [\delta_{t-1}(i)a_{ij}],$$

其中,$\delta_1(i)=\pi_i b_i(o_1),1\leqslant i,j\leqslant N,\psi_1(i)=0$.

综上所述,软件状态评估预测的整体系统架构如图 6 所示:首先采集系统运行时的特征参数;然后,利用指数平滑法对特征参数进行预测;接下来再对特征数据进行预处理,以获取符合 HMM 模型要求的输入(即,利用 *K*-means 聚类结果将预测得到的特征划分到相应的观测状态空间);HMM 模型根据输入数据计算出对应的系统内部状态.HMM 模型和三次指数平滑法初始参数通过历史数据训练获得,在系统运行中,根据预设条件更新参数,从而使模型能够动态地适应外部的变化.更新的触发条件为:预测的准确度是否低于某一阈值.

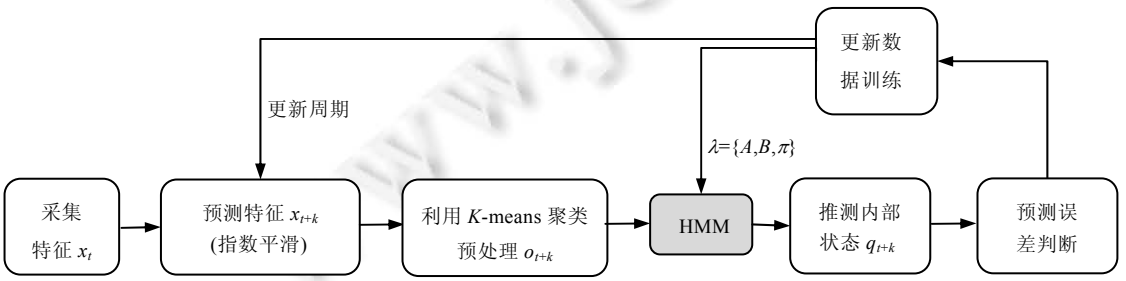


Fig.6 System architecture

图 6 系统体系架构

4 实 验

本节我们将通过实验来测试本文所提出的基于隐马尔可夫模型的软件系统状态预测方法.

4.1 实验环境部署

本文搭建了一个仿真的 B/S 信息管理系统来模拟真实运行环境,其实验环境部署如 7 图所示.

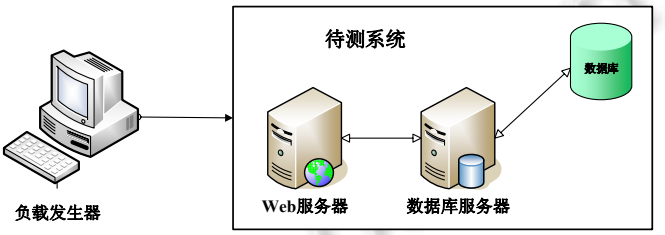


Fig.7 Experiment system

图 7 实验环境

整个实验环境由负载发生器、Web 服务器以及数据库服务器构成.Web 服务器以及数据库服务器配置了 Intel E2160 CPU 2.30Hz、DDR3 4G 内存以及千兆网卡.Web 服务器操作系统为 Windows 2003,并安装了 WebLogic 9.2 以及 Tomcat 7;数据库服务器部署了 Oracle 11g.负载发生器配置了 Intel E2160 CPU 1.8GHz、DDR3 4G 内存以及千兆网卡.负载发生器配软件部署了负载发生工具 Loadrunner 11[23,24].Loadrunner 的实例测试流程如下:

1) 首先,控制器调度压力测试,向虚拟用户生成器下达测试指令;
2) 随后,生成器开始模拟大量真实用户,从而产生压力.当整个待测系统受到来自用户的压力,便开始产生各种性能状态;
3) 性能监控器实时地捕获这些性能状态,随即反馈到控制器上;
4) 监控器将测试结果收集并保存起来,最终产生性能分析报告.

测试用例的选取,是性能测试的首要任务.测试用例的选取要符合的标准是典型的业务流程、用户操作使用频繁、对系统性能影响较大、性能测试压力符合业务系统用户实际的操作.按照该标准,我们以某安全监督与管理业务应用系统为例,选取其中一个主要模块作为评价案例.该模块提供的操作有:读取文件、发布文件、删除文件、修改文件、阅读文件之后可以提交一些评阅意见、统计文件的阅读次数、按照日期排序等.每个操作设计为一个事务,编写了测试用例.

4.2 测试状态建立

在实验开始前,利用负载发生器生成部分数据用于 HMM 模型参数训练以及观测状态空间的构造.HMM 模型参数训练方法如第 2.3 节所述,这里详细介绍利用 K-means 聚类算法构造观测状态空间的过程.

本实验采集了连续的 m=345 个时刻的系统运行历史数据,将同一时刻的特征参数构造为一个样本 $x^{(i)}$, $x^{(i)}\in R^{45}$,i=1,2,3,…,m, 训练集合 $X=\{x^{(1)},x^{(2)},\ldots,x^{(m)}\}$.训练步骤如下:

1. 随机选取 K 个聚类质心点$\mu_j\in R^{45}$,j=1,2,…,K;
2. 根据公式 $c^{(i)}=\arg\min\|x^{(i)}-\mu_j\|^2$ 计算每个聚类质心点的模,将样本 $x^{(i)}$归为最小 $c^{(i)}$对应的类;
3. 将每个类的质心调整到所属样本的中心位置:

$$\mu_j=\frac{\sum_{i=1}^{m}1\{c^{(i)}=j\}x^{(i)}}{\sum_{i=1}^{m}1\{c^{(i)}=j\}};$$

4. 重复步骤 2、步骤 3,直至样本收敛至 K 个簇.

完成数据处理后,可得到观测状态集合.

在采用 *K*-means 聚类过程中,最难的部分是如何选取 *K* 值.*K* 值的确定,很大程度上依赖于数据集的分布和规模.增大 *K* 值,能够减小聚类误差.一种极端的情况是:当 *K* 等于数据(样本)个数时,即每一个数据点被划分为一个独立的类,此时误差为 0.*K* 值的选择,必须在最大化的压缩数据和最小化聚类误差之间取得平衡.*K* 值选取方法如下.

- 定义聚类误差函数:

$$J(c^{(1)},...,c^{(m)},\mu_1,...,\mu_K)=\frac{1}{m}\sum_{i=1}^{m}\| x^{(i)}-\mu_{c^{(i)}} \|^2;$$

- 选取不同 *K* 值,求出对应聚类后对应的误差函数值 *J*,做出 *K*-*J* 关系图,如图 8 所示.

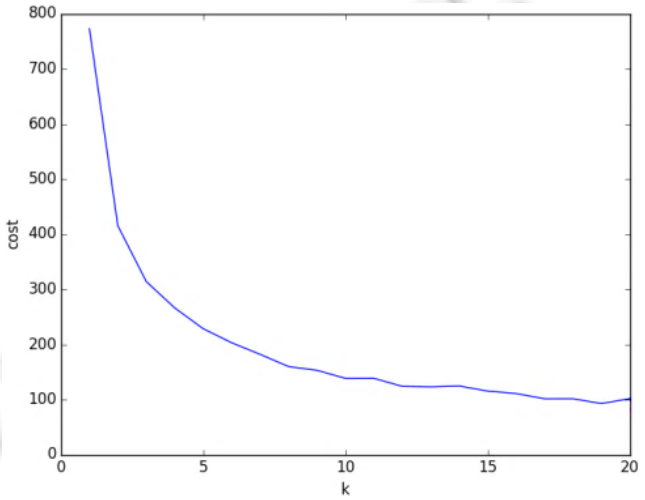


Fig.8 Clustering *K*-*J*

图 8 聚类 *K*-*J* 关系图

从图 8 可以看出:当 *K* 达到 8 时,聚类误差函数将不再有明显下降.因此,最佳聚类 *K*=8.根据聚算法的参数和运行环境,聚类分析的时间开销会有较大差异.在本实验中,聚类分析的时间开销约为 3 小时.

4.3 预测结果分析

本实验测试的目的是,验证所提出的方法对基于 B/S 架构软件系统运行状态的预测能力.实验通过负载发生器逐渐向系统添加负载,模型实时地对系统运行状态进行监控和分析,并预测系统运行状态.

我们设计了 3 个测试场景:单业务作为测试对象(删除文件和修改文件操作)以及混合业务场景压力测试(混合删除、修改文件操作).

- 第 1 个测试场景测试删除文件操作.

我们尝试逐渐增加负载,每 10s 加载 10 个虚拟用户,用户思考时间设置为实际值的 2%~5%.从图 9 上图可以看到:在时间点 100 左右,删除操作的平均事务响应时间约为 44s,属于不正常范围业务系统的响应时间,此时系统出现异常状态.我们注意到,图 9 的隐马尔可夫模型预测输出状态已提前发生改变,从而较好地预测出业务系统的运行状态.

- 第 2 个测试场景测试修改文件操作.

我们尝试突然增加负载,每 10s 加载 20 个虚拟用户,用户思考时间设置为实际值的 2%~5%.从图 10 可以看到:在时间点 50 左右,修改操作的平均事务响应时间为 46s,属于不正常范围业务系统的响应时间,此时系统出现异常状态.注意到,图中隐马尔可夫模型预测输出状态已提前发生改变,从而较好地预测出业务系统的运行状态.

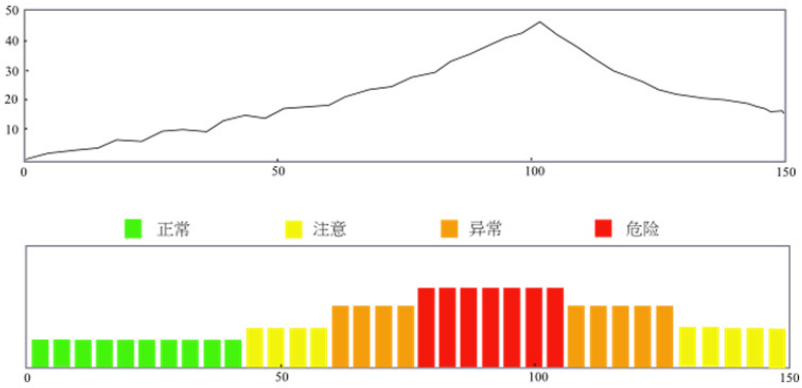


Fig.9 Test 1: Transaction response time and system output of forecast

图 9 测试 1:事务响应时间和模型预测输出

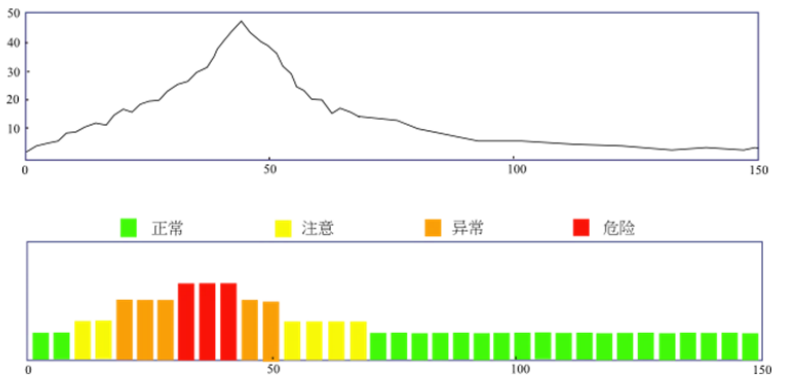


Fig.10 Test 2: Transaction response time and system output of forecast

图 10 测试 2:事务响应时间和模型预测输出

- 第 3 个测试场景为混合业务场景压力测试.

同时测试文件的修改与删除操作.每 10s 加载 10 个虚拟用户直到 50s 后,释放 30 个虚拟用户,然后在 10 秒后重新每秒加载 20 个用户.用户思考时间设置为实际值的 2%~5%.

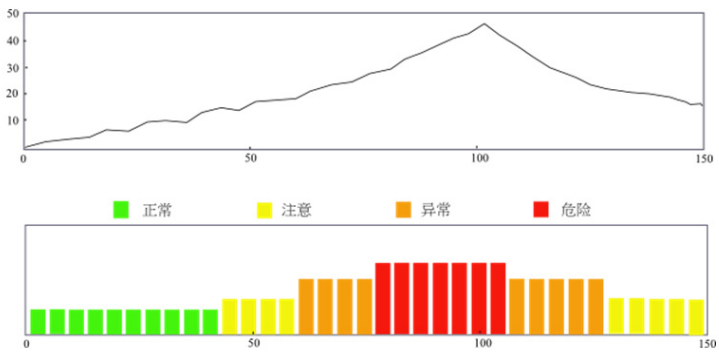


Fig.11 Test 3: Transaction response time and system output of forecast

图 11 测试 3:事务响应时间和模型预测输出

从实验数据分析我们看到:操作的平均事务响应时间在 10s 以内时,模型预测系统状态属于正常范围;10s~20s范围模型预测系统状态为注意;20s~30s范围模型预测系统状态为异常状态;30s以上为模型预测系统危险状态.另外,我们选取了系统状态预测为异常状态时的 CPU 使用率和 JVM 空间占用率数据分析,在该时间段,CPU 占用率超过了 75%,JVM 空间占用率超过 85%,系统处于严重状态.因此,模型评估和预测结果与我们的常识一致.上述实验结果表明:模型有效地评估了对系统的状态,特别是对危险状态进行了预测,因而可以较好地实时监控软件系统.

5 结束语

软件系统的状态评估机制是保障软件系统稳定运行的重要手段,然而随着软件系统规模不断增大,系统复杂性不断的提升,许多传统的评估方法的预测效果越来越不尽人意.这些方法主要的研究思路过于注重对软件系统的各个子系统的外在特征参数进行分析,建立简单模型对系统内部状态进行评估预测,这样往往不能充分挖掘大量数据中隐含的信息.本文则提出了一种基于隐马尔可夫模型的软件系统状态评估预测方法,该方法利用隐马尔可夫模型建立起系统外在状态(观测状态)和内部状态(隐藏状态)之间联系,从整体的角度对系统状态进行预测,而不拘泥于局部的特征,从而提高软件状态预测方法的准确性,便于软件系统管理和维护工作的开展.在今后工作中,我们将考虑进一步开发动态负载均衡算法以及相关程序包,并且结合软件系统的特征和本质对本算法进行优化,使其具有一定的普遍性和适用性.

References:

[1] Zhao D, Zhang WD, Liu J. The safety production information system based on B/S structure. China Public Security (Academy Edition), 2007,4:97−99 (in Chinese with English abstract).

[2] Chen DJ, Cao WG. The design and achievement of enterprise stock-sell-storage management system based on browser/server pattem. Modular Machine Tool & Automatic Manufacturing Technique, 2006,5:110 (in Chinese with English abstract).

[3] Oracle. Introduction to Oracle WebLogic Server, 11g Release. 2011.

[4] Zong Y, Jin P, Chen EH, Li H, Liu RJ. Fuzzy co-clustering algorithm for WebLogic. Journal of Electronics & Information Technology, 2012,34(3):543−548 (in Chinese with English abstract).

[5] Liu WX, Zhou JN, Zhang J. Application security of oracle database stored-procedure. Computer Systems & Applications, 2013,2: 80−83 (in Chinese with English abstract).

[6] Stephens SM, Chen JY, Davidson MG, Thomas S, Trute BM. Oracle database 10g: A platform for BLAST search and regular expression pattern matching in life sciences. Nucleic Acids Research, 2005,33(Suppl.1):D675−D679. [doi: 10.1093/nar/gki114]

[7] Rabiner LR, Juang BH. An introduction to Hidden Markov models. IEEE ASSP Magazine, 1986,3(1):4−16. [doi: 10.1109/MASSP.1986.1165342]

[8] Rabiner LR. A tutorial on Hidden Markov models and selected applications in speech recognition. Proc. of the IEEE, 1989,77(2): 257−286. [doi: 10.1109/5.18626]

[9] Li J, Najmi A, Gray RM. Image classification by a two-dimensional Hidden Markov model. IEEE Trans. on Signal Processing, 2000,48(2):517−533. [doi: 10.1109/78.823977]

[10] Sohrab PS, Xiang X, DeLeeuw RJ, Khojasteh M, Lam WL, Ng R, Murphy KP. Integrating copy number polymorphisms into array CGH analysis using a robust HMM. Bioinformatics, 2006,22(14):431−439. [doi: 10.1093/bioinformatics/btl238]

[11] Zhang JH, Zhang WB, Xu JW, Wei J, Zhong H, Huang T. Approach of virtual machine failure recovery based on Hidden Markov model. Ruan Jian Xue Bao/Journal of Software, 2014,25(11):2702−2714 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4548.htm [doi: 10.13328/j.cnki.jos.004548]

[12] Xie JY, Gao HC. Statistical correlation and *K*-means based distinguishable gene subset selection algorithms. Ruan Jian Xue Bao/Journal of Software, 2014,25(9):2050−2075 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4644.htm [doi: 10.13328/j.cnki.jos.004644]

[13] Wagstaff K, Cardie C, Rogers S, Schroedl S. Constrained *k*-means clustering with background knowledge. In: Proc. of the ICML. 2001. 577−584.

[14] Kanungo T, Mount DM, Netanyahu NS, Piatko CD, Silverman R, Wu AY. An efficient k-means clustering algorithm: Analysis and implementation. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2002,24(7):881−892. [doi: 10.1109/TPAMI.2002.1017616]

[15] Huang JZ, Ng MK, Rong H, Li Z. Automated variable weighting in *k*-means type clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2005,27(5):657−668. [doi: 10.1109/TPAMI.2005.95]

[16] Jia B, Zhu XY, Luo YP, Hu DC. Accurate Baum-Welch algorithm free from overflow. Ruan Jian Xue Bao/Journal of Software, 2000,11(5):707−710 (in Chinese with English abstract). http://www.jos.org.cn/ch/reader/view_abstract.aspx?flag=1&file_no=20000519&journal_id=jos

[17] Miklós I, Meyer IM. A linear memory algorithm for Baum-Welch training. BMC Bioinformatics, 2005,6(1):231−238. [doi: 10.1186/1471-2105-6-231]

[18] Stratonovich RL. Conditional Markov processes. Theory of Probability and its Applications, 1960,5(2):156−178. [doi: 10.1137/1105015]

[19] Brown RG. Smoothing forecasting and prediction of discrete time series. In: Proc. of the Englewood Cliffs. Prentice-Hall, 1963.

[20] Billah B, King ML, Snyder RD, Koehler AB. Exponential smoothing model selection for forecasting. Int'l Journal of Forecasting, 2006, 22(2):239−247. [doi: 10.1016/j.ijforecast.2005.08.002]

[21] Wang GQ, Wang S, Liu HY, Xue YD, Zhou P. Self-Adaptive and dynamic cubic ES method for wind speed forecasting. Power System Protection and Control, 2014,15:117−122 (in Chinese with English abstract).

[22] Kong DG, Tan XB, Xi HS, Shuai JM, Gong T. Hidden Markov model for multi-thread programs time sequence analysis. Ruan Jian Xue Bao/Journal of Software, 2010,21(3):461−472 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3521.htm [doi: 10.3724/SP.J.1001.2010.03521]

[23] Van BL, Garcia-Salicetti S, Dorizzi B. On using the Viterbi path along with HMM likelihood information for online signature verification. IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics, 2007,37(5):237−1247. [doi: 10.1109/TSMCB.2007.895323]

[24] Dai XJ, Zhang N. Performance testing and optimization of data analysis platform based on LoadRunner. Computer Technology and Development, 2013,23(7):202−206, 210 (in Chinese with English abstract). [doi: 10.3969/j.issn.1673-629X.2013.07.052]

[25] Yang P, Li J. Using LoadRunner to test Web's load automatically. Computer Technology and Development, 2007,1(80):242−244 (in Chinese with English abstract).

[26] Levinson SE, Rabiner LR, Sondhi MM. An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition. The Bell System Technical Journal, 1983,62(4):1035−1074. [doi: 10.1002/j.1538-7305.1983.tb03114.x]

[27] Baum LE, Sell GR. Growth transformations for functions on manifolds. Pacific Journal of Math, 1968,27(2):211−227.

[28] Kalekar PS. Time series forecasting using holt-winters exponential smoothing. Kanwal Rekhi School of Information Technology, 2004.

[29] Feng XP, Zhang TF. Comparison of four clustering methods. Microcomputer & Its Applications, 2010,29(16):1−3 (in Chinese with English abstract).

附中文参考文献:

[1] 赵迪,赵望达,刘静.基于 B/S 架构的安全生产监督管理信息系统.中国公共安全(学术版),2007,4:97−99.

[2] 陈帝江,曹文钢.基于 B/S 模式的进销存管理系统的设计与实现.作何机床与自动化加工技术,2006,5:110−112.

[4] 宗瑜,金萍,陈恩红,李红,刘仁金.面向 WebLogic 的模糊协同聚类算法.电子与信息学报,2012,34(3):543−548.

[5] 刘伟祥,周建宁,张捷.ORACLE 数据库存储过程应用安全.计算机系统应用,2013,2:80−83.

[11] 张建华,张文博,徐继伟,魏峻,钟华,黄涛.一种基于隐马尔可夫模型的虚拟机失效恢复方法.软件学报,2014,25(11):2702−2714. http://www.jos.org.cn/1000-9825/4548.htm [doi: 10.13328/j.cnki.jos.004548]

[12] 谢娟英,高红超.基于统计相关性与 *K*-means 的区分基因子集选择算法.软件学报,2014,25(9):2050−2075. http://www.jos.org.cn/1000-9825/4644.htm [doi: 10.13328/j.cnki.jos.004644]

[16] 贾宾,朱小燕,罗予频,胡东成.消除溢出问题的精确 Baum-Welch 算法.软件学报,2000,11(5):707−710. http://www.jos.org.cn/ch/reader/view_abstract.aspx?flag=1&file_no=20000519&journal_id=jos

[21] 王国权,王森,刘华勇,薛永端,周平.基于自适应的动态三次指数平滑法的风电场风速预测.电力系统保护与控制,2014,15:117−122.

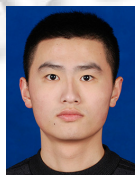
[22] 孔德光,谭小彬,奚宏生,帅建梅,宫涛.多线程程序时序分析的隐 Markov 模型.软件学报,2010,21(3):461−472. http://www.jos.org.cn/1000-9825/03521.htm [doi: 10.3724/SP.J.1001.2010.03521]

[24] 戴晓婧,张宁.基于 LoadRunner 的数据分析平台的性能测试及优化.计算机技术与发展,2013,23(7):202−206. [doi: 10.3969/j.issn.1673-629X.2013.07.052]

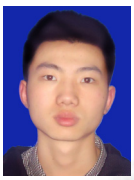
[25] 杨萍,李杰.利用 LoadRunner 实现 Web 负载测试的自动化.计算机技术与发展,2007,17(1):242−244.

[29] 冯晓蒲,张铁峰.4 种聚类方法之比较.微型机与应用,2010,29(16):1−3.

吴佳(1980－),女,四川荣昌人,博士,副教授,CCF 专业会员,主要研究领域为机器智能,数据挖掘.



陈瀚霖(1995－),男,CCF 学生会员,主要研究领域为数据挖掘,机器学习.



曾惟如(1995－),男,主要研究领域为神经网络,机器智能,数据分析.

唐雪飞(1964－),男,博士,副教授,主要研究领域为云计算,数据处理与数据可视化,代码自动化.