使用带分类参数贝叶斯网的虚拟机性能预测

尚聪聪,郝 佳,张彬彬,岳 昆

(云南大学 信息学院,昆明 650500)
E-mail:1390095882@qq.com

摘 要: 准确量化虚拟机特征和性能之间的依赖关系,并预测特定特征配置下的虚拟机性能,是虚拟资源细粒度分配的基础.然而,传统贝叶斯网(Bayesian Network,BN)虽然能够表达虚拟机特征和性能之间的不确定性依赖关系,但当某组虚拟机特征配置未出现在训练集中时,BN 便无法预测该配置下的虚拟机性能.此外,当虚拟机特征配置组合情况过多时,会导致虚拟机性能节点出现组合爆炸的情况.为此,本文提出一个带分类参数的 BN 模型.该模型首先利用随机森林分类算法,对虚拟机进行分类,然后根据分类结果和对应性能值,构建一个带分类参数的 BN 模型,从而实现任意特征配置下的虚拟机性能预测.同时,也降低了由于组合爆炸而带来的性能预测困难.实验结果表明了本文提出方法的有效性和准确性.

关 键 词: 虚拟机;性能预测;随机森林;贝叶斯网
中图分类号: TP391 **文献标识码:** A **文 章 编 号:** 1000-1220(2019)07-1416-07

Performance Prediction of Virtual Machines Via the Class Parameter Augmented Bayesian Network

SHANG Cong-cong, HAO Jia, ZHANG Bin-bin, YUE Kun
(School of Information Science and Engineering, Yunnan University, Kunming 650500, China)

Abstract: Accurately quantifying the dependencies among properties and performance of virtual machines (VMs) as well as predicting their performance under specific configurations is the basis for achieving fine-grained VM resources allocation. However, although the traditional Bayesian Network (BN) method can represent the uncertain dependences among properties and performance of the VMs, when a group of VM properties do not appear in the dataset, BN cannot predict the corresponding VM's performance. In addition, when there are too many VM property values, the combination of VM performance may explode, which will increase the difficulty of VM performance prediction. Thus, we propose a Class Parameter Augment Bayesian Network model (CBN). The CBN model first uses random forest (RF) algorithm to classify the VM properties, and then builds CBN according to the classification results and the corresponding performance values, so as to achieve the accurate prediction of VM performance under arbitrary VM configuration. At the same time, CBN reduces the difficulty of performance prediction caused by combination explosion. The experimental results show the effectiveness and accurateness of our proposed method.

Key words: virtual machine; performance prediction; random forest; Bayesian network

1 引 言

基础设施即服务(Infrastructure as a Service,IaaS)是云计算[1]的一种资源使用模式,云计算通过虚拟化技术向用户提供计算、存储等资源,用户按需租用这些资源并支付费用.目前大部分云服务提供商,例如阿里云[1]、亚马逊云[2] 等,都是以虚拟机的形式,向用户提供可选的若干种 CPU、内存以及网络带宽等资源配置套件[2].这种业务模式,并未真正实现资源的按需分配,难以适应动态变化的资源需求,导致资源利用率低下.为了提高资源利用率以及实现按应用需求分配虚拟机所需资源,首先需要准确预测特定虚拟机特征配置下的性能[3].

然而,准确预测虚拟机特定特征配置下的性能,存在以下困难.一方面,虚拟机的性能与特征之间存在复杂的非线性依赖关系[4,5],虚拟机的各个特征之间相互作用共同决定了虚拟机性能.另一方面,由于虚拟机上所运行的负载具有复杂性、动态性以及不确定性,这也增加了虚拟机性能预测的难度.其中,应用负载的复杂性,是指多个虚拟机上的应用负载所需资源不同,可能会导致物理主机的资源使用超过所能承受的峰值;应用负载的动态性,是指不同负载的运行情况会随着时间变化而改变,可能会产生多个应用之间的通信,影响虚拟机性能;应用负载的不确定性,是指应用负载在不同时刻所

[1] Alibaba Cloud. https://www.aliyun.com/,2018/06/05.

[2] Amazon EC2. http://aws.amazon.com/ec2/,2018/06/05.

收稿日期:2018-10-05 收修改稿日期:2018-12-03 基金项目:国家自然科学基金项目(61402398)资助;云南大学青年英才培育计划(WX173602)资助;云南大学科研基金项目(2017YDJQ06)资助. 作者简介:尚聪聪,男,1992 年生,硕士研究生,研究方向为云计算、数据分析;郝 佳,女,1993 年生,博士研究生,研究方向为人工智能、数据挖掘;张彬彬,女,1982 年生,博士,讲师,研究方向为虚拟化、云计算;岳 昆,男,1979 年生,博士,博士生导师,CCF 高级会员,研究方向为海量数据分析与服务、大数据知识工程、社会计算与智能信息处理.

需的资源大小不确定,可能会导致固定资源特征分配下虚拟机的性能产生波动.

学界已开展了很多虚拟机性能预测的研究,一些研究通过建立虚拟机特征和性能之间的线性模型,来实现虚拟机性能预测. 文献[6]提出基于硬件计数器的虚拟机性能估算方法;文献[7]提出基于指数平滑模型的虚拟机预测算法. 然而,虽然线性模型使用简单,但它难以有效拟合特征和性能之间的非线性依赖关系,从而导致模型的预测结果不够理想.

另一些工作通过建立虚拟机特征与性能之间的非线性关系模型来实现性能预测. 其中,文献[8]提出基于奇异值近似分解的虚拟机性能预测方法,分析虚拟机性能的变化趋势,但难以将结果进行直观展示;文献[9]基于马尔科夫方法来预测虚拟机特定状态下的性能,但对时间序列太过依赖;文献[10]提出基于遗传优化的人工神经网络方法来预测不同配置组合情况下虚拟机性能,并从中找出具有最小资源开销的特征配置组合;文献[11]提出基于人工神经网络的迭代模型来预测给定特征配置下的虚拟机性能,虽然可以对虚拟机特征和性能之间存在的非线性关系建模,但耗费大量计算资源、且容易出现过拟合. 综上所述,上述方法和模型中,虚拟机特征和性能之间的不确定性依赖关系仍然难以有效表达,且当虚拟机性能产生波动时,这些模型则无法准确预测这种性能波动情况.

文献[12]提出基于贝叶斯网(Bayesian networks,BN)来对虚拟机特征和性能之间存在的不确定性依赖关系建模的方法,用以虚拟机性能评估. 贝叶斯网模型用条件概率表达各个信息要素之间的相关关系,能在有限、不完整、不确定的信息条件下进行学习和推理. BN 是一个有向无环图(Directed Acyclic Graphic,DAG),图中的节点表示随机变量,节点间的边表示变量之间的依赖关系. 每个节点都有一张条件概率表(Conditional Probability Table,CPT),用来刻画这种依赖的程度. BN 能够有效地表达虚拟机各个特征以及特征与性能之间存在的非线性依赖关系,以及这种依赖关系的强弱,基于 BN 可进行概率推理计算[13].

然而,对于一般的 BN 模型来说,模型的构建依赖于数据的训练结果,当虚拟机特征取值的组合未出现在训练数据集中时,BN 便无法预测这些特征所对应的性能. 此外,当影响虚拟机性能的特征节点数量较多时,会导致性能节点的 CPT 组合情况复杂,增加参数计算以及性能预测的复杂性.

因此,本文采用一种带分类参数的 BN 模型(Class parameter augmented BN,CBN). 该模型首先对虚拟机进行分类,然后基于分类结果与对应的虚拟机性能值,构造 CBN 模型以实现对任意特征配置下的虚拟机性能预测. 经过对常见的分类模型比较,基于集成学习的随机森林(Random Forest,RF)模型与其他分类算法相比,能够处理高维度的复杂特征数据,预测速度快且准确率高,还能够平衡不均衡数据的预测误差,具有分类结果稳定且不易过拟合等优点[14]. 因此,本文采用随机森林作为 CBN 模型的分类器.

综上,本文的主要工作包括:

1)基于虚拟机特征数据训练随机森林模型作分类器.

2)基于虚拟机的分类和性能数据,训练 CBN 模型,用于表达虚拟机特征和性能之间的依赖关系.

3)基于 CBN 的概率推理预测任意虚拟机特征配置下的虚拟机性能和性能波动情况.

本文第 2 节给出 CBN 模型的构造方法,第 3 节给出基于 CBN 预测虚拟机性能的方法,第 4 节给出实验结果,第 5 节总结全文内容并展望将来的工作.

2　CBN 模型构建

影响虚拟机性能的特征包括硬件特征、软件特征以及运行时环境特征[9]. 这些特征的不同组合值,构成不同配置的虚拟机. 当某个应用在特定的虚拟机配置上运行时,应用的运行时间可以反映当前虚拟机的性能. 为了能够方便地获取虚拟机特征-性能数据,我们把应用在特定特征配置虚拟机上的运行时间作为当前虚拟机的性能值.

假设共有 m 种可调整的虚拟机特征,其中包括虚拟 CPU 的个数、内存大小、同时运行的虚拟机个数等,分别表示为 $V_1,V_2\cdots,V_m$. 假设每一个特征都有 i 种取值情况,如特征 V_m 的取值为 $v_{m1},v_{m2},\cdots,v_{mi}$,则 m 个特征可构成 i^m 种不同虚拟机特征配置. 本文通过 RF 分类算法,将 i^m 种虚拟机分为 r 个类别. 在 CBN 模型中,m 个虚拟机经过分类后构成一个分类节点 Z,Z 有 r 种可能取值,表示为 $z_1,z_2,\cdots,z_r$. 下面给出 CBN 的定义.

定义 1. CBN 为一个有向无环图,表示为 $G_{CBN}=(U,E,\theta)$,其中:

· U 为 G_{CBN}中的节点集合,包括虚拟机分类节点 Z 和虚拟机性能节点 T,即 $U=Z\cup T$.

· E 表示 U 中各个节点之间边的集合. 用 $e(U_i,U_j)$来表示一条由节点 U_i 指向节点 U_j 的有向边,则 $e(U_i,U_j)\in E,i\neq j$. 在 CBN 模型中,只包含一条由分类节点 Z 指向性能节点 T 的边.

· θ 表示 G_{CBN}中各个节点的参数集合,由各个节点的条件概率表构成,度量了虚拟机性能节点 T 对虚拟机分类节点 Z 的依赖程度的大小.

在 CBN 中,无父亲的节点 Z 的参数为边缘概率分布值 $P(Z)$,节点 Z 的子节点 T 的参数值为条件概率分布值 $P(T|Z)$. $P(Z)$和 $P(T|Z)$构成了 CBN 的参数 θ.

CBN 的构建包括结构构建和参数计算,结构构建部分包括 RF 分类器的训练,以及根据分类结果的结构学习;参数计算部分包括对虚拟机分类节点 Z 和特征节点 T 的后验概率计算.

2.1　CBN 模型中随机森林分类器训练

随机森林回归算法以 CART 决策树[14] 为基础,利用 Bootstrap 重采样方法,从含有 L 个样本的训练数据集 D 中,有放回地抽取 s 个样本组成新的样本集合 S,该样本集合称为 Bootstrap 样本. 分别对每个 Bootstrap 样本构建一棵 CART 决策树,最终根据所有决策树分类结果进行投票,将投票结果作为随机森林的最终分类结果[15].

构建决策树时,我们选用基尼系数(GINI)作为当前虚拟机特征分裂的依据. 选择具有最小 *Gain_GINI* 值的虚拟机特征 V_m 及其相应的特征值 v_{mi},作为当前 CART 分类树的最优分裂节点及其相应的分裂特征值. *Gain_GINI* 的值越小,则证明被分裂后的样本“纯净度”越高,则选择该特征值作为分裂

依据的效果越好. 对于含有 s 个虚拟机特征-性能样本的集合 S 来说,它的 *GINI* 系数计算方式如式(1)所示.

$$GINI(S) = 1 - \sum_{k=1}^{r} p_k^2 \tag{1}$$

在(1)中,p_k 表示虚拟机分类为第 k 类的概率,即 Z 为 z_k ($k \in \{1,2,\cdots,r\}$)的概率值.

对于含有 s 个样本的集合 S 来说,根据虚拟机特征 V_m 的第 i 个属性值,即 v_{mi},将 S 划分为 S_1 和 S_2 两部分后,*Gain_GINI* 的的计算方式如(2)所示.

$$Gain\_GINI_{v_{mi}}(S) = \frac{n_1}{s} GINI(S_1) + \frac{n_2}{s} GINI(S_2) \tag{2}$$

其中,n_1 和 n_2 分别为 S_1 和 S_2 中样本的个数,$GINI(S_1)$ 和 $GINI(S_2)$ 的计算方式类似于公式(1). 对于特征 V_m 来说,通过公式(2)分别计算当选择 $v_{m1}, v_{m2}, \cdots, v_{mi}$ 作为分裂特征值时,对应的 *Gain_GINI* 值的大小,并选择使得 *Gain_GINI* 最小的特征值,作为虚拟机特征 V_m 的最优分裂值. 即,

$$\min_{v_{mi}}(Gain\_GINI_{v_m}(S)) \tag{3}$$

对于样本集 S 来说,计算所有特征 V_m 的最优二分方案,选择其中的最小值,作为样本集合 S 的最优二分方案,即,

$$\min_{v_1,v_2,\cdots,v_m}(\min_{v_{mi}}(Gain\_GINI(S))) \tag{4}$$

上述方案中,虚拟机特征 V_m 和它的特征值 v_{mi},即为样本 S 的最优分裂特征及其最优分裂特征值.

通过上述方法,可构造一棵 CART 决策树. 当给定一组虚拟机特征配置时,可用该树预测出当前特征配置的所属类别. 当随机森林中一共包含 t 棵 CART 分类树时,它们的分类结果的投票结果即为虚拟机的分类.

综上,虚拟机分类的随机森林模型构建步骤如下:

1)从训练数据集 D 中,采用 Bootstrap 方法抽取 s 个数据样本作为第 i 棵树的训练集 D_{s_i} ($0 < s \leqslant N, 1 \leqslant i \leqslant t$),并用 D_{s_i} 构造第 i 棵树的根节点,将根节点放入队列 *queue* 中.

2)如果 *queue* 非空,则取出一个节点 *node*. 基于 *node* 中的虚拟机特征-性能数据样本 D_{node} 和 m 个虚拟机特征节点,依次计算特征节点 V_j ($1 \leqslant j \leqslant m$)的每一个特征值 v_{jk} ($1 \leqslant k \leqslant r$)所对应的 *Gain_GINI* 值大小,并找出最小的 *Gain_GINI* 值 *min* 和对应的特征节点 V_j 和特征值 v_{jk}. 如果 *min* 大于阈值 q,则根据分裂特征 V_j 和特征值 v_{jk} 将当前节点分裂出两个孩子节点,并将两个孩子节点加入队列 *queue*,否则不做操作. 重复执行上述过程,直到 *queue* 为空,完成当前决策树的构造.

3)重复步骤 1)和 2),直至建立 t 棵 CART 分类树,最终构成虚拟机分类的随机森林模型 RF.

算法 1 具体描述了上述思想.

算法 1. 基于随机森林的虚拟机分类

输入: D:虚拟机特征-性能数据集;

L:虚拟机特征-性能数据集中的数据条数

t:随机森林中决策树的数量

m:虚拟机特征的个数

r:虚拟机特征的取值个数

q:*GINI* 系数最小的阈值

$Gm[m]$:用于存储虚拟机特征的最小 *Gain_GINI* 值

$Vm[m]$:用于存储最小 *Gain_GINI* 值对应的特征值

$Gr[r]$:用于存储同一个虚拟机特征按不同取值分裂的 *Gain_GINI* 值

$Vr[r]$:用于存储虚拟机一个特征所有取值

输出: RF:基于随机森林的虚拟机分类模型

步骤:

```
1. for i←1 to t do
2.   D_si←Bootstrap(S,D)//为第 i 棵树选择样本 D_si
3.   root←makeNode(D_si)//集合 D_si作为根节点
4.   queue←root //将 root 加入队列
5.   while queue ! =Φ do
6.     node←pop(queue)//队列取出一个节点
7.     D_node←OutData(node)//获取当前 node 的数据
8.     for j←1 to m do
9.       r←CountNum(D_node,j)//当前特征取值个数
10.      for k←1 to r do
11.        Gr[k]←Calculate(Gain_GINI(D_node, j,k))
             //计算虚拟机特征 V_j 的取值为 v_jk时的 Gain_GINI
             值的大小存储到数组 Gr[r]中
12.        Vr[k]←findValue(D_node,j,k)//第 k 个特征值
13.        k←k+1
14.      end for
15.      Gm[j]←findNum(MIN(Gr[r]))
           //找到 Gr[r]数组中最小的 Gain_GINI 值
16.      Vm[j]←relevantValue(Gm[j], Gr[r], Vr[k])
           //找到最小 Gain_GINI 值对应的特征取值
17.      j←j+1
18.    end for
19.    min←findNum(MIN(Gm[m]))//最小 Gain_GINI 值
20.    if min > q
21.      V_j ←relevantFeature(min, Gm[m])
           //找到最小 Gain_GINI 值对应的特征
22.      v_jk←relevantValue (min, Gm[m], Vm[j])
         //找到 min 对应的虚拟机特征的取值
23.      lchild,rchild←makeNode(split(node, V_j, v_jk))
         //根据分裂特征和分裂值,将节点 node 分成左右节点
24.      queue←lchild,rchild //将节点存储到队列中
25.    end if
26.  end while
27.  RF←root //将 root 加入随机森林中
28. end for
29. return RF
```

基于算法 1 可实现对多个虚拟机进行分类. 当给出一组未知虚拟机特征时,分别由随机森林中的 t 棵树分类,分类结果按少数服从多数进行投票,投票结果即表示该组特征的分类结果. 算法 1 运行时,假设每一棵决策树需要计算 m 个特征的 *Gain_GINI* 值系数,则基于含有 N 个样本的训练数据集 D,来构建含有 t 棵决策树的随机森林时,该算法的时间复杂度为 $O(t(mt * \log N))$.

例 1.

使用表 1 中的数据,根据算法 1 构建 CART 决策树. 首先计算每个属性的每个取值的 *Gain_GINI* 值. 根据公式(2)可以算出:

CPUnum 取 1 为分裂值时,*Gain_GINI* 值 =

2/4(1 - (1^2 + 0^2 + 0^2)) + 2/4(1 - ((1/2)^2 + (1/2)^2)) = 1/4

CPUnum 取 2 为分裂值时,*Gain_GINI* 值 = 1/4
MemorySize 取 1024 为分裂值时,*Gain_GINI* 值 = 1/2
MemorySize 取 512 为分裂值时,*Gain_GINI* 值 = 1/2
CPUFreq 取 3.3 为分裂值时,*Gain_GINI* 值 = 1/4
CPUFreq 取 3.6 为分裂值时,*Gain_GINI* 值 = 1/4

表 1　虚拟机特征与分类实例

Table 1　Example for classification of virtual machines

CPUnum（个）	*Memory Size*(MB)	*CPUFreq*（GHz）	*Z*
1	512	3.3	z_1
2	1024	3.6	z_3
2	512	3.6	z_2
1	1024	3.3	z_1

根据算法 1,我们选取 *Gain_GINI* 值最小的属性和取值作为最优分裂特征和分裂值. 本例我们选择 *CPUnum* 是否取 1 作为分裂特征和分裂值,将数据分为实例分为两部分. 第一部分包含 z_1 与 z_4,第二部分包含 z_2 与 z_3. 对于第一部分,类别标签只有 z_1,所以无需再分. 对于第二部分重复上述步骤,得到 *MemorySize* 是否取 1024 为最优分裂特征和分裂值. 由此构建出如图 1 所示的决策树.

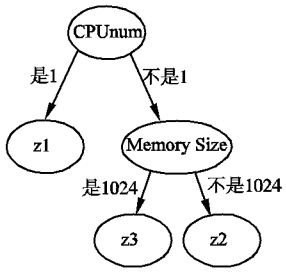

图 1　CART 决策树实例

Fig. 1　An example of CART decision tree

2.2　CBN 的结构构造与参数学习

根据先验知识,可确定虚拟机特征将对虚拟机性能产生不确定性影响,且它们之间存在相互依赖关系. 所以 CBN 的结构中只存在一条由 Z 指向 T 的有向边,即 $e(Z,T)$. 基于算法 1 可将所有的虚拟机分为 r 个类别,即 $Z\subseteq\{z_1,z_2,\cdots,z_r\}$,而性能节点有 j 种取值,即 $T_1,T_2\cdots T_j$. 例 1 是一个 CBN 的特征选择以及结构构造示例.

待分类的虚拟机特征为虚拟 CPU 个数(*CPUnum*)、内存大小(*MemorySize*)、同时运行的虚拟机个数(*coVMS*)和物理主机的 CPU 主频(*CPUFreq*),取值情况如表 2 所示. 运行算法 1 后,这些不同的特征配置被分为 4 个类别,即 z_1,z_2,z_3,z_4,它们所对应的性能值分别为 T_1,T_2,T_3,T_4. 表 1 展示了具体取值情况.

基于这 4 个特征的分类结果 Z 和所对应的性能值 T,可以构造 CBN 的结构.

例 2.

对于 CBN 的参数计算来说,我们运用最大似然估计(Maximum Likelihood Estimation,MLE)来计算 CBN 中各个节点的参数,即节点的 CPT. 计算公式如(5)所示.

$$\theta_{ijk}=\frac{Num((U_i=k),Pa(U_i=j))}{Num(Pa(U_i=j))} \tag{5}$$

其中,θ_{ijk}表示当节点 U_i 取值为 k,且 U_i 的父节点取值为 j 时节点 U_i 的参数. $Num((U_i=k),Pa(U_i=j))$表示当节点 U_i 取值为 k,且 U_i 的父节点取值为 j 时,虚拟机特征-性能数据集 D 中的数据条数. 而 $Num(Pa(U_i=j))$表示 U_i 的父节点取值为 j 时的数据条数. 在 CBN 中,所有节点的 CPT,构成了 CBN 的参数值 θ.

表 2　虚拟机特征、分类与性能实例

Table 2　Example for features and the classification of virtual machines

CPUnum（个）	*Memory Size*(MB)	*coVMS*（个）	*CPUFreq*（GHz）	*Z*	*T*
1	512	1	3.3	z_1	T_1
2	1024	2	3.6	z_2	T_2
3	1500	3	3.6	z_3	T_3
4	2048	1	3.3	z_4	T_4

3　基于概率推理的虚拟机性能预测

基于 CBN 结构及参数值,可以预测任意虚拟机特征配置下的的性能. 当某一组虚拟机特征配置为$\{v_{11},v_{21},\cdots,v_{m1}\}$时,经过 RF 算法分类后,该虚拟机被分为类别 z_i. 基于 CBN 可得到 z_i 所对应的虚拟机性能值分别为 $T_1,T_2,\cdots,T_i$ 的条件概率. 具有最大条件概率值的性能即为 CBN 预测的虚拟机性能值.

例如,CBN 的结构构建和参数计算情况如图 2 所示. 图中节点 Z 为虚拟机分类节点,所有的虚拟机特征配置被分为 3 个类别,即 z_1,z_2 和 z_3,经过计算后,这三个类别的取值概率分别为 0.1,0.4 和 0.5. 而虚拟机的性能节点 T 的取值共有 3 种,通过 MLE 可计算相应类别下的条件概率值. 图 1 中节点 T 的 CPT 展示的是当虚拟机类别为 z_1 时,性能分别为 T_1,T_2 和 T_3 的条件概率值.

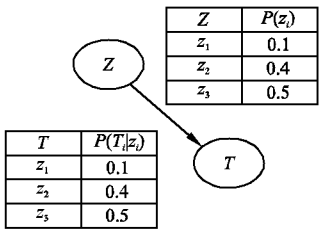

图 2　CBN 示例

Fig. 2　Example for CBN model

当 Z 的取值为 z_i 时,性能节点 T 的取值为 T_i 的条件概率计算公式如(6)所示.

$$P(T=T_i|Z=z_i)=\frac{P(T=T_i)*P(Z=z_i|T=T_i)}{P(z_i)} \tag{6}$$

使用 RF 算法对虚拟机的某一组特征进行分类后,结果记为 z_i,然后用 CBN 分别预测当前配置性能为 $T_1,T_2,\cdots,T_i$ 的概率. 预测过程即是计算 $P(T=T_1|Z=z_i))$,$P(T=T_2|Z=z_i))$,$\cdots$,$P(T=T_i|Z=z_i))$的过程. 最终,我们选择具有

最大后验概率的性能值,作为当前配置下虚拟机性能的预测值.

算法 2 描述了具体的性能预测过程.

算法 2. 基于 CBN 的虚拟机性能预测

输入: $v_{11},v_{21}\cdots.v_{m1}$:待预测虚拟机的虚拟机配置

j:性能值个数

$Tj[j]$:存储虚拟机所有的性能值

$Pj[j]$:用于存储每个性能值的条件概率

输出: X:CBN 预测的最大概率值

T_i:基于 CBN 所预测的性能值

步骤:

1. $z_i \leftarrow RF(v_{11},v_{21}\cdots.v_{m1})$
   //*RF* 根据 m 个特征取值得到分类结果
2. **for** $k \leftarrow 1$ to j **do**
3. $P_j[k] \leftarrow Calculate(P(T=T_k \mid Z=z_i))$
   //分类结果为 z_i 时,性能为 T_k 的条件概率
4. **end for**
5. $X \leftarrow findMax(P_j[j])$//找到性能预测最大概率
6. $T_i \leftarrow findPerformance(X, P_j[j], T_j[j])$
   //根据 X 找到对应的性能值,作为性能预测值
7. **return** X and T_i

基于算法 2,当虚拟机分类为 z_i 时,分别计算它所对应的 j 个性能值的条件概率,则算法 2 的时间复杂度为 $O(j)$.

例 3.

假定某台虚拟机配置为 CPUnum 为 2 个,MemorySize 为 1024MB,CPUFreq 为 3.3GHz. 根据例 1 的出的 CART 决策树可以得到类别为 z_3. 再根据相应的 CPT 表,查询 z_3 的性能值,就可以得到该虚拟机的性能预测值.

4 实验结果

实验环境如下:Intel(R)Core(TM)i5-6200U 处理器;2.40GHz 的 CPU 主频,8GB 内存,Windows10 64 位操作系统,Python 3.5.2:Anaconda 4.2.0(64-bit)为开发平台.

为了测试本文提出的 CBN 模型在预测虚拟机性能时的有效性和准确性,我们使用阿里云公布的批处理任务运行数据集 BatchInstance 做测试[3]. 该数据集包含了部署在阿里云中的某个实际运行的集群(共包含 13000 台物理主机)上的虚拟机,在 12 个小时内运行批处理任务的信息统计情况. 这个数据集共有 34762 条数据,其中包含了任务开始与结束的时间戳、任务 ID、CPU 利用率、内存利用率等特征,以及所对应的任务运行时间.

为了验证 CBN 预测的准确性,选择任务 ID、最多 CPU 利用个数、平均 CPU 利用个数、最大内存利用情况以及平均内存利用情况,作为虚拟机待分类特征,分别表示为 *job_id*, *real_cpu_max*, *real_cpu_avg*, *real_mem_max*, *real_mem_avg*. 同时,用虚拟机上任务的响应时间 *response_time* 作为待预测的虚拟机性能. 我们将所有虚拟机性能数据分为 20 个类别,分别表示为 $C_1,C_2,\cdots,C_{20}$. 采用十折交叉验证的思想,将数据均分为 10 份,轮流将其中 9 份作为训练数据集来建立 CBN 模型,一份作为测试数据集,进行模型构建和验证. 每次验证都能得到对应的虚拟机性能预测准确率. 我们把实验分为以下两类,首先按照特征取值对虚拟机分类,采用随机森林的分类结果与决策树的分类结果做对比,验证随机森林模型对虚拟机分类的准确性. 其次,将 CBN 对虚拟机性能的预测结果和线性模型的结果做对比,说明 CBN 对于虚拟机性能预测的准确性.

4.1 随机森林算法分类准确性

为了验证随机森林算法对虚拟机分类的准确性,首先根据 BatchInstance 数据集中的 *response_time* 的大小,人为地将不同虚拟机特征配置分为 20 个类别. 基于十折交叉验证的思想[16],将数据均分为 10 份,轮流将其中 9 份作为训练数据,1 份作为测试数据,基于这 10 组数据分别运行随机森林算法和决策树分类算法,并统计两个算法分类的准确率. 图 3 是两个算法分类的准确性对比情况,横坐标表示实验结果,纵坐标表示准确率.

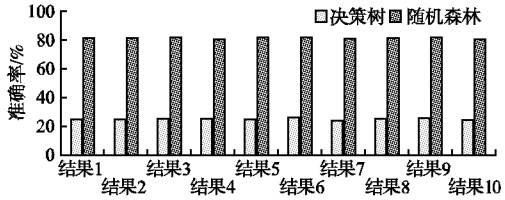

图 3 随机森林和决策树的平均分类准确率

Fig. 3 Classification accuracy rate of random forest and decision tree

从图 3 中,可以看出,随机森林算法在进行虚拟机分类时,准确率平均可以达到 80.87%,而决策树的分类准确率平均只有 25.05%. 由此可以说明我们提出的基于随机森林的虚拟机分类算法具有较高的准确率. 另外,十组数据中,随机森林算法分类准确率最大值与最小值之差为 1.1%,而决策树算法为 1.9%. 由此可以说明随机森林算法对数据的适应性更高.

为了验证随着数据量的增长,两个模型的分类准确率是否有所变化,我们做如下测试:将数据集以 5000 为一个增长单位,依次验证数据量为 5000、10000、15000、20000、25000、30000 条时,随机森林和决策树对虚拟机分类的准确性. 结果如图 4 所示.

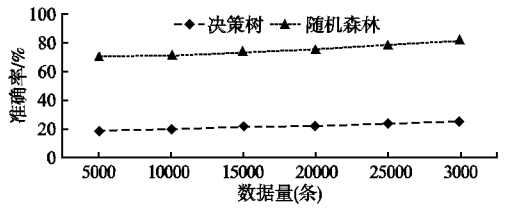

图 4 分类模型准确率与数据量变化关系对比

Fig. 4 Relationships between classification model and amount of data

从图 4 中可以得出,对虚拟机分类时,随机森林算法和决

[3] Alibaba ClusterData, https://github.com/alibaba/clusterdata, 2018/08/27.

策树算法分类的准确率均随着数据量的增长而增长. 随着数据量从5000 增长到30000,随机森林算法的分类准确率提高了10.8%,而决策树算法的分类准确率提高了6.4%.

4.2 CBN 模型预测准确性

为了验证CBN 模型预测虚拟机性能的准确性,我们用CBN 的预测结果和线性模型的预测结果对比说明. 图5 为基于十折交叉验证,用CBN 和线性模型来预测虚拟机性能的准确率. 从图5 可以得出,线性模型在预测虚拟机性能时,平均准确率只能达到22.7%. 而CBN 模型在预测虚拟机性能时,平均准确率达到76.36%.

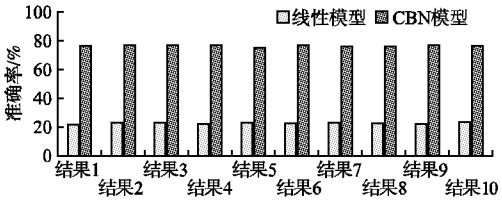

图5　CBN 模型和线性模型预测结果准确率

Fig. 5　Prediction accuracy rate of CBN and linear model

同样地,将数据量分别设置为5000、10000、15000、20000、25000、30000 条时,对比CBN 和线性模型的虚拟机性能预测准确性. 结果如图6 所示.

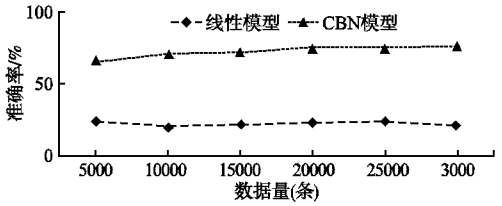

图6　性能预测准确率与数据量变化对比

Fig. 6　Relationship between accuracy of CBN and amount of data

图6 中可以看出,CBN 模型对于虚拟机性能的预测准确率远高于线性模型,且准确率随着数据量的增长而增长. 当数据量从5000 增加到30000 时,CBN 模型的预测准确率提高了12%. 而线性模型的预测准确性却没有太大改变,最高的预测准确率只达到了24%.

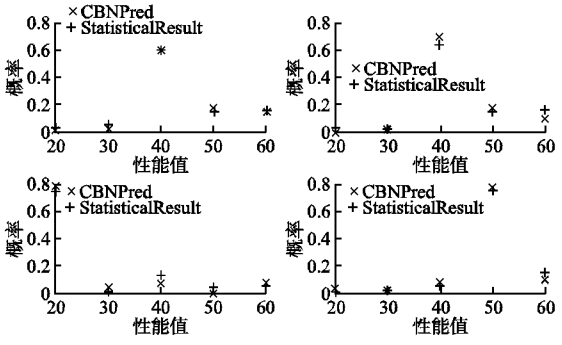

图7　CBN 预测概率分布与实际概率分布

Fig. 7　CBN predictions and experimental data distribution

由于CBN 可以预测出虚拟机性能的波动情况,我们随机选择四组数据,用CBN 推理出各个可能取得性能值的概率分布情况,并用推理结果和实际数据的分布情况作对比,验证CBN 预测的虚拟机性能概率分布是否和实际情况相符合. 具体情况如图7 所示,图中性能值为虚拟机上应用的运行时间.

从图7 中可以看出,CBN 预测虚拟机性能的结果,与实际数据的分布情况相符合.

为了验证CBN 预测的虚拟机性能分布情况是否与实际的数据分布情况相符,分别计算以上四组数据中,最大概率估计值相对于从实际数据中统计得到的概率值的误差和相对误差.

误差的计算方式为 $error = P_C - P_S$,而相对误差的计算方式为 $relative\ error = |P_C - P_S|/P_S$,$P_C$ 表示由CBN 模型所预测得到的性能概率值,P_S 表示从实际数据统计出来的性能概率值. 计算结果如表3 所示,为了区分4 组数据,使用1,2,3,4 为4 组数据编号.

表3　CBN 预测结果的误差与相对误差

Table 3　Error and relative error of CBN predictions

数据集	response time /s	P_C	P_S	error	relative error
1	40	0.61	0.6	0.01	1.67%
2	40	0.7	0.66	0.06	9.01%
3	20	0.79	0.75	0.04	5.33%
4	50	0.77	0.75	0.02	2.66%

从表3 中可以看出CBN 对于虚拟机性能分布概率的预测误差中,相对误差最低的只有1.67%,而最大的误差达到9.01%. 通过计算,可以得出CBN 预测的平均误差为:0.0325,而相对误差的平均值为:4.67%.

通过以上实验,验证了我们所提出方法在虚拟机分类,以及虚拟机性能预测时的有效性和准确性.

5　总结与展望

本文针对虚拟机特征与性能之间存在不确定性依赖关系,导致虚拟机性能难以被准确预测的问题,提出了一种基于带分类参数的BN 模型,来实现虚拟机性能预测. 该模型克服了传统BN 对于训练集中未出现过的数据,无法进行预测的局限性,以及当虚拟机特征节点过多时,性能预测的复杂度增加、准确率下降的情况. 实验结果表明本文提出的方法预测虚拟机性能的高效性和准确性.

然而,本文提出的随机森林模型是将虚拟机作为一个整体进行分类,针对虚拟机不同类型的特征差异性所导致的分类误差尚未充分考虑. 因此,在后续方案中,我们考虑将多个虚拟机的分类细化,以实现更加准确地预测虚拟机的性能.

References:

[1] Danilov A, Andersen J, Molodkina E, et al. The NIST definition of cloud computing[J]. Communications of the ACM, 2011, 53: 50. doi: 10.6028/NIST.SP.800-145.

[2] Expósito R R, Taboada G L, Ramos S, et al. Performance evaluation of data-intensive computing applications on a public IaaS cloud [J]. Computer Journal, 2018, 59(3): 287-307.

[3] Zhao Hong-wei, Shen De-rong, Tian Li-wei. Research on resources forecasting and scheduling method in cloud computing environment [J]. Journal of Chinese Computer Systems, 2016, 37(4): 659-663.

[4] Menon A, Santos J R, Turner Y, et al. Diagnosing performance overheads in the xen virtual machine environment[C]//International Conference on Virtual Execution Environments, 2005: 13-23.

[5] Iosup A, Ostermann S, Yigitbasi N, et al. Performance analysis of cloud computing services for many-tasks scientific computing[J]. IEEE Transactions on Parallel & Distributed Systems, 2011, 22(6): 931-945.

[6] Wang Sa, Zhang Wen-bo, Wu Heng, et al. Approach of quantifying virtual machine performance interference based on hardware performance counter[J]. Journal of Software, 2015, 26(8): 2074-2090.

[7] Wu Yi-hua, Cao Jian, Li Ming-lu. Energy efficient allocation of virtual machines in cloud computing environments based on demand forecast[J]. Journal of Chinese Computer Systems, 2013, 34(4): 778-782.

[8] Li Feng-ze, Yang Da, Zhou Peng, et al. Modeling application performance in a virtualized environment [J]. Computer Systems & Applications, 2015, 24(9): 9-15.

[9] Hammer H L, Yazidi A, Begnum K. An inhomogeneous hidden Markov model for efficient virtual machine placement in cloud computing environments[J]. Journal of Forecasting, 2017, 36(4): 407-420.

[10] Kousiouris G, Cucinotta T, Varvarigou T. The effects of scheduling, workload type and consolidation scenarios on virtual machine performance and their prediction through optimized artificial neural networks[J]. Journal of Systems & Software, 2011, 84(8): 1270-1291.

[11] Kundu S, Rangaswami R, Dutta K, et al. Application performance modeling in a virtualized environment[C]//International Symposium on High Performance Computer Architecture, 2010: 1-10.

[12] Hao J, Zhang B, Yue K, et al. Performance measurement and configuration optimization of virtual machines based on the Bayesian network[C]//International Conference on Cloud Computing and Security, 2017: 641-652.

[13] Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference[J]. Computer Science Artificial Intelligence, 1988, 70(2): 1022-1027.

[14] Breiman L. Random forest[J]. Machine Learning, 2001, 45(1): 5-32.

[15] Wu Chen-wen, Wang Wei, Li Chang-sheng, et al. One feature selection method combined random forest with neighborhood rough set [J]. Journal of Chinese Computer Systems, 2017, 38(6): 1358-1362.

[16] Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection[C]//International Joint Conference on Artificial Intelligence, 1995: 1137-1143.

附中文参考文献：

[3] 赵宏伟，申德荣，田力威. 云计算环境下资源需求预测与调度方法的研究[J]. 小型微型计算机系统，2016，37(4)：659-663.

[6] 王 卅，张文博，吴 恒，等. 一种基于硬件计数器的虚拟机性能干扰估算方法[J]. 软件学报，2015，26(8)：2074-2090.

[7] 吴毅华，曹 健，李明禄. 云计算环境下基于需求预测的虚拟机节能分配方法研究[J]. 小型微型计算机系统，2013，34(4)：778-782.

[8] 黎丰泽，杨 达，周 鹏，等. 虚拟环境下虚拟机应用性能建模[J]. 计算机系统应用，2015，24(9)：9-15.

[15] 吴辰文，王 伟，李长生，等. 一种结合随机森林和邻域粗糙集的特征选择方法[J]. 小型微型计算机系统，2017，38(6)：1358-1362.