一种位置服务测试数据生成技术研究

侯可佳 黄 军 白晓颖 周立柱
(清华大学计算机科学与技术系 北京 100084)
(清华大学信息科学与技术国家实验室 北京 100084)

摘 要 随着移动互联网的高速发展,移动服务越来越成为人们生活、工作中不可或缺的一部分,绝大部分移动服务都要以用户的位置信息为基础,位置服务(Location-Based Services,LBS)则通过调用 GPS 等服务在确定用户位置的基础上,向用户提供导航、服务推荐等功能.位置服务的基础性使得其正确性和完整性至关重要,同时其开放性和复杂性对测试提出了更高的要求.文中以反地理编码服务为例,探讨了位置空间测试数据的生成方法.基于模拟退火算法(Simulated Annealing,SA)的优化过程,提出了两种适应度函数定义方法,一种是基于有效点密集区假设,以提高位置空间覆盖率为优化目标;另一种是基于 LBS 故障模式假设,以提高潜在故障的检测能力为优化目标.研究引入了贝叶斯分类器,将预测值作为探索点的能量值,指导搜索过程.研究选取了实际应用的位置服务平台进行了实验验证,实验表明,与简单随机算法相比,该方法在可容忍的时间代价下,可以有效提高测试效率和错误检测能力.

关键词 移动平台服务;位置服务;朴素贝叶斯理论;数据生成;模拟退火算法
中图法分类号 TP311 **DOI 号** 10.11897/SP.J.1016.2016.02161

Geographical Data Generation for Testing Location-Based Services

HOU Ke-Jia HUANG Jun BAI Xiao-Ying ZHOU Li-Zhu
(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)
(*National Laboratory for Information Science and Technology, Tsinghua University, Beijing* 100084)

Abstract With the rapid development of mobile cloud, mobile services have been an integrated part of people's daily life and work. Location-Based Services (LBS) provide information services, like navigation and recommendation, for given physical locations gathered through mobile network. It is critical enabling technique for more and more mobile applications. Hence, the correctness and completeness of LBS are important for mobile services. However, due to its openness and complexity, LBS testing faces new challenges. Taking reverse-GeoCoding service as an example, the paper investigates methods for geographic test data generation. Based on simulated-annealing method, two algorithms are proposed for the definition of fitness functions. One is based on the assumption of intensive area of valid positions, with spatial coverage as the optimization objective. The other is based on the assumption of LBS fault models, with the optimization objective to enhance defects detection probabilities. To guide effective search process, the energy in the annealing algorithm is defined by the predicted probability using Naïve Bayes classifier. Experiments are exercised on real mobile service platforms, which show encouraging results that the proposed methods can enhance test effectiveness and defect detection capabilities with tolerable time cost.

Keywords mobile platform service; location-based service; Naïve Bayes; test generation; simulated annealing

收稿日期:2015-03-30;在线出版日期:2015-11-06.本课题得到国家自然科学基金(61472193)、北京市自然科学基金(4132062)、国家"九七三"重点基础研究发展规划项目基金(2011CB302505)、国家"八六三"高技术研究发展计划项目基金(2013AA01A215)资助.**侯可佳**,女,1983 年生,博士研究生,主要研究方向为软件测试.E-mail:houkejia01@163.com.**黄 军**,男,1991 年生,硕士研究生,主要研究方向为软件测试、移动服务计算.**白晓颖**,女,1973 年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为软件测试、服务计算.**周立柱**,男,1947 年生,硕士,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为数据库技术、万维网搜索、万维网信息集成.

1 引 言

移动云计算作为一种计算模型,能够实现对移动设备计算能力透明、弹性的扩展,通过无处不在的无线网络访问云存储和计算资源[1].移动云计算为用户提供了强大的计算及数据存储能力,降低了对移动计算终端在计算能力和存储能力等方面的要求,拓展了移动终端的业务范围.由于移动计算终端的移动特性,使得位置信息成为各种移动应用正常运行的基础,移动应用能够利用用户的位置信息为用户提供路径导航、服务推荐等功能.目前,很多移动平台均提供开放的位置服务(Location Based Service,LBS),例如在国内得到广泛应用的 Baidu、Tencent 和 Amap 等,每个服务提供商均面向 Web 和手机终端提供两种不同的接口服务.不同服务提供商提供的服务之间以及同一个服务提供商提供的两种不同服务版本之间,均可能存在较大差异.不同服务提供的位置查询服务在查询范围和精度上会存在差异,例如服务可以提供在某个矩形或者圆形区域的位置查询,或者提供在某个行政区域划分内的位置查询.移动应用的位置服务质量也会在正确性、完整性、时效性等方面存在较大差异[2-3].我们用 1000 组随机生成的位置数据对上述服务提供商所提供的位置服务进行了初步测试,通过对比发现:对于任意两个服务,其结果均存在显著差异(差异能达到 10%以上).移动位置服务具有如下特性:

(1)位置服务需要具备快速响应能力.当用户在移动过程中使用应用服务时,位置服务必须能够实时显示出用户的位置变化.

(2)位置服务必须能够提供轻量级的数据信息.移动终端的屏幕、内存均比较小,移动通信网络带宽窄并且费用昂贵,因此位置服务提供的位置信息必须进行压缩和精简,去除不必要的信息,为用户节省费用及等待时间,提高用户使用体验.

位置服务的基础性,使得其正确性和完整性至关重要,同时,平台服务的开放性和复杂性又对传统的测试技术提出了新的挑战:

(1)位置服务测试工作量大.位置服务具有广阔的输入空间,即使在一个较小的区域内,仍然存在无数个不同的位置点,因此,如何定义地理区域覆盖率,如何生成和选择优质的测试数据以发现更多的软件缺陷,就成为位置服务测试的一个难题.

(2)位置服务的运行结果难以评价.位置服务的运行结果依赖于服务的位置信息数据库,数据库为位置点存储与之对应的经纬度地理编码、行政地址编码等相关信息,数据库的正确性和完整性将直接影响位置服务的运行结果,测试人员无法获得完整、准确的数据库信息,因此难以对位置服务的运行结果进行评价.

(3)位置服务需要持续在线测试.由于位置地理信息的不断变化,位置服务需要及时更新、升级.例如在城市中,某一地点的经营主体和内容会发生频繁变化,或者某个经营主体由于搬迁使其地址信息发生了变化,再如新建建筑也会使地址信息发生变化.

针对上述问题,本文将位置服务的测试生成问题转化为优化搜索问题,分别以区域覆盖和故障检测为优化目标,生成一系列位置数据作为位置服务的输入参数,这些输入数据能够对区域达到一定的覆盖率要求,或者有较大概率能够发现服务的隐藏缺陷.本文提出的算法将建立在有效点聚集假设和缺陷密集假设的基础上,利用已生成数据推测未生成数据的概率,并指导搜索过程的推进.

本文第 2 节介绍位置服务和模拟退火算法两方面的背景知识;第 3 节对位置服务测试问题进行形式化定义,并提出基于贝叶斯分类器的模拟退火搜索算法,利用该算法从有效点生成效率和故障探测效率两方面探讨算法的具体实现;第 4 节报告实验的开展和结果评价;第 5 节对相关工作进行总结和讨论;第 6 节是对本研究工作的总结讨论.

2 背景知识

2.1 位置服务

位置服务,又称为定位服务,以 GPS 服务为基础,是一种典型的移动平台服务,服务被部署在移动云平台上,利用强大的云计算能力为用户提供位置查询、兴趣点查询、位置导航、地理编码及反地理编码等与位置相关的服务接口.目前几乎所有的移动应用程序都需要利用位置服务来辅助其完成功能,或者通过位置服务来拓展业务范围,提升服务质量.例如用户可以在聊天软件中调用位置服务确定自己所处的位置,然后发送给朋友,免去了手动输入地址的繁琐和可能出现的错误;再如移动应用可以在获取用户所处的位置后,为其按照距离远近推荐餐饮、娱乐等用户需要的服务.由此可见,位置服务已经渗

透到了我们生活的方方面面，为我们的工作和生活提供了便利.

基于移动云计算的位置服务通过大量接口为用户提供服务，其工作过程如下：首先移动终端上的应用程序向云服务提供商发送请求来调用他们的接口；其次服务提供商接收请求，利用其强大的计算能力处理用户请求，并将结果返回给移动用户. 位置服务的运行依赖于服务内部的编码数据库，该数据库中存储了物理位置信息，也就是位置的经纬度信息与行政位置信息，例如某个城市某条街道之间的映射关系，将行政位置信息转化为物理位置信息的服务为地理编码服务，将物理位置信息转化为行政位置信息的服务为反地理编码服务. 图 1 展示了反地理编码服务的一般工作过程，该服务将从 GPS 定位服务获取的物理位置信息，即经纬度坐标转化为相应的行政地址信息，例如经纬度坐标(29.919136379842868, 117.05622844224358)指示的地点为“安徽省池州市东至县安庆高速公路”，经纬度坐标(28.30449900496494, 113.93426589494133)指示的地点为“湖南省长沙市浏阳市官渡镇桃树坪”. 该服务以经纬度为输入参数，这些输入参数可以手工直接输入，也可通过 GPS 服务获得.

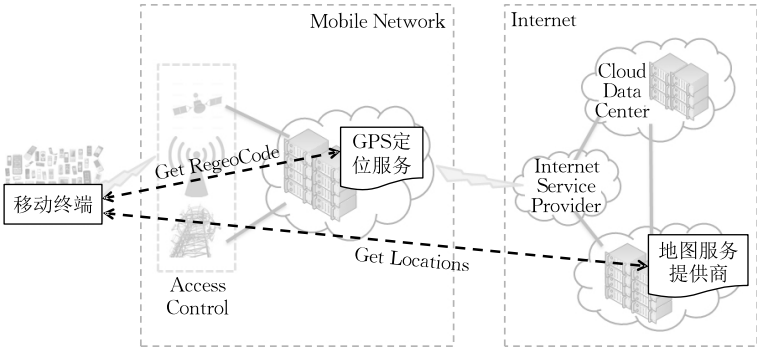


图 1　反地理编码服务

2.2　模拟退火算法

模拟退火算法(Simulated Annealing，SA)[4-5]的基本思想是模拟固体退火过程. 固体退火是将固体加热，使其温度达到充分高，随着温度的升高，固体内部的粒子逐渐变为无序状态，然后再将固体徐徐降温，在降温的过程中，固体内部的粒子逐渐趋于有序状态，在每个温度都达到平衡，直至常温，固体达到基础状态. 类比优化搜索过程，在一定温度下，搜索从一个状态随机地变化到另一个状态，随着温度的不断降低，直至最低温度，搜索过程以接近 1 的概率停留在最优解. SA 算法是对爬山算法[5]的优化改进，爬山算法在搜索过程中只接受更优的邻近解，直至搜索不到更优解为止，算法简单易实现，但是却极有可能停止在局部最优解. SA 算法引入一个温度参数，当搜索到较差的邻近解时，利用温度参数和目标函数值之差共同确定一个概率参数，利用此概率参数决定是否接受较差的邻近解，概率参数 P 的定义为

$$P=\mathrm{e}^{-\frac{\delta}{t}},$$

其中，δ 为邻近解与当前解的目标函数之差，t 为温度参数，该参数对应于固体退火过程中的温度，随着搜索过程的不断推进而不断减小，直至算法达到终止条件. 在温度下降足够慢时，算法找到全局最优解的概率接近 1.

3　基于模拟退火的位置测试数据生成

本文对位置服务的测试方法如图 2 所示，从图中可以看出，该方法将服务的运行结果作为输入数据生成的反馈信息，为进一步生成输入数据提供参考信息和指导作用. 其中“位置服务平台”节点可能包含一个或多个位置服务，“服务结果评价”则根据不同的测试目的，从不同角度对服务的运行结果进行评价. “贝叶斯分类器”用来预测在某个区域中取到特定类型位置点的概率，服务运行结果和贝叶斯分类器共同为搜索的进一步优化提供指导.

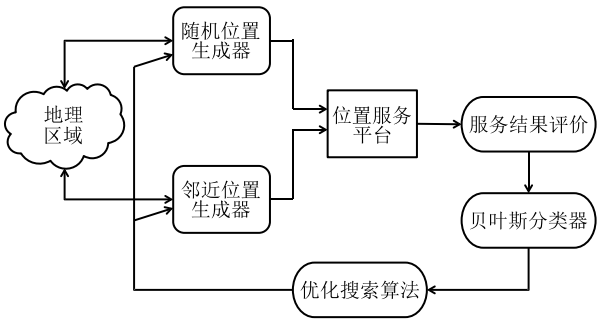


图 2　位置服务测试方法

3.1 问题定义

对一个给定的地理区域 D，$S=\{s_1,s_2,\cdots\}$ 是 D 中一系列位置点的集合，D 的覆盖区域通过经纬度的范围来限定，即 $D.rangeLng=[Lng_l,Lng_h]$，$D.rangeLat=[Lat_l,Lat_h]$，其中 Lng_l 和 Lat_l 分别是经度和纬度的下限，Lng_h 和 Lat_h 则分别为经度和纬度的上限. $s_i.lng$ 为点 s_i 的经度值，$s_i.lat$ 为点 s_i 的纬度值，对于 $s_i\in S$，则有 $Lng_l\leq s_i.lng\leq Lng_h$，并且 $Lat_l\leq s_i.lat\leq Lat_h$.

$L=\{l_1,l_2,\cdots\}$ 为一系列开放的位置服务，这些服务以位置点 s_i 为输入，我们将服务 l_j 对于输入 s_i 的返回结果定义为 $R(l_j,s_i)$.

根据服务测试的不同目的需要为服务生成不同类型的测试数据，这些数据具有不同的属性特征，我们通过示性函数 $value(s)$ 来评价数据的属性特征，函数的不同定义能够表示不同的数据属性，根据函数取值的不同能够实现对测试数据的划分. 本研究的目标就是针对示性函数 $value(s)$ 的某个特定取值 v，以尽量少的代价生成尽量多的数据，使得 $value(s)=v$. 在算法执行过程中，已生成测试数据能够指导测试生成过程的执行，根据已生成数据 s_1，$s_2,\cdots,s_j$ 的示性函数值来预测即将生成的数据 s_{j+1} 是否同样能够满足 $value(s_{j+1})=v$ 的要求，从而大幅提高特定数据的生成效率.

3.2 模拟退火算法生成过程

测试生成算法对输入域进行划分，根据测试目的为位置点定义属性特征，基于朴素贝叶斯理论(Naïve Bayes)[6]对位置点进行分类，并计算位置点的类别概率，并利用此概率定义位置点的能量值函数，利用能量函数引导模拟退火算法的进行. 假设被测服务的输入域是地理区域 D，测试生成的过程如下：

(1)将 D 等分为 $n\times m$ 个小的子区域，每个子域为 $D_{ij}(0<i\leq n,0<j\leq m)$，用 $Matrix(D,n,m)$ 表示这样的划分.

(2)按照不同的测试目的，为区域内的点 s，定义示性函数 $value(s)$，利用示性函数定义位置点 s 的属性，

$$value(s)=\begin{cases}1, & s\text{ 具有某属性特征时}\\ 0, & \text{否则}\end{cases}.$$

(3)利用示性函数定义贝叶斯分类器(Naïve Bayes Classifier，NBC)，测试目的不同，则贝叶斯分类器的含义不同，具体含义将在后序章节的具体实现中介绍：

$$P(D_{ij}\mid value(s)=1)=\sum_{s\in D_{ij}}value(s)\Big/\sum_{s\in D}value(s),$$

$$P(value(s)=1)=\sum_{s\in D}value(s)/|D|,$$

$$P(D_{ij})=|D_{ij}|/|D|,$$

$$P(value(s)=1\mid D_{ij})=P(D_{ij}\mid value(s)=1)\times P(value(s)=1)/P(D_{ij}).$$

(4)利用贝叶斯分类器确定在某个区域取到某一类位置点的概率，以每个位置点的概率值定义其能量值，采用模拟退火算法生成测试数据，以能量值函数作为搜索算法的目标函数，指导算法搜索过程的推进. 对于区域 D_{ij} 内已探索的 q 个点 $History=\{s_1,s_2,\cdots,s_q\}$，搜索得到 s_q 的邻近点 s_{q+1}，且 $s_{q+1}\in D_{ij}$，则其能量值的定义为 $Energy(s_{q+1})=value(s_{q+1})\times P(value(s_{q+1})=1\mid D_{ij})$，能量值的变化为

$$\Delta Energy=Energy(s_{q+1})-Energy(s_q).$$

模拟退火算法根据能量值的变化来确定下一步搜索是继续在本区域进行还是进入新的未探索区域重新开始搜索. 如果能量值增大，则继续在本区域搜索；如果能量值减小，则以一定概率继续在本区域进行搜索，否则在其余未探索子区域中选取 D_{ij} 重新开始搜索，并且在所有未探索子区域中 $P(value(s)=1\mid D_{ij})$ 最大. 算法的具体过程如下：

算法 1. 生成具有某特定属性的测试数据.

输入：指定区域 D，用经纬度范围表示

输出：指定区域选取的若干点的经纬度值

```
Select a set of points History={s1, s2, ..., sq} in D randomly
Select an initial temperature t>0
REPEAT
  Divide D into n×m sub-areas D[1][1], D[1][2], ..., D[n][m]
  NBC training by the set History
  REPEAT
    Select D' from D, where D' is not searched and
      P(value(s)=1|D') ≥
      (P(value(s)=1|D[i][j]) | 0<i≤n, 0<j≤m)
    Select a point s from D' randomly
    S←s; e←Energy(S)
    it←0
    REPEAT
      Snew←Neighbor(S)
      ΔEnergy=Energy(Snew)−Energy(S)
      IF (ΔEnergy>0)
        THEN S←Snew, Update History and
        P(value(s)=1|D)
        ELSE Generate random number r, 0≤r<1
```

IF ($e^{-\frac{\Delta Energy}{t}} < P(value(s)=1|D')$)
THEN $S \leftarrow Snew$,
Update $History$ and
$P(value(s)=1|D')$
ELSE Break
ENDIF
ENDIF
$it \leftarrow it+1$
UNTIL enough data generated in D' or it reached the threshold
UNTIL enough data in D are generated
Update $P(value(s))$, where $s \in D$
$D \leftarrow D[i][j]$, where $P(value(s)=1|D[i][j])$ is the maximum
$History \leftarrow$points already searched in D
Decrease t according to cooling schedule
UNTIL t reached the threshold or D is small enough

在算法 1 中包含 3 层循环，最内层循环表示在一个子域中的搜索过程，由能量值的变化来引导，每生成一个令 $value(s)=1$ 的点将增加该区域的置信度，而生成一个 $value(s)=0$ 的点则会降低该区域的置信度. 本层循环终止有 3 种情况：(1) 不接受新生成的较差解；(2) 在该子域中取到足够多的点；(3) 在该划分层次的循环次数已达到最大限制. 中间层循环表示对区域 D 的各个子域进行搜索，直到在 D 中取得了足够的点. 最外层循环则是对子域进行迭代划分的过程，本层循环的终止条件包含两种情况：(1) 温度参数减小到预设定的阈值；(2) 区域划分细化到了预定程度.

3.3　以区域覆盖为优化目标的算法

对被测服务输入域的覆盖分为两个算法来完成，首先是对有效点密集区的探索，该算法的目的是要尽可能多的找到有效点，有效点聚集区的确定是后续测试生成的基础，使测试生成能够直接在算法确定的有效点聚集区进行，大大提升了有效点的生成概率. 如果在探索有效点聚集区的过程中已经生成了足够多的有效点，也可以直接利用这些数据对服务进行测试. 利用有效点对多个位置服务接口进行测试，通过对比不同接口的返回结果，评价服务的正确性.

图 3 为探索有效点密集区算法执行过程中一个

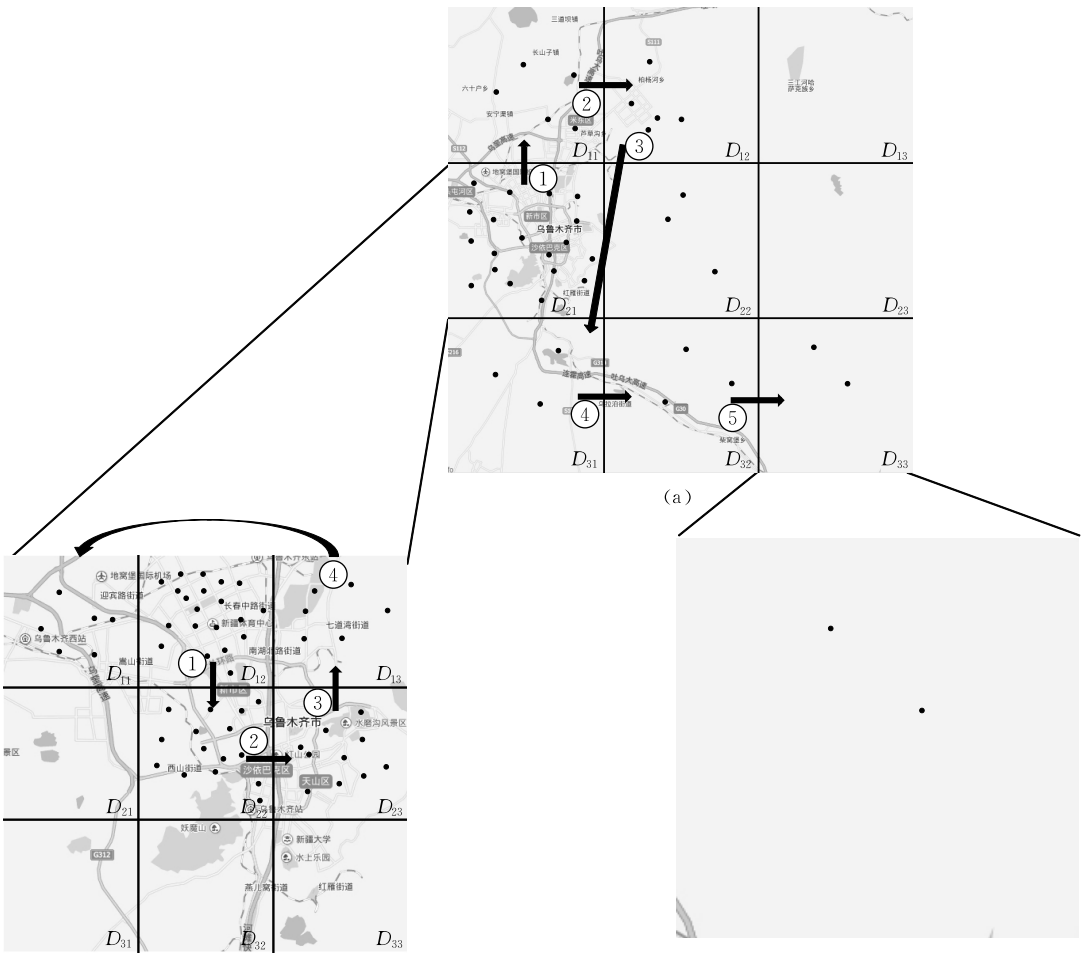


图 3　算法执行示意图

片段的示意图,图中(a)表示的地理范围最大,包含乌鲁木齐及其周边地区,将该地区划分为 3×3 个子域,包含有效点密集区和稀疏区,从图中可以看出,该算法能够在有效点密集区取得更多的点,从而保证生成参数的质量.图中的箭头表示搜索路径,说明对各个子域的搜索顺序.其中 D_{21} 为有效点密集区,在该子域中得到的点最多,(b)即是对 D_{21} 进一步划分得到的结果,在对此区域进一步划分后,能够看出在此区域仍然可以区分出有效点密集区和稀疏区,同样的,在有效点密集区能够取得更多的点.(c)显示了(a)中 D_{33} 区域放大后的情况,从图中可以清楚的看出该区域为有效点稀疏区,因此不对该区域进行进一步的划分.

其次是对有效点稀疏区的探索,有效点稀疏区域往往包含很少的有效点,因此服务也常常会在这样的区域遗留数据空白,导致服务数据信息的不完整.该算法的目的是尽可能广的覆盖有效点稀疏区,利用有效点稀疏区内取的点来检验服务信息的完备程度.算法认为在有效点密集区选取少量的点即可代表该区域,而在有效点稀疏区却需要选取大量的点对服务进行测试,以发现服务的潜在问题.该算法与探索有效点聚集区算法存在以下明显区别:

(1)此算法搜索过程朝能量值减少的方向进行.当算法认为当前搜索区域为有效点聚集区时,则在剩余的子域中选取一个无效点概率最大,并且与当前搜索区域距离最远的区域进行搜索.

(2)子区域探索完成的标准包括两种:① 算法认为该区域为有效点聚集区;② 算法认为该区域为有效点稀疏区,并且在该区域已探索足够多的点.

3.3.1 有效点聚集区假设

在地球上存在人口稠密的人类聚集区,也存在人口稀疏区,这两类区域在经纬度定义方面是一致的,但在行政地址编码方面却存在着巨大差异:

(1)人口稠密区域,例如城市,行政地址编码的区域粒度较小,距离很近的两个建筑通常都会有不同的地址信息,也就是说经纬度的较小改变也会使服务返回不同的输出结果,我们称这样的区域为有效点聚集区.

(2)人口稀疏区域,例如沙漠、戈壁、海洋等,通常不存在行政地址编码信息,也就是说服务返回结果为空.由于这样的区域通常非常广阔,就导致相差巨大的两组输入参数也可能返回相同的执行结果,我们称这样的区域为有效点稀疏区,例如太平洋,纬度跨过 151 度,经度横跨 177 度,在这个区域内就可能生成大量导致输出结果为空的数据.

(3)有效点稀疏区域也可能存在小范围的有效点聚集区,例如在太平洋中分布着众多岛屿,这些岛屿包括人口稀疏岛屿和人口稠密岛屿,人口稠密岛屿则必然是有效点聚集区,例如台湾岛就是太平洋中的一个人口稠密岛屿.

区域覆盖算法建立在两个基本假设的基础上:(1)在某区域内发现的有效点越多,则该区域越有可能是有效点聚集区域;(2)有效点密集区附近的区域也极有可能是有效点密集区.

3.3.2 贝叶斯分类器设计

区域覆盖算法主要关注服务对一个给定的点是否能够返回一个有效的地址信息,因此对于每个选定的点 s,定义示性函数如下:

$$value(s)=\begin{cases}0, & s.addr=\text{null}\\ 1, & s.addr!=\text{null}\end{cases},$$

其中 $s.addr$ 为点 s 的行政地址编码,为被测服务反地理编码服务运行的输出结果.对于点 s,如果返回的地址信息不为空,则示性函数 $value(s)$ 的返回值为 1,说明该点为有效点;否则为 0,说明该点为无效点.根据某个点所处的区域,利用贝叶斯分类器能够预测该点为有效点/无效点的概率.为预测有效点概率,贝叶斯分类器的具体设计如下:

(1)在区域 D 中随机选取 q 个点作为训练样本,并且对于划分 $Matrix(D,n,m)$,落在每个子域中的点的个数相同,即 $|D_{11}|=|D_{12}|=\cdots=|D_{nm}|$,并且 $|D_{11}|+|D_{12}|+\cdots+|D_{nm}|=q$,其中 $|D_{ij}|$ 表示在子域 D_{ij} 中选取的点的个数;

(2)计算样本中有效点落在区域 D_{ij} 中的概率 $P(D_{ij}|value(s)=1)=\sum_{s\in D_{ij}}value(s)\Big/\sum_{s\in D}value(s)$;

(3)计算训练样本中有效点的概率

$$P(value(s)=1)=\sum_{s\in D}value(s)/|D|;$$

(4)计算训练样本中的点落在区域 D_{ij} 中的概率 $P(D_{ij})=|D_{ij}|/|D|$;

(5)根据以上计算,预测区域 D_{ij} 中点 s 为有效点的概率

$$P(value(s)=1|D_{ij})=P(D_{ij}|value(s)=1)\times P(value(s)=1)/P(D_{ij}).$$

由于无效点的示性函数值为 0,因此定义无效点的贝叶斯分类器时需要借助有效点的数量,具体定义为

$$P(D_{ij}|value(s)=0)=\Big(|D_{ij}|-\sum_{s\in D_{ij}}value(s)\Big)\Big/\Big(|D|-\sum_{s\in D}value(s)\Big),$$

$$P(value(s)=0)=1-\sum_{s\in D}value(s)/|D|,$$

$$P(D_{ij})=|D_{ij}|/|D|,$$

$$P(value(s)=0|D_{ij})=P(D_{ij}|value(s)=0)\times P(value(s)=0)/P(D_{ij}),$$

其中 $P(D_{ij}|value(s)=0)$为无效点落在 D 中的先验概率，$P(value(s)=0)$为训练样本中无效点的概率，$P(value(s)=0|D_{ij})$是区域 D_{ij} 中的点 s 为无效点的后验概率.

3.4　以故障检测为优化目标的算法

测试人员并没有关于全国甚至全球所有地理坐标及其行政地址编码信息的数据库，因此当服务对某一组输入数据返回执行结果后，测试人员也无法判断其结果是否正确，测试是否通过. 为判定测试结果，本文引入投票机制，不同服务提供商提供的反地理编码服务具有功能上的等价性，在互联网上能够形成互为备份的容错体系架构，服务的运行结果具有可比性，用户可以对不同服务的结果进行对比判断，从中选取用户认为最优的结果. 利用相同的输入数据调用不同的反地理编码服务，将这些服务的运行结果进行对比，如果所有服务的运行结果均相同，则认为这些结果均正确，如果存在不同的返回结果，则认为存在隐藏故障，需要进一步研究分析. 本文将服务故障总结为以下 3 类：

(1) SF_1：对于同一个位置点 x，服务 A 返回结果为空，认为其为无效点，而服务 B 返回结果为一个有效的行政地址，即认为该点为有效点；

(2) SF_2：对于同一个位置点 x，被测服务的返回值均不为空，但服务 A 返回结果为服务 B 返回结果的子串，也就是说服务 A 的结果指示的行政区域较服务 B 的结果范围更大，而服务 B 的返回结果指示的地址更加精确；

(3) SF_3：对于同一个位置点 x，被测服务的返回值均不为空，但服务 A 和服务 B 的返回结果不同，并且此错误类型不包含 SF_2 的情况.

3.4.1　位置服务故障模型假设

故障驱动算法要以尽量少的测试数据发现尽量多的服务潜在缺陷，我们称能够发现服务潜在缺陷的数据为缺陷数据. 位置点的行政地址编码具有一定的区域连续性，例如“北京市海淀区清华大学南楼 11 号楼”和“北京市海淀区清华大学南楼 12 号楼”是位于清华大学校内南北相邻的两栋楼，地址编码也是相邻的，如果某位置服务为其中一个地点的返回结果是错误的，则为另一个地点的返回结果也极有可能是不正确的. 由此我们可以为位置服务的故障模型提出如下假设：

(1) 缺陷密集假设，我们认为缺陷数据存在一定的聚集效应，也就是说如果点 s 为缺陷数据，则其附近的点为缺陷数据的概率也较高；

(2) 对于某个区域 D_{ij}，如果在其中找到的缺陷数据占比很低，则认为在该区域搜索到缺陷数据的概率也很低，应该转移到其他区域进行搜索.

3.4.2　贝叶斯分类器设计

根据故障驱动算法的目的，将位置点 s 的示性函数定义为

$$value_F(s)=\begin{cases}1, & \exists l_i,l_j\in L, R(l_i,s)\neq R(l_j,s)\\ 0, & \text{其他}\end{cases}.$$

该函数的含义为：对于给定点 s 及一组被测服务，如果这些服务的返回结果均相同，则示性函数 $value_F(s)$的值为 0；只要这些结果中存在差异，则函数值为 1. 当示性函数值为 1 时，说明不同服务对于同样的输入产生了不同结果，也就是说该输入点 s 能够发现服务缺陷，为缺陷数据. 根据示性函数的定义，地图中的点可以分为两类：缺陷数据和非缺陷数据. 根据某个点所处的区域，利用贝叶斯分类器能够预测该点为缺陷数据的概率.

叶斯分类器的设计同 3.3.2 节类似，只是其中的示性函数定义不同，即

$$P(D_{ij}|value_F(s)=1)=\sum_{s\in D_{ij}}value_F(s)\Big/\sum_{s\in D}value_F(s),$$

$$P(value_F(s)=1)=\sum_{s\in D}value_F(s)/|D|,$$

$$P(D_{ij})=|D_{ij}|/|D|,$$

$$P(value_F(s)=1|D_{ij})=P(D_{ij}|value_F(s)=1)\times P(value_F(s)=1)/P(D_{ij}).$$

$P(D_{ij}|value_F(s)=1)$为从区域 D_{ij} 中取到缺陷数据的先验概率，而 $P(value_F(s)=1|D_{ij})$则是从区域 D_{ij} 中随机选取一个点 x，该点是缺陷数据的后验概率.

3.5　案例分析

本节将具体介绍以上两种优化搜索算法的执行过程. 将经纬度范围分别为[110,150]和[35,53]的矩形区域作为输入域，首先将该区域划分为 2×2 个子区域，分别为 D_1([110,130],[35,44])、D_2([110,130],(44,53])、D_3((130,150],[35,44])、D_4((130,150],(44,53])，从每个子区域中随机选取 25 个点，共 100 个点对贝叶斯分类器进行训练，从中选取一个概率值最高的子区域进行优化搜索.

(1) 以区域覆盖为优化目标的算法

利用训练数据集计算从 D_1、D_2、D_3、D_4 中取到有效点的概率，分别为 0.07、0.09、0、0，可以看出从 D_2 中能够取到有效点的概率最高，因此首先对 D_2 进行进一步优化搜索. 表 1 是从搜索过程中截取的一个片段，其中第 1 行，虽然返回值不为空，但在该子区域取到有效点的概率为 0，因此其能量值为 0；第 2 行则因为其返回结果为空，因此能量值为 0. 当

搜索得到的新解不被接受时，则转向下一个子区域进行继续搜索，直到达到算法要求的终止条件.

表 1 区域覆盖优化算法搜索过程

步骤序号	区域	位置坐标	接口返回结果	能量值	是否接受
1	[110,120],(44,48.5]	(45.49,116.53)	内蒙古自治区锡林郭勒盟东乌珠穆沁旗嘎达布其镇	0	否
2	[110,120],(48.5,53]	(52.44,110.88)	无	0	否
3	(120,130],(44,48.5]	(47.61,129.80)	黑龙江省伊春市金山屯区金山街道	0.3158	是
4	(120,130],(44,48.5]	(48.49,129.00)	黑龙江省伊春市汤旺河区高峰林场	0.3158	是
5	(120,130],(44,48.5]	(46.99,129.19)	黑龙江省伊春市南岔区亮子河林场	0.3158	是

(2) 以故障检测为优化目标的算法

利用训练数据集计算从 D_1、D_2、D_3、D_4 中取到缺陷数据的概率，分别为 0.09、0、0、0，可以看出从 D_1 中取到缺陷数据的概率最大，因此首先对 D_1 进行优化搜索. 表 2 是从搜索过程中截取的一个片段，其中第 3 行的新解被接受，由于此时已达到该子区域的采样上限，因此转向下一个子区域进行搜索；第 4 行的解虽然使能量值降低，但仍然被接受，这是由于模拟退火算法能够以一定概率接受较差的解.

表 2 故障检测优化算法搜索过程

步骤序号	区域	位置坐标	接口返回结果	能量值	是否接受
1	[110,120][35,39.5]	(39.49,113.79	API3 结果更详细	0.16	是
2	[110,120][35,39.5]	(37.90,110.36)	API3 结果更详细	0.16	是
3	[110,120][35,39.5]	(35.64,116.66)	API3、API4 结果更详细，并且结果不同	0.16	是
4	(120,130],(39.5,44]	(41.30,126.74)	4 个接口结果均相同	0	是
5	(120,130],(39.5,44]	(41.91,125.83)	API3 结果更详细	0.08	是

4 实验与评估

4.1 实验设置

本实验以高德 Android Location API、百度 Android Map API、百度 Web、腾讯 Android Map SDK 这 4 个位置服务为被测服务，以算法 1 生成的测试数据集分别调用这 4 个服务，通过对输出结果的比较，评价服务的正确性. 算法中各参数具体取值的设定如下：

(1) 输入参数的取值范围，经度范围为[110,150]，纬度范围为[35,53].

(2) 在输入域范围内随机选取的初始点集的规模为 100 个点，即参数 q 为 100.

(3) 经过多次实验，将区域划分参数 n 和 m 的取值确定为 $n=m=2$，此时算法能达到更加理想的收缩效果.

(4) 初始温度 t 设定为 250，降温系数为 0.98.

(5) 算法最内层循环的循环次数上限设定为 100.

(6) 每个子域中采样点的数量上限设定为 10，区域划分每一层采样点的数量上限设定为 100.

(7) 算法终止条件中温度系数的阈值为 0.05，区域划分的阈值为经度或者纬度跨度为 0.05.

以上循环次数以及采样点阈值的设定均考虑了移动终端以及无线网络的限制条件，将对 API 的访问次数控制在一个可以接受的范围内.

4.2 实验结果

实验选取的输入域在地球上南北跨度大约为 2000 公里，东西跨度大约为 3500 公里，覆盖面积约为 700 万平方公里，其中包括人口稠密的城市、农村以及人烟稀少的森林、海洋等. 下面总结了有效点聚集区探索算法和故障驱动算法的实验结果，实验结果基于算法的多次运行，对多次运行结果取平均值获得.

4.2.1 以区域覆盖为优化目标的算法

贝叶斯分类器训练过程的训练数据集中包含 100 组数据，除此之外有效点密集区探索算法在确定有效点聚集区的过程中，共生成 661 组测试数据，其中包含有效点 614 个，无效点 47 个. 测试数据可以从这 661 组数据中选取，或者可以从已确定的有效点聚集区中重新生成. 由于在确定有效点聚集区时已生成大量有效点，因此本次实验将直接利用这 661 组数据进行测试，分别调用 4.1 节中介绍的 4 个位置服务，然后对比这 4 个服务返回结果的异同. 由于商业数据的敏感性问题，本文将上述 4 个服务随机排序，并在进行结果对比时将服务名称隐去，以 API1、API2、API3、API4 来指代上述 4 个服务. 服务执行结果分列于表 3～表 6 中，对上述 4 个服务的输出结果进行比较，结果不同的测试数据的数量总结在表 3 中，表中数字的含义为对应行和列所表示的接口之间输出结果不同的个数，例如第 1 行第 2 列的数字 142 表示 API1 和 API2 的输出结果

中，有 142 组结果不同，第 1 行第 3 列则表示 API1 和 API3 的输出结果中有 603 组不同.

表 4～表 6 则是对 3.4 节定义的 3 种服务故障进行的统计.

表 3　服务结果差异统计

	API1	API2	API3	API4
API1	—	142	603	108
API2	142	—	601	134
API3	603	601	—	601
API4	108	134	603	—

表 4　针对 SF_1 的服务结果差异统计

	API1	API2	API3	API4
API1	—	4	2	8
API2	4	—	2	8
API3	2	2	—	6
API4	8	8	6	—

表 5　针对 SF_2 的服务结果差异统计

	API1	API2	API3	API4
API1	—	110	461	49
API2	110	—	596	105
API3	461	596	—	468
API4	49	105	468	—

表 6　针对 SF_3 的服务结果差异统计

	API1	API2	API3	API4
API1	—	28	140	51
API2	28	—	3	21
API3	140	3	—	129
API4	51	21	129	—

从以上结果对比可以得出结论如下：

(1) 通过对 4 个服务结果进行对比，发现 API3 与这 3 个服务的差异性很大，基本都达到了 91.23%，仅考察结果的差异性，还很难评价服务的质量，必须将这些不同的结果进行分类，分别从 SF_1、SF_2、SF_3 这 3 个方面进行考察.

(2) SF_1：对于无效点的判定，各服务之间差别很小. 差别最大的是API1和 API4 以及 API2 和 API4，有 8 个不同的结果，占全部测试数据的 1.21%. 差别最小的为 API1 和 API3 以及 API1 和 API4，有 2 个不同的结果，占全部测试数据的 0.30%.

(3) SF_2：API3 与其他 3 个服务相差较大，结果不同的数据最多占到测试数据的 90.17%. 通过对原始数据的分析，发现 API3 确定的地址信息更加详细，也就是说 API3 定义地址信息的区域粒度更小.

(4) SF_3：相差最多的两个服务为 API1 和 API3，包含 140 个不同的输出结果，占全部测试数据的 21.18%，相差最少的两个服务为 API2 和 API3，包含 3 个不同的输出结果，占全部测试数据的 0.45%，此类错误说明结果不同的两个服务中至少有一个服务存储的地址信息是错误的.

通过以上比较，我们发现目前市场上主流的反地理编码服务通常都能够提供正确的地址信息，但有很多服务提供的地址信息不够详细，例如 API1、API2 和 API4，只能定位到一个比较宽泛的区域范围，这样的结果对于下一步的地址导航、服务推荐等结果的正确性均有不同程度的影响.

4.2.2　以故障检测为优化目标的算法

故障驱动算法共生成 730 组数据，其中包括 100 组贝叶斯分类器训练数据，及 630 组测试数据. 用生成的这 630 组测试数据调用被测服务，表 7 是对 API1、API2、API3、API4 这 4 个服务返回的不同结果的比较，从表中可以看出在任意两个服务之间均存在不同数量的结果差异，差异最大的是 API1 和 API1 以及 API3 和 API4，结果不同的数据占总数据量的 93.81%.

表 7　故障驱动算法服务结果差异统计

	API1	API2	API3	API4
API1	—	87	591	193
API2	87	—	588	196
API3	591	588	—	591
API4	193	196	591	—

表 8、表 9、表 10 则是分别针对 SF_1、SF_2、SF_3 这 3 种故障类型进行的对比统计，从表中数据可以看出：

(1) SF_1：接口之间差异最大的有 141 个结果不同，占全部测试数据的 22.38%，差异最小的有 2 个结果不同，占测试数据的 0.32%.

(2) SF_2：API3 与其他几个服务的结果相差很大，其中与 API2 的差异最大，包含 584 个不同的返回结果，占全部测试数据的 92.70%.

(3) SF_3：相差最多的两个服务是 API1 和 API3，包含 82 个不同结果，占全部数据的 13.02%，相差最少的两个服务是 API2 和 API3，包含 2 个不同结果，占全部数据的 0.32%.

表 8　故障驱动算法 SF_1 统计

	API1	API2	API3	API4
API1	—	6	4	141
API2	6	—	2	141
API3	4	2	—	139
API4	141	141	139	—

表 9 故障驱动算法 SF_2 统计

	API1	API2	API3	API4
API1	—	58	505	24
API2	58	—	584	48
API3	505	584	—	398
API4	24	48	398	—

表 10 故障驱动算法 SF_3 统计

	API1	API2	API3	API4
API1	—	23	82	28
API2	23	—	2	7
API3	82	2	—	54
API4	28	7	54	—

通过上述比较发现，两种算法生成的测试数据对服务进行测试，测试结果基本一致，测试结果能够反映算法的缺陷发现能力，说明这两种算法缺陷发现的能力相当.

4.3 结果评估

4.3.1 以区域覆盖为优化目标的算法

作为比较基准，本实验利用随机算法(Random Algorithm，RA)在指定区域中生成了 660 组数据，其中包含有效点 218 个，无效点 442 个. 我们用有效点占比来定义该算法测试数据的生成效率：

$$Efficiency=\frac{|S_{\text{Valid}}|}{|S|}\times 100\%,$$

其中，

(1) $S=\{s_i\}$ 为算法生成的全部测试数据，$|S|$ 为生成的测试数据的数量；

(2) $S_{\text{Valid}}=\{s_j\}$ 为生成数据中的有效点集合，也就是对于 $\forall s_j\in S_{\text{Valid}}, s_j\in S$，并且 $value(s_j)=1$. $|S_{\text{Valid}}|$ 为 S_{Valid} 中包含的测试数据的数量.

利用上述公式分别计算 SA 和 RA 的数据生成效率：

$$Efficiency_{\text{SA}}=\frac{614}{661}\times 100\%=92.89\%,$$

$$Efficiency_{\text{RA}}=\frac{218}{660}\times 100\%=33.03\%.$$

SA 生成的数据数量由多种因素共同决定，RA 生成的数据数量只能依靠循环次数控制，因此不一定能够生成与 SA 相同数量的数据. 通过以上计算可以看出 SA 的数据生成方法能够更好地按照测试人员的意图生成符合要求的数据，并且能够在算法早期就生成足够多符合要求的数据，使算法能够提前终止搜索，节约了宝贵的测试资源.

本文还通过修改每层采样上限和每个子区域采样上限两个参数值，探讨这两个参数取值对算法的影响. 表 11 的实验结果是将每个子区域采样上限确定为 10，增加每层采样上限，从表中可以看出随着每层采样上限的增加，SA 算法的参数生成效率也有所提高，该参数从 50 增加到 100、从 100 增加到 200，生成效率分别增加了 0.94 和 0.1 个百分点. 对于 RA，该参数的改变也提高了参数的生成效率，从表中可以看出，参数值从 50 增加到 100，生成效率增加了 5.62 个百分点，参数值从 100 增加到 200，生成效率增加了 0.35 个百分点.

表 11 增加每层采样上限对算法的影响

	数据总量	有效点	生成效率/%
SA-50	566	531	93.82
RA-50	566	169	29.86
SA-100	668	633	94.76
RA-100	668	237	35.48
SA-200	681	646	94.86
RA-200	681	244	35.83

表 12 的实验结果是将每层采样上限固定为 200，然后增加每个子区域的采样上限值，考察该参数对 SA 算法的影响. 从表中可以看出，随着参数值的增加，测试数据的生成效率会随之增加：参数值从 5 增加到 10，生成效率增加了 3.86 个百分点，参数值从 10 增加到 20，生成效率增加了 2.41 个百分点.

表 12 增加每个子区域采样上限对算法的影响

	数据总量	有效点	生成效率/%
SA-5	378	344	91.00
SA-10	681	646	94.86
SA-20	1244	1210	97.27

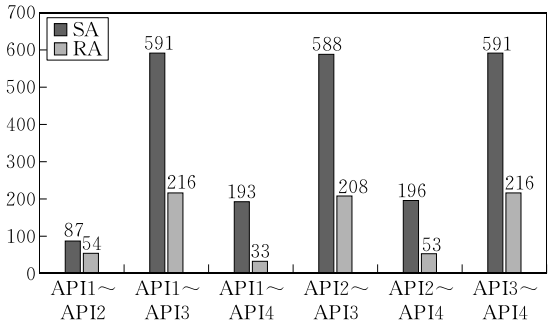
4.3.2 以故障检测为优化目标的算法

作为基准比较，同样用 RA 生成 630 组数据，对上述 4 个服务进行测试，并统计返回结果不同的数据数量，如表 13 所示.

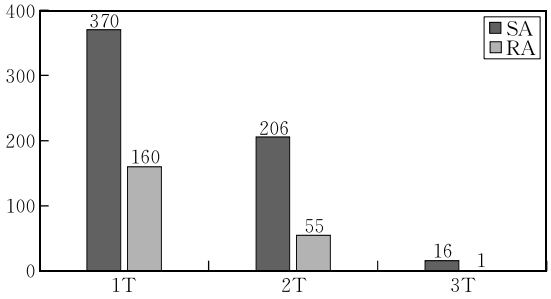
表 13 随机算法服务结果差异统计

	API1	API2	API3	API4
API1	—	54	216	33
API2	54	—	208	53
API3	216	208	—	216
API4	33	53	216	—

图 4(a)呈现了 SA 和 RA 生成的测试数据，探测出的任意两个接口之间结果差异的对比. 图 4(b)是能够探测出不同种类结果差异的测试数据的数量对比，其中 $nT(n=1,2,3)$ 表示能够探测出 $n(n=1,2,3)$ 种结果差异的测试数据的数量. 从图中可以看出，SA 生成的测试数据比 RA 能够探测出更多的结果差异，也就是能够发现更多的潜在服务缺陷.



(a) 接口对之间的对比



(b) 3种差异类型的对比

图 4　SA 和 RA 结果不同的对比

我们将故障驱动算法测试数据的生成效率定义为

$$Efficiency=\frac{|S_{Diff}|}{|S|}\times 100\%,$$

其中，

(1) $S=\{s_i\}$ 为算法生成的全部测试数据，$|S|$ 为生成的测试数据的数量；

(2) $S_{Diff}=\{s_j\}$ 为能够探测出服务结果差异的测试数据，也就是对于 $\forall s_j\in S_{Diff}$，$s_j\in S$，并且 $value_F(s_j)=1$. $|S_{Diff}|$ 为 S_{Diff} 中包含的测试数据的数量.

利用上述公式计算 SA 和 RA 的数据生成效率，通过计算结果可以看出，SA 的生成效率要优于 RA.

$$Efficiency_{SA}=\frac{370+206+16}{630}\times 100\%=93.97\%,$$

$$Efficiency_{RA}=\frac{160+55+1}{630}\times 100\%=34.29\%.$$

针对以故障检测为优化目标的算法，同样修改了每层采样上限和每个子区域采样上限两个参数值，进行了多次实验并对比结果. 表 14 的实验结果是将每个子区域采样上限确定为 10 的基础上，改变每层采样上限，从表中可以看出随着每层采样上限参数的增加，SA 算法的参数生成效率有所提高，该参数从 50 增加到 100，生成效率增加了 0.19 个百分点，而从 100 增加到 200，生成效率只增加了 0.09 个百分点. 对于 RA，该参数的改变并没有对实验结果有明确的影响，从表中可以看出，参数值从 50 增加到 100，生成效率增加了 0.99 个百分点，而当参数值从 100 增加到 200，生成效率反而减少了 2.69 个百分点.

表 14　增加每层采样上限对算法的影响

	数据总量	1T	2T	3T	生成效率/%
SA-50	611	469	106	0	94.11
RA-50	611	154	47	1	33.06
SA-100	649	489	122	1	94.30
RA-100	649	165	56	0	34.05
SA-200	660	444	179	0	94.39
RA-200	660	158	49	0	31.36

表 15 的实验结果是将每层采样上限固定为 200，然后增加每个子区域的采样上限值，考察该参数对 SA 算法的影响. 从表中可以看出，随着参数值的增加，测试数据的生成效率会随之增加，参数值从 5 增加到 10，生成效率增加了 4.25 个百分点，而当参数从 10 增加到 20 时，生成效率增加了 2.56 个百分点.

表 15　增加每个子区域采样上限对算法的影响

	数据总量	1T	2T	3T	生成效率/%
SA-5	365	132	176	21	90.14
SA-10	660	444	179	0	94.39
SA-20	1215	952	225	1	96.95

5　相关工作

5.1　移动应用测试

移动互联网和移动终端的飞速发展，使得相应的移动应用也呈现出井喷式的增长趋势. 文献[7]给出了移动应用通俗易懂的定义：专门为智能手机、平板电脑等可移动设备开发的软件应用，并且这些软件应用能够以情景信息为输入. 该文献将移动应用分为两大类：运行于移动终端的应用和通过网络访问的应用. 分析了移动应用测试面临的挑战，并提出解决这些问题的方法以及今后的发展趋势. 目前移动应用测试研究主要分布在以下几个领域：移动应用的白盒测试和单元测试，移动应用的黑盒测试和 GUI 测试，移动应用的服务质量需求测试，无线网络测试，移动可用性测试以及移动测试自动化框架等. 文献[8]总结了 4 类主流的移动测试技术：(1) 基于仿真的测试，利用移动终端仿真器进行模拟，使得测试过程能够在个人计算机上进行；(2) 基于设备的测试，将测试直接部署在真实的移动终端

上;(3)云测试,其核心是要构建移动设备云,使用户能够以按需租用的方式使用测试服务;(4)基于群智的测试技术,其核心是利用群众的智慧,通过网络利用来自世界各地的自由职业者或者签约软件测试工程师来共同完成测试任务.

文献[9]对位置服务测试进行了相关研究,针对移动设备现实应用环境的复杂多变性以及测试实验开展的安全性和合法性等问题,为移动位置服务创建了一种测试环境,用来考察移动设备、设备使用者以及设备所处环境之间的交互活动,测试环境包括:(1)城市虚拟现实模型.能够按照测试目的创建街道、指示牌以及标志性建筑等,能够模拟用户在街道中穿梭等个体活动;(2)移动设备.作为信息提供源,能够模拟位置服务应用程序;(3)软件.用来记录参与者的行为及其与测试环境之间的交互活动.虚拟现实环境的创建能够保持环境的稳定,在相同环境下,对不同用户行为进行对比分析,从而给出位置服务质量的评价.

5.2 模拟退火算法

模拟退火算法最初由 Metropolis 等人在 1953 年研究二维相变时提出,Kirkpatrick 等在 1983 年利用该算法设计大规模集成电路,并最早将其应用于优化问题的求解中[10].另外,Černý[11]于 1985 年独立提出了 SA 算法,将其用于求解 TSP 问题,并获得成功.1987 年 Szu[12]提出一种退货率与时间成反比的快速模拟退火算法 FSA.1988 年 Aarts 等人[13]出版的书籍以及 Laarhoven 等人[14]出版的书籍系统全面地介绍了模拟退火算法,这标志着该算法的描述和应用得到了广泛的验证.迄今为止,SA 算法已经在计算机领域中,包括人工智能、网络通信等许多涉及优化问题的领域中得到广泛应用[15-16],也衍生出了各种不同的改进版本[17-19].Gallego 等人[20]则提出一种模拟退火算法的并行化实现方法,该方法将模拟退火算法应用于网络传输的扩展,并将传统模拟退火算法部署在分布式处理器中,在温度参数的每一档,分别在每个处理器上利用模拟退火算法进行搜索,然后每个处理器分别将其搜索的结果发送给其中一个处理器,该处理器从这些解中选出最优解,并将此解分发给其他所有处理器,所有处理器以此解作为初始解进行新一轮的搜索过程.

将模拟退火算法应用于软件工程中需要解决的两个基本问题是:(1)问题的描述;(2)目标函数的定义.通常情况下,软件工程师对于其要解决的软件工程问题都能找到合适的表达形式,在目标函数定义的确定上则有很多不同的选择,如何确定一个目标函数,使其方便度量,并能准确评价搜索结果的优劣是应用模拟退火算法的关键[21].

文献[22]将模拟退火算法应用于软件的缺陷查找,将软件规约形式化地定义为前置/后置条件,并利用前置/后置条件定义搜索算法的目标函数,利用目标函数指导搜索过程的进行.该文献利用类似算法形式化定义软件运行异常的前提条件,并以此作为搜索算法的目标函数.文献[23]介绍了一种测试数据自动生成工具 GADGET,该工具最初仅应用遗传算法来实现测试数据的生成,但之后该工具实现了包括模拟退火算法在内的四种优化算法,实现复杂软件程序测试数据的自动生成.文献[24]利用模拟退火算法等构建覆盖数组完成组合测试,算法的当前解即为目前已产生的测试数据集合,将参数中未被覆盖的组合数作为算法的目标函数值,搜索的方向为目标函数值降低的方向,直至目标函数值为 0 时算法终止.

本问题中,我们将测试生成过程类比为寻优过程,基于 SA 算法,加入贝叶斯分类器来预测和评判较好区域的概率,从而为状态评价提供依据,为新状态选择提供方向,是对 SA 算法的改进和优化.SA 算法的参数选择对算法效率影响较大,温度 T 的初始值设置过高,则搜索到全局最优解的可能性大,但将耗费大量计算时间,反之虽可节约计算时间,但全局搜索性能可能受到影响,实际使用中一般需要若干次实验调整初始温度.SA 算法的优点是质量高,初始值鲁棒性强,简单通用易实现.其缺点也十分明显,由于要求较高的初始温度、较慢的降温速率、较低的终止温度以及各温度下足够多次的抽样,因此优化过程较长.由于我们并不是要寻找最优解,而是记录寻优过程,因此最终停留状态的好坏对本问题并不十分重要,我们通过引入贝叶斯分类器提高退火效率,并对抽样数据的数量加以限制,在可接受的时间代价下大大提高了原有 SA 算法中退火过程产生可行解的数量和概率.

6 总 结

移动服务已经成为很多人工作、生活不可获取的一部分,移动服务测试对保证移动服务的正确性、可用性等有非常重要的作用.本文以一个典型的移动位置服务为例,采用基于贝叶斯分类器的模拟退火搜索方法为服务生成测试数据,并对所选的 4 个

服务进行输出结果对比，通过结果对比判定服务的正确性和返回结果的精确性.

参 考 文 献

[1] Bajad R A, Srivastava M, Sinha A. Survey on mobile cloud computing. International Journal of Engineering Sciences & Emerging Technologies, 2012, 1(2): 8-19

[2] Küpper A. Location-Based Services: Fundamentals and Operation. New York, USA: John Wiley & Sons Inc., 2005

[3] Rao B, Minakakis L. Evolution of mobile location-based services. Communications of the ACM, 2003, 46(12): 61-65

[4] Xie Yun. A survey of simulated annealing algorithm. Micro Computer Information, 1998, 14(5): 7-9(in Chinese)
(谢云. 模拟退火算法综述. 微计算机信息, 1998, 14(5): 7-9)

[5] Phil M. Search-based software test data generation: A survey. Software Testing, Verification and Reliability, 2004, 14(2): 105-156

[6] McCallum A, Nigam K. A comparison of event models for Naïve Bayes test classification//Proceedings of the AAAI-98 Workshop on Learning for Text Categorization. Madison, USA, 1998: 41-48

[7] Kirubakaran B, Karthikeyani V. Mobile application testing—Challenges and solution approach through automation//Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME). Stockholm, Sweden, 2013: 79-84

[8] Gao J, Bai X, Tsai W, et al. Mobile application testing: A tutorial. Computer, 2014, 47(2): 46-55

[9] Li C, Longley P. A test environment for location-based services applications. Transactions in GIS, 2006, 10(1): 43-61

[10] Kirkpatrick S. Optimization by simulated annealing: Quantitative studies. Journal of Statistical Physics, 1984, 34(5-6): 975-986

[11] Černý V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of Optimization Theory and Applications, 1985, 45(1): 41-51

[12] Szu H, Hartley R. Fast simulated annealing. Physics Letters A, 1987, 122(3-4): 157-162

[13] Aarts E, Korst J. Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing. New York, USA: John Wiley & Sons Inc., 1989

[14] Laarhoven P J M V, Aarts E H L. Simulated Annealing: Theory and Applications. Dordrecht, Holland: D. Reidel Publishing Co., 1987

[15] Aarts E H L, Laarhoven P J M V. Simulated annealing: A pedestrian review of the theory and some applications//Devijver P A, Kittler J eds. Pattern Recognition Theory and Applications. Berlin Heidelberg: Springer, 1987: 179-192

[16] Eglese R W. Simulated annealing: A tool for operational research. General Information, 1990, 46(3): 271-281

[17] Dueck G, Scheuer T. Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing. Journal of Computational Physics, 1990, 90(1): 161-175

[18] Goffe W L, Ferrier G D, Rogers J. Global optimization of statistical functions with simulated annealing. Journal of Econometrics, 1994, 60(94): 65-99

[19] Laarhoven P J M V, Aarts E H L, Lestra J K. Job shop scheduling by simulated annealing. Operations Research, 1992, 40(1): 113-125

[20] Gallego R A, Alves A B, Monticelli A, et al. Parallel simulated annealing applied to long term transmission network expansion planning. IEEE Transactions on Power Systems, 1997, 12(1): 181-188

[21] Harman M. The current state and future of search based software engineering//Proceedings of the 29th Conference on Future of Software Engineering. Washington, USA, 2007: 342-357

[22] Tracey N, Clark J, Mander K. Automated program flaw finding using simulated annealing//Proceedings of the ACM/SIGSOFT International Symposium on Software Testing and Analysis (ISSTA). Clearwater Beach, USA, 1998: 73-81

[23] Michael C, Mcgraw G. Automated software test data generation for complex programs//Proceedings of the 13th IEEE International Conference on Automated Software Engineering. Honolulu, USA, 1998: 136-146

[24] Mugridge W B, Cohen M B, Gibbons P B, et al. Constructing test suites for interaction testing//Proceedings of the International Conference on Software Engineering. Portland, USA, 2003: 38-48

HOU Ke-Jia, born in 1983, Ph. D. candidate. Her research interest is software testing.

HUANG Jun, born in 1991, M. S. candidate. His research interests include software testing and mobile service.

BAI Xiao-Ying, born in 1973, Ph. D., associate professor. Her research interests include software testing and service computing.

ZHOU Li-Zhu, born in 1947, M. S., professor, Ph. D. supervisor. His research interests include database system, digital library and massive information processing.

Background

With the expansion of mobile cloud, location-awareness has been an enabling technique for a variety of mobile applications. Mobile devices gather geographical positions and transfer the addresses to Location-Based Services (LBS) through mobile network. LBS can then provide information services such as transportation, entertainment and healthcare for the given physical locations. It is considered as the "killer application" of mobile commerce.

Many platforms have been built to provide open LBS services. For example, Baidu, Tencent and Amap are three widely used LBS platforms in China. However, the quality of location-awareness of Mobile applications varies across different LBS platforms in terms of accuracy, completeness and up-to-dateness. For examples, how well do the location-based information services meet users' expectations? How many, and how precisely, are the locations covered in the LBS system? How well are the data in sync with latest updates? Research has been conducted on LBS quality from various perspectives, such as content, usage, hardware and communication network.

Validation of the quality attributes needs extensive an continuous testing. However, LBS testing faces the challenge of large input space of geographical locations. It is usually impossible to enumerate locations, and thus hard to measure and evaluate test adequacy in terms of spatial-coverage. In addition, mobile applications are usually characterized by online evolutions.

In response to the needs and difficulties, the research investigates search-based methods for LBS test data generation. Taking reverse-GeoCoding service as a motivating example, the paper proposed SA algorithms with different assumptions and fitness functions.

This work is supported in part by a grant from the National Natural Science Foundation of China (61472193), the Beijing Natural Science Foundation (4132062), the National Grand Fundamental Research (973 Program) of China (2011CB302505), and the National High Technology Research and Development Program (863 Program) of China (2013AA01A215).