屏蔽输入参数敏感的异常点检测新方法*)

陶运信　皮德常

(南京航空航天大学信息科学与技术学院　南京 210016)

摘　要　大多数基于密度的异常点检测算法需要设置两个输入参数，并对输入参数很敏感，用户设置不正确会导致算法不能发现所有有意义的异常点，甚至是发现错误的异常点，这使得评价一个数据挖掘算法的"3-E"标准中"易于使用"这一点不能得到满足。为此，首先根据对象的邻域、反邻域和局部密度构造基于邻域的局部密度因子 NLDF，NLDF 可指示异常点的异常程度，然后提出一种屏蔽输入参数敏感的异常点检测算法 ODINP。ODINP 的一个非常显著的优点就是只需要一个参数 k 并且对 k 不敏感。该算法在保持已有基于密度的异常点检测算法高效性的同时，具有很高的异常点检测精度。大规模、任意形状和高维数据集的测试结果表明该算法是有效的、可行的。

关键词　数据挖掘，异常点检测，参数，邻域，密度

New Approach to Detect Outlier which is Insensitive to Input Parameter

TAO Yun-xin　PI De-chang

(College of Information Science and Technology, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, Chian)

Abstract　Most density-based outlier detection algorithms require the setting of two input parameters and are sensitive to input parameters. Incorrect setting may cause an algorithm to fail in finding all meaningful outliers and even find wrong outliers, which cannot satisfy the easy to use of "3-E" criteria. Therefor, constructed neighborhood based local density factor NLDF taking account of neighborhood, reverse neighborhood and local density, NLDF can denote the degree of outlierness of an object. Afterward, an novel outlier detection algorithm named ODINP that insensitive to input parameter was proposed. ODINP keeps the efficiency of the existing density-based outlier detection algorithms and owns high precision. Just a parameter k and insensitive to k is a significantly advantage of ODINP. Extensive experiments on large-scale, different shape and high-dimensional data sets demonstrated that the algorithm is effective and feasible.

Keywords　Data mining, Outlier detection, Parameter, Neighborhood, Density

1　引言

大规模和高维数据被收集并存储到空间数据库中，迫切需要有效的、可行的和易于使用的方法，以挖掘出数据中潜在的信息。当今，数据挖掘和知识发现的大部分研究主要集中于发现常规的模式或频繁的事件，但是在信用卡欺诈识别、入侵检测、军事侦察、灾害天气预报和医疗分析等应用领域，异常的模式、罕见的事件比常规的模式、频繁的事件更有价值。异常点检测就是旨在发现数据集中与其余数据相比显著相异的、异常的或不一致的对象。

近几年，研究人员已经提出了不少异常点检测算法，如基于统计的、深度的、距离的和密度的方法[1-3]。基于密度的方法是异常点检测算法家族中的一类，其基本思想来自基于密度的聚类方法，比较有代表性的算法是 LOF[1]，ODBSN[2]，DBSCAN[3]等，其中 DBSCAN 是基于密度的聚类算法，也可以用来挖掘异常点。

LOF 查找异常点需执行一个两步算法：第一步对每个对象查询 MinPtsUB 近邻，返回一个大小为 $n*$ MinPtsUB 的距离数据库 M；第二步取[MinPtsLB, MinPtsUB]这一集合内所有 MinPts，计算每个对象的 MinPts 近邻的 LOF 值，将最大值作为它的最终 LOF 值并输出到一个文件中。为了根据最终的 LOF 值确定异常点，还需要一个 LOF 阈值，即算法需要两个参数。DBSCAN 算法需要两个全局参数 Eps 和 MinPts，虽然它提供排序 k-dist 图这种可视化方法来辅助参数 Eps 的确定，但是在实际操作中，通过这种方法确定的 Eps 与"理想的"Eps 之间常有一定的差距，这会造成结果的很大不同。ODBSN 先根据边长为 Eps 的方形邻域内的对象个数是否大于 MinP 进行异常点粗选，再计算候选异常点的局部偏离指数进行异常点精选。Eps 的改变导致 MinP 大小也要相应改变，还可能会导致发现的异常点有很大不同。上述算法的一个共同缺陷就是需要两个参数并且对输入参数敏感。在实际使用中，当用户不明白算法中输入参数的作用时，不正确的设置会导致错误的挖掘结果，这就使得评价一个数据挖掘算法的"3-E"标准[4]中"易于使用"这一点不能得到满足。多个参数还会使通过不同方法发现的结果很难比较。

OPTICS[5]虽然能较好地解决 DBSCAN 对输入参数 Eps 敏感的缺陷，并与 DBSCAN 算法有相同的时间复杂度，但由于采用了复杂的处理方法以及额外的磁盘 I/O 操作，使得 OPTICS 实际运行效率远远低于 DBSCAN。为此，蔡颖琨等人[6]对基于密度的聚类算法参数敏感性这一缺陷进行了改进，但这是针对单个聚类算法，不具有通用性。文献[7]讨论了一种最理想的情形，即无参数的数据挖掘算法。作为向这

*)国家高技术研究发展计划(863 计划)项目 (2007AA01Z404) 资助。**陶运信**　硕士研究生，主要从事移动对象聚类与异常点检测研究；**皮德常**　副教授，硕士生导师，主要从事数据挖掘和移动对象数据库技术研究。

个问题过渡的一个步骤，在一系列不同问题和数据类型上，验证了无参数或者少量参数的算法能与参数依赖性很大的算法相抗衡。上述所做的研究要么针对单个聚类算法，要么就针对具体的问题和数据类型。据我们所知，基于密度的异常点检测算法参数敏感性至今仍未得到很好解决。

本文提出一个新算法 ODINP，该算法使用邻域、反邻域和局部密度构造对象的基于邻域的局部密度因子 NLDF，再根据 NLDF 使用邻域扩展的方法排除聚类点。NLDF 指示异常点的异常程度，避免了以往全局异常点的二值属性(数据集中的任一个对象，要么是异常点，要么不是)，保持了 LOF 中异常程度可度量的优点。本算法一个非常显著的优点就是只需要一个参数 k 并且对它不敏感，满足了"3-E"标准中的易于使用这一点。在保持了以往基于密度的异常点检测算法的高效性的同时，也具有很高的异常点检测精度。

本文其它内容组织如下：第 2 节在引入相关定义的基础上，给出基于邻域的局部密度因子的构造方法；第 3 节提出屏蔽输入参数敏感的异常点检测算法，即 ODINP 算法，并分析其时间和空间复杂度；第 4 节通过大量实验验证了算法的有效性、可行性和可伸缩性；最后总结全文。

2 基于邻域的局部密度因子的构造

给定一个大小为 n 的空间数据库 $D=\{d_1,d_2,\cdots,d_n\}$，p，q，o 和 o' 是 D 中的对象，下文中使用欧氏距离来表示两个对象之间的距离。我们在引入 k 近邻和反 k 近邻定义后，给出 k 邻域和反 k 邻域的定义。尽管在文献[1,4,10]中已经出现过类似的定义，但是为了让读者理解新构造的基于邻域的局部密度因子和 ODINP 算法，这里将做简单介绍。

定义 1(k 近邻)　p 的 $k(k>0)$ 个最近邻的集合称为 p 的 k 近邻集合，记作 $kNN(p)$。$kNN(p)$ 是满足以下条件的 D 中对象的集合：

(1) $|kNN(p)|=k$；

(2) $p\notin kNN(p)$；

(3)如果 o 和 o' 分别是 p 的第 k 个和第 $(k+1)$ 个最近邻，那么 $dist(p,o)\leqslant dist(p,o')$。

定义 2(反 k 近邻)　p 的反 k 近邻是其它对象 k 近邻中包含 p 的这些对象的集合，表示为 $RkNN(p)$，即 $RkNN(p)=\{q\in D\mid p\in kNN(q)\text{ and }p\neq q\}$。

定义 3(k 邻域)　对 D 中的每个对象 p，如果存在 $o\in kNN(p)$，$r=dist(p,o)$，使得对于任意 $o'\in kNN(p)$，有 $dist(p,o')\leqslant r$，那么 k-邻域定义为 $kNB(p)=\{q\in D\mid dist(q,p)\leqslant r\text{ and }q\neq p\}$，称为 p 关于 $kNN(p)$ 的 k 邻域，记作 $kNB(p)$。

定义 4(反 k 邻域)　p 的反 k 邻域是其它对象 k 邻域中包含 p 的这些对象的集合，表示为 $RkNB(p)$，即 $RkNB(p)=\{q\in D\mid p\in kNB(q)\text{ and }p\neq q\}$。

定义 5(局部密度)　p 的局部密度定义为

$$LD_k(p)=\frac{\sum_{q\in kNB}\frac{1}{dist(p,q)}}{|kNB(p)|}$$

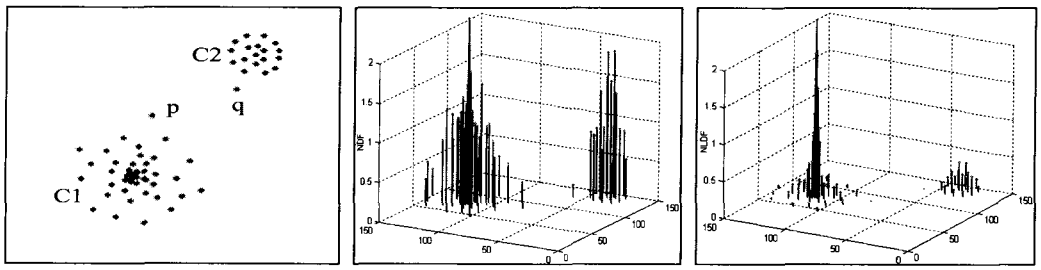
定义 6(基于邻域密度因子)　p 的邻域密度因子定义为

$$NDF_k(p)=\frac{|RKNB(p)|}{|KNB(p)|}$$

通常，不同背景的研究人员在研究异常点检测时，都会给出自己的定义，其中具有代表性的定义有两个。Hawkin[8]认为，异常点是严重偏离其他对象的观察点，以至于让人们怀疑它是由不同的机制产生的。Knorr 等人[9]给出基于距离的异常点 $DB(p,d)$ 定义，他们认为，如果数据集中至少有 p 部分对象与对象 o 的距离大于 d，则对象 o 是一个带参数 p 和 d 的基于距离的异常点。上述两个定义只考虑了数据集的全局属性，这样定义的异常点为全局异常点，只有二值属性。LOF 根据邻域和可达距离来定义异常点，避免了全局异常点的二值属性。文献[10]分析了当稀疏聚类中的对象靠近稠密聚类时，LOF 会导致错误的异常点判断，并使用对称邻域关系对异常点查找进行改进。受上述缺陷和相关研究的启发，我们根据对象的邻域、反邻域和局部密度构造基于邻域的局部密度因子 NLDF，它不需要计算 LOF 中的可达距离，亦不需要计算 ODBSN 中一个对象的方形邻域中的所有对象的局部密度，比二者定义的异常因子都简单。实验结果表明，根据 NLDF 来查找异常点更有效。

定义 7(基于邻域的局部密度因子)　p 的基于邻域的局部密度因子定义为

$$NLDF_k(p)=LD_k(p)*NDF_k(p)$$



(a) 数据集　(b)对象的 NDF 值　(c)对象的 NLDF 值

图 1　一个解析性例子

NLDF 是定义 5 中的局部密度与 NBC 中的基于邻域密度因子 NDF 的乘积，但它与 NDF 是不同的，乘积项 $LD_k(p)$ 是二者的本质区别。考虑 $LD_k(p)$ 后，NLDF 可用于表示异常点的异常程度，NLDF 值越小，异常程度越高，相反地，NLDF 值越大，异常程度越低。根据 NDF 的定义，它的取值范围一般在 0.5 到 2.0 之间，步长为 0.1，所有可能的取值一般不超过 16 个，一个很明显的缺点就是很多对象的 NDF 值都相等，区分性很差，而定义 NLDF 时加上 LD 后，区分性就很强。图 1(a)是一个包含 2 个聚类 $C1$ 和 $C2$，2 个异常点 p 和 q 的数据集，图 1(b)和(c)绘出所有对象的 NDF 和 NLDF 值，表 1 为数据集中部分对象的 NDF，LD 和 NLDF 值。从表 1 中可明显看出 NDF 区分性很差，考虑 LD 后，NLDF 的区分性就很强。此外，两个异常点的 NLDF 值都很小且 $NLDF(q)<NLDF(p)$，这也说明了异常程度与 NLDF 值成反比这一性质。

表 1　部分对象的 NDF,LD 和 NLDF 值

对象/值	|kNB|	|RkNB|	NDF	LD	NLDF
	10	11	1.1	0.245993	0.270592
C1 中部	10	11	1.1	0.272074	0.299281
分对象	11	18	1.63636	0.625732	1.02393
	11	18	1.63636	0.522164	0.854451
	10	8	0.8	0.0984776	0.0787821
C2 中部	10	8	0.8	0.103658	0.0829261
分对象	10	8	0.8	0.115664	0.0925314
	10	9	0.9	0.118691	0.106822
p	10	2	0.2	0.0403124	0.00806249
q	10	1	0.1	0.0536322	0.00536322

3　屏蔽输入参数敏感的异常点检测方法

3.1　ODINP 算法

异常点检测分成 3 步:第 1 步是对每个对象进行 kNB 查询,为了减少时间开销,在进行 kNB 查询的过程中动态计算 RkNB 和该对象的局部密度。第 2 步是计算 NLDF 和它的阈值 thNldf,因为 NBC 中 NDF 的阈值取 1,根据第 2 节对 NDF 和 NLDF 的分析,thNldf 可按下面方法取得:对数据集中所有 NDF=1 的对象,取这些对象的 NLDF 值中最小的一个;如果数据集中不存在 NDF=1 的对象,那么对数据集中所有 NDF<1 的对象,取这些对象的 NLDF 值中最大的一个。因为 $\sum_{p\in D}|kNB(p)|=\sum_{p\in D}|RkNB(p)|$,所以当数据集中不存在 NDF=1 的对象时,一定存在 NDF<1 的对象,也就是说按照上述方法取得的 thNldf 值一定存在。第 3 步根据 NLDF 对数据点进行基于密度的排除,经过这一筛选后的点为异常点,它的异常程度可通过 NLDF 来表示。算法的伪代码如下:

```
ODINP(SetOfPoints,k)
  kNBAndRkNBQuery(indexFile,SetOfPoints,kNB,RkNB,ld,k);
  CalculateNldf(SetOfPoints.size,kNB,RkNB,ld,nldf,thNldf);
  FOR i FROM 1 TO SetOfPoints.size DO
    IF SetOfPoints[i]->getClusterId() != UNCLASSIFIED
      || nldf[i] < thNldf THEN Continue;
    END IF;
    SetOfPoints[i]->setClusterId(CLUSTER_LABEL);
    //Creating and initializing a queue and path array
    FOR j FROM 1 TO kNB[i].count DO
      SetOfPoints[kNB[i].idx[j]]->setClusterId(CLUS-
      TER_LABEL);
      //Adding to queue
      IF nldf[kNB[i].idx[j]] >= thNldf THEN
        path[rear++] = kNB[i].idx[j];
      END IF;
    END FOR;
    //Eliminating the cluster point
    WHILE (front != rear) DO
      frontId = path[front++];
      FOR j FROM 1 TO kNB[frontId].count DO
        IF SetOfPoints[kNB[frontId].idx[j]]->get-
          ClusterId() != UNCLASSIFIED THEN
          continue;
        END IF;
        SetOfPoints[kNB[frontId].idx[j]]->setClus-
        terId(CLUSTER_LABEL);
        IF nldf[kNB[frontId].idx[j]] >= thNldf
          THEN
            path[rear++] = kNB[frontId].idx[j];
        END IF;
      END FOR;
    END WHILE;
END FOR;
//Label outlier
  FOR i FROM 1 TO SetOfPoints.size DO
    IF SetOfPoints[i]->getClusterId() == UNCLASSIFIED
      THEN
      SetOfPoints[i]->setClusterId(OUTLIER_LABEL);
      outlierCount++;
    END IF;
  END FOR;
END;//ODINP
```

3.2　算法复杂性分析

在 ODINP 算法的 3 个步骤中,第 1 步是最费时的,这是因为它要进行邻域查询操作,邻域查询操作的时间复杂度为 $O(n)$。若为所有点建立空间索引如 SR 树[11],邻域查询操作的代价将通过在 SR 树中使用剪枝技术而降为 $O(\log n)$,而且建立的空间索引结构以索引文件的形式存在,不需要在每次运行算法时都去建立。第 2 步只是对每个对象计算 NLDF,时间复杂度为 $O(n)$。第 3 步对异常点设置标志,这一步的时间复杂度也为 $O(n)$。所以,整个算法的时间复杂度为 $O(n\log n)$,保持了以往基于密度的异常点检测算法的高效性。

在算法的执行过程中,需要在内存中存放 SetOfPoints,kNB,RkNB,LD 和 NLDF,其中 LD 可共用为 NLDF 分配的内存。在第 3 步需要借助一个队列来保存 NLDF 大于等于 thNldf 的对象,为了降低内存使用,我们并不为队列动态分配内存来存放这些对象,而是动态分配一个数组来存放相应对象的 ID 号。因此,算法的空间复杂度为 $O(n)$,当内存空间不足时,可以用时间来换取空间。

4　实验评价

这一节对算法的有效性、可伸缩性与执行效率进行实验评价。有效性是指算法能正确发现异常点的能力;可伸缩性分为行可伸缩性和维可伸缩性,前者指算法对大规模数据集的处理能力,后者指算法对高维数据集的处理能力;执行效率主要考虑算法的运行时间。我们选择基于 SR 树的 LOF 算法、DBSCAN 算法与 ODINP 算法进行比较,有 3 个主要原因:(1) SR 树是 R* 树[12]和 SS 树[13]的扩展,将(超)矩形和(超)球形两种不同的数据划分方法结合起来,改善了区域之间的不相交性,增强了近邻查询的性能并可有效支持高维近邻查询;(2) 3 个算法都是基于密度的方法;(3)3 个算法的时间复杂度都为 $O(n\log n)$,是效率比较高的算法。实验中,算法用 VC6.0 实现,所有的测试是在 CPU 为 Pentium4 3.0G,主存为 512M,操作系统为 Windows XP Professional 的机器上进行。

4.1　有效性实验

有效性实验的测试数据集来自著名算法 DBSCAN 中数据库 1(ds1)、数据库 2 (ds2)、数据库 3 (ds3)和 NBC 中的数据集(ds4),如图 2(a)—(d)所示。ds3 和 ds4 中分别包含 26 和 21 个异常点,可以直接用于有效性实验,在 ds1 和 ds2 中我们采用 ODBSN 一文中的做法,加入一定量的异常点。$k=10$ 时,异常点检测结果如图 3 所示,圆点为异常点,ODINP 算法能全部正确检测到 ds1 和 ds2 中的各 10 个异常点、ds3 中

的 26 个异常点以及 ds4 中的 21 个异常点。

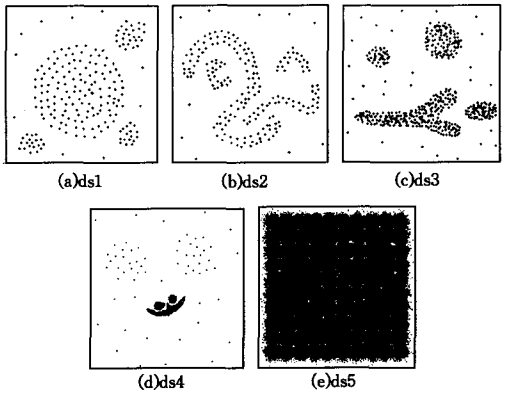


图 2　数据集

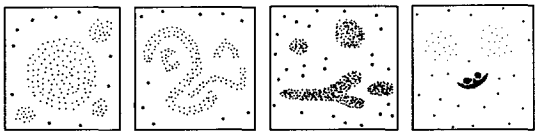


图 3　算法 ODINP 检测到的异常点示意图

最小簇的大小可作为选择 k 值的启发信息，图 4 示意了算法精度与 k 值的变化关系，当 k 值位于 4 与最小簇的大小之间时，算法具有很高的异常点检测精度。此外，从图中可看出，若 k 值太小，如小于 4，就有部分聚类点被误认为是异常点，这是因为 k 值太小导致在计算聚类边缘点的 NLDF 值时不能更多考虑它所在簇中对象的信息。若 k 值太大，如超过最小簇的大小，就会有部分位于最小簇边缘的异常点丢失，这是因为位于最小簇边缘的异常点的 k 邻域包含整个最小簇中对象，并且它也位于最小簇中所有对象的反 k 邻域中，这就会平滑 NDF 对 NLDF 值的作用。

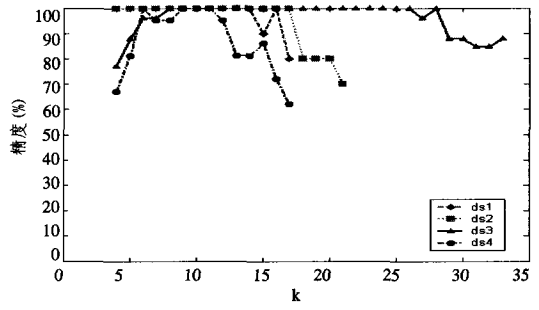


图 4　算法精度与 k 值关系

4.2　可伸缩性和效率评价

为了说明算法保持以往基于密度的异常点检测算法的高效性，我们按 BIRCH[14] 中所提到的模拟数据发生器产生了 10 * 10 个聚类中心，呈网格分布，均值在聚类中心的正态分布数据集 ds5，如图 2(e)所示。数据量依次为 10 000，20 000，…，100 000 个，随机分布的异常点比例为 1%。DBSCAN，LOF 和 ODINP 算法的执行效率如图 5 所示。

为了测试算法的效率和维可伸缩性，我们使用 UCI 机器学习数据库中 Corel 图像特征数据集，它共包含 68040 个从 Corel 图像集合中提取的特征，测试结果如图 6 所示。从图 6 中我们可以看出，在中高维情况下算法运行时间与对象个数的变化曲线接近线性。

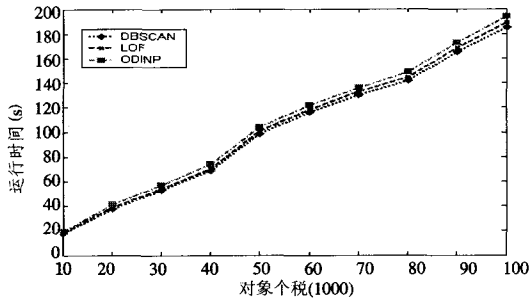


图 5　算法执行效率比较

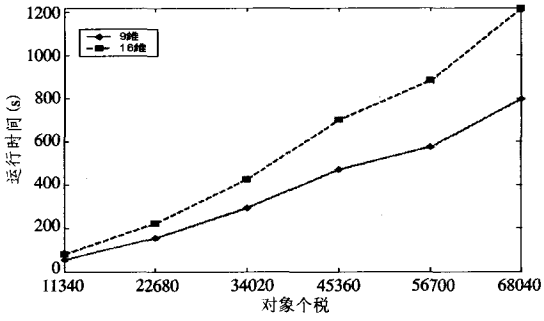


图 6　ODINP 算法运行时间

结束语　对于许多数据挖掘的应用来说，异常点检测是一项重要任务。已有的基于密度的异常点检测算法需要设置两个输入参数，并对输入参数敏感，用户不正确的设置会导致算法不能发现所有有意义的异常点，甚至是发现错误的异常点，而且多个参数还使得使用不同算法发现的异常点之间无法进行比较。受相关研究的启发，本文在分析 NDF 的缺陷后，根据对象的邻域、反邻域和局部密度构造基于邻域的局部密度因子 NLDF，NLDF 避免了全局异常点的二值属性和 LOF 中当稀疏聚类中的对象靠近稠密聚类时错误的异常点判断。吸收基于密度方法的优点，提出一种屏蔽输入参数敏感的异常点检测算法，该算法一个非常显著的优点就是只需要一个参数 k，在启发信息的帮助下，算法对 k 不敏感。理论分析和实验结果表明，该算法保持了以往基于密度的异常点检测算法的高效性，并具有很高的异常点检测精度，可有效地从大规模、任意形状的和高维数据集中发现有意义的异常点。

参 考 文 献

[1] Breunig M M, Kriegel H-P, Ng R T, et al. LOF: Identifying Density-Based Local Outliers // Proc. 2000 ACM SIGMOD Int'l Conf. on Management of Data. Dallas, TX, 2000: 93-104

[2] 黄添强，秦小麟，叶飞跃. 基于方形邻域的离群点查找新方法. 控制与决策，2006，21(5)：541-545

[3] Ester M, Kriegel H-P, Sander J, et al. A Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise // Proc. 2nd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Portland, Oregon, 1996: 226-231

[4] Zhou S, Zhao Y, Guan J, et al. A Neighborhood-Based Clustering Algorithm // Proc. 9th Pacific-Asia Conf. on Knowledge Discovery and Data Mining. Hanoi, Vietnam, 2005: 361-371

[5] Ankerst M, Breunig M, Kriegel H-P, et al. Optics: Ordering Points to Identify the Clustering Structure // Proc. 1999 ACM SIGMOD Int'l Conf. on Management of Data. Philadelphia, PA, 1999: 49-60

(下转第 206 页)

问题。

参 考 文 献

[1] Picard R W. Affective Computing. Cambridge[M]. London, England: MIT Press, 1997

[2] Bernstein D A, Stewart A C, Roy EJ, et al. Psychology (forth edition)[M]. New York: Houghton Miffin Company, 1997: 360-361

[3] Ortony A, Clore GL, Collins A. The cognitive structure of emotions[M]. New York: Cambridge University Press, 1988: 68-83

[4] Ortony A. On making believable emotional agents believable [A]// Trapple R P, ed. Emotions in humans and artifacts. Cambridge: MIT Press, 2003: 189-211

[5] Bates J. The role of emotion in believable characters [J]. Communications of the ACM, 1994, 37(7): 122-125

[6] Loyall AB. Some requirements and approaches for natural language in a believable agent [A]// Trappl R, Petta P, eds. Creating Personalities for Synthetic Actors: Towards Autonomous Personality Agents. Berlin: Springer Verlag, 1997: 113-119

[7] Reilly WSN. Believable Social and Emotional Agents. Ph. D. Dissertation. Carnegie-Mellon University, 1996

[8] Badler N I, Reich B D, Weber B L. Towards personalities for animated agents with reactive and planning behaviors [A]// Trappl R, Petta P, eds. Creating Personalities for Synthetic Actors: Towards Autonomous Personality Agents. Berlin: Springer Verlag, 1997: 43-57

[9] Chi D, Costa M, Zhao LW, et al. The Emote Model for Effect and Shape [A]// Proceedings of SIGGRAPH'2000 Conference. New Orleans, Louisiana USA, 2000: 173-182

[10] Ball G, Breese J. Emotion and Personality in a conversational agent [A]// Cassell J, Sullivan J, Prevost S, et al., eds. Embodied conversational agents. Cambridge: MIT Press, 2000: 189-219

[11] Perlin K, Goldberg A. Improv: a system for scripting interactive actors in virtual worlds [A]// Proceedings of SIGGRAPH'1996 Conference. New Orleans, Louisiana USA, 1996: 205-216

[12] Rousseau D, Hayes-Roth B. A Social-Psychological Model for Synthetic Actors [A]// Proceedings of the Second International Conference on Autonomous Agents. Minneapolis, MN, May 1998

[13] Moffat D. Personality parameters and programs [A]// Trappl R, Petta P, eds. Creating Personalities for Synthetic Actors: Towards Autonomous Personality Agents. Berlin: Springer Verlag, 1997: 120-165

[14] Magnenat-Thalmann N, Thalmann D. Handbook of Virtual Humans [M]. John Wiley & Sons, 2004

[15] Egges A, Kshirsagar S, Thalmann N M. Generic personality and emotion simulation for conversational agents [J]. Computer Animation and Virtual Worlds, 2004, 15(1): 1-13

[16] Gratch J, Marsella S. A Domain-independent framework for modeling emotion [J]. Journal of Cognitive Systems Research, 2004, 5(4): 269-306

[17] Read S J, Miller L S, Rosoff A, et al. Integrating Emotional Dynamics into the PAC Cognitive Architecture [A]// Proceedings of the 15th Annual Conference on Behavioral Representation in Modeling and Simulation. Orlando, FL: Institute for Simulation and Training, 2006

[18] Unuma M, Anjyo K, Takeuchi R. Fourier Principles for Emotion-Based Human Figure Animation [A]// Proceedings of SIGGRAPH'1995 Conference. Los Angeles, CA, 1995: 91-96

[19] Rose CF, Cohen M, Bodenheimer B. Verbs and Adverbs: Multidimensional Motion Interpolation [J]. IEEE Computer Graphics& Application, 1998, 18(5): 32-40

[20] Parke FI, Waters K. Computer Facial Animation[M]. Wellesley, Boston, USA: AK Perters, 1996: 105-147

[21] Pandzic IS, Forchheimer R. MPEG-4 Facial Animation: The Standard, Implementation and Applications [M]. John Wiley & Sons, 2002

[22] Blumberg B, Galyean T. Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments [A]// Proceedings of SIGGRAPH'1995 Conference. Los Angeles, CA, 1995: 47-54

[23] Velasquez J D. Modeling emotions and other motivations in synthetic agents [A]// Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97). 1997: 10-15

[24] 王国江, 王志良, 陈锋军, 等. 基于 Markov 决策过程的交互虚拟人情感计算模型[J]. 计算机科学, 2006, 33(12): 135-138

[25] 刘箴, 潘志庚. 智能体情绪行为动画模型 [J]. 中国图象图形学报, 2003, 8(7): 817-822

[26] 裴玉茹, 查红彬. 真实感人脸的形状与表情空间 [J]. 计算机辅助设计与图形学学报, 2006, 18(5): 613-619

(上接第 195 页)

[6] 蔡颖琨, 谢昆青, 马修军. 屏蔽输入参数敏感性的 DBSCAN 改进算法. 北京大学学报: 自然科学版, 2004, 40(3): 480-486

[7] Keogh E, Lonardi S, Ratanamahatana C A. Towards Parameter-free Data Mining // Proc. 2004 ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Washington, USA, 2004: 206-215

[8] Hawkins D M. Identification of Outliers[M]. London: Chapman and Hall, 1980

[9] Knorr E M, Ng R T. Algorithm for Mining Distance-Based Outliers in Large Datasets// Proc. of the 24th Int'l Conf. on Very Large Database. New York, USA, 1998: 392-403

[10] Jin W, Tung A K H, Han J, et al. Ranking Outliers Using Symmetric Neighborhood Relationship // Proc. 10th Pacific-Asia Conf. on Knowledge Discovery and Data Mining. Singapore, 2006: 93-104

[11] Katayama N, Satoh S. The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries// Proc. 1997 ACM SIGMOD Int'l Conf. on Management of Data. AZ, USA, 1997: 369-380

[12] Beckmann N, Kriegel H-P, Schneider R, et al. The R*-tree: An Efficient and Robust Access Method for Points and Rectangles [C]// Proc. 1990 ACM SIGMOD Int'l Conf. on Management of Data. New Jersey, USA, 1990: 322-331

[13] White D A, Jain R. Similarity Indexing with the SS-tree // Proc. of the 12th Int'l Conf. on Data Engineering. New Orleans, USA, 1996: 516-523

[14] Zhang T, Ramakrishnan R, Livny M. BIRCH: An efficient data clustering method for very large databases // Proc. of the ACM SIGMOD International Conference on Management of Data. Montreal, Canada, 1996: 103-114