一种基于随机游动的聚类算法

李 强 何 衍 蒋静坪
(浙江大学电气工程学院 杭州 310027)

摘 要：该文提出一种改进的随机游动模型，并在此模型的基础上，发展了一种数据聚类算法。在此算法中，数据集中的样本点根据改进的随机游动模型，生成有权无向图 $G(V,E,d)$，其中每个样本点对应图 G 的一个顶点，并且假设每个顶点为可以在空间中移动的 Agent。随后计算每个顶点向其邻集中顶点转移的概率，在随机选定邻集中的一个顶点作为转移方向后，移动一个单位距离。在所有样本点不断随机游动的过程中，同类的样本点就会逐渐的聚集到一起，而不同类的样本点相互远离，最后使得聚类自动形成。实验结果表明，基于随机游动的聚类算法能使样本点合理有效地被聚类，同时，与其他算法对比也说明了此算法的有效性。

关键词：数据聚类；随机游动；无监督学习
中图分类号：TP391.4 **文献标识码**：A **文章编号**：1009-5896(2009)03-0523-04

A Random Walk Based Clustering Algorithm

Li Qiang He Yan Jiang Jing-ping
(*College of Electrical Engineering*, *Zhejiang University*, *Hangzhou* 310027, *China*)

Abstract: In this paper, a modified model of random walk is proposed, and then a clustering algorithm is developed based on this model. In the algorithm, at first a weighted and undirected graph $G(V,E,d)$ is constructed among data points in a dataset according to the model, where each data point corresponds to a vertex in the graph, and is regarded as an agent who can move randomly in space. Next, the transition probabilities of data points are computed, and then each data point chooses a neighbor randomly in its neighborhood as a transition direction and takes a step to it. As all data points walk in space at random repeatedly, the data points that belong to the same class are located at a same position, whereas those that belong to different classes are away from one another. Consequently, the experimental results demonstrate that data points in datasets are clustered reasonably and efficiently. Moreover, the comparison with other algorithms also provides an indication of the effectiveness of the algorithm.

Key words: Data clustering; Random walk; Unsupervised learning

1 引言

数据聚类是模式识别领域内一个重要的研究课题。回顾聚类算法的发展，可以发现一个显著的变化，即在过去很长一段时间里，要聚类或分类的样本点都是固定不动的，人们设计各种算法去找到复杂的聚类或分类平面。然而，近年来，一些研究者提出，为什么样本点不能像 Agent 一样，可以自己根据一些规则，在空间中自由移动而自动聚集到一起呢？因此，研究者根据他们的新思想，开发出一些令人兴奋的新的聚类算法[1−3]。这些革新的聚类算法都有一个显著的特征，那就是样本点本身可以根据一些规则或定律在空间中移动，从而自动实现数据聚类。

近十年来，随机游动在计算机，物理，生物等领域都有广泛的应用，然而，随机游动在模式识别领域的应用却并不太多。例如，Yen 等[4]研究了一种基于随机游动的新的距离度量，并将其应用到经典的 K-means 聚类算法中。Harel 等[5]提出一种基于有权图上随机游动的决定性探索算法，它利用一个分割算符(separating operators)聚类空间数据。Erkan 等[6]将 Harel 的随机游动模型推广到有向图的情况，并给出一个基于语言模型的文件聚类算法，作为这一模型的应用实例。

这些算法都是建立在经典随机游动模型上的，与他们的算法不同，本文提出一种改进的随机游动模型，并基于此模型提出一种数据聚类算法。在本文的算法中，数据集中的样本点(图的顶点)被看作是可以在空间中随机游动的 Agent。根据一些随机游动的规则，每一时刻，所有的样本点随机选择向它的一个邻居的方向移动一个单位距离 Δu，所以图的大小和形状都是随时间变化的。在样本点不断移动的过程中，同类的样本点就会逐渐地聚集到一起，而不同类的样本点相互远离，从而使得聚类自动形成。

2007-10-15收到，2008-09-23改回
国家自然科学基金(60405012, 60675055)资助课题

2 随机游动模型

2.1 图上的随机游动

图 $G(V,E,\omega)$ 是一个有 n 个顶点，m 条边的有权无向图，其中 V 是顶点集，E 是边集，$\omega: V\times V\to \mathbb{R}$ 是连接权函数。进一步假设有一个粒子 A，初始时在图 G 上的顶点 v_0 处，p_{ij} 表示粒子 A 向它的一个邻居移动的转移概率，其计算公式如下[7]：

$$p_{ij}=\begin{cases}\omega_{ij}/\omega_i, & j\in \Gamma(i)\\ 0, & \text{其他}\end{cases} \tag{1}$$

ω_i 为邻集权重之和，即 $\omega_i=\sum_{j\in\Gamma(i)}\omega_{ij}$。

如果粒子 A 以概率 p_{ij} 从顶点 v_i 游动到顶点 v_j，并不断重复这一过程，那么被粒子 A 访问过的顶点就组成了一个随机序列 $X_n,\ n=0,1,2,\cdots$。

定义 1[7] 有权无向图 $G(V,E,\omega)$ 上的随机游动为一随机序列：$X_0,X_1,\cdots,X_n,\cdots$，其中 X_n 表示粒子在 n 时刻的位置。并且，如果粒子在 n 时刻位于顶点 v_i，那么下一时刻位于顶点 v_j 的概率为 p_{ij}。

2.2 改进的随机游动模型

对于任意数据集 $S=\{\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_n\}$，初始时，样本点间并无边的连接关系。因此，首先要定义一个连接权(距离)函数 $d: V\times V\to\mathbb{R}$，计算出所有点间的权重(距离)，然后设定一个最大感知距离 R。如果两样本点间的距离小于 R，那么就在这两个样本点间添加一条边 $e(i,j)$。当所有样本点都按此方法添加边之后，就产生一个有权无向图 $G(V,E,d)$。

定义 2 数据集 $S=\{\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_n\}$ 中的所有样本点，生成的有权无向图 $G(V,E,d)$：

$$\left.\begin{aligned}&V=\{v_i\mid i=1,2,\cdots,n\}\\ &E=\bigcup_{i=1}^{n}E_i, E_i=\{e(i,j)\mid j\in V, d(i,j)<R\}\\ &\Gamma(i)=\{j\mid e(i,j)\in E_i, j\in V\}\\ &D=\bigcup_{i=1}^{n}D_i, D_i=\{d(i,j)\mid e(i,j)\in E_i, i,j\in V\}\end{aligned}\right\} \tag{2}$$

这里，每个样本点 $\boldsymbol{X}_i$ 对应图 G 中的一个顶点 v_i，$\Gamma(i)$ 为顶点 v_i 的邻集。

在经典的图上随机游动中，图本身的连接关系、权重等是固定不变的，通过放入一个外加的粒子，在图上根据转移概率进行游动。然而，在本文改进的随机游动模型中，所有顶点(样本点)都被考虑为可以移动的 Agent。这样，随着各顶点(Agent)的移动，在样本点生成的图 G 中，顶点的位置、连接关系、权重等都是随时间变化的。另外，在经典的图上随机游动中，粒子的转移概率只是权重 ω_{ij} 的函数。而在改进的随机游动模型中，顶点 v_i 向顶点 v_j 方向转移的概率，不仅与权重 $d(i,j)$ 有关，而且还与顶点 v_j 的连接密度 L_{ij} 有关。

定义 3 $\forall e(i,j),e(i,k)\in E_i, v_i\in V, j,k\in\Gamma(i)$，顶点 v_j 的连接密度 L_{ij} 为

$$L_{ij}=\left|\left\{l\mid l=\angle\big(e(i,j),e(i,k)\big)<\angle\alpha, j\neq i, k\neq i\right\}\right| \tag{3}$$

这里，$\angle\big(e(i,j),e(i,k)\big)$ 表示边 $e(i,j)$ 和 $e(i,k)$ 的夹角，符号 $|\bullet|$ 表示集合的势。

图 G 中的每个顶点 v_i，可以选择向它邻集 $\Gamma(i)$ 中任何邻居顶点的方向移动。初始时，顶点 v_i 到自身的距离 $d(i,i)=0$，根据定义 2，那么它本身也包含在其邻集 $\Gamma(i)$ 中，所以顶点 v_i 也可以向它自己的方向转移。这样，为保证顶点 v_i 向自身的转移概率不为零，我们必须重新定义顶点 v_i 到自身的距离 $d(i,i)$。

定义 4 $\forall v_i\in V$，顶点 v_i 到自己的距离为

$$d(i,i)=\sum_{j\in\Gamma(i),j\neq i}d(i,j)//\big(\big|\Gamma(i)\big|-1\big) \tag{4}$$

在定义了顶点到自身的距离之后，顶点 v_i 向其邻集中的邻居顶点转移的概率 p_{ij}，就可以根据下面的定义计算出来。

定义 5 $\forall v_i,v_j\in V, e(i,j)\in E_i, d(i,j)\in D_i$，令

$$\left.\begin{aligned}&Y_i=\sum_{e(i,j)\in E_i}Y_{ij}\\ &Y_{ij}=\begin{cases}L_{ij}/d(i,j), & d(i,j)<R\\ 0, & \text{其他}\end{cases}\end{aligned}\right\} \tag{5}$$

那么，顶点 v_i 向其邻集 $\Gamma(i)$ 中的顶点转移的概率为

$$p_{ij}=\begin{cases}Y_{ij}/Y_i, & j\in\Gamma(i)\\ 0, & \text{其他}\end{cases} \tag{6}$$

所以，转移概率矩阵 $\boldsymbol{P}=(p_{ij})_{i,j\in V}$。

顶点 v_i 的游动方向由一个事件组 $\mathrm{EVE}_i=\{\mathrm{eve}_{ij}\mid j=1,\cdots,h_i,\ h_i=|\Gamma(i)|\}$ 决定，事件组中的每一个事件 eve_{ij} 对应一个可转移的方向。对于顶点 v_i，每一时刻，事件组 EVE_i 中有且只有一个事件发生，用 $B(\mathrm{EVE}_i)=\mathrm{eve}_{ij}$ 表示事件组中对应顶点 v_j 的事件发生，$B(\bullet)$ 是一个事件产生函数。值得注意的是，顶点 v_i 并不是一步直接转移到顶点 v_j 的位置，而只是向顶点 v_j 的方向移动一个单位距离 $\triangle u_i$。并且，如果两顶点间的距离 $d(i,j)<\beta$，β 为一个防止两顶点碰撞的阈值，或得到的事件为 $B(\mathrm{EVE}_i)=\mathrm{eve}_{ii}$，即向自身转移，那么顶点 v_i 将留在原地不动。当所有顶点均完成一次转移之后，顶点的位置将全部被更新。

定义 6 对于 $\forall v_i,v_j\in V, e(i,j)\in E_i, d(i,j)\in D_i, B(\mathrm{EVE}_i)=\mathrm{eve}_{ij}$，顶点 v_i 的位置按如下公式更新：

$$\left.\begin{aligned}&\triangle\boldsymbol{X}_i=\begin{cases}\dfrac{\boldsymbol{X}_j-\boldsymbol{X}_i}{d(i,j)}\triangle u_i, & d(i,j)\geq\beta\text{并且}\\ & B(\mathrm{EVE}_i)\neq\mathrm{eve}_{ii}\\ 0, & \text{其他}\end{cases}\\ &\boldsymbol{X}_{i,\mathrm{new}}=\boldsymbol{X}_i+\triangle\boldsymbol{X}_i\end{aligned}\right\} \tag{7}$$

当所有样本点更新自己的位置以后，模型的一次迭代完成。

3 基于随机游动的聚类算法

假设一个未标记的数据集 $S=\{\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_n\}$，每个样本点有 p 个属性。在基于改进的随机游动模型的聚类算法中，数据集中的每个样本点被考虑为一个可在整个空间中移动的 Agent，并分别对应有权无向图 $G(V,E,d)$ 中的一个顶

点。

3.1 算法描述

(1)设定最大感知距离 R，防止碰撞阈值 β 和转移步长 Δu_i；

(2)选择相似度(距离)函数，计算数据集中样本点的相似度矩阵 $\boldsymbol{D}$，合并小于碰撞阈值 β 的点；

(3)根据定义 2，确定每个顶点 v_i 的邻集 $\Gamma(i)$；

(4)按定义 5，计算每个顶点 v_i 向其邻集 $\Gamma(i)$ 中的所有顶点方向转移的概率 p_{ij}；

(5)掷一个有 $h_i=\left|\Gamma(i)\right|$ 面的有偏骰子，以此作为事件产生函数 $B(\mathrm{EVE}_i)$，决定顶点 v_i 的方向转移；

(6)如果顶点 v_i 与被选定的顶点 v_j 间的距离 $d(i,j)<\beta$，或 $B(\mathrm{EVE}_i)=\mathrm{eve}_{ii}$，那么顶点 v_i 的坐标不进行更新。否则，按定义 6 更新顶点坐标；

(7)按更新后的顶点坐标，重新计算相似度矩阵 $\boldsymbol{D}$；

(8)转回第(3)步。

3.2 算法分析

在我们的算法中，最大感知距离 R 的值，由相似度矩阵 $\boldsymbol{D}$ 的上三角($\boldsymbol{D}^{\Delta}$)或下三角($\boldsymbol{D}_{\Delta}$)部分的均值和中位数之差 $\mathrm{dif}=\mathrm{mean}(\boldsymbol{D}^{\Delta})-\mathrm{median}(\boldsymbol{D}^{\Delta})$ 来计算。如果 dif 的值很小，说明类间距离很小，那么 R 就需要取一个比较小的值。而当 dif 的值较大时，说明类间距离相对较大，或是样本点的分布比较稀疏，那么 R 就要取一个相对大的值。对于同一个数据集，R 的值也部分决定了聚类类数的多少。一般来说，随着 R 的增大，聚类的类数减少。因此，可根据实际要求，通过调整 R 的值，得到不同的聚类类数。

另外，顶点 v_i 的转移概率 p_{ij}，不仅与它到各顶点的距离 $d(i,j)$ 有关，还与定义 3 中的连接密度 L_{ij} 有关。连接密度 L_{ij} 的值，说明了在边 $e(i,j)$ 的周围邻边的条数，即以边 $e(i,j)$ 为 $\angle\alpha$ 的角平分线，落入此范围内的边 $e(i,k)$ 的条数，$k\in\Gamma(i),k\neq i$。这样就克服了只使用距离值来计算转移概率时，如果两点的距离近，那么转移概率就高的缺点。另一方面，当顶点 v_i 与顶点 v_j 的距离 $d(i,j)<\beta$，或 $B(\mathrm{EVE}_i)=\mathrm{eve}_{ii}$ 时，即使选中此方向，顶点 v_i 也不向此方向转移。这就使得相互靠近的顶点，仍有分离的可能，从而避免顶点过早地聚集到一起。

顶点 v_i 的转移方向由一个事件产生函数 $B(\mathrm{EVE}_i)$ 来确定。在算法中，我们通过掷一个有 $h_i=\left|\Gamma(i)\right|$ 面的有偏骰子来决定，骰子的每一面对应事件集 EVE_i 中的一个事件。并且，骰子是有偏的，这是因为向邻集中的每个顶点的转移概率是不同的。这样，顶点就不会总是向概率最大的顶点方向转移，从而有可能使得聚类结果跳出局部最优。另外，顶点并不是一步就转移到另一个顶点，而只是向这个顶点的方向游动一步 Δu_i，这就增大了样本点对解空间的探索范围。样本点在每步转移以后，自己在样本空间的位置，以及与邻居点的位置关系都发生变化。因此，它受到的局域作用必然发生变化，从而导致样本点的行为比静态时复杂得多，一些好的结果也在此过程中自然得到。

3.3 参数的影响

算法中的步长 Δu_i 决定了顶点每一次转移的长度。我们从 UCI 数据库[8]中选择了 4 个数据集：Soybean，Iris，Wine 和 Breast 作为实验数据集，Δu_i 分别取 0.1,0.2,0.3，来观察 Δu_i 的变化对数据聚类正确率的影响。在 Δu_i 取定一个值后，分别在每个数据集上，独立运行此算法 20 次，然后将得到的聚类正确率(见第 4 节定义 7)按由小到大的顺序排列，所有结果如图 1 所示。

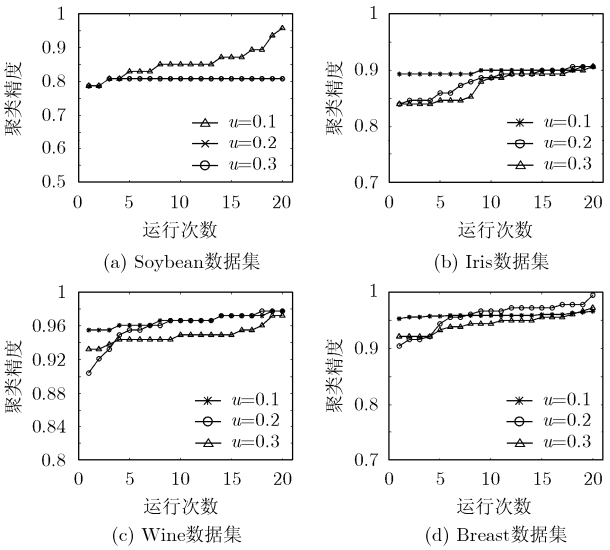

图 1　Δu_i 变化对聚类正确率的影响

从图 1 可以看出，随着 Δu_i 的增大，算法的聚类正确率略有下降的趋势。步长 Δu_i 的增大，使得顶点每次转移后，位置坐标的变化随之增大。而一般来说，数据集内样本点间的距离相对较小，一个较大的 Δu_i 值，会造成某些边界上的样本点向其他类转移后，再回到本类的机会减小。而如果步长 Δu_i 取一个相对小的值，即使这个顶点向其他类转移了一步或几步，但其坐标的变化相对较小，这就使得此顶点还有较大机会转移回本类。另一方面，大的 Δu_i 值，会使数据集聚类速度加快，但从上一节的分析可知，这也同时降低了由顶点转移而产生的有权无向图 $G(V,E,d)$ 的变化，从而减小了对解空间搜索的范围。因此，在兼顾聚类速度的基础上，尽可能选择小一些的 Δu_i 值。

4　实验

为测试算法，我们从 UCI 数据库[8]中选择 4 个数据集：Soybean，Iris，Wine 和 Breast cancer Wisconsin，并在所有数据集上完成实验。对于 Breast 数据集中丢失的属性，用一个随机数替换。

在所有实验中，数据集中的样本点被看作是图 $G(V,E,d)$ 的顶点，顶点的初始位置从原始数据集中直接取

值。而最大感知距离 R，根据各数据集 dif 的大小设定。由 3.3 节的分析知，步长 $\triangle u_i$ 应取一个相对小的值，以得到更高的聚类正确率，所以实验中令 $\triangle u_i = 0.1$。数据集中样本点间的相似度，由改进的高斯距离函数 $d(i,j)=\exp(||\boldsymbol{X}_i - \boldsymbol{X}_j|| /2\sigma^2)$ 得到，因为它克服了欧氏距离在两样本点非常相似时，函数值过小的缺点。改进的高斯距离最小值为 1，给计算带来许多方便。聚类算法结束后，将此算法得到的结果用聚类正确率的形式来表示：

定义 7[6] 令 cluster_i 为算法分配给数据集中样本点 X_i 的类标签，c_i 为样本点 X_i 在数据集中的实际类标签。那么，聚类正确率为

$$\left.\begin{aligned}&\text{accuracy} = \sum_{i=1}^{n} \lambda\big(\text{map}(\text{cluster}_i), c_i\big) \Big/ n \\ &\lambda\big(\text{map}(\text{cluster}_i), c_i\big) = \begin{cases} 1, & \text{map}(\text{cluster}_i) = c_i \\ 0, & \text{其他} \end{cases}\end{aligned}\right\} \tag{8}$$

其中映射函数 $\text{map}(\bullet)$ 将算法得到的类标签集，映射到数据集的实际类标签集。

为说明算法的有效性，我们也与其它两种聚类算法(DMVC[9]和 LDA-Km[10])在同一数据集上的结果作了简单的比较。由于随机游动算法具有不确定性，所以将此算法独立运行 20 次，表 1 中列出了这 20 次结果的均值与方差以及得到的最好结果。

表 1 算法聚类正确率比较

算法	Soybean	Iris	Wine	Breast
随机游动	85.11± 4.84%	89.8± 0.44%	96.60± 0.69%	95.89± 0.29%
最好结果	95.75%	90.67%	97.75%	96.57%
DMVC	100%	84%	95.5%	96.3%
LDA-Km	76.6%	98%	82.6%	—

5 结论

本文提出一种改进的随机游动模型，此模型首先定义由数据集的样本点产生有权无向图 $G(V,E,d)$ 的方法，然后通过连接密度 L_{ij} 和权重 $d(i,j)$ 计算每个顶点的转移概率，最后利用事件产生函数 $B(\text{EVE}_i)$，确定顶点的转移方向，并给出顶点的更新公式。在此模型的基础上，研究了一种数据聚类算法。算法中，数据集中的每个样本点对应有权无向图 $G(V,E,d)$ 的一个顶点，而且样本点自身是可以移动的 Agent。在空间中，顶点 v_i 根据转移概率选择其邻集中的一个顶点 v_j，作为转移方向，并向这个顶点方向移动一个单位距离 $\triangle u_i$。在所有样本点不断移动的过程中，同类的样本点就会逐渐的聚集到一起，而不同类的样本点相互远离，最后使得聚类自动形成。

本文实验结果表明，此算法能够使样本点合理有效地被聚类。同时，如果有一些关于样本点的先验知识，比如知道类的大致形状，那么根据这些先验知识，可以有针对性地选择相似度测量函数，从而得到更好的结果。

参 考 文 献

[1] Merwe V D and Engelbrecht A P. Data clustering using particle swarm optimization. Proceedings of IEEE Congress on Evolutionary Computation 2003 (CEC2003), Canbella, Australia, 2003: 215–220.

[2] Labroche N, Monmarché N, and Venturini G. AntClust: Ant Clustering and Web Usage Mining. Genetic and Evolutionary Computation Conference, Chicago, IL, USA, 2003: 25–36.

[3] Cui X H, Gao J Z, and Potok T E. A flocking based algorithm for document clustering analysis. *Journal of Systems Architecture*, 2006, 52(8): 505–515.

[4] Yen L, Vanvyve D, and Wouters F, *et al*.. Clustering using a random walk based distance measure. Proceedings of the 13th Symposium on Artificial Neural Networks, Bruges (Belgium), 2005: 317–324.

[5] Harel D and Koren Y. Clustering spatial data using random walks. Proceedings of the Seventh ACM SIGKDD Conference, NY, USA, ACM Press, 2001: 281–286.

[6] Erkan G. Language Model-Based Document Clustering Using Random Walks. Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, New York, June 2006: 479–486.

[7] Lovász L. Random walks on graphs: A survey. Bolyai Society Mathematical Study, Combinatorics, Paul Erdös is eighty, Keszthely, Hungary, 1993: 1–46.

[8] Blake C L and Merz C J. UCI Repository of machine learning databases, 1998.

[9] Song L, Smola A, and Gretton A, *et al*.. A dependence maximization view of clustering. Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, 2007: 815–822.

[10] Ding C and Li T. Adaptive dimension reduction using discriminant analysis and *K*-means clustering, Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, 2007: 521–528.

李 强： 男，1978 年生，博士生，研究方向为模式识别、量子计算.

何 衍： 男，1973 年生，副教授，研究方向为信息融合、多机器人协作.

蒋静坪： 男，1935 年生，教授，博士生导师，研究方向为智能系统与智能控制、电力传动自动化.