* 1

一种基于静电场概念的聚类算法

李小喜[1]，　辛永平[1]，　陈疆萍[1]，　张明学[2]
(1. 空军工程大学　导弹学院，陕西　三原　713800；2. 空军工程大学　理学院，陕西　西安　710051)

摘　要：在静电场中质心位于静电平衡或那些能够被接受的平衡位置处，基于这一思想提出了一种发现簇中心的新方法。根据静电场中电荷间的引力作用来确定质心位置，然后根据相应的准则(如最小距离准则等)使用选定的质心对数据点进行聚类。最后将提出的方法与 K – means 算法进行实验对比，结果表明该方法克服了 K – means 算法存在的问题，例如，对噪声和初始聚类中心敏感以及易于陷入局部最优等。该方法具有很高的效率，并且对多维数据集有强的鲁棒性。

关键词：静电场；静电平衡；K – means 算法；簇中心
DOI：10.3969/j.issn.1009 – 3516.2010.05.010
中图分类号：TP311　　**文献标识码**：A　　**文章编号**：1009 – 3516(2010)05 – 0044 – 04

聚类分析[1]是知识发现(KDD)中一项重要研究内容，旨在将数据集合划分为若干类的过程，使得类内差异小，类间差异大。通常用数据之间的距离来描述相似度，距离越大，相似度越小，反之则越大。理想的聚类算法应该具有可扩展性、能发现任意形状、用户输入参数少、对噪声不敏感、能处理高维数据、可解释性和可用性。国内外学者已经提出了不少相关的算法，大体上可分为划分方法、层次方法、基于密度的方法、基于网格的方法和基于模型的方法。典型算法分别有 K – means 算法[2]、CURE 算法[3]、DBSCAN 算法[4]、CLIQUE 算法[5]和 BIRCH 算法[6-7]，等等。这些算法以自己的特色和方式解决了一类特殊问题。

K – means 算法是最常使用的聚类算法之一。在 K – means 算法中用户必须事先给出要生成的簇的数目，然后数据点再依据最小距离准则被分配给各个中心形成簇。算法不断地更新簇中心，直到 2 次相继迭代中质心位置不再变化。虽然 K – means 算法是一种有效的聚类算法，但是它存在对噪声、离群点数据和初始簇中心的选择敏感的缺陷。

本文提出了一种新的基于静电场理论的聚类方法，利用静电场中正负电荷之间的引力，使得电荷在电场中移动，最终趋于平衡的思想。假设数据集中的数据点带单位负电荷，簇中心带正电荷，根据电荷间的引力作用，最终簇中心移动到其平衡位置。

1　Force 聚类算法

该算法利用了电磁场理论中电荷间作用力的性质。电荷间作用力的方向受电场中其他电荷的影响。基本假设：①簇质心带有大量的、正的、易变的、动态(位置上)电荷；②数据点带有单一的、负的、稳定的、静态(位置上)电荷。

当带正电的中心点随机落入带负电的数据点中时，便形成一个电场，在电场中，中心点在电荷间作用力的影响下向平衡的位置移动。在平衡状态下，中心点的位置保持不变。我们称这种状态为静电平衡。中心

* **收稿日期**：2010 – 03 – 24
基金项目：国防重点实验室基金资助项目(9140C8301011001)
作者简介：李小喜(1983 –)，男，甘肃平凉人，硕士生，主要从事数据挖掘与效能评估研究.
E – mail：lixiaoxi1983@163.com

点之间的作用力是相斥的,而中心点和数据点是相互吸引的。作用力 $F = c\frac{q_1 q_2}{r^2}$,式中:c 为常量;r 是两电荷之间的距离;q_1 和 q_2 为电荷所带电量。当距离 r 趋于 0 时,用一充分小的距离常数 R_0 代替。电荷之间作用力的方向可用单位向量$\frac{(\boldsymbol{r}_1 - \boldsymbol{r}_2)}{(\| \boldsymbol{r}_1 - \boldsymbol{r}_2 \|)}$表示,这里 $\boldsymbol{r}_1$ 和 $\boldsymbol{r}_2$ 是电荷的坐标向量。

中心点自然地向着数据点散布的区域移动,同时各中心点之间又是相互排斥的,因此它们不可能处于同一簇中。也就是说,不论中心点的初始位置在哪儿,其都会向着簇中心移动。数据点总是和距离它们最近的中心相关联,因此簇是基于最小距离约束而形成的[8-10]。

数据点所带电荷为一常量,但是中心点所带电荷数是动态变化的。每个中心点所带电荷是根据数据点的数目(假设与其对应的簇中)来确定的。例如,如果有 N_j 个数据点对应中心点 j,Q_j 为该中心点所带电荷(设每个数据点所带电荷为 1),则:

$$Q_j = \alpha N_j \quad , \quad 0 < \alpha < 1 \tag{1}$$

由此可知,中心点所带电荷的总量总是略小于数据点所带电荷总量。这意味着中心点必然被数据点吸引,并且中心点之间斥力总是小于数据点对其的引力。如果 $\alpha \geqslant 1$,斥力将会使中心点远离理想的簇中心,以致于达到稳态时质心不能位于簇中心。如果 $\alpha \ll 1$,2 个中心点电荷可能被吸引到一个簇中或者离得太近而成为一个点。如果 $0 \ll \alpha < 1$,这时中心点可以充分接近理想的质心。每个中心点所受的合力可由下式计算:

$$\boldsymbol{F}_j = \boldsymbol{F}_j^D + \boldsymbol{F}_j^C \tag{2}$$

式中:$\boldsymbol{F}_j^C$ 为每个中心点受其他中心点的作用力的合力;$\boldsymbol{F}_j^D$ 为每个中心点受数据点的作用力的合力。因此总的作用力可由下式计算得到:

$$\begin{aligned} \boldsymbol{F}_j &= \sum_{i \neq j, J \in D \cup C} \frac{Q_j Q_i (\boldsymbol{c}_j - \boldsymbol{p}_i)}{R_{ij}^2 \| \boldsymbol{c}_j - \boldsymbol{p}_i \|} \\ R_{ij} &= \begin{cases} \| \boldsymbol{c}_j - \boldsymbol{p}_i \| & \| \boldsymbol{c}_j - \boldsymbol{p}_i \| > R_0 \\ R_0 & \| \boldsymbol{c}_j - \boldsymbol{p}_i \| \leqslant R_0 \end{cases} \end{aligned} \tag{3}$$

式中:$\boldsymbol{c}_j$ 和 $\boldsymbol{p}_i$ 为中心点和数据点的位置向量;每个数据点 i 所带电量 Q_i 为 -1;R_0 为最小距离。

根据作用在质心上的力,算法可以估计出质心移动的方向。其移动的速度和每次迭代的步长受多种因素的作用,例如,质心的重量、所带的电荷量等等。取固定步长为 η,此时质心的移动方向是唯一的确定质心位置的参数。质心的位置方程为:

$$\boldsymbol{c}_j^{(\tau+1)} = \boldsymbol{c}_j^{(\tau)} + \eta_1 \frac{\boldsymbol{F}_j}{\| \boldsymbol{F}_j \|} \tag{4}$$

式中:$\boldsymbol{c}_j^{(\tau+1)}$ 为簇质心的新位置;$\boldsymbol{c}_j^{(\tau)}$ 为其先前的位置;$\frac{\boldsymbol{F}_j}{\| \boldsymbol{F}_j \|}$为质心在上一位置时作用力方向的单位向量。

经过每次迭代后,可以确定新的质心位置,根据最小距离约束确定新的簇。在新的簇中,由式(1)可计算得到质心的电荷数以及作用力的大小。当 2 次连续的迭代过程中,每个簇中心位置的移动量都小于设定的阈值时,算法停止。这一算法的最大优点之一就是在发现簇中心的过程中,初始质心位置不同,算法将产生不同的路径,并且步长不同路径也会不同。和 K – means 算法相比 Force 算法是一种全局搜索算法,其计算复杂度为 $O(nkt')$,其中 n 是对象的总数,k 是簇的个数,t'是迭代的次数,而 K – means 算法是局部搜索算法,其计算复杂度为 $O(nkt)$,其中 n 是对象的总数,k 是簇的个数,t 是迭代的次数。由于 Force 算法的迭代次数 t'比 K – means 算法的迭代次数 t 要小,所以其计算时间优于 K – means 算法。

2　算法的改进

2.1　步长的选取

当质心距离实际的簇中心较远时,需经过许多步才能移动到最终的位置。为了提高算法的性能,需要对步长作适当的调整,当质心远离实际的簇中心时,要选择更大的步长。也就是说,那些距离簇中心较远的中

心点,作用力也是比较小的,因此步长应该和作用力成反比,故对式(4)修正如下:

$$\boldsymbol{c}_j^{(\tau+1)} = \boldsymbol{c}_j^{(\tau)} + \eta_1 \frac{\boldsymbol{F}_j}{\| \boldsymbol{F}_j \|} + \eta_2 \frac{\boldsymbol{F}_j}{\| \boldsymbol{F}_j \|^2} \tag{5}$$

2.2 根据经验估计初始位置来减小运行时间

为了加快算法的速度,本节给出一个简便、有效的方法,即将初始中心点选择在数据点散布区域的中间位置。首先,要找到所有数据点的引力中心,再将初始中心点分散放置在这一区域。这一改进有效地减少了算法的迭代次数,并且最终结果的准确性也未受到影响。计算这些初始质心的方法十分简单而且容易实现。如果初始中心点位于大量数据点聚集的区域,聚类就可以很快完成。

3 实验性能评估

这里将对算法在各种不同的初始质心位置、不同的噪声水平及不同的分布情况下的性能进行实验,并与 K – means 算法的性能进行了比较,这里 K – means 算法由 Matlab 统计工具箱中的代码实现。

我们通过测量算得的质心和实际簇中心的欧式距离来评价算法在不同初始质心的性能,设初始中心点位于点(a,a)和$(a,1-a)$,这里 a 从 0 向 1 变化,每次增加 0.01。

3.1 不同初始位置对算法性能的影响

图 1 给出了 K – means 算法的欧几里得误差,由图 1 可知,Force 算法中欧几里得误差始终趋向于一平衡点,误差范围为 $2\eta_1$(由式(5)可知)。

在平衡状态时,由于作用力 F_j 很大,这时式(5)中的第 3 项趋向于 0,而 η_1 始终发挥着重要的作用。因此,经由不同轨迹得到的质心的误差最大为 $2\eta_1$。

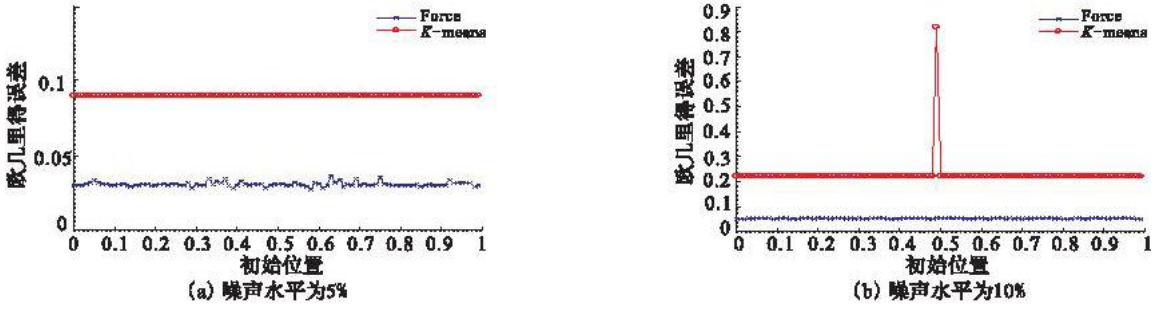

图 1 K – means 算法和 Force 算法在不同初始位置时的欧几里得误差

Fig. 1 Euclidean error of the found centers for the K – means and Force algorithm for different initial guesses

在本实验中,低噪声状态下 K – means 算法中欧几里得误差始终趋向于同一点。但是,当噪声水平增大时,算法出现了一次识别错误,由图 1 可以看出噪声使得算法趋向于一个错误的位置。而 Force 算法对噪声干扰具有很强的鲁棒性。

3.2 不同噪声水平对算法性能的影响

通过计算 2 个测试数据集在不同噪声水平和不同噪声分布下的欧几里得误差,对噪声的影响进行进一步的研究。

对测试数据集 1 中,计算在不同噪声水平下由 K – means 算法和 Force 算法得到的质心和实际质心的欧几里得距离。这里,噪声点在沿着 2 个坐标轴的(0,1)区域内均匀分布。图 2 比较了 2 种算法在不同噪声水平时的欧几里得误差。该误差是对 50 次不同初始质心进行计算得到的平均误差值。噪声水平是噪声点相对于所有数据点的比率。可以认为,随着噪声

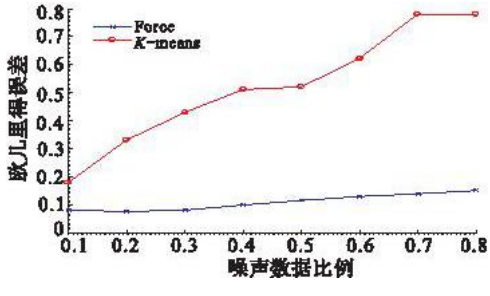

图 2 2 种算法在不同噪声水平时的欧几里得误差

Fig. 2 Euclidean error of the K – means and force algorithm as the noise level increases

水平的增大误差也在不断增大。从图中可以看出,K – means 算法的欧几里得误差增加较快而 Force 算法的误差增加非常缓慢。

测试数据集 2 中,噪声分布的均值位置自分布空间左侧向右侧移动,测试结果见图 3。

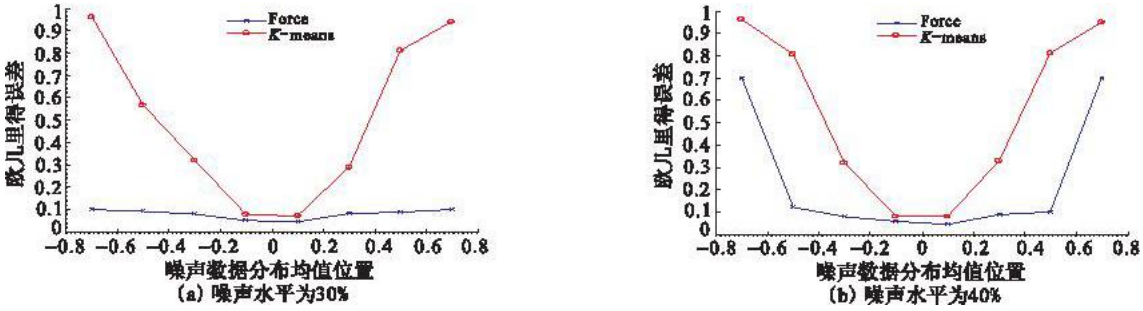

图 3　噪声分布的均值位置由 –0.7 变化到 0.7 时的欧几里得误差

Fig. 3　Mean location of noise distribution varies between –0.7 and 0.7

测试空间中,噪声分布的均值位置从 –0.7 到 0.7 变化,并且噪声样本在均值位置周围(0,1)的区间内呈同一分布。从图 7 可以看出在 0 点附近 2 种算法都很小,但随着噪声分布中心的远离,K – means 算法的误差快速增加。噪声率为 30% 时 Force 算法的误差变化较为缓和,但在噪声率为 40% 时,只有当噪声中心点在距离数据集的质心 0.5 的范围内移动时变化较小,当超出此范围时,误差变化十分大。

4　结束语

本文提出了一种新的无监督学习聚类方法,利用静电学中的规则,发现静电场中的平衡点。通过实验可以看出,该方法相对于 K – means 算法对噪声具有很好的鲁棒性。下一步的工作将研究初始中心的选择,以提高算法的运行速度。

参考文献:

[1] Han J W, Kambr M. Data Mining Concepts and Techniques[M]. Beijing: Higher Education Press, 2001.

[2] Kaufan L, Rousseeuw P J. Finding Groups in Data: An Introduction to Cluster Analysis[M]. New York: John Wiley & Sons, 1990.

[3] Guha S, Rastogi R, Shim K. CURE: An Efficient Clustering Algorithm for Large Databases[C]// Proceedings of the ACM SIGMOD International Conference on Management of Data. Seattle: ACM Press, 1998:73 – 84.

[4] Ester M, Kriegel H P, Sander J, et al. A Density Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise[C]//Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining. Portland: AIAA Press, 1996:226 – 231.

[5] Agrawal R, Gehrke J, Gunopolos D, et al. Automatic Subspace Clustering of High Dimensional Data for Data Mining Application[C]//Proceedings of the ACM SIGMOD International Conference on Management of Data. Seattle: ACM Press, 1998:94 – 105.

[6] 焦李成,刘芸,刘静,等. 智能数据挖掘与知识发现[M]. 西安:西安电子科技大学出版社,2006.
JIAO Licheng, LIU Yun, LIU Jing, et al. Intelligent Date Mining and Knowledge Discovery[M]. Xi'an: Xidian University Press, 2006. (in Chinese)

[7] Zhang T, Ramakrishnan R, Livny M. BIRCH: An Efficient Data Clustering Method for Very Large Databases[C]//Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data. Quebec: ACM Press, 1996:103 – 114.

[8] Babu G P, Murty M N. A Near Optimal Initial Seed Value Selection in K – means Algorithm Using A Genetic Algorithm [J]. Pattern Recogn Lett, 1993, 14(10):763 – 769.

[9] 孙吉贵,刘杰,赵连宇. 聚类算法[J]. 软件学报,2008, 19(1):48 – 61.
SUN Jigui, LIU Jie, ZHAO Lianyu. Clustering Algorithms Research[J]. Journal of Software, 2008, 19(1):48 – 61. (in Chinese)

(下转第 52 页)