一类数据空间网格化聚类算法的均值近似方法*

李存华+, 孙志挥

(东南大学 计算机科学与工程系,江苏 南京 210096)

A Mean Approximation Approach to a Class of Grid-Based Clustering Algorithms

LI Cun-Hua+, SUN Zhi-Hui

(Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

+ Corresponding author: Phn: 86-518-5817691, Fax: 86-518-5806171, E-mail: cli@hhit.edu.cn

http://www.seu.edu.cn

Received 2002-04-23; Accepted 2002-11-04

Li CH, Sun ZH. A mean approximation approach to a class of grid-based clustering algorithms. *Journal of Software*, 2003,14(7):1267~1274.

http://www.jos.org.cn/1000-9825/14/1267.htm

Abstract: In recent years, the explosively growing amount of data in numerous clustering tasks has attracted considerable interest in boosting the existing clustering algorithms to large datasets. In this paper, the mean approximation approach is discussed to improve a spectrum of partition-oriented density-based algorithms. This approach filters out the data objects in the crowded grids and approximates their influence to the rest by their gravity centers. Strategies on implementation issues as well as the error bound of the mean approximation are presented. Mean approximation leads to less memory usage and simplifies computational complexity with minor lose of the clustering accuracy. Results of exhaustive experiments reveal the promising performance of this approach.

Key words: clustering; grid; density-based; mean approximation; error evaluation

摘 要: 随着聚类分析对象数据集规模的急剧增大,改进已有的算法以获得满意的效率受到越来越多的重视.讨论了一类采用数据空间网格划分的基于密度的聚类算法的均值近似方法.该方法过滤并释放位于稠密超方格中的数据项,并利用其重心点近似计算其对周围数据元素的影响因子.给出均值近似在聚类算法中的实现策略及其误差估计.均值近似方法在有效减少内存需求、大幅度降低计算复杂度的同时对聚类精确度影响甚微.实验结果验证了该方法能够取得令人满意的效果.

关键词: 聚类;网格;基于密度的;均值近似;误差估计

中图法分类号: TP311 **文献标识码**: A

* Supported by the National Natural Science Foundation of China under Grant No.79970092 (国家自然科学基金); the Natural Science Foundation of the Education Board of Jiangsu Province of China under Grant No.02KJB520012 (江苏省教育厅自然科学基金)

LI Cun-Hua was born in 1963. He is a Ph.D. candidate at the Department of Computer Science and Engineering, Southeast University. His research interests are database, information system and KDDM. **SUN Zhi-Hui** was born in 1941. He is a professor and doctoral supervisor at the Department of Computer Science and Engineering, Southeast University. His current research areas are database, information system and KDDM.

© 中国科学院软件研究所 http://www.jos.org.cn

1 Introduction

Clustering analysis has been recognized as a primary data mining tool for knowledge discovery in numerous application fields such as pattern recognition, genome analysis, and market research. However, because of fast technological progress, the amount of data stored in the database increases sharply. Although various algorithms are constructed, few of them show preferable efficiency when facing large scale datasets. Another problem a clustering algorithm must face in dealing with large datasets is memory availability. Inefficient usage of the limited memory space can degrade the behavior of a clustering algorithm considerably.

Grid-partition method has been employed to develop a series of powerful clustering algorithms[1]. In this paper, we analyze several typical grid-based algorithms and introduce a novel approach to boost the behavior of them. The key idea of our approach is to apply mean approximation on the crowded grids. We analyze this approximation theoretically along with extensive experiments. Results show that mean approximation on the crowded grids can elevate the efficiency of an algorithm greatly with relatively little loss of its accuracy. To our knowledge, this is the first paper that deals with this problem explicitly in the data mining literature.

1.1 Related work

Many clustering algorithms start with partitioning the data space of the dataset into a set of hypercubes (we call them grids, here after, for convenience). For this reason, we call these algorithms grid-based. Generally, with the grid-partition mechanism, an algorithm holds information of the data objects scanned from the database into the grids, performs data retrieval, aggregates statistical values or proceeds other grid-wise computations of the data objects. This approach enables the algorithm to manage the data objects effectively, such as storing the data objects in K*-tree or X-tree[2] which is very efficient for storage and fast retrieval. The most prominent representatives of these algorithms are Optimal Grid-Clustering[3], Wavecluster[4], CLIQUE[5], STING[6] and DENCLUE[7].

In this paper, a special spectrum of the grid-based algorithms are concerned. These algorithms manage the dataset with the grids, but cluster the data object point-wisely by computing density functions defined over the underlying attribute space. Literarily, these algorithms are categorized as density-based because they only use grid-based technique at their initial stage while taking analytical methods as their major principle instrument[1]. A set of this kind of hybrid algorithms has been presented in the databases literature[3,7~10]. They share the following two common ideas:

1) Data space partition: all of the algorithms segment the data space into grids and assign each data object to a proper grid. In this way, these algorithms limit the search combinations greatly thus relieve the burden of complicated calculation. Further, these algorithms are independent of data ordering. This is another advantage over those Hierarchical or relocation approaches, which are very sensitive with respect to data ordering.

2) Density computing: these algorithms cluster the dataset analytically by computing density functions defined over the underlying attribute space. Firstly, the influence of each data object is formally modeled using a mathematical function, which is called influence function. The influence function describes the impact of a data object within its neighborhood. Then, the clusters are defined as the data sets in the dense regions which are separated using certain techniques. Apparently, this kind of approach has a firm mathematical foundation, which enables the algorithms to define and find arbitrary shaped clusters as well as to withstand the noise in the dataset.

These algorithms are often commented highly and proven to be powerful in dealing with various clustering applications with large and noisy dataset. However, with a close look into the algorithms, we find that they are still improvable for even better achievement. This is the proposal of this paper.

1.2 Our contribution

We introduce a concept called gravity center, the mean value of the data objects in a grid, and demonstrate why and how the gravity center can play an important role in grid-based clustering. We analyze the error introduced by

© 中国科学院软件研究所 http://www.jos.org.cn

employing mean approximation on the grids and give the error bound mathematically. We also forward two strategies to imbed the approximation into different phases of a clustering algorithm. We present by exhaustive experiments to show that the error of mean approximation is minor so that it affects little of the accuracy of the clustering results.

The paper is organized as follows. We first give an overview of a typical algorithm that falls within our scope in Section 2. In sections 3, we analyze the error involved in the mean approximation and forward its upper error bound. Implementation techniques on how to realize mean approximation in clustering are also concerned. In section 4, we present the results of experimental evaluations. Finally we give comments and conclude the paper.

2 Review of the Grid-Based Algorithms

In this section, we focus exclusively on the most well known representative of the algorithms addressed above, the DENCLUE (DENsity-based CLUstEring) developed by Hinneburg *et al*., The definitions and ideas it involved are essential for us to forward our results.

Let $A=\{A_1,A_2,...,A_k\}$ be a set of domains under a metric space, and $S=A_1\times A_2\times...\times A_k$ be the minimum bounding hyper-rectangle of the data space. The input D is a set of k-dimensional data objects and $|D|=N$.

Definition 1. (Influence function) The influence function of a data object $q\in D$ is a function $f_B^q: D\to R_0^+$, such that $\forall p\in D$, $f_B^q(p)=f_B(q,p)$.

The influence functions can be square wave function: $f_{Square}^q(p)=1$, if and only if $dist(q,p)\le\sigma$, or the Gaussian influence function $f_{Gauss}^q(p)=e^{-\frac{d(p,q)^2}{2\sigma^2}}$, etc. An influence function has the property: $f_B^q(p_1)\ge f_B^q(p_2)$, if and only if $dist(q,p_1)\le dist(q,p_2)$. A detailed introduction to the influence function can be found in Ref.[11].

Definition 2. (Density function) Given dataset D and the underlying attribute space S, for $\forall p\in S$, the density function at point p is defined as: $f_B^D(p)=\sum_{i=1}^N f_B^{q_i}(p)$.

With the density function defined, a series of density attractors can be obtained, where density attractors are local maxima of the density function. In DENCLUE, a cluster is defined mathematically as the subset of data objects that are attracted by their common density attractor (center defined) or set of attractors (arbitrary-shape).

Definition 3. (Center-Defined and arbitrary-shape cluster) A center-defined cluster (wrt to σ,ξ) for a density attractor p^* is a subset $C\subseteq D$, with $p\in C$ being density attracted by p^* and $f^D(p^*)\ge\xi$. An arbitrary-shape cluster (wrt to σ,ξ) for the set of density attractors X is a subset $C\subseteq D$, where

1. $\forall p\in C,\exists p^*\in X: f^D(p^*)\ge\xi, p$ is density attracted to p^* and
2. $\forall p_1^*,p_2^*\in C,\exists$ a path $P\subset F^k$ from p_1^* to p_2^* with $\forall p\in P: f^D(p)\le\xi$.

Based on the above definitions, the DENCLUE algorithm first partitions the data space S into grids. Then, with one scan through the dataset D, all data objects are mapped into proper grids and all the populated grids are numbered depending on their relative position from a given origin. For efficiency reason, the populated grids are classified into highly populated grids (with $N_c\ge\xi_c$) and sparse grids. Only grids that are highly populated are clustered, while the points in sparse grids are ignored as outliers. Practically, DENCLUE uses local density and gradient to approximate the global density and gradient by considering the influence of nearby objects exactly whereas neglecting those lying farther then the distance threshold $\sigma_{near}=\lambda\sigma$.

As addressed in section 1.1, DENCLUE inherits advantages from both grid partition and density consideration. It has a firm mathematical foundation and generalizes other clustering methods, such as DBSCAN, k-Means clusters. The algorithm is stable with respect to outliers and capable of finding arbitrarily shaped clusters. While no clustering algorithm could have less than $O(N)$ complexity, the runtime of DENCLUE scales sub-linearly with $O(N\log N)$.

© 中国科学院软件研究所 http://www.jos.org.cn

However, there are still improvements that can be done to boost it for more efficiency or to scale it up to even larger datasets. Firstly, DENCLUE needs to hold all of the original data to fulfill its clustering procedure. Thus, its behavior depends on the memory availability. For a large dataset that can not be fitted into the main memory, swap in and swap out of the data objects can degrade its behavior sharply. Secondly, for each data object to be clustered, the algorithm has to compute its local density value by summing up all the influences of nearby objects with a point wise manner, whether those points are crowded together or distributed sparsely. It pays no attention to the statistical information of the grids around a data object. This negligence also complicates the algorithm markedly.

In next section, we introduce mean representation of the crowded grid. We analyze how to use mean approximation in density computation and the error drawn by such kind of approximation.

3 Mean Approximation and Its Error Bound

3.1 Key ideas of our approach

The key idea of this paper is to take more advantages from the grid mechanism. The considerations include:

1) The data space is usually not uniformly occupied with the dataset. This inhomogeneity results in diversity of data objects in different grids. However, when the grid width is small, it is reasonable to assume that a single grid is occupied homogenously with the data objects.

2) With a predefined threshold ξ, it is provable that when a grid is crowded enough with the data objects, its member objects must belong to a same cluster. Therefore, these objects can be freed from memory even before the clustering procedure started. The only information to be kept is the point count and the gravity center of the points within this kind of grid.

3) To cluster an object in a less-crowded grid, we need to compute its density value. Since the objects in a crowded grid are substituted simply by their center-of-gravity. The point-to-point computations related to these objects are simplified as a single gravity-center-to-point computation.

3.2 Definitions and main results

Let D be a set of N k-dimensional data objects in an Euclidean space. Given a real number σ, the domain space S is partitioned into a set of non-overlapping k-dimensional grids with equal width σ in each dimension.

Definition 4. (Gravity center of a grid) Let C be a grid with m data objects $p_i = (x_{i1},...,x_{ik}) \in C$, $i = 1,2,...,m$. The gravity center G_c of C is the representative of the m data objects, i.e., $G_c = (y_1,...,y_k) = (\frac{1}{m}\sum_{i=1}^{m} x_{i1},...,\frac{1}{m}\sum_{i=1}^{m} x_{ik})$.

Definition 5. Let C be a grid with gravity center G_c. For $\forall q \notin C$, the influence and the mean influence of C on to q is defined respectively by $f^C(q) = \sum_{i=1}^{m} f^{p_i}(q)$, $\tilde{f}^C(q) = m \cdot f^{G_c}(q)$.

Theorem 1. Given the threshold ξ as in Definition 3 and the grid C of width $\sigma > 0$, $\exists \lambda > 0$ such that if $m \geq \lambda$, then all the m data objects in C must belong to same cluster.

Proof. Since the length of the longest diagonal of the grid C is $\sqrt{k}\sigma$ in a k-dimensional data space, thus, $\forall p,q \in C$, $d(p,q) \leq \sqrt{k}\sigma$. If we choose $\lambda = \xi \cdot f_B^{-1}(\sqrt{k}\sigma)$, then for $\forall p \in C$ with $m \geq \lambda$, we have

$$f_B^D(p) = \sum_{q \in D} f_B^q(p,q) \geq \sum_{q \in C} f_{\mathbf{B}}^q(p) \quad \geq \quad m \cdot f_B(\sqrt{k}\sigma) \geq \lambda \cdot f_B(\sqrt{k}\sigma) \quad = \quad \xi .$$

Let p^* be the density attractor of p. Since $f^D(p^*)$ is the local maximum of the density function, we have $f^D(p^*) \geq f^D(p) \geq \xi$. Thus p is a member point of the cluster attracted by p^*. If there exist $p' \in C$, $p' \neq p$ and p' is attracted by attractor $p_1^* \neq p^*$, from Definition 3, we know p', p must also belong to the same multi-attractor cluster because the overall density of C is large than ξ. □

© 中国科学院软件研究所 http://www.jos.org.cn

Corollary. For the Gaussian influence function, $\lambda = \xi \cdot e^{k/2}$; For the Square wave influence function, $\lambda = \xi$.

From Theorem 1 and its corollary, it is clear that for a chosen influence function, all the points in a "dense" grid can be taken for sure as members of a cluster. Thus, these points can be directly assigned to a cluster and laid aside in the data scanning phase. However, if these points are freed and their influences on the nearby points are substituted by the gravity center, how much error will be introduced? Theorems 2 and 3 deal with this approximation.

Theorem 2. Let C be a grid uniformly occupied with m data objects. For each point $q \notin C$, $\exists y_0 \in C$ such that $f_B^C(q) = mf_B^{y_0}(q)$.

Proof. We denote by V the volume of grid C and ∇V an arbitrary micro-cube of C with volume ∇V. Since the data objects are uniformly distributed in C, the number of points within ∇V is $m\nabla V / V$. If we assume $\nabla V \to 0$, then the overall influence of the points in ∇V is approximated by:

$$f_B^{\nabla V}(q) = \sum\nolimits_{y \in \nabla V} f_B(q,y) \approx mf_B(q,y)\nabla V / V,$$

where q is an arbitrary data object outside grid C. Thus

$$f^C(q) = \sum_{\nabla V} \sum_{y \in \nabla V} f_B(q,y) = \sum_{\nabla V} mf_B(q,y)\nabla V / V = \frac{m}{V}\int_V f_B(q,y)\mathrm{d}y \text{ as } \nabla V \to 0.$$

From Cauchy's integral theorem, $\exists y_0 \in C$ such that $f^C(q) = \frac{m}{V} f_B(q,y_0) \cdot V = mf_B^{y_0}(q)$. □

For the density defined using the Gaussian influence function, we further have the follow theorem.

Theorem 3. $\forall q \notin C$, let $f_{Gauss}^C(q)$ and $\widetilde{f}_{Gauss}^C(q)$ be defined as in Definition 5 for the Gaussian density function respectively, then the relative error bound for the substitution of $f_{Gauss}^C(q)$ by $\widetilde{f}_{Gauss}^C(q)$ is given by: $R_B = 1 - e^{-\frac{\delta^2 \pm 2\delta d(G_c,q)}{2\sigma^2}}$, where δ is the distance from the Cauchy's median point y_0 to the gravity center G_c.

Proof. From Theorem 2, $\exists y_0 \in C$ such that $f_{Gauss}^C(q) = mf_{Gauss}^{y_0}(q)$. Therefore, the relative error bound

$$R_B = \left| \frac{f_{Gauss}^{y_0}(q) - mf_{Gauss}^{G_c}(q)}{mf_{Gauss}^{G_c}(q)} \right| = \left| \frac{me^{-\frac{d(y_0,q)^2}{2\sigma^2}} - me^{-\frac{d(G_c,q)^2}{2\sigma^2}}}{me^{-\frac{d(G_c,q)^2}{2\sigma^2}}} \right| = \left| \frac{e^{-\frac{(d(G_c,q)\pm\delta)^2}{2\sigma^2}} - e^{-\frac{d(G_c,q)^2}{2\sigma^2}}}{e^{-\frac{d(G_c,q)^2}{2\sigma^2}}} \right| = 1 - e^{-\frac{\delta^2 \pm 2\delta d(G_c,q)}{2\sigma^2}}$$ □

For the error bound deduced above, we further have $Lim_{\delta \to 0} R_B = 0$. This fact is especially revelatory for us to make sure the puniness of the relative error when applying approximation to a point close to the grid. For a point far away from the grid, the absolute error introduced is also negligible because of the negative exponential property of the influence function.

3.3 Employs mean approximation to boost the algorithms

Based on the results of section 3.2, we forward the approximation approach to the algorithms that fall in the scope of this article. These revisions do not inflict the principal mechanisms of the algorithms.

1) Data freeing while scanning: In the data scanning stage, free all the data objects dynamically of those girds of which the point counts exceed a predefined value λ. Instead, the gravity centers of the data objects in these grids have to be updated whenever new data objects fall into them.

2) Approximated density computation: Two strategies are available to simplify the density and gradient computation procedure in the clustering stage. The first one is the complete gravity-center-to- point approximation. That is, for each point $p \in D$, the density value at p is approximated by $\widetilde{f}^D(p) = \sum_C m_c \cdot f^{G_c}(p)$, where the sum is

© 中国科学院软件研究所 http://www.jos.org.cn

done on all the populated grids. This approach may result in less accurate clusters. However, the complexity of the density computation is reduced to $O(kN)$. Another approach is to apply gravity-center-to-point strategy only to the dense grids (with objects more than λ) from which the data objects are freed already. i.e.,

$$\widetilde{f}^{D}(p)=\sum_{m_c\geq\lambda}Pts_c\cdot f^{G_c}(p)+\sum_{m_c<\lambda}\sum_{q\in C}f^{q}(p).$$

For a skewed dataset with many "dense" grids, this strategy also relieves much of the burden from the computation.

4 Experimental Results

To test the efficiency and effectiveness of the mean approximation approach, we run sets of experiments on both real world and synthetic datasets. All the experiments were conducted on a HP 3000 933 MHz with 256M memory running Windows 2000. The times are well-clock time including CPU and I/O times.

The first experiment was done with a 2-D dataset containing 400 objects on the square $[0,20]\times[0,20]$ jointly generated by normal, Gaussian and random data generators (See Fig.1(a)). We partitioned the square with width $\sigma=0.5$ on both dimensions. As a result, there are 11 among the 1600 grids having more than 5 points occupied and the total points in them is 70. Figure 1(b) shows the Gaussian density function of the origin dataset; Fig.1(c) shows the complete gravity-center-to-point density function. Surprisingly, the two figures show almost no difference between each other. This experiment proves the applicability of our mean approximation approach.

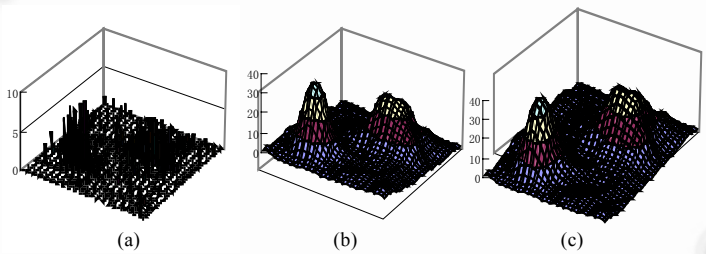

Fig.1 Mean Approximation on a 2-D dataset. (a) Distribution of the objects; (b) Density function of the dataset; (c) Density function with mean approximation on all non-empty grids

Secondly, we processed error evaluations with a single grid (dimension k=2, 10, 50 respectively) of edge- width $\sigma=0.5$ occupied by varied points (m) and a point of different point-to-gravity-center distance $d(q,G_c)$ outside the grid. We computed both exact and gravity-center-to-point influence (i.e., $f^{C}(q)$ and $\widetilde{f}^{C}(q)$) of the grid to the object. The absolute error ($a.e$) and relative error ($r.e$) of each run is listed in table 1. From the table, we see that for a data object very close to the grid, the relative error is less then 1% except the case k=2, m=10. For a point far from the grid, the absolute error is very little although the relative error may be big. The result of this experiment highly favors the mean approximation approach.

The third experiment compares efficiency of mean approximation over the original algorithm of DENCLUE. The data used was the Forest Cover Type Dataset from the UCI KDD achieve, which consists of 581 012 rows of records. Our reconstructed dataset includes the first three numerical values (elevation, aspect and slope) along with the attribute of cover type. The normalized attribute space was partitioned with width $\sigma=0.02$ along each numerical dimension. Figure 2 indicates the percentage of dense grid and percentage of data object occupied in these grids under varied density thresholds. Figure 3 shows the time used by computing local density function of the

© 中国科学院软件研究所 http://www.jos.org.cn

dataset. The top curve shows the time used without any approximation on all objects with locality $\sigma = 0.02$ (i.e., $d(p,q) < 0.02$). The two bottom curves show the time used by mean approximated computation with locality $\sigma = 0.1$. Although the locality of mean approximated computation is 5 times larger than the non-approximated run, the partial approximated runs are still 10 to 50 times faster than the non-approximated runs depending on the given density threshold. Again, the complete point-to-gravity-center runs are nearly 3 times faster then the partial runs with same density threshold.

Table 1 Absolute and relative error under different grid density and gravity-center-to-point distance

k	m	$d(q,p_c)$	$\tilde{f}^C(q)$	$f^C(q)$	*a.e.*	*r.e.*(%)
2	10	0.2647	8.692	8.566	0.126	1.4
		2.279	≈0	≈0	–	–
	500	0.259	436.2	433.9	2.277	0.52
		2.259	0.028	0.035	–0.00	–22
10	10	0.275	9.702	9.674	0.028	0.2
		3.568	2.798	2.801	–0.00	–0.09
	500	0.259	486.6	485.0	1.6	0.3
		3.969	0.916	0.941	–0.02	–2.7
50	10	0.402	9.871	9.866	0.005	0.05
		3.745	3.255	3.258	0.003	0.1
	100	0.268	99.42	99.27	0.156	0.15
		7.455	1.171	1.174	0.003	0.25

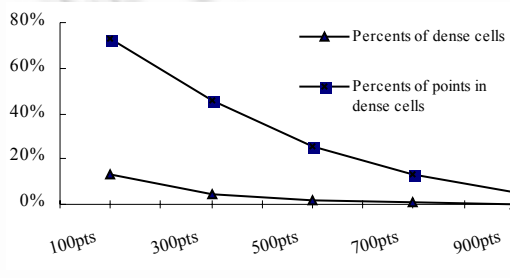

Fig.2 Percentages of dense grids and their data objects under varied density thresholds

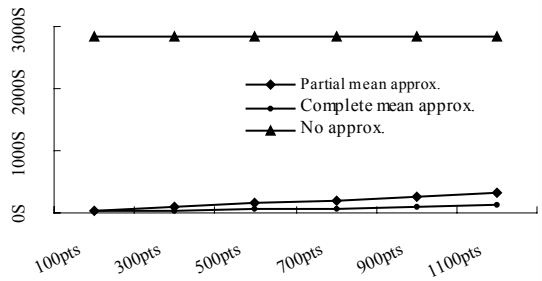

Fig.3 Time used for none approximation, partial and complete approximation runs

5 Conclusions

Recently, the identification of clustering as a central task in data mining has attracted researchers to investigate the scaling of clustering methods to large datasets. The key concerns of scaling a clustering algorithm up to large dataset are memory usage and time efficiency. This paper focuses on the effects of mean approximation to a series of clustering algorithms that are based on both density function computation and grid-partition mechanism. We analyze the error introduced by mean approximation and give a proof for its error bound mathematically. Furthermore, our extensive experiments demonstrate high capability, minor loss of accuracy of this approach. Mean approximation can be implemented partially or completely into the algorithms to gain significant improvement both in memory usage and time efficiency. Due to the promising performance of mean approximation, we believe that our approach is quite reasonable and applicable to scaling a spectrum of clustering algorithms up to much larger datasets.

References:

[1] Han J, Kamber M. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2000. 335~398.

© 中国科学院软件研究所 http://www.jos.org.cn

[2] Berchtold S, Keim D, Kriegel HP. The X-tree: An index structure for high-dimensional data. In: Proceedings of the International Conference on Very Large Databases. Bombay, India, 1996. 28~39.
[3] Hinneburg A, Keim DA. Optimal gird-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In: Proceedings of the 25th International Conference on Very Large Databases. Edinburgh, Scotland, 1999. 506~517
[4] Sheikholeslami G, Chatterjee S, Zhang A. Wave-Cluster: A multi-resolution clustering approach for very large spatial databases. In: Proceedings of the 24th International Conference on Very Large Databases. New York, 1998. 428~439.
[5] Aggrawal R, Gehrke J, Gunopulos D, Raghawan P. Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of the ACM SIGMOD International Conference on Management of Data. Seattle, WA, 1998. 94~105.
[6] Wang W, Yang J, Muntz R. STING: A statistical information grid approach to spatial data mining. In: Proceedings of the 23rd International Conference on Very Large Databases. Athens, Greece, 1997. 186~195.
[7] Hinneburg A, Keim DA. An efficient approach to clustering in large multimedia databases with noise. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD'98). New York, 1998. 58~65.
[8] Xing EP, Karp RM. CLIFF: Clustering of high dimensional microarray data via iterative feature filtering using normalized cuts. BIOINFORMATICS, 2001,1(1):1~9.
[9] Hinneburg A, Keim DA, Brandt W. Clustering 3D-structures of small amino acid chains for detecting dependences from their sequential context in proteins. In: Proceedings of the IEEE International Symposium on BioInformatics and Biomedical Engineering. Washington, DC, 2000. 43~49.
[10] Xu X, Ester M, Kriegel H, Sander J. A distribution-based clustering algorithm for mining in large spatial databases. In: Proceedings of the 14th International Conference on Data Engineering, ICDE'98. Orlando, FL, 1998. 324~331.
[11] Silverman B. Density Estimation for Statistics and Data Analysis. Chapman & Hall, 1986. 72~113.

中国人工智能学会 2003 年全国学术大会 (CAAI-10)

征 文 通 知

为了总结 CAAI-9 以来的新进展，交流我国科技-教育-企业工作者在人工智能领域的自主创新成就，探讨人工智能未来的发展，共享转化科技成果以及在推进信息化过程中推进智能化的经验，中国人工智能学会决定于 2003 年 11 月 19 日~21 日在广州市召开第 10 届全国学术大会(CAAI-10)，由广东工业大学承办。欢迎从事人工智能领域研究、教学、应用的科技工作者,大专院校师生、企业家以及一切爱好和有志于人工智能事业的朋友踊跃投稿。

大会将邀请著名科学家做前沿报告，同时将举行“中韩智能系统学术研讨会”。凡被程序委员会录用的论文，将由北京邮电大学出版社正式出版专书《中国人工智能进展:2003》，并将从这些论文中评选授奖论文。

学术大会征文范围包括(但不限于):

理论创新:逻辑学、离散数学、模糊集-粗糙集、认知学、控制论、系统学、信息-知识-智能理论、可拓学、哲学、信息化与智能化。

技术创新:机器学习、智能机器人、专家系统、知识工程与分布智能、智能控制与智能管理、神经网络与计算智能、自然语言理解、机器翻译、机器感知与虚拟现实、生物信息学与人工生命、计算机辅助教育、智能 CAD、智能制造、可拓工程、智能信息网络、智能系统工程、集对分析与联系数。

应用发展:机器人足球、人工智能产品标准与产业发展、人工智能教育、人工智能普及、智能技术在各个领域的应用。

征文截止日期：2003 年 7 月 31 日

详情请访问中国人工智能学会网站：http://caai.org.cn