Experimental Results of Randomized Clustering Algorithm

Mary Inaba*, Hiroshi Imai* and Naoki Katoh†

Abstract

This paper describes computational results for k-clustering algorithm using random sampling technique [2] to show its practical usefulness. By computational experiments, first, we show that small size of samples are actually enough for 2-clustering problem. Then, we apply this algorithm for k-clustering problem in a recursive manner and use the output as the initial solution of the existing local improvement technique, called k-means. We compare the result with variance-based algorithm [1, 4] which is commonly used.

1 Randomized 2-clustering algorithm

Clustering is the grouping of similar objects. A k-clustering of a set is a partition of its elements into k clusters that is chosen to minimize some dissimilarity cost in each cluster. It is very fundamental and used in various fields in computer science such as pattern recognition, learning theory, image processing, computer graphics, etc.

Variance-based clustering problem for a set S of n points x_i is to find a k-clustering of S into S_j $(j = 1, \ldots, k)$ minimizing the clustering cost $\sum_{j=1}^{k} V(S_j)$ where

$$V(S_j) = \sum_{p_i \in S_j} \|x_i - \bar{x}(S_j)\|^2,$$

$\|\cdot\|$ is the L_2 norm, and $\bar{x}(S_j)$ is the centroid of points in S_j, i.e., $\frac{1}{|S_j|}\sum_{p_i \in S_j} x_i$. For this problem, an optimal k-clustering is induced by the Voronoi Diagram generated by k centroids of the clusters. For this problem, we have proposed the following 2-clustering randomized algorithm in [2]. We implement this algorithm for the planar case and add a local improvement step for the inner loop.

1. Sample a subset T of m points from S at random;
2. For every linearly separable 2-clustering (T_1, T_2) of T, execute the following:
   (a) Compute the centroids t_1 and t_2 of T_1 and T_2;
   (b) Find a 2-clustering (S_{t_1}, S_{t_2}) of S divided by the perpendicular bisector of $\overline{t_1 t_2}$;
   (c) Compute the centroids s_1 and s_2 of S_{t_1} and S_{t_2};
   (d) Find a 2-clustering (S_1, S_2) of S divided by the perpendicular bisector of $\overline{s_1 s_2}$;
   (e) Compute the value of $V(S_1) + V(S_2)$;
3. Output the 2-clustering of S with minimum value.

The main idea of this randomized algorithm is to use all pairs of centroids of linearly separable 2-clusterings for the sampled point set T to obtain S_{t_1} and S_{t_2}, then do local improvement in (2.c) and (2.d). Even without (2.c) and (2.d), we have the following [2]:

Theorem 1 *When there is an optimum 2-clustering (S_1^*, S_2^*) of S such that $|S_1^*|$, $|S_2^*| \geq \gamma|S| = n$ for a constant γ $(0 < \gamma \leq 1/2)$, this randomized algorithm, sampling m points, finds a 2-clustering whose clustering cost is within a factor of $1 + O(1/m)$ from the minimum clustering cost with high probability in $O(m^2 n)$ time.* □

Note that it is not the ratio of the size of samples to that of the original set, but just the absolute value of the sizes of samples which affects the expected value.

2 Effect of the size of samples

All algorithms are implemented in C language, and run on Sun SPARCstation 20 using gcc compiler.

First, experiments for 2-clusterings are done to show the quantitative relationship between the sample size and the cost of computed clustering. As the set S of points to be clustered, we consider (S) 2^{10} and (L) 2^{14} points taken from (U) uniform distribution in the unit square square, (N) standard normal distribution truncated to the unit square, (C) uniform distribution in two disks with randomly chosen 2 centers and random radii in the unit square, and (M) 20726 points taken from Tokyo/Kanto district road data rotated by 45 degree. Examined sizes of sample point set are $m = 2^i$, $i = 2, \ldots, 10$, and the error ratio of the clustering cost for 2^i to that for 2^{10} is shown in Table 1. Note that $m = 2^{10}$ provides an optimal clustering for the case S.

Table 1: Error percentage of clustering costs for sampled points of 2^i, $i = 2, 3, 4, 5, 7, 9$ to the case of 2^{10} (the rightmost column shows the percentage of clustering costs between our algorithm and the variance-based algorithm)

$\|T\|$	2^2	2^3	2^4	2^5	2^7	2^8	2^9	2^4/VB
SU	2.6397	0.1198	0.0457	0.0127	0.0015	0.0011	0.0000	100.01
SN	0.9164	0.2375	0.0373	0.0121	0.0033	0.0010	0.0000	99.69
SC	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.00
LU	0.5809	0.3527	0.1442	0.0019	0.0002	0.0000	0.0000	100.14
LN	0.4721	0.0546	0.0099	0.0008	0.0000	0.0000	0.0000	99.57
LC	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	100.00
M	0.0862	0.0527	0.0082	0.0040	0.0000	0.0000	0.0000	97.62

Uniform distribution is one of the most difficult case for clustering, and even in this case, with the use of steps (2.c), (2.d), we may expect that error is less than 0.2 percent if we select 16 sample points for the set T in many cases, independent of the size of given data set S. This is much better than the theorem guarantees, probably caused by the local improvement steps.

3 Recursive method for k-clustering

A k-clustering can be obtained by applying the 2-clustering algorithm described above in a top down recursive manner. We used a heap to obtain the subset with maximum cost to be divided in the next step.

*Department of Information Science, University of Tokyo, Tokyo 113, Japan. E-mail: {mary,imai}@is.s.u-tokyo.ac.jp

†Department of Management Science, Kobe University of Commerce, Kobe 651-21, Japan. E-mail: naoki@kucgw.kobeuc.ac.jp

Computational Geometry'96, Philadelphia PA, USA

© 1996 ACM 0-89791-804-5/96/05..$3.50

For the k-clustering problem, the variance-based algorithm [1, 4], to be denoted VB, is well-known. It also obtains a k-clustering by recursively applying a 2-clustering algorithm which searches the best partition among $2n$ lines perpendicular to one of the two axes by sweeping.

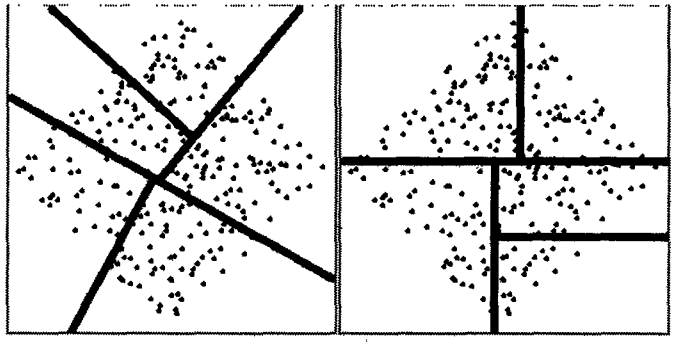

Figure 1: 5-clustering

A typical example to show the difference of these two methods is now given. Fig.1 shows a 5-clustering obtained by our algorithm (left) and variance-based algorithm (right). Data is a set of 256 points in the square rotated by 45 degree. The cost $V(S_1) + V(S_2)$ by our algorithm is about 83.7% of that by variance-based algorithm. For a 4-clustering, a nicer balanced 4-partition is obtained, and the cost of ours is 79.6% of that of variance-based algorithm. This also indicates the limit of top-down recursion by 2-clusterings.

We next compare our algorithm with variance-based algorithm for data used in the previous section. As for 2-clustering, the ratio of clustering cost found by our algorithm using 2^4 samples to that by the variance-based algorithm is shown in the rightmost column in Table 1. The ratio is very close to 1, unlike the above extreme case. For larger k, we show some comparisons in Table 2 below.

4 Relations with k-means algorithm

The k-means algorithm is an iterative improvement algorithm for the k-clustering problem. It starts with an initial k-clustering and then iterate to move each representative point to the centroid of each cluster in the Voronoi partition generated by the current representative points, until a local minimum solution is found. See [3] for example. It can be described by a pseudo-program as follows:

Find an initial k-clustering S_j $(j = 1, \ldots, k)$ for n points x_i $(i = 1, \ldots, n)$ by some method;
repeat
  Compute the centroid $\bar{x}(S_j)$ of each cluster;
  Update the k-clustering to the Voronoi partition by the Euclidean Voronoi diagram of $\bar{x}(S_j)$;
until a local minimum is found.

For this kind of algorithm, the initial solution plays an important role. If we select inappropriate initial points, the number of iterations becomes large and computed clustering bad. Hence, the variance based algorithm [4] is commonly used to obtain an initial solution of k-means problem.

We compare our algorithm to variance-based algorithm for providing good initial solutions for the k-means. Since ours uses random sampling, we perform 10 trials and take the average (av.), minimum (min.), maximum (max.), and standard deviation (dev.) of initial cost, that is, output of our algorithm, local optimum cost (final), that is, output of k-means algorithm, and number of iterations (#iter). Inputs are (L) 2^{14} points with (U,C) types and (M) 20726 points of road map as in the previous section where in (C) 12 disks are generated. The size of samples is 16, and we examined $k = 12, 25, 50$ and 100. We here show part of the experimental results. All costs are normalized by the cost obtained by the variance-based algorithm.

Table 2: Computational results for k-means algorithm

		VB	av.	min.	max.	dev.
$k = 12$	init	252	98.51%	96.30%	100.70%	1.34%
uniform	final	232	99.05%	97.45%	100.20%	0.85%
(LU)	#iter.	10	18.2	12	31	6.09
$k = 12$	init	63.6	96.67%	96.50%	96.77%	0.08%
cluster	final	60.6	99.94%	99.90%	99.95%	0.01%
(LC)	#iter.	13	8.5	8	9	0.5
$k = 12$	init	75.7	93.84%	92.69%	95.59%	0.86%
map	final	74.0	90.93%	89.87%	93.39%	1.46%
(M)	#iter.	7	9.3	8	13	1.41
$k = 100$	init	29.4	98.05%	96.66%	98.92%	0.70%
uniform	final	27.0	98.86%	97.76%	100.08%	0.67%
(LU)	#iter.	18	25.0	16	32	5.56
$k = 100$	init	19.4	97.57%	96.52%	99.78%	0.99%
cluster	final	18.0	98.78%	97.92%	99.63%	0.56%
(LC)	#iter.	17	15.4	8	25	4.47
$k = 100$	init	7.42	97.56%	95.77%	99.15%	0.96%
map	final	6.97	98.62%	97.45%	99.70%	0.72%
(M)	#iter.	7	7.8	5	19	3.99

5 Evaluation of the experimental results

- The small size of samples suffices in the randomized algorithm.
- For the 2-clustering, the randomized algorithm performs better on the average and more robust than variance-based algorithm when optimum solution is not perpendicular to one of the axes.
- On the average, initial k-clusterings produced by our algorithm are better than those by VB, but, after applying the k-mean algorithm, this superiority becomes less except the case of $k = 12$ and map data.

The randomized algorithm is also observed to be more powerful for smaller k a little bit, and more powerful modification for larger k may be further considered. For example, an algorithm finding k candidate representative points directly from random samples, without using the top-down binary partition technique, would deserve consideration.

Acknowledgment This work was supported in part by the Grant-in-Aid of the Ministry of Education, Science and Culture of Japan.

References

[1] P. Heckbert: *Color Image Quantization for Frame Buffer Display.* Bachelor's Thesis, Department of Mathematics, Massachusetts Institute of Technology, 1980.

[2] M. Inaba, H. Imai, and N. Katoh: Applications of Weighted Voronoi Diagrams and Randomization to variance-Based k-Clustering. *Proc. 10th ACM Symp. on Computational Geometry*, 1994,332-339.

[3] S. Z. Selim, M. A. Ismail: K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1984,81–87.

[4] S. J. Wan, S. K. M. Wong and P. Prusinkiewicz: An algorithm for multidimensional data clustering. *ACM Trans. on Mathematical Software*, Vol.14, No.2 (1988), 153–162.