Dependence of Clustering Algorithm Performance on Clustered-ness of Data

Bin Zhang
Data Mining Solutions Department
Hewlett-Packard Research Laboratories
bzhang@hpl.hp.com

Abstract Intuitively, clustering algorithms should work better on the datasets that have well separated clusters. But we found the contrary for the center-based clustering algorithms, including K-Means, K-Harmonic Means and EM. We generated *1200* synthetic datasets with varying ratio of inter-cluster variance over within-cluster variance, which we call the clustered-ness of the dataset. We run K-Means, K-Harmonic Means and EM on these datasets and found that the ratio of the performance over the global optimum grows with increasing clustered-ness. Dependence of clustering algorithm performance on other parameters -- quality of initialization and dimensionality of data -- are also demonstrated.

Keywords – Clustering, K-Means, K-Harmonic Means, EM, Data Mining

1.0 Introduction

Clustering is one of the principle workhorse techniques in the field of data mining [FPU96], statistical data analysis [KR90], data compression and vector quantization [GG92], and many others. K-Means (*KM*), first developed more than three decades ago [M67], and the Expectation Maximization (*EM*) with linear mixing of Gaussian density function [DLR77] are two of the most popular clustering algorithms. See [GG92] for more complete references on K-Means and [MK97][RW84] on EM. K-Harmonic Means is another center-based clustering algorithm, developed by this author [ZHD00][Z00]. The details of each algorithm is reviewed in Section 2.

With guarantee of convergence to only a local optimum, the quality of the converged results, measured by the performance function of the algorithm, could be far from the global optimum. Understanding when these algorithms converge to a good local optimum and when they do not is very important to the practitioners. Such understanding may also provide hints on how the algorithms might be improved.

Let $X = \{x_i \mid i = 1,..., N\}$ be a dataset with K clusters[1] and $M = \{m_k \mid k = 1,..., K\}$ be a set of K centers to mark the location of the clusters. All three algorithms minimize a function of the following form over M (details in Section 2)

$$Perf(X, M) = \sum_{x \in X} d(x, M), \tag{1}$$

where $d(x,M)$ measures the "distance" from a data point to the set of centers. If the global optimum is reached, the centers will be positioned in such a way that each data point is close to one of the centers. However, these algorithms converge only to local optima of their performance functions. The "defects" seen in the local optima are that some clusters trap more centers than their fair share and other clusters attract less or none of their share[2]. When this happens, the value of the performance function will be higher than the global optimum. Intuitively, the farther separated the clusters are the more the performance

[1] See [TWH00] for how to detect the number of clusters in a dataset.
[2] This picture is most clear when K is set to the correct number of well separated clusters in the dataset.

function increases on each of the defects. The experimental results in Section 3 give a quantitative illustration of this relationship.

1.1 Quality Ratio of A Local Optimum

We measure the quality of a local optimum by the ratio, which we call *quality ratio QR,* of the performance value at the local optimum over the global optimum. Let M_{loc} be a local optimum,

$$QR(M_{loc}) = \frac{Perf(X, M_{loc})}{\min_{M} Perf(X, M)}. \tag{2}$$

Quality ratio has been used widely in the development of approximate algorithms. The most widely known is the two-times-optimum approximate solution to the Traveling Salesman Problem. Such a ratio can also be used in experimental studies when a good estimation of the global optimum is available. This is the case in this paper with the use of synthetic datasets.

We found that the average[3] quality ratio is a function of an important characteristic of the dataset, which we call the clustered-ness of the dataset.

1.2 Clustered-ness of A Dataset

For a partition of the dataset, $S_k \subset X$, $k=1,...,K$, the variance of the dataset decomposes into the sum of the within partition-variance and the inter-partition variance [DH72],

$$\sigma^2(X) = \sum_{k=1}^{K} p_k \sigma^2(S_k) + \sum_{k=1}^{K} p_k (m_k - m)^2 \tag{3}$$

where $p_k = |S_k| / |X|$ and

$$m_k = \sum_{x \in S_k} x \Big/ |S_k| \quad and \quad m = \sum_{x \in X} x \Big/ |X| \tag{4}$$

the centroids of S_k and X. The first term in (3) is exactly the same as the K-Means performance function. Minimizing the first term will match the partitions with the clusters in the dataset. Then the first term is called the within-*cluster* variance and the second term the inter-cluster variance. Minimizing the within-partition variance is the same as maximizing the ratio of inter-partition variance over the within-partition variance because the total variance, the summation in (3), remains constant. We call the ratio the *K*-clustered-ness or simply clustered-ness of the dataset when there is no confusion about *K*,

$$Clusteredness \quad C(X) = \max_{M} \frac{\sum_{k=1}^{K} p_k (m_k - m)^2}{\sum_{k=1}^{K} p_k \sigma^2(S_k)}. \tag{5}$$

The larger the *clustered-ness* the better separated are the clusters. It is natural to expect clustering algorithms to work better on the datasets that have well separated clusters. But we found the contrary that the quality ratio, *QR*, grows with increasing clustered-ness of the dataset in our experiments with all three algorithms -- the better separated the clusters are in a dataset, the worse on average the algorithm will perform, measured by the quality ratio. The experimental results are presented in Section 3.

[3] Average is done over different initializations to the algorithm. It may also be averaged over different datasets with the same (or similar) clustered-ness.

The clustered-ness of a dataset depends on *K*, the number of clusters we look for. An example is given in Figure 1. The dataset, *S,* has 8 one-dimensional data points. The distance between any close pair of points is *2* and $d_1 > d_2 > 2$.

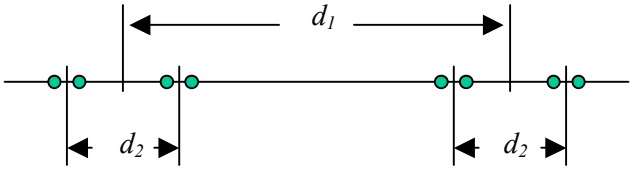

Figure 1. An example: The clustered-ness of a dataset depends on *K*.

For *K=2*, the four points on the left is one cluster and the four on the right is another. For *K=4*, every close pair of points is a cluster. The clustered-ness' are

$$C_2 \equiv Clusteredness(S, K=2) = \frac{d_1^2}{d_2^2 + 4}, \qquad C_4 \equiv Clusteredness(S, K=4) = \frac{d_1^2 + d_2^2}{4}.$$

The following table shows three possible combinations of C_2 / C_4:

d_1=8	d_2=4	C_2=16/5	C_4=20
d_1=large	d_2=4	C_2=large	C_4=large
d_1=2d_2	d_2=large	C_2=small	C_4=large

The fourth combination, C_2=large/C_4=small, is not possible because $C_2 < C_4$ in this example. In general as $K \to \infty$, the inter-cluster variance approaches the variance of the dataset and the within-cluster variance goes to zero, therefore the clustered-ness of a dataset goes to infinity.

The rest of the paper is organized as follows: Section 2 introduces three clustering algorithms, K-Means, (generalized) K-Harmonic Means, and EM. Section 3 presents the results of the experiments. Section 4 concludes the paper.

2.0 Center-Based Clustering Algorithms

Among many clustering algorithms, the center-based clustering algorithms stand out on two most important aspects – a clearly defined objective function that the algorithm minimizes and a low runtime cost. The time complexity per iteration for all three algorithms is *linear* in the size of the dataset, *N*, the number of clusters, *K*, and the dimensionality of data, *D*. The number of iterations it takes for the algorithms to converge is very insensitive to *N*, the size of the dataset.

2.1 K-Means (or LBG) Algorithm

K-Means' performance function is the within-partition variance,

$$Perf_{KM}(X, M) = \sum_{k=1}^{K} \sum_{x \in S_k} \| x - m_k \|^2, \tag{6}$$

where $S_k \subset X$ is the subset of *x's* that are closer to m_k than all other centers (the Voronoi partition). (6) can also be written as

$$Perf_{KM}(X, M) = \sum_{i=1}^{N} MIN\{\| x_i - m_l \|^2 \mid l = 1, ..., K\}. \tag{7}$$

K-Means algorithm calculates its centers' locations in the following iterative steps:

Step 0: Initialize the centers in M
Step 1: Partition the dataset X into K partitions $S_k=\{x|\ argmin_k||x-m_k||=k\}$ (Voronoi partition)
Step 2: Calculate new m_k as the centroid of S_k

$$m_k = \frac{\sum_{x \in S_k} x}{|S_k|}.$$

Repeat Step 1 and Step 2 until there is no more change to the partitions. The local optimal K-Means converges depends on the initialization of the centers. Randomly sampling K points from the dataset as the initialization of the K centers is commonly used.

2.2 K-Median – A Variant of K-Means

Comparing the K-Median algorithm with K-Means, the calculation of the centroids in the Step 2 above is replaced by a search algorithm, searching for the location of m_k (within the data set) that minimizes the total distances from it to all the data points in the partition. The advantage is the removal of the dependency on the geometry of the Euclidian space, which makes K-median meaningful under a metric space. The disadvantage is the $O(N^2)$ time-cost of the search comparing with the linear cost of calculating the centroid. This makes K-Median about 50 times more expensive in time than K-Means for the datasets of size 2500 used in all experiments in this paper. Since we need to run each algorithm on hundreds of datasets, K-Median's high cost prevents us from considering it further.

2.3 K-Harmonic Means

The performance function of KHM_p is defined by:

$$Perf_{KHMp}(X,M) = \sum_{i=1}^{N} HA\{||x_i - m_l||^p \mid l = 1,...,K\} = \sum_{i=1}^{N} \frac{K}{\sum_{l=1}^{K} \frac{1}{||x_i - m_l||^p}}. \quad (9)$$

where $p > 2$. The quantity under the outer summation in (9) is the harmonic average of the distances from data point x to all the centers. Harmonic average of K numbers is sandwiched by the *MIN(),*

$$MIN(||x_i - m_l||^p \mid l = 1,...,K\} \le \frac{K}{\sum_{l=1}^{K} \frac{1}{||x_i - m_l||^p}} \le K * MIN(||x_i - m_l||^p \mid l = 1,...,K\},$$

which makes it serving a similar purpose the K-Means' performance function (7) but leads to an algorithm much less sensitive to initialization (see Figure 3 to Figure 5 in Section 3, or [Z01]).

Taking partial derivatives of the KHM_p's performance function (9) with respect to the center positions m_k, $k=1,...,K$, and set them to zero, we have

$$\frac{\partial Perf_{KHM}(X,M)}{\partial m_k} = -K \sum_{i=1}^{N} \frac{p(x_i - m_k)}{d_{i,k}^{p+2} (\sum_{l=1}^{K} \frac{1}{d_{i,l}^{\ p}})^2} = \vec{0} \quad (10)$$

where $d_{i,l} = ||x_i - m_l||$. "Solving" for the centers from (10), we get the KHM_p iterative algorithm:

$$m_k = \sum_{i=1}^{N} \frac{1}{d_{i,k}^{p+2} (\sum_{l=1}^{K} \frac{1}{d_{i,l}^{p}})^2} x_i \Bigg/ \sum_{i=1}^{N} \frac{1}{d_{i,k}^{p+2} (\sum_{l=1}^{K} \frac{1}{d_{i,l}^{p}})^2}. \quad (11)$$

For more details, see [Z00].

2.4 Expectation-Maximization (EM)

We limit ourselves to the EM algorithm with linear mixing of K identical spherical bell-shape (Gaussian distribution) functions. The performance function of EM is the log-likelihood,

$$Perf_{EM}(X, M) = -\log\left\{\prod_{i=1}^{N}[\sum_{l=1}^{K} p_l * \frac{1}{\left(\sqrt{\pi}\right)^D} EXP(-\| x_i - m_l \|^2)]\right\}. \quad (12)$$

EM algorithm is a recursive algorithm with the following two steps:

E-Step:
$$p(m_l \mid x_i) = \frac{p(x_i \mid m_l) * p(m_l)}{\sum_{i=1}^{N} p(x_i \mid m_l) * p(m_l)}, \quad (13)$$

where $p(x|m)$ is the prior probability with Gaussian distribution, $p(m_l)$ is the mixing probability.

M-Step:
$$m_l = \frac{\sum_{i=1}^{N} p(m_l \mid x_i) * x_i}{\sum_{i=1}^{N} p(m_l \mid x_i)}, \quad and \quad p(m_l) = \frac{1}{N} \sum_{i=1}^{N} p(m_l \mid x_i), \quad (14) \text{and } (15)$$

where N is the size of the whole data set. For more details, see [MK97] and the references there.

3.0 Experiments

We run all three algorithms -- K-Means, K-Harmonic Means and EM -- on 1200 datasets, with three different initializations for each dataset -- bad, better and good. A total of 10800 individual experiments were conducted.

3.1 Datasets

We set the size of the datasets *N=2500*, which is large enough to create non-trivial clusters and small enough to be affordable to run many experiments. We set the number of clusters *K=50* and the dimensionality *D*= 2, 5, and 8. For each dimensionality, we randomly generated 400 datasets, *Dataset(i), i=1, …, 400*, with their clustered-ness uniformly distributed in *[0,60]*. This range of clustered-ness covers from datasets with all clusters overlapping almost completely to datasets with very well separated clusters. The following Matlab function is used to generate all the datasets.

```
function [dataset,centers] = ClusGen(K, N, D, r)
    % K = # clusters, N = #data points, D = dimensionality,
    % r = a parameter to control the within cluster variance/inter-cluster variance.
    %Step 1: Generate cluster centers.
    centers = r * rand(K,D);
    % K center locations are generated and scaled up by the factor r.
    % Step 2: Generate the random sizes of the K clusters.
```

```
s = 2*rand(K,1)+1; s = round(N*s/sum(s));  N1 = sum(s);  diff = abs(N-N1);
s(1:diff) = s(1:diff) + sign(N-N1);  % adjust the size so that they add up to N.
%Step 3: Generate clusters one-by-one.
for k=1:K
    cluster = randn(s(k),D); % normal distribution.
    % move the clusters to the kth center location.
    mean = sum(cluster)/s(k);        Sk = repmat(centers(k,:)-mean,s(k),1)+cluster;
    % merge the cluster into the dataset.
    dataset = [dataset' Sk']';
end;  % of for loop.
%End of the cluster dataset generator.
```

3.2 Initialization

We use three types of random initializations with each dataset, *Init(i,j), j=1, 2, 3*, for *i=1,…,400*. Separating initializations into three bins help to reduce the variance on the quality ratio we collect in each bin, so that the dependence on the clustered-ness is less interfered by the dependence on initialization, which is very strong for K-Means and EM. Three types of initializations are

Type-1: Bad -- all 50 centers are initialized to be within a small region relative to the data.
Type-2: Better -- all centers are randomly initialized, covering the region of data. The centers have a bigger spread than the data itself.
Type-3: Good -- the centers are initialized to 50 randomly chosen data points from the dataset.

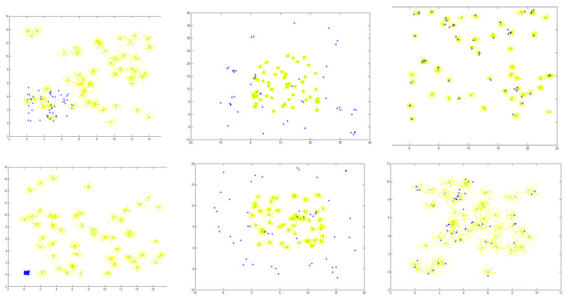

Figure 2. Two samples of each type of initializations are shown here. Ordered from left to right: Type-1, Type-2, and Type-3. The light yellow backgrounds are the data points and the dark blue dots are the initial center locations.

3.3 A Common Performance Measure

Each algorithm optimizes its own performance function. To compare different algorithms under the same measure, we use the *square-root* of K-Means' performance function to measure the quality of the clusters (centers) derived by all algorithms. We choose K-Means' performance function because it is more popular, more intuitive and simpler than others. Taking square root is to remove the quadratic behavior of the function and restore the linear behavior. To calculate the quality ratio, the global optimal performance, which is measured by the same function, is derived by running K-Means starting from the location of the "true" centers of the clusters returned by the ClusGen() function[4]. Except for the datasets that has very small clustered-ness (below 10), the global optimum we calculated are very good approximations.

[4] which is, of course, only an approximation of the true global optimum.

To plot the quality ratio as a function of the clustered-ness, we equally partition the clustered-ness interval, *[0,60]*, into *12* bins. The datasets and the experiments on them are also partitioned into *12* groups accordingly, G_l, *l=1,…,12*. Each group has about *400/12* datasets because the clustered-ness was uniformly generated.

The average and coefficient of standard deviation of the quality ratios in the *l*-th group under the *j*-th type initialization is calculated as

$$Perf(X,M) = \sqrt{Perf_{KM}(X,M)}, \quad QR_{i,j} = \frac{Perf(X,M)}{\min_{M} Perf(X,M)},$$
$$avg_{l,j} = \frac{\sum_{i \in G_l} QR_{i,j}}{|G_l|}, \quad \sigma_{l,j}^2 = \frac{\sum_{i \in G_l} (QR_{i,j} - avg_{l,j})^2}{|G_l|}, \quad \theta_{l,j} = \frac{\sigma_{l,j}}{avg_{l,j}}. \tag{17}$$

3.4 Results

Each algorithm, K-Means, K-Harmonic Means (*p=3*) and EM, was run up to 100 iterations on all pairs *(Dataset(i), Init(i,j))*, which is sufficient for the centers to stabilize. The average quality ratio is plotted against the average clustered-ness of the dataset from Figure 3 to Figure 5. Each Figure has three plots, one for each type of initialization. The scales of the vertical axis are different for different plots. Dimensionality of the datasets is 2 in Figure 3, 5 in Figure 4 and 8 in Figure 5.

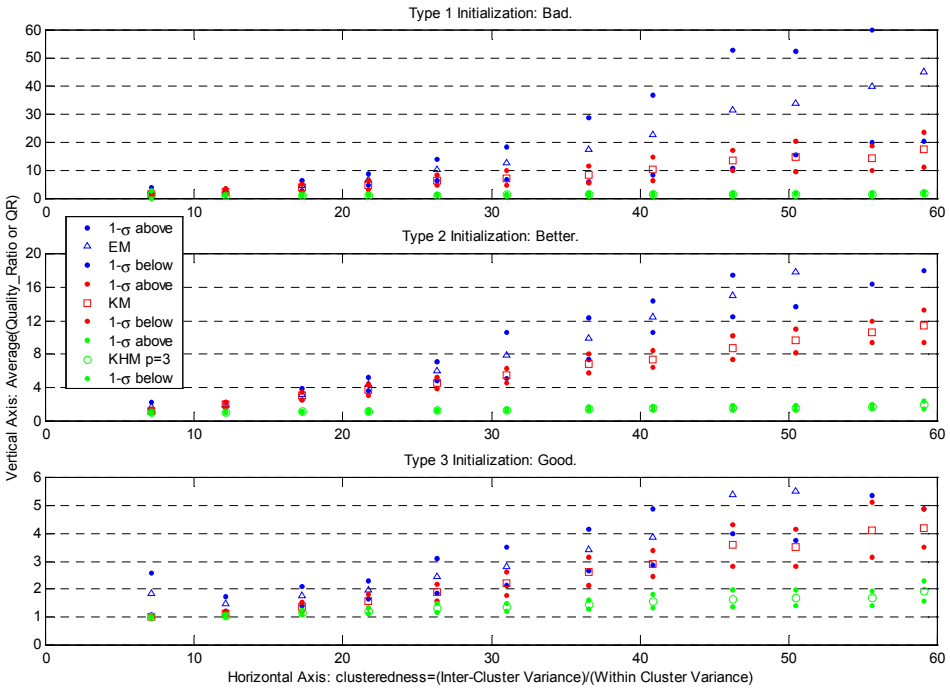

Figure 3. The dependence of clustering algorithm performance on the clustered-ness of the dataset. One-sigma (on each side) confidence intervals are printed as dots of the same color. The dimensionality of the datasets equals to 2. For KHM, *p=3*.

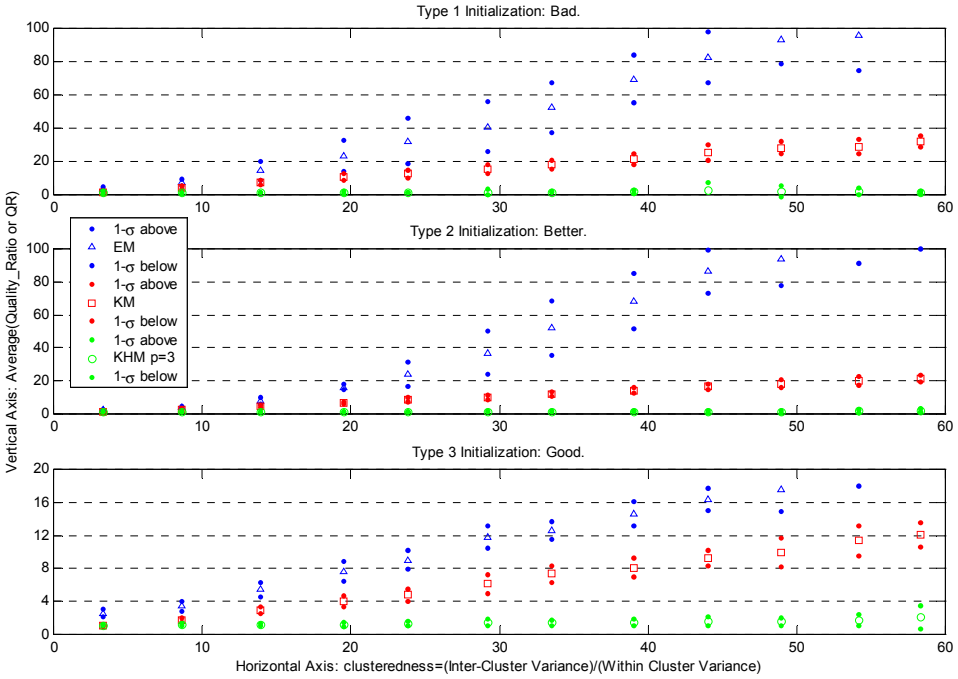

Figure 4. The dependence of clustering algorithm performance on the clustered-ness of the dataset. The dimensionality of the datasets equals to 5. For KHM, *p=3*.

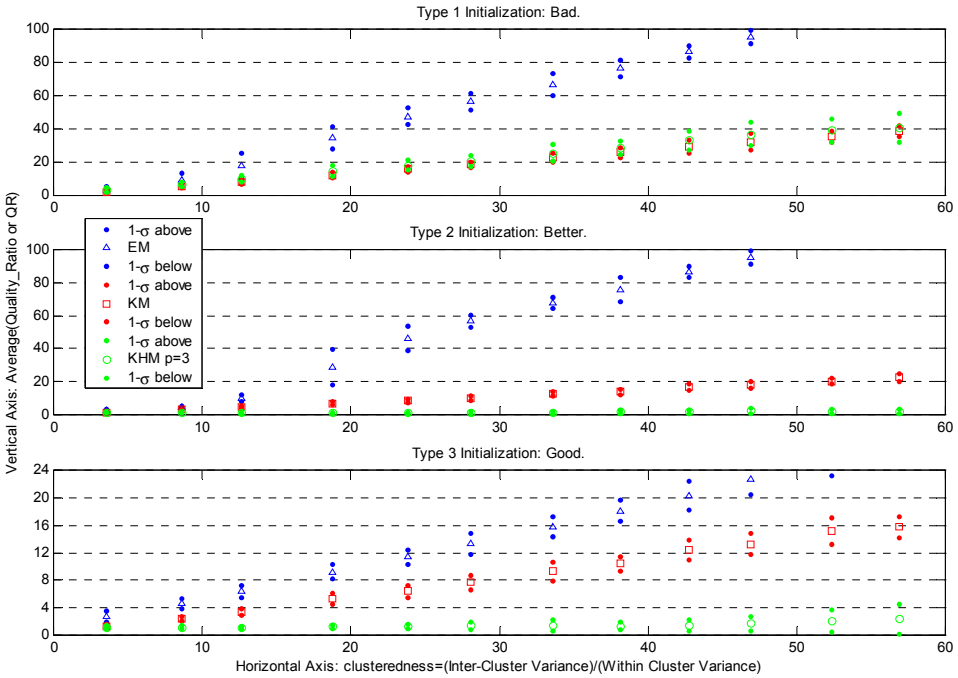

Figure 5. The dependence of clustering algorithm performance on the clustered-ness of the dataset. The dimensionality of the datasets equals to 8. For KHM, *p=3.5*.[5]

[5] As the dimensionality of data become higher, the parameter *p* in KHM need to be set higher to achieve the best result. It exact relationship between the dimensionality and the best *p* is not known. We

3.5 An explanation

The almost linear growth of the quality ratio is partially caused by the way the performance is measured. If we compare the local optimum the algorithm converged to with the global optimum, we see that some centers are "misplaced" in the local optimum. Such misplaced centers are either squeezed into a cluster that already has one or more centers, or more than one clusters share one center (especially for K-Means) and the center is not in any of the clusters. The better separated the clusters are, the more (linearly) each of such defects contributes to the increase of the quality ratio.

The quality ratio grows at different rates for different algorithms. *QR's* growth rate also depends on the quality of initialization and dimensionality of the data. This rate reflects the average number and the severity of the defects.

3.6 On high dimensional data

From our results in the Section 3, we see that the performance of all three algorithms deteriorates as the dimensionality become higher. This does not prevent us from seeing K-Means, KHM and EM working on some high dimensional datasets. Many such cases have been reported in the literature. We have run K-Means and K-Harmonic Means *(p=6)* on the 1998 KDD-CUP data mining contest dataset [Z01].

4.0 Conclusion

We identified a parameter (characteristic) of data that influences of the average performance of the center-based clustering algorithms.

References

[A73] Anderberg, M. R. 1973. Cluster analysis for applications. Academic Press, New York. xiii + 35p.

[B99] Bay, S. D. (1999). The UCI KDD Archive [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science.

[BFR98] Bradley, P., Fayyad, U. M., and Reina, C.A., "Scaling EM Clustering to Large Databases," MS Technical Report, 1998.

[BF98] Bradley, P., Fayyad, U. M., C.A., "Refining Initial Points for KM Clustering", MS Technical Report MSR-TR-98-36, May 1998.

[BFR98a]Bradley, P.,Fayyad, U.M., and Reina, C.A., "Scaling Clustering to Large Databases", KDD98, 1998.

[DH72] Duda, R., Hart, P., "Pattern Classification and Scene Analysis", John Wiley & Sons, 1972.

[DLR77]Dempster, A. P., Laird, N.M., and Rubin, D.B., "Miximum Likelyhood from Incomplete Data via the EM Algorithm", Journal of the Royal Statistical Society, Series B, 39(1):1-38, 1977.

[FPU96]Fayyad, U. M., Piatetsky-Shapiro, G. Smyth, P. and Uthurusamy, R., "Advances in Knowledge Discovery and Data Mining", AAAI Press 1996

[FS96] Freund, Y., & Schapire, R.E., "Experiments with a new boosting algorithm," Proceedings of the 13th International Conference On Machine Learning, pp. 148, 1996.

[FZ00] Forman, G. & Zhang, B., "Linear Speedup for a parallel non-approximating recasting of center-based clustering algorithms, including K-Means, K-Harmonic Means and EM," KDD-2000 Workshop on Distributed and Parallel Knowledge Discovery, Sixth ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August, 2000, Boston, USA.

[GG92] Gersho & Gray, "Vector Quantization and Signal Compression", KAP, 1992

[GMW85] Gill, P.E., Murray, W. and Wright, H.M., "Practical Optimization", Academic Press, 1981.

---

observed that a gradual degradation of KHM performance as p deviates from its optimum. Searching p in the neighborhood of *sqrt(D)* is a rule-of-thumb.

[G85] Gonzales,T.F,"Clustering to Minimize the Maximum Intercluster Distance",Theo.Comp.Sci.38, p293-306, 85.

[KR90] Kaufman, L. and Rousseeuw, P. J., "Finding Groups in Data : An Introduction to Cluster Analysis", John Wiley & Sons, 1990.

[M67] MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. Pp. 281-297 *in*: L. M. Le Cam & J. Neyman [eds.] Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1. University of California Press, Berkeley. xvii + 666 p.

[MA] McKenzie, P. and Alder, M., "Initializing the EM Algorithm for Use in Gaussian Mixture Modeling", The Univ. of Western Australia, Center for Information Processing Systems, Manuscript.

[MK97]McLachlan, G. J. and Krishnan, T., "The EM Algorithm and Extensions.", John Wiley & Sons, Inc., 1997

[PM99] Pelleg, D. and Moore, A, "Accelerating Exact K-Means Algorithms with Geometric Reasoning", KDD-99, Proc. of the Fifth ACM SIGKDD Intern. Conf. On Knowledge Discovery and Data Mining, page 277-281.

[RW84]Rendner, R.A. and Walker, H.F., "Mixture Densities, Maximum Likelihood and The EM Algorithm", SIAM Review, vol. 26 #2, 1984.

[SI84] Selim, S.Z. and Ismail, M.A., "*K*-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality", IEEE Trans. On PAMI-6, #1, 1984.

[TWH00] Tibshirani, R., Walther, G., and Hastie, T., "Estimating the number of clusters in a dataset via the Gap Statistic." Available at http://www-stat.stanford.edu/~tibs/research.html. March, 2000.

[ZHD00] Zhang, B., Hsu, M., Dayal, U., "K-Harmonic Means", International Workshop on Temporal, Spatial and Spatio-Temporal Data Mining, TSDM2000, Lyon, France Sept. 12, 2000.

[ZHF00] Zhang, B., Hsu, M., Forman, G., "Accurate Recasting of Parameter Estimation Algorithms using Sufficient Statistics for Efficient Parallel Speed-up", PKDD 2000, Lyon, France, Lecture Notes in AI #1910, Springer, p243.

[Z01] Zhang, B., "Generalized K-Harmonic Means – Dynamic Weighting of Data in Unsupervised Learning,", the First SIAM International Conference on Data Mining (SDM'2001), April 5-7, 2001, Chicago, USA.

Appendix. The Tables of Results

We provide the tables of results that were used for the plots in this paper. These tables are for more detailed examination of the results by the reviewers, optional for the final publication.

Table 1. Type 1 Initialization. Dimension=2.

	EM		K-Means		K-Harmonic Means	
Avg. C	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.
7.1	2.70	0.4711	1.42	0.2229	1.41	0.8788
12.1	2.64	0.2827	2.59	0.3792	1.06	0.0358
17.3	4.64	0.3753	3.91	0.3377	1.17	0.0909
21.7	6.85	0.3054	4.93	0.3114	1.28	0.1004
26.4	10.24	0.3596	6.45	0.2718	1.36	0.1276
31.0	12.65	0.4593	7.36	0.3661	1.43	0.1560
36.5	17.53	0.6520	8.58	0.3560	1.67	0.1876
40.9	22.63	0.6297	10.59	0.3948	1.58	0.1539
46.2	31.77	0.6597	13.57	0.2774	1.68	0.1764
50.5	33.86	0.5442	14.90	0.3616	1.75	0.1972
55.6	39.88	0.5029	14.48	0.2985	1.76	0.2334
59.1	45.11	0.5467	17.44	0.3473	1.85	0.2244

Table 2. Type 2 Initialization. Dimension=2.

	EM		K-Means		K-Harmonic Means	
Avg. C	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.
7.1	1.72	0.2874	1.26	0.1803	1.01	0.0139
12.1	1.95	0.1622	1.95	0.1527	1.07	0.0401
17.3	3.21	0.2166	3.03	0.1682	1.15	0.0642
21.7	4.46	0.1863	3.77	0.1635	1.21	0.0934
26.4	6.00	0.1869	4.55	0.1492	1.31	0.0711
31.0	7.91	0.3478	5.46	0.1669	1.35	0.1121
36.5	9.90	0.2514	6.91	0.1698	1.52	0.1455
40.9	12.50	0.1521	7.44	0.1407	1.59	0.1401
46.2	14.97	0.1641	8.77	0.1629	1.59	0.1781
50.5	17.79	0.2328	9.60	0.1500	1.66	0.1495
55.6	20.44	0.1998	10.66	0.1196	1.78	0.1299
59.1	23.06	0.2220	11.39	0.1719	1.96	0.2253

Table 3. Type 3 Initialization. Dimension=2.

	EM		K-Means		K-Harmonic Means	
Avg. C	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.
7.1	1.84	0.4072	1.02	0.0249	1.01	0.0101
12.1	1.47	0.1823	1.12	0.0777	1.06	0.0371
17.3	1.75	0.1916	1.36	0.1228	1.15	0.0730
21.7	1.97	0.1630	1.56	0.1556	1.22	0.0863
26.4	2.47	0.2541	1.88	0.1599	1.33	0.1225
31.0	2.80	0.2444	2.21	0.1921	1.36	0.1039
36.5	3.40	0.2207	2.63	0.1916	1.46	0.1149
40.9	3.85	0.2612	2.91	0.1635	1.57	0.1564
46.2	5.41	0.2628	3.58	0.2075	1.66	0.1871
50.5	5.53	0.3212	3.49	0.1920	1.69	0.1666
55.6	6.81	0.2111	4.11	0.2411	1.68	0.1619
59.1	6.63	0.2638	4.20	0.1636	1.93	0.1924

Table 4. Type 1 Initialization. Dimension=5.

	EM		K-Means		K-Harmonic Means	
Avg. C	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.
3.3	3.33	0.3625	1.59	0.4073	1.32	0.2599
8.6	7.59	0.2661	4.24	0.2677	1.24	0.3333
13.9	14.53	0.3906	7.19	0.1802	1.21	0.1221
19.5	23.14	0.4055	10.80	0.1876	1.38	0.4337
23.8	32.13	0.4240	12.45	0.1864	1.29	0.2209
29.2	40.94	0.3704	15.60	0.1686	1.66	0.9640
33.5	52.43	0.2856	18.15	0.1454	1.37	0.2547
39.0	69.59	0.2081	21.21	0.1514	1.69	0.7107
44.0	82.64	0.1844	25.40	0.1751	2.64	1.8518
48.9	93.66	0.1601	28.10	0.1295	2.24	1.4712
54.2	95.74	0.2217	28.99	0.1590	1.99	0.9088
58.3	112.60	0.0797	32.06	0.0972	1.60	0.3862

Table 5. Type 2 Initialization. Dimension=5.

	EM		K-Means		K-Harmonic Means	
Avg. C	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.
3.3	2.43	0.1341	1.30	0.2606	1.13	0.0804
8.6	3.89	0.1823	2.77	0.1993	1.19	0.0805
13.9	8.22	0.2312	4.91	0.1592	1.25	0.1292
19.5	16.37	0.1182	6.80	0.1055	1.26	0.1597
23.8	24.32	0.3043	8.68	0.1378	1.29	0.2109
29.2	37.17	0.3501	10.24	0.1185	1.47	0.2835
33.5	52.29	0.3151	11.94	0.1117	1.47	0.2435
39.0	68.56	0.2470	14.25	0.1074	1.39	0.2711
44.0	86.36	0.1514	16.46	0.0970	1.51	0.3474
48.9	94.14	0.1696	18.41	0.1366	1.56	0.3580
54.2	105.51	0.1325	20.12	0.1228	1.81	0.4519
58.3	112.66	0.1132	21.55	0.0870	2.30	0.5285

Table 6. Type 3 Initialization. Dimension=5.

	EM		K-Means		K-Harmonic Means	
Avg. C	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.
3.3	2.61	0.1671	1.11	0.1333	1.13	0.0770
8.6	3.47	0.1763	1.79	0.1492	1.22	0.0838
13.9	5.45	0.1519	2.94	0.1445	1.19	0.1080
19.5	7.64	0.1564	4.03	0.1670	1.22	0.1984
23.8	9.06	0.1311	4.79	0.1475	1.30	0.2084
29.2	11.77	0.1143	6.14	0.1832	1.49	0.2653
33.5	12.66	0.0835	7.32	0.1359	1.41	0.1939
39.0	14.63	0.1013	8.12	0.1373	1.46	0.2982
44.0	16.39	0.0790	9.23	0.0997	1.63	0.3323
48.9	17.56	0.1485	9.92	0.1786	1.56	0.2711
54.2	20.34	0.1124	11.39	0.1584	1.76	0.3835
58.3	22.29	0.0954	12.06	0.1252	2.12	0.6567

Table 7. Type 1 Initialization. Dimension=8.

	EM		K-Means		K-Harmonic Means	
Avg. C	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.
3.6	3.62	0.4449	2.24	0.4690	3.54	0.4460
8.6	9.39	0.4209	5.51	0.1502	7.98	0.0811
12.6	17.82	0.4390	8.19	0.1917	11.88	0.1266
18.8	34.85	0.1956	12.25	0.1281	17.26	0.1159
23.8	47.61	0.1095	15.86	0.1093	21.82	0.0816
28.1	56.47	0.0876	18.40	0.0927	25.72	0.0805
33.6	66.86	0.1005	22.41	0.1205	31.07	0.0767
38.1	76.44	0.0677	25.91	0.1134	34.45	0.0934
42.7	86.60	0.0422	29.32	0.1284	39.33	0.0812
46.9	95.21	0.0407	32.33	0.1501	43.73	0.0995
52.3	106.32	0.0401	35.56	0.0940	47.91	0.0779
56.9	114.17	0.0332	38.69	0.0782	51.61	0.0782

Table 8. Type 2 Initialization. Dimension=8.

	EM		K-Means		K-Harmonic Means	
Avg. C	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.
3.6	2.44	0.2580	1.51	0.3048	1.53	0.2272
8.6	4.53	0.1815	3.13	0.1205	1.26	0.1346
12.6	10.11	0.2209	4.66	0.1649	1.32	0.2515
18.8	28.90	0.3713	7.05	0.1256	1.37	0.2715
23.8	46.53	0.1587	8.61	0.1198	1.39	0.3302
28.1	56.89	0.0646	10.06	0.1239	1.42	0.4529
33.6	67.65	0.0503	12.77	0.1256	1.59	0.5065
38.1	75.98	0.0979	13.99	0.1226	1.73	0.6002
42.7	86.60	0.0422	16.71	0.1046	2.08	0.5097
46.9	95.21	0.0407	18.02	0.1112	2.44	0.6311
52.3	106.32	0.0401	20.30	0.0877	2.06	0.6302
56.9	114.17	0.0332	22.55	0.1050	1.91	0.6473

Table 9. Type 3 Initialization. Dimension=8.

	EM		K-Means		K-Harmonic Means	
Avg. C	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.	Avg. QR	Coef. Std.
3.6	2.71	0.2857	1.31	0.2358	1.39	0.1106
8.6	4.62	0.1472	2.39	0.1212	1.31	0.2010
12.6	6.38	0.1368	3.43	0.1332	1.27	0.2214
18.8	9.22	0.1117	5.31	0.1660	1.31	0.2459
23.8	11.36	0.0925	6.38	0.1292	1.43	0.3594
28.1	13.30	0.1157	7.68	0.1387	1.36	0.3979
33.6	15.81	0.0914	9.29	0.1472	1.44	0.5697
38.1	18.10	0.0846	10.39	0.1074	1.50	0.4793
42.7	20.25	0.1036	12.43	0.1205	1.55	0.6534
46.9	22.75	0.0971	13.29	0.1143	1.68	0.6273
52.3	25.23	0.0791	15.13	0.1237	1.93	0.6942
56.9	27.08	0.0908	15.76	0.0970	2.48	0.8813