A New Improved K-means Algorithm with Penalized Term

Zejin Ding
Dept. of Computer Science
Georgia State University
Atlanta, GA 30302–3994
zding1@student.gsu.edu

Jian Yu
School of Computer Science
and Technology
Beijing Jiaotong University
Beijing, China 100044
jianyu@center.bjtu.edu.cn

Yan-Qing Zhang
Dept. of Computer Science
Georgia State University
Atlanta, GA 30302–3994
yzhang@cs.gsu.edu

Abstract

K-means Algorithm is a popular method in cluster analysis. After reviewing different K-means algorithms, we propose the new penalized K-means algorithm. Originally inspired by the Maximum Likelihood(ML) method, a prior probability distribution assumed by classic K-means algorithm about the clustering data set was discovered, and then the new objective function for the penalized K-means algorithm was introduced. By minimizing this function with genetic algorithm, results show that this method is better than K-means algorithm in some perspectives.

1. Introduction

Clustering is a task of dividing the objects in a dataset into several subsets, where the objects in each subset are similar to each other, and different from objects in other subsets. Clustering algorithms have been widely used in many fields, such as data mining, pattern recognition, image segmentation, biology, and so on. Overview of clustering algorithms can be found in the literature [2]. Generally, one of the most popular clustering algorithms is the K-means (or C-means) algorithm, which was developed by MacQueen in 1967 [6]. This is because K-means algorithm has many advantages, such as easily realizing, quickly converging, etc. However, K-means algorithm also has its drawbacks: each sample only belongs to one cluster; it is prone to find clusters with spherical shape and sensitive to noisy data and so on. Certainly, many improved algorithms based on K-means have been presented. Usually all these methods are partitional algorithms, so the paper only discuss this family.

For the traditional K-means algorithm, initialization plays a key role in the performance. Different initialization often lead to different results, and thus some improved methods are proposed to avoid such sensitivity. For example, it's possible to run several times of K-means algorithm with different starts and choose the result with the minimum sum-squared-error; Bradley [1] introduced a sub-sampling version of random restart to deal with large data sets.

K-means algorithm with iterative approaches converge easily into local optimal minimum. An alterative method is to use genetic algorithms. Krishna presented the Genetic K-Means algorithms in [4] and showed that GKM produces better result. Commonly, there are two main strategies of involving genetic algorithm. One is to treat every sample in the dataset as the chromosomes, called string representation; the other is to use the k centers as the chromosomes [10]. Recently, Laszlo and Mukherjee[5] even used Hyper-Quadtrees to represent cluster structure in the chromosome strings of genetic algorithm. Obviously, the former method has long chromosomes and thus is inefficient when the samples are numerous, and the latter also suffer similar problems when the data dimension is huge.

Meanwhile, classic K-means algorithm uses Euclidean distance as its distance function. Thus new distances are introduced in several papers, such as Manhattan distance, Minkowski distance, etc [8]. By changing Euclidean distance to Manhattan distance, the k-centroid algorithm appears, which is also popular. In[12], Wu creates a new distance function for better robust ability; and in [11], Su defines a non-metric distance to solve cluster symmetry. In more recent study, Huang add a new variable weight to each attribute of a sample [3], and this method can also be viewed as creating a new distance function.

In this paper, we will present a new K-means algorithm originated from a different perspective. Generally speaking, one prototype represents one cluster in partitional clustering algorithms. If a partitional clustering algorithm has the prior probability distribution about the data set, and the real distribution of the data set do match such assumption, then the algorithm usually reach good clustering result; otherwise, vice versa. More importantly, if the distribution of data set is known at first, an algorithm can be easily obtained. By Maximum Likelihood method, prototypes can be estimated

0-7695-3032-X/07 $25.00 © 2007 IEEE
DOI 10.1109/GrC.2007.39

by making them as the parameters of the distribution. Does K-means algorithm have its own assumed distribution? We find no direct answer in literature. So if such distribution can be deduced appropriately, it will be helpful for studying K-means algorithm. Now, the main problem becomes to how to construct the probability distribution of the K-means algorithm. This paper will discuss it in detail, and propose the Penalized K-means algorithm.

The organization is as follows. In Section 2, K-means algorithm will be reviewed briefly. In Section 3, we will propose a new K-means algorithm in detail. In order to better understand the parameter β, analysis will be developed in Section 4. Also, several experiments will show the performance of the algorithm in Section 5. Finally, Section 6 summarizes the paper and give further discussion.

2. K-means Algorithm

K-means algorithm groups a data set with n samples into k clusters, where k is a given parameter. This algorithm is based on minimizing the sum of the square error among interior clusters and often uses iterative method to reach its minimum. Suppose that $X = \{x_1, x_2, ..., x_n\}$ represents a s-dimensional data set with n samples, and is divided into k clusters $v = \{v_1, v_2, ..., v_k\}$ where each prototype v_i represents a cluster, then the objective function of K-means algorithm can be written as follows:

$$J_{k-means}(X, v) = \sum_{j=1}^{n} \min_{1 \le i \le k} (\parallel x_j - v_i \parallel)^2 \qquad (1)$$

The main iterative steps of K-means can be described as follows:

Step 1: update membership value $u_{ij}^{(l+1)}$ with the formula:

$$u_{ij}^{(l+1)} = \begin{cases} 1 & if \quad i = \{m | \min_{1 \le m \le k} ||x_j - v_m^{(l)}||\} \\ 0 & otherwise \end{cases} \qquad (2)$$

Step 2: update new prototypes $v_i^{(l+1)}$ by:

$$v_i^{(l+1)} = \sum_{j=1}^{n} u_{ij}^{(l+1)} x_j \Big/ \sum_{j=1}^{n} u_{ij}^{(l+1)} \qquad (3)$$

Initialization and termination step are not shown here for limitation. It is well known that K-means algorithm often traps to local minimum and doesn't cluster well on overlapping data sets.

3. The Penalized K-means Algorithm

Up to now, no distribution assumption associated with K-means algorithm has been found in literature. We try to find such distribution by maximum likelihood method. According to ML method, it assumes that samples in the data set are independently and identically drawn from a probability distribution. A sample x_j drawn from such distribution can be represented by the probability $Pr(x_j|v)$ in mathematics language, and then the overall likelihood of the data set is the probability to be drawn from the following probability model:

$$L(X|v) = \prod_{j=1}^{n} Pr(x_j|v), \ln(L(X|v)) = \sum_{j=1}^{n} \ln(Pr(x_j|v)) \qquad (4)$$

Obviously, the log-likelihood $\ln(L(X|v))$ can be considered as an *objective function*, and by maximizing this function it can estimate the prototypes $v_1, v_2, ..., v_k$. Actually, this procedure also lead to the Expectation-Maximization (EM) method, see[7]. However, the probability $Pr(x_j|v)$ is not easy to find out, especially for the data set with no prior knowledge. Sometimes it's assumed that $Pr(x_j|v)$ is a Gaussian probability for easily computing.

Nevertheless, the probability distribution assumed by typical K-means algorithm hasn't been found in literature. So if such distribution can be uncovered or deduced, a new algorithm can be established by substituting this distribution into the ML method. To compare K-means objective function term(1) and log-likelihood function term(4), it's easy to find that if we set $Pr(x_j|v) = \exp(-\beta \min_{1 \le i \le k} \parallel x_j - v_i \parallel^2)$, then these two terms are almost the same except the sign, and minimizing term(1) is equal to maximizing term(4). Since $Pr(x_j|v) = \exp(-\beta \min_{1 \le i \le k} \parallel x_j - v_i \parallel^2)$ is not a p.d.f., it should be regularized.

The regularized probability density function is:

$$Pr(x_j|v) = (H(v))^{-1} e^{-\beta \min_{1 \le i \le k} \|x_j - v_i\|^2} \qquad (5)$$

where $H(v) = \int_{R^s} e^{-\beta \cdot \min_{1 \le i \le k} \|x - v_i\|^2} dx$.

Parameters β is a scale factor. It will influence significantly the shape of the above p.d.f. and be discussed in Section 4.

To substitute term(5) into term(4), we will get:

$$\begin{aligned} \ln(L(X|v)) &= \sum_{j=1}^{n} [-\beta \min_{1 \le i \le k} \parallel x_j - v_i \parallel^2 - \ln(H(v))] \\ &= -\beta J_{k-means}(X, v) - n \ln(H(v)) \end{aligned} \qquad (6)$$

Maximizing the above objective function, it will get the prototypes $v_1, v_2, ..., v_k$. For simplicity, the term(6) can also be expressed as:

$$J_{PKM}(X, v) = \beta J_{k-means}(X, v)/n + \ln(H(v)) \qquad (7)$$

It's clear that minimizing $J_{PKM}(X, v)$ identically equals maximizing $\ln(L(X|v))$. Thus, minimizing term (7) can lead to a new algorithm, called Penalized K-means algorithm(PKM) here. In fact, minimizing term (7) is equal to K-means algorithm under some conditions and we won't prove it here for limitation.

Considering $J_{PKM}(X, v)$, we note that it's hard to compute $H(v)$ for the high-dimensional or unknown domain in the calculus, so it's not easy to get $H(v)$ directly. Nevertheless, it's possible to find ways to estimate its unbiased value.

Lemma 1[9]: In statistics, by applying Monte Carlo method, a calculus $I = \int_a^b g(x)dx$ can be estimated with the following way:

$$I = \int_a^b g(x)dx \approx \widetilde{I} = (b-a) \cdot \frac{1}{N} \sum_{i=1}^{N} g(r_i) \qquad (8)$$

where r_i is a random number which uniformly distributes from (a, b), and N is the total number of r_i. When $N \to \infty$, $\widetilde{I}$ will tend to I with probability 1. $\dashv$

Here, the range of the data set is R^s in $H(v)$, so we define a mapping function for using **Lemma 1**, that is: $x = \tan(\pi \cdot y)$ where y is uniformly distributed in the interval $[-0.5, 0.5]^s$, and the term $H(v)$ can be expressed as:

$$H(v) = \int_{[-0.5,0.5]^s} \frac{e^{-\beta \min\limits_{1\le i\le k} \|\tan(\pi\cdot y)-v_i\|^2}}{\prod_{d=1}^{s} \left[\pi(1+\tan^2(y_{.d}))\right]} dy \qquad (9)$$

According to **Lemma 1**, it's easy to get:

$$H(v) \approx \widetilde{H}(v) =$$

$$\frac{1}{N} \sum_{h=1}^{N} \left[e^{-\beta \min\limits_{1\le i\le k} \|\tan(\pi\cdot y_h)-v_i\|^2} \cdot \prod_{d=1}^{s} \left[\pi(1+\tan^2(y_{hd}))\right] \right] \qquad (10)$$

In this term, only v_i are unknown for a given sample set y^0. So if v_i are fixed, it's easy to compute $H(v)$. Thus, the estimation of $H(v)$ is finished.

Recall the new objective function by substituting $\widetilde{H}(v)$ into it; we finally have a new objective function of the clustering algorithm:

$$J_{PKM}(X, v) = \frac{\beta}{n} J_{k-means}(X, v) \qquad (11)$$
$$+ \ln\left(\frac{1}{N} \sum_{h=1}^{N} \left[e^{-\beta \cdot minv} \cdot \prod_{d=1}^{s} \left[\pi(1+\tan^2(y_{hd}))\right] \right] \right)$$

where $minv = \min\limits_{1\le i\le k} \| \tan(\pi \cdot y_h) - v_i \|^2$ and y are samples from a uniform distribution within the interval $[-0.5, 0.5]^s$ and N is a large predefined number(typically twice or more times than the number of samples). Minimizing this function, it will get the prototypes $v_1, v_2, ..., v_k$ and cluster the dataset. Moreover, the β can also be determined by minimizing the function if unknown. Now, the Penalized K-means cluster algorithm is designed.

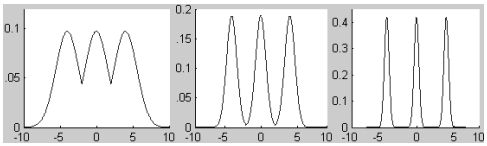

Figure 1. probability density figure with same centers [-4,0,4] and different $\beta\{1/5; 1; 5\}$

4. The Scale Factor β

In this algorithm, β is a scale factor. More interestingly, it is possible to learn the β by searching the minimum of the objection function of Penalized K-means. Meanwhile, the K-means algorithm doesn't influence by the factor if we substitute β into its objective function; while in the Penalized K-means algorithm, β is a new parameter for us to know more about the feature of the data set.

It's clear that the β has an important effect on the assumed p.d.f. From Fig 1, we will find that the p.d.f. changes a lot when β changes.

Obviously, β shows the separation among inter-clusters of a data set in some extent. The bigger the β, the more separate the clusters will be; when β becomes smaller, clusters will be more overlapped. Therefore, after introducing β in the p.d.f., the data set can be depicted more accurately by our algorithm. Meanwhile, because β does influence the value of the Penalized K-means object function nonlinearly by existing both in the front part and the penalized term in this function, it will approximately get factor β and centers $v_1, v_2, ..., v_k$ simultaneously by minimizing $J_{PKM}(X, v)$. Furthermore, finding β by this function is fairly possible because the paper have made an assumption in the beginning that the probability density of the data is mostly like the following: $Pr(x_j|v) = (H(v))^{-1} e^{-\beta \min\limits_{1\le i\le k} \|x_j - v_i\|^2}$.

5. Experiments

In this section, some experiments are conducted in order to test the performance of the new algorithm. The standard Genetic Algorithm is used to minimize $J_{PKM}(X, v)$ to get prototypes $v_1, v_2, ..., v_k$ and the factor β and then the data are clustered by assigning each sample to the nearest v_i. We treat the centers and factor as the chromosomes instead of labels of every sample.

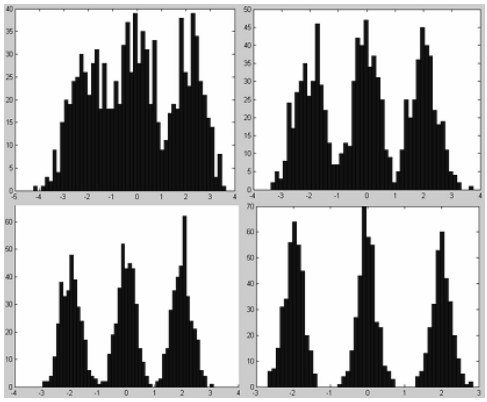

Figure 2. histograms of 4 group data with centers[-2,0,2] and $\beta = \{1, 2, 4, 8\}$

5.1. Artificial data with different factor β

First, 4 groups of 1-dimensional data are identically drawn from the p.d.f. $Pr(x_j|v) = (H(v))^{-1} \cdot e^{-\beta \min\limits_{1\le i\le k} \|x_j - v_i\|^2}$. Each group has 1000 samples round 3 specified centers $[v_1, v_2, v_3] = [-2, 0, 2]$, and β are different, which are {1, 2, 4, 8}, see Fig. 2.

Then, 10 groups of Y are uniformly drawn from the interval [-0.5 0.5], with the dimension of 2000*1. Then we program the objective function of Penalized K-means as a .m file and minimize it by GA Toolbox in Matlab 7.0. Default parameters are set except the stopping generations which are 1000.

The following results(Table 1) indicate that our algorithm could efficiently and approximately get the centers and the factor β. In order to avoid the sensitivity of specific Y, we use the 10 different groups of Y to get 10 groups of centers and β for each data set, then the 10*3 centers are treated as 1-dimensional data set and clustered by classic K-means algorithm. Meanwhile, the average value of 10 β is taken as the final result.

We also sample more overlapping and separated data with different centers for testing. The following results(Table 2) are from data with centers [-1.5,0,1.5], [-4,0,4], and β{1, 2, 4, 8}. Other conditions are the same as above.

5.2. Artificial data with different centers and specified β

In the following experiments, the data sets are created by given factor β, but with different centers. Again, we first sample 4 groups of 1-dimensional data obeying the p.d.f. $Pr(x_j|v) = (H(v))^{-1} e^{-\beta \min\limits_{1\le i\le k} \|x_j - v_i\|^2}$. Here,

Table 1. Results after PKM and K-means(KM) by using data in Fig.2

Data		Centers			β
Group1	PKM	-2.0544	-0.0582	2.0942	1.0185
β=1	KM	-2.1744	-0.0514	2.1922	—
Group1	PKM	-2.0345	-0.0234	2.0209	2.0777
β=2	KM	-2.0546	-0.0258	2.0417	—
Group1	PKM	-2.0039	-0.0109	1.9848	4.2964
β=4	KM	-2.0005	-0.0102	1.9882	—
Group1	PKM	-2.0009	-0.0173	1.9925	7.8381
β=8	KM	-1.9979	-0.0155	1.9961	—

Table 2. Results of PKM and K-means(KM) with original data centers [-1.5,0,1.5], [-4,0,4], and β{1, 2, 4, 8}

Data		Centers			β
Group2	PKM	-1.4248	0.0289	1.4054	0.8262
β=1	KM	-1.7376	-0.0435	1.6876	—
Group2	PKM	-1.4453	-0.0172	1.5219	2.0944
β=2	KM	-1.5241	-0.0047	1.5876	—
Group2	PKM	-1.5016	-0.0109	1.4929	4.3003
β=4	KM	-1.5087	-0.0072	1.5039	—
Group2	PKM	-1.4960	-0.0119	1.4929	8.1345
β=8	KM	-1.4947	-0.0095	1.4904	—
Group3	PKM	-4.0516	0.0406	4.0020	1.0272
β=1	KM	-4.0608	0.0349	4.0180	—
Group3	PKM	-4.0238	-0.0264	3.9958	1.8722
β=2	KM	-4.0288	-0.0282	3.9993	—
Group3	PKM	-4.0048	0.0005	4.0381	4.0313
β=4	KM	-4.0061	0.0004	4.0338	—
Group3	PKM	-3.9791	-0.0145	4.0045	7.6670
β=8	KM	-3.9830	-0.0127	4.0089	—

each group has 1000 samples round 3 specified centers, which are [-2,0,2], [-3, 0,3], [-4,0,4], [-8,0,8], and β is set with 1. 10 groups of centers are gained with 10 groups of Y(2000*1 dimension), and then these centers are clustered by K-means again.

From above results in Table 3, it's obvious that the Penalized K-means performs well when the data are overlapped in a certain extent, but a little weaker than K-means when data are separated clearly. IRIS data set with single attribute have also been tested, and the results(in Table 4) show that the Penalized K-means algorithms has better performance.

It is still a challenging problem to sample a data set which draws from the p.d.f. $Pr(x_j|v)$ with high dimensions. In future, we will do more similar experiments.

Table 3. results of Data Group4 with different Y of 2000*1 dimension

Data	Algorithms	Centers		
Group4	PKM	-2.0746	0.0836	1.9867
[-2,0,2]	K-means	-2.1755	0.0478	2.1155
Group4	PKM	-3.0543	-0.0163	2.9846
[-3,0,3]	K-means	-3.0682	-0.0212	3.0056
Group4	PKM	-4.0326	-0.0137	4.0003
[-4,0,4]	K-means	-4.0427	-0.0170	4.0182
Group4	PKM	-8.0726	0.0613	8.0466
[-8,0,8]	K-means	-8.0264	0.0645	7.9722

Table 4. Result of IRIS data on the 3rd attribute(Er are wrong labeled numbers)

Data Y	v_1	v_2	v_3	Er
each Y is	1.4642	4.3096	5.3872	7
2000*1	1.4665	4.2165	5.5015	7
dimension.	1.4554	4.4279	5.6612	10
PKM	1.4788	4.3729	5.5686	8
each Y is	1.4447	4.3765	5.5095	8
4000*1	1.4812	4.4277	5.6441	10
dimension.	1.4353	4.3188	5.5343	8
PKM	1.4653	4.3575	5.5331	8
K-means	**1.4620**	**4.4318**	**5.8265**	**16**

6. Conclusion and further Discussion

In this paper, a new K-means clustering algorithm has been proposed based on Maximum Likelihood method. We give a reasonable conjecture of p.d.f. assumed by K-means algorithm, and get a new objective function. And by minimizing it through genetic search for clustering, experiments show that the Penalized K-means algorithm has its special merit in some aspects. There are still two problems about PKM algorithm which are worthy of studying. One is that Penalized K-means algorithm need more computing than classic K-means methods; the other is that we have to check the fitness between high dimensional data and the distributions assumed by clustering algorithm. More research need to be done in the future.

Acknowledgement

Zejin Ding is supported by MBD (Molecular Basis of Disease) fellowship of Georgia State University. The work is partially supported by the Fok Ying Tung Education Foundation under Grant No. 101068, Program for New Century Excellent Talents in University in 2006, Grant NCET-06-0078, The Special Research Fund of Doctoral Program of Higher Education of China under Grant 20050004008, 973 Program under Grant No. 2007CB311002.

References

[1] P. Bradley, U. Fayyad. Refining initial points for K-Means clustering. *In Proc. of the* 15^{th} *Inte'l Conf. on Machine Learning.* Madison, WI, 91–99, July, 1998.

[2] J. Han, M. Kamber, A. Tung. Spatial clustering methods in data mining: A survey. In: H. Miller, J. Han, eds. *Geographic Data Mining and Knowledge Discovery*, Taylor and Francis, NY, 188–217, 2001.

[3] Z. Huang, M. Ng, H. Rong, Z. Li. Automated Variable Weighting in k-Means Type Clustering. *IEEE Trans. on Pattern Analysis And Machine Intelligence*, 27(5):657–668, May, 2005.

[4] K. Krishna, M. Murty. Genetic K-Means Algorithm. *IEEE Trans. on SMC-Part B: Cybernetics*, 29(3):433–439, June, 1999.

[5] M. Laszlo, S. Mukherjee. A Genetic Algorithm Using Hyper-Quadtrees for Low-Dimensional K-means Clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(4):533–543, April, 2006.

[6] J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proc. of* 5^{th} *Berkeley Symp. on Math. Stat. and Prob.*, 1:281–297, 1967.

[7] G. McLachlan, T. Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, Inc., NY, 1997.

[8] W. Pedrycz. Clustering and fuzzy clustering. In: W. Pedrycz ed. *Knowledge-Based Clustering: From Data to Information Granules*. John Wiley & Sons,NY,2005.

[9] C. Robert, G. Casella. *Monte Carlo Statistical Methods* (2^{nd} *edition)*. Springer-Verlag, NY, 2004.

[10] S. Bandyopadhyay, U. Maulik. An evolutionary technique based on K-Means algorithm for optimal clustering in R^N. *Information Sciences*, 146(1-4):221–237, October, 2002.

[11] M. Su, C. Chou. A Modified Version of the K-Means Algorithm with a Distance Based on Cluster Symmetry. *IEEE Trans. on Pattern Analysis And Machine Intelligence*, 23(6):674–680, June, 2001.

[12] K. Wu, M. Yang. Alternative c-means clustering algorithms. *Pattern Recognition*, 35(10):2267–2278, October, 2002.