A Near-Optimal Centroids Initialization in K-means Algorithm Using Bees Algorithm

M. Mahmuddin, Y. Yusof

IT Building
College of Arts and Sciences
Universiti Utara Malaysia
06010, Sintok
Kedah, Malaysia
Email:{ady, yuhanis}@uum.edu.my

ABSTRACT

The K-mean algorithm is one of the popular clustering techniques. The algorithm requires user to state and initialize centroid values of each group in advance. This creates problem for novice users especially to those who have no or little knowledge on the data. Trial-error attempt might be one of the possible preference to deal with this issue. In this paper, an optimization algorithm inspired from the bees foraging activities is used to locate near-optimal centroid of a given data set. Result shows that propose approached prove it robustness and competence in finding a near optimal centroid on both synthetic and real data sets.

Keywords
Optimization, K-means, Nature-Inspired Algorithm

1. 0 INTRODUCTION

Clustering is one of the important towards to knowledge acquisition for unlabelled data in machine learning. The learning process in clustering is different and more problematic compare to the labeled one. The unlabelled data does not require class in training patterns (or instances) while the labeled data to produce a learned model. Learning on labeled data known as 'supervised learning', needs a substantial number of training patterns of labeled data to fulfill the classification performance. An inadequate number of training labeled data affects the performance of the learned model.

Unlabelled data is regarded as incomplete information because it has no information of the output label for each training pattern. In the real world, unlabelled data from observation of many research fields are easier to collect. Manual labeling of the unlabelled data consumes a lot of time and is sometimes an impossible process. For example, it is impractical to label a million websites manually in a web-page classification problem. The same problem can also be seen in some other current applications, including text categorization, bioinformatics and image classification. These data types are extremely large and it is hard to label every one of them.

The K-means algorithm is known as one of the popular clustering techniques at least for two reasons: easy to understand and its robustness. However, K-means suffers with some weaknesses and one of them is it requires user to define total number of group that might exist in the data. This total number of group is basically the centroid (centre of cluster) initialization values. This condition causes problem for those who are new with the algorithm.

Cluster analysis is a process to categorize a given data set $D_i = \{x_1, x_2, \ldots, x_N\}$ in *n*-dimensional space from *N* number of objects into *k* homogenous clusters. The typical categorizing practice is to assign each object to the nearest centroids, $C_i = \{C_1, C_2, \ldots, C_k\}$ such that an object, x_i is belong to the *y*-th cluster when $I(y \mid x_i) = 1$, depending on the minimum values of a distortion error as in the Eq. 1.

$$I(y \mid x_i) = \begin{cases} 1 & if\ y = \arg\min_y \left\| x_i - c_y \right\|^2 \\ 0 & otherwise \end{cases} \qquad (1)$$

The K-means algorithm is sensitive to centroids initialization performed at the beginning of clustering condition . However, the cluster initialization is important in order to ensure the final cluster centre as accurate as possible. In this paper, a nature-inspired optimization algorithm is applied to K-means algorithm. The idea behind it is that the Bees Algorithm with automatically find a near optimal centroid of the analyzed data.

This paper is organized as follow: Section 2 explains ideas behind this work. A series of simulation is undertaken in order to prove the robustness and effectiveness of the proposed algorithm is discussed in Section 3. Conclusion of the work can be seen in Section 4.

2.0 OPTIMIZATION OF CENTROID USING BEES ALGORITHM

Initialization of parameters in K-means can be traced as early as the algorithm is introduced. Forgy uses a random seed approach (see in) in presumption that if a point is randomly chosen, it is more likely to find a good quality centroid in area of density data objects. MacQueen continues the use of random sampling approach by introducing the seed idea. He proposed the use k number of cluster of objects in a data set as the initial seed. The centroid of clusters is updated by calculating the mean values of all points in the cluster. This process continues until all of the objects in the data set have been assigned to their own cluster and the centroids' position shows no changes.

2.1 The Bees Algorithm

There are properties of the real bees food foraging (searching) activities are fundamentally important in developing Bees Algorithm. This searching is different from the Ant Colony Optimization where the ant searching is based on trail-laying trail-following behavior . Pheromone is placed by the ant to be as guidance trails for the next ant to follow. The amount of pheromone become intense when other ant leave their pheromone if explore the same trail. At some stage, more ants will choose to the most intense trail rather to other path. This activity is known as stigmergy communication.

Bees do not apply the stigmergy searching concept. Searching as mentioned in previous section is sourced from the 'dance' communication at dance area in the hive. Some significant similarities for the real bees' activities are inherited during development of the Bees Algorithms. Figure 1 shows a typical pseudo-code of the Bees Algorithm. More details of this algorithm can be read in .

2.2 Optimizing of K-means Using Bees Algorithm

Babu and Murty use genetic programming (GA) in finding near-optimal seed selection. Population of the seed selections is measured by running the K-means algorithms until it converges and then calculates a distortion value. A solution is generated in each generation population of solutions. This repetition generation process continues until a predetermined total number of generation is reached.

One of the disadvantages of using the GA approach is the difficulty of identifying a binary string to denote the centroid, especially with a floating point type of data. This situation will discourage the novice GA user who has only a basic knowledge. The users also have to set a few GA's parameters (such as crossover and mutation rate) before it can work at the best level. 'Trial & Error' and 'Mix & Match' are sometimes needed when tuning the best combination values of the parameters and this is time consuming.

```
Initialise parameters.
Step 1. Initialise population.
   Step 2. Evaluate fitness of the
                population.
Do
Step 3. Select elite bees and
          neighbourhood search.
Step 4. Select other sites for
          neighbourhood search.
Step 5. Recruit bees around selected
          sites and evaluate fitness.
Step 6. Select fittest bee's site from
          each site.
Step 7. Assign remaining bees to
          search randomly and
          evaluate their fitness.
While stopping criterion not met.
```

Figure 1: A typical simplified pseudo-code of the Bees Algorithm

K-Bees algorithm performance evaluation is based on total distance of each object to its centroid. The K-Bees algorithm is not constructed to find the optimal centroids. Modification of K-Bees is performed by calculating the distance of each data objects to its nearest centroid by using Eq. 1 and Eq. 2. The object that produces the smallest distance belongs to the particular cluster.

$$DiErr = \frac{1}{N}\sum_{y=1}^{k}\sum_{i=1}^{N} I(y \mid x_i)\left\|x_i - c_y\right\|^2 \qquad (2)$$

3.0 SIMULATION

In order to evaluate the algorithm capabilities to calculate and find the near-optimal centroids, a series of experiment is undertaken using synthetic and real data sets. Two-dimensional synthetic data sets with different characteristic have been chosen for this experiment. The *simple* data consists of 30 objects that have three distinctive clusters and the total number of objects in each cluster is equally distributed. The *unbalanced* data has four distinctive clusters with various number objects in the clusters. The *overlapped* data has two distinctive clusters with another two clusters and two overlapped other. The *noisy* data has five clusters with a lot of noisy objects around these clusters. And the *mixed* data consist of noisy objects and overlapped clusters. Object distribution of each data set can be seen in Table 1. And to illustrates further of these objects can be seen in Figure 2 of each synthetic data object of *simple, unbalanced, noisy, overlapped* and *mixed* respectively.

The algorithm is also tested on real data sets. For this purpose, three different data sets have been chosen, namely Iris , Ruspini and Vowel data.

Table 1: *Summary of data sets use in the experiment*

Data set	# objects/cluster/ attribute	Data type	Objects per cluster
Simple	30/3/2	Real	10 per cluster
Unbalanced	100/4/2	Real	11, 15, 33, 41
Overlapped	100/4/2	Real	17, 21, 33, 29
Noisy	100/5/2	Real	16, 17, 17, 16, 20, and 14 noisy objects
Mixed	100/6/2	Real	11, 19, 13, 16, 19, 15 and 8 noisy objects
Iris	150/3/4	Real	50 per cluster
Ruspini	75/4/2	Con.	23, 20, 17, 15
Vowel	871/6/3	Int.	72, 89, 172, 151, 207, 180

Cont. = continuous and Int. = integer data type respectively.

4.0 RESULT

Table 2 and Table 3 shows the accumulated error from the true centroid using the proposed approach of synthetic and real data sets respectively.

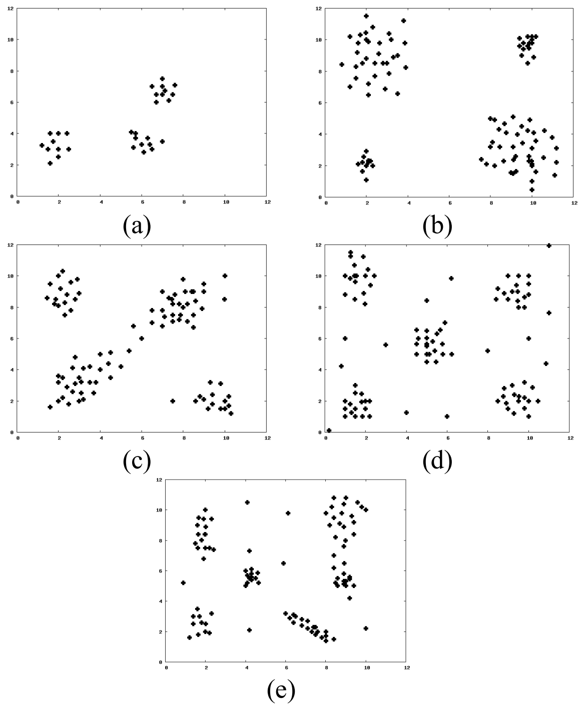

Figure 2: Synthetic data sets that is used (a) simple (b) unbalanced (c) overlapped (d) noisy and (e) mixed data sets

Table 2: *Comparison between the true centre and obtained centre of five synthetic data sets*

Data set	Cluster	True Centre	Obtained Centre	Error
Simple	Cluster1	(2, 2)	(2.006, 1.992)	(0.006, -0.008)
	Cluster2	(10, 9)	(9.959, 8.922)	(-0.041,-0.079)
	Cluster3	(13, 5)	(12.923, 5.061)	(-0.077, 0.061)
Unbalanced	Cluster1	(1.978, 2.121)	(2.039, 2.193)	(0.061, 0.072)
	Cluster2	(9.797, 9.637)	(9.771, 9.681)	(-0.026, 0.044)
	Cluster3	(2.408, 8.948)	(2.422, 8.921)	(0.015, -0.027)
	Cluster4	(9.428, 3.001)	(9.440, 2.933)	(0.012, -0.068)
Noisy	Cluster1	(1.587, 1.716)	(1.593, 1.758)	(0.006, 0.042)
	Cluster2	(9.401, 9.049)	(9.440, 9.030)	(0.039, -0.019)
	Cluster3	(9.460, 2.126)	(9.376, 2.197)	(-0.084, 0.071)
	Cluster4	(1.658, 9.804)	(1.598, 9.697)	(-0.060, -0.107)
	Cluster5	(5.212, 5.618)	5.241, 5.732)	(0.030, 0.114)
Overlapped	Cluster1	(3.199, 3.356)	(3.212, 3.338)	(0.013, -0.018)
	Cluster2	(7.861, 8.061)	(7.866, 8.073)	(0.005, 0.013)
	Cluster3	(2.554, 8.812)	(2.242, 8.864)	(-0.312, 0.053)
	Cluster4	(9.595, 2.028)	(9.588, 2.030)	(-0.007, 0.003)
Mixed	Cluster1	(1.750, 2.510)	(1.877, 2.713)	(0.127, 0.203)
	Cluster2	(8.923, 9.474)	(8.772, 9.475)	(-0.151, 0.001)
	Cluster3	(4.280, 5.575)	(4.379, 5.771)	(0.099, 0.196)
	Cluster4	(2.188, 8.513)	(2.038, 8.479)	(-0.150, -0.034)
	Cluster5	(7.202, 2.288)	(7.354, 2.275)	(0.152, -0.013)
	Cluster6	(8.876, 5.483)	(8.864, 5.455)	(-0.012, -0.028)

The total error generated in synthetic data sets produce very low amount of error was generated each of these synthetic data sets. amplifies this by showing the minimum, maximum and mean values of error for each data set. The table also shows that the *overlapped* data has the biggest error range, while the *mixed* data set gives the second highest. The rest generally generate almost the same small range of error.

This result also demonstrates that an overlapped data set is a difficult type of data to be analyzed. The reason for this can be accounted for by the macro view of human eyes. Human eyes can only spot objects as one group when they look from the top, even though there are many objects partly covered by other objects around them

Table 3: *Comparison between the true centre and obtained centre of four real data sets*

Data set	Cluster	True Centre	Obtained Centre	Error
Iris	Cluster1	(5.006, 3.418, 1.464, 0.244)	(5.030, 3.355, 1.467, 0.207)	(0.020, -0.063, 0.003, -0.037)
	Cluster2	(5.936, 2.77, 4.26, 1.326)	(5.924, 2.751, 4.411, 1.387)	(-0.012, -0.019, 0.151, 0.061)
	Cluster3	(6.588, 2.974, 5.552, 2.026)	(6.726, 3.120, 5.621, 2.174)	(0.138, 0.146, 0.069, 0.149)
Ruspini	Cluster1	(20.150, 64.950)	(19.701, 63.106)	(-0.449, -1.844)
	Cluster2	(43.913, 146.043)	(41.854, 147.261)	(-2.059, 1.217)
	Cluster3	(98.176, 114.882)	(99.401, 117.255)	(1.225, 2.372)
	Cluster3	(68.933, 19.400)	(68.662, 17.960)	(-0.272, -1.440)
Vowel	Cluster1	(603.472, 1468.056, 2379.306)	(387.963, 2153.878, 2676.139)	(-215.509, 685.823, 296.834)
	Cluster2	(698.315, 1240.449, 2338.202)	(622.210, 1301.211, 2330.036)	(-76.105, 60.761, -8.167)
	Cluster3	(342.209, 2202.035, 2805.058)	(359.789, 2291.234, 2973.822)	(17.580, 89.199, 168.763)
	Cluster4	(358.146, 978.477, 2494.503)	(448.893, 993.123, 2677.936)	(90.747, 14.646, 183.434)
	Cluster5	(504.831, 1866.570, 2617.440)	(496.444, 1839.553, 2554.854)	(-8.387, -27.017, -62.586)
	Cluster6	(481.944, 1057.278, 2501.611)	(396.322, 1013.068, 2332.378)	(-85.622, -44.210, -169.233)

Table 3 reveals the result of the proposed approach to real data sets. Again, a low error is generated by Iris and Ruspini data sets compared to the true centroid. For example, in the Iris data set the obtained centroids produce an error of around 1.2% to 2.9% of the total error compared to the true centroid values. The worst result is obtained from the Vowel data set. This data set produces 1.9% to 26.7% error. Vowel data is massively overlapped between each object, which is shown in (b). For the Ruspini data set, the total error obtained was 1.6% to 2.7%.

5.0 CONCLUSION

Cluster initialization is one of the main issue in K-means algorithm. This work proposes a nature-inspired optimization technique to overcome this drawback of the K-means algorithm. The result from a series of experiment undertaken on various data sets (real and synthetic) shows that the proposed algorithm is capable to find a near-optimal solution that produce minimal error of each cluster from it respective centroid.

RERFRENCES