Hybrid Solution of K-Means with PSO and Cuckoo Search

Alper TADAY
Computer Engineering Department
TOBB University of Economics and Technology
Ankara,Turkey
st101101020@etu.edu.tr

Ecem Elvin ÇEVİK
Computer Engineering Department
TOBB University of Economics and Technology
Ankara,Turkey
st111101037@etu.edu.tr

Fırat TOP
Computer Engineering Department
TOBB University of Economics and Technology
Ankara,Turkey
st101101047@etu.edu.tr

ABSTRACT

This paper presents a k-means clustering approach for classifying and shows how the traditional k-means clustering algorithm can be modified to be used as a classifier algorithm. The proposed model comprises particle swarm optimization (PSO) with the traditional k-means algorithm to provide the requirements of a classifier. Also it combines cuckoo-search based evolutionary algorithm with some centroid-calculation heuristics.

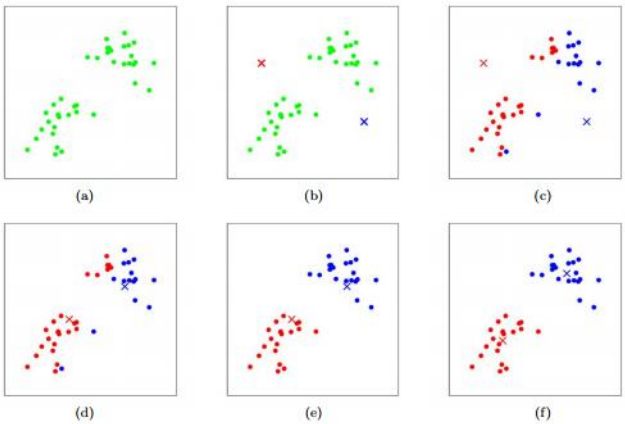
1. INTRODUCTION

This work promotes to the application of a popular clustering technique, k-means clustering, modified with the use of PSO. The classifier is projected using the proposed PSO based k-means clustering algorithm. This proposed algorithm states the optimal cluster centers based on train data set. Classification precision and false dismissals provide affirmation to the performance of the algorithm for both training and testing steps. The proposed clustering approach is implemented and the traditional k-means algorithm compared with the results.

In this work, PSO based k-means clustering algorithm is determined as a suitable tool for the design of classifier. This gives a brief outline of k-means clustering, use of PSO in improving the performance of k-means algorithm for classification.

2. K-MEANS

K-means is one of the simplest algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. [1]



The algorithm is composed of the following steps: [2]

1. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.
2. Assign each object to the group that has the closest centroid.
3. When all objects have been assigned, recalculate the positions of the K centroids.
4. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

3. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is a quantitative method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Elementally, PSO is a population based stochastic optimization technique. Easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control.

Algorithm 1 PSO

```
For each particle
    Initialize particle
END
Do
    For each particle
        Calculate fitness value
        If the fitness value is better than the
        best fitness value (pbest) in history
        Set current value as the new pbest
    End
    Choose the particle with the best fitness value
    of all the particles as the gbest
    For each particle
        Calculate particle velocity according equation 1
        Update particle position according equation 2
    End
    While maximum iterations or minimum
    error criteria is not attained [3]
```

4. CUCKOO SEARCH

It was inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds (of

other species). Some host birds can engage direct conflict with the intruding cuckoos.

Cuckoo search idealized such breeding behavior, and thus can be applied for various optimization problems. It seems that it can outperform other metaheuristic algorithms in applications.

An important advantage of this algorithm is its simplicity. In fact, comparing with other population or agent-based metaheuristic algorithms such as particle swarm optimization. There is essentially only a single parameter P_a in CS (apart from the population size n). Therefore, it is very easy to implement.

Cuckoo search uses the following representations:

Each egg in a nest represents a solution, and a cuckoo egg represents a new solution. The aim is to use the new and potentially better solutions (cuckoos) to replace a not-so-good solution in the nests. In the simplest form, each nest has one egg. The algorithm can be extended to more complicated cases in which each nest has multiple eggs representing a set of solutions.

CS is based on three idealized rules:

1. Each cuckoo lays one egg at a time, and dumps its egg in a randomly chosen nest;
2. The best nests with high quality of eggs will carry over to the next generation;
3. The number of available hosts nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability $P_a \in (0, 1)$. Discovering operate on some set of worst nests, and discovered solutions dumped from farther calculations. [4]

5. HYBRID OF K-MEANS AND PSO ALGORITHM

The hybrid of k-means and PSO is proposed to be initialized with k-means module and then PSO is applied on the initial results generated by k-means module. In k-means module the recalculation of the cluster centroid is done as:

$$c_j = \frac{1}{nj} \sum_{\forall d_j \in S_j} d_j$$

where dj denotes the document vectors that belong to cluster Sj ; cj stands for the centroid vector; nj is the number of document vectors belong to cluster Sj. The fitness function used to minimize in the PSO module is the ADDC (Average Distance Documents to the cluster centroid) which is computed as follows:

$$f = \frac{\sum_{i=1}^{N_c} \left\{ \frac{\sum_{j=1}^{p_i} d(o_i, m_{ij})}{p_i} \right.}{N_c}$$

Where mij denotes the jth document vector, which belongs to cluster i; Oi is the centroid vector of the ith cluster; d(oi,mij) is the distance between document mij and the cluster centroid Oi.; Pi stands for the number of documents, which belongs to cluster Ci; and Nc stands for the number of clusters.

Algorithm 2 K/PSO

1: [K-Means Module] Select K-points as initial centroids
2: Repeat
3: Form K-clusters by assigning each point to its closest centroid
4: Recompute the centroid of each cluster
5: UNTIL centroid does not change
6: [PSO Module] Run PSO on initial clusters generated by K-Means
7: Initialize the Particles (Clusters)
8: Initialize Vi(t), Vmax , c1 and c2
9: Initialize Population size and iterations
10: Initialize clusters to input data
11: Obtain the original position
12: Iterate Swarm
13: Find the winning points
14: Update Velocity and Position using equation 1 and 2
15: Evaluate the strength of Swarm
16: Iterate Generation
17: Consume weak particles
18: Recalculate the position
19: Exit when the maximum number of iterations fulfilled or any other stopping criteria is reached. [5]

6. HYBRID OF CUCKOO SEARCH AND PSO ALGORITHM

In this section, we explore the details of the proposed hybrid algorithm. The nature of cuckoo birds is that they do not raise their own eggs and never build their own nests, instead they lay their eggs in the nest of other host birds. If the alien egg is discovered by the host bird, it will either throw these alien eggs away or simply abandon its nest and build a new nest elsewhere. Thus cuckoo birds always are looking for a better place in order to decrease the chance of their eggs to be discovered. In the proposed hybrid algorithm, the ability of communication for cuckoo birds has been added. The goal of this communication is to inform each other from their position and help each other to immigrate to a better place. Each cuckoo bird will record the best personal experience as pbest during its own life. In addition, the best pbest among all the birds is called gbest. The cuckoo birds' communication is established through the pbest, gbest, and they update their position using these parameters and also the velocity of each one. The update rule for cuckoo is position is as the following:

$$V_{t+1}^{i} = W_t^{i} * V_t^{i} + C_1 * \text{rand}( ) * (\text{pbest} - x_t^{i}) + C_2 * \text{rand}( ) * (\text{gbest} - x_t^{i}) \quad \textbf{[Equation 1]}$$

$$x_{t+1}^{i} = x_t^{i} + V_{t+1}^{i} \quad \textbf{[Equation 2]}$$

Where W is inertia weight which shows the effect of previous velocity vector (V_t^i) on the new vector, C_1 and C_2 are acceleration constants and rand( ) is a random function in the range [0, 1] and x_t^i is current position of the cuckoo.

Algorithm 3 CS/PSO

1: **begin**
2: Objective function f(x), $x = (x_1, \ldots, x_d)^T$;
3: Initial a population of n host nest x_i (I = 1,2, …, n);
4: **While** (t < MaxGeneration) or (stop criterion);
5: Get a cuckoo (say **i**) randomly by Levy flights and record pbest;
6: Evaluate its quality/fitness F_i;
7: Choose a nest among n (say **j**) randomly;
8: **if** ($F_i > F_j$),
9: Replace **j** by the new solution;
10: **end**
11: Move cuckoo birds using equation 1 and 2;
12: Abandon a fraction (P_a) of worse nests [and build new ones at new locations via Levy flights];
13: Keep the best solutions (or nests with quality solutions);
14: Rank the solutions and find the current best;
15: **end while**
16: Post process results and visualization;
17: **end** [6]

7. TEST AND RESULTS

The hybrid CS/PSO and K/PSO algorithms have been implemented in Java using NetBeans 8.1 IDE and Eclipse. A GUI based clustering approach has been developed. We test our algorithms with 2 dataset. Here is some screenshots from GUI:

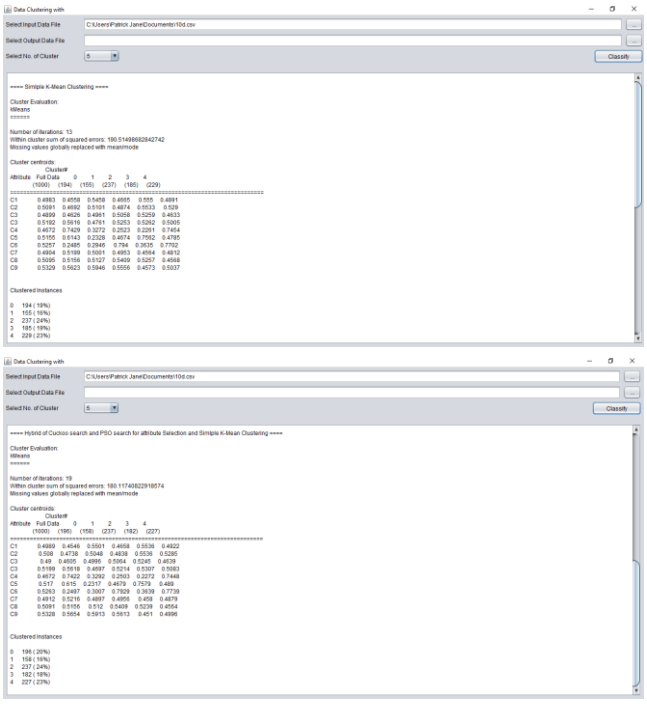


Table I

Simple K-Mean Clustering & Number of Cluster: 3

Number of iterations: 10

Within cluster sum of squared errors: 217.6101309161789

Cluster centroids:

Attribute	Full Data (1000)	0 (278)	1 (266)	2 (456)
C1	0.4983	0.4875	0.4694	0.5217
C2	0.5091	0.5305	0.4702	0.5187
C3	0.4899	0.4664	0.482	0.5088
C4	0.5192	0.4929	0.5328	0.5273
C5	0.4672	0.7084	0.6551	0.2105
C6	0.5155	0.4802	0.5594	0.5114
C7	0.5257	0.7714	0.2339	0.5461
C8	0.4904	0.4971	0.4709	0.4977
C9	0.5095	0.4755	0.5036	0.5337
C10	0.5329	0.5102	0.5592	0.5313

Clustered Instances:

0 278 (28%)
1 266 (27%)
2 456 (46%)

Table II

Hybrid of Cuckoo search and PSO search for attribute Selection and Simple K-Mean Clustering & Number of Cluster: 3

Number of iterations: 11

Within cluster sum of squared errors: 202.14329153330806

Cluster centroids:

Attribute	Full Data (1000)	0 (279)	1 (266)	2 (455)
C1	0.4997	0.4923	0.4703	0.5214
C2	0.5093	0.5315	0.4707	0.5182
C3	0.49	0.4675	0.4812	0.5089
C4	0.52	0.4945	0.5313	0.5291
C5	0.4668	0.7075	0.6546	0.2094
C6	0.5147	0.4773	0.5587	0.512
C7	0.5258	0.7732	0.2341	0.5446
C8	0.4891	0.4952	0.4711	0.496
C9	0.5096	0.4725	0.5062	0.5343
C10	0.5328	0.511	0.5599	0.5304

Clustered Instances

0 279 (28%)
1 266 (27%)
2 455 (46%)

Table III

Simple K-Mean Clustering & Number of Cluster: 6

Number of iterations: 24

Within cluster sum of squared errors: 181.4657573178813

Cluster centroids:

Attribute	Full Data (1000)	0 (179)	1 (150)	2 (200)	3 (180)	4 (142)	5 (149)
C1	0.4983	0.4507	0.5457	0.4405	0.5764	0.4915	0.4972
C2	0.5091	0.4716	0.5323	0.4803	0.5379	0.5469	0.4987
C3	0.4899	0.4701	0.487	0.505	0.5369	0.4324	0.4942
C4	0.5192	0.5638	0.4851	0.5315	0.5241	0.5223	0.4745
C5	0.4672	0.7366	0.2934	0.2308	0.2171	0.6725	0.7422
C6	0.5155	0.611	0.2783	0.3989	0.7773	0.7624	0.2444
C7	0.5257	0.2187	0.2575	0.7918	0.4155	0.7665	0.7109
C8	0.4904	0.5125	0.4768	0.5035	0.4486	0.4951	0.4951
C9	0.5095	0.5123	0.5258	0.5399	0.5196	0.4827	0.4623
C10	0.5329	0.5392	0.636	0.5199	0.4621	0.5636	0.4951

Clustered Instances

0 179 (18%)

1 150 (15%)

2 200 (20%)

3 180 (18%)

4 142 (14%)

5 149 (15%)

Table IV

Hybrid of Cuckoo search and PSO search for attribute Selection and Simple K-Mean Clustering & Number of Cluster: 6

Number of iterations: 24

Within cluster sum of squared errors: 160.5807335438179

Cluster centroids:

Attribute	Full Data (1000)	0 (182)	1 (158)	2 (197)	3 (173)	4 (145)	5 (144)
C1	0.4987	0.5071	0.5909	0.4856	0.5453	0.4703	0.3783
C2	0.5094	0.5284	0.5619	0.4863	0.498	0.5228	0.46
C3	0.4898	0.5458	0.4704	0.5319	0.4763	0.4032	0.4863
C4	0.5195	0.4854	0.5032	0.5568	0.5172	0.5237	0.528
C5	0.4684	0.531	0.2367	0.202	0.4812	0.6879	0.7696
C6	0.5163	0.2254	0.3217	0.6186	0.8107	0.7644	0.3553
C7	0.5258	0.7996	0.2668	0.7109	0.2253	0.7709	0.3259
C8	0.4904	0.4902	0.4792	0.5172	0.4828	0.4959	0.47
C9	0.5097	0.5275	0.5299	0.5089	0.5462	0.4872	0.4453
C10	0.5318	0.5833	0.638	0.4078	0.5166	0.5896	0.4801

Clustered Instances

0 182 (18%)

1 158 (16%)

2 197 (20%)

3 173 (17%)

4 145 (14%)

5 145 (14%)

8. CONCLUSION

In this paper, we aim to present techniques and algorithms separately which improve clustering quality. These are k-means, PSO and Cuckoo Search. We determine to improve clustering algorithm by combining with these algorithms. CS/PSO based k-means algorithm is more useful than traditional k-means as we expected.

9. REFERENCES

[1] https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm

[2] http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html

[3] http://www.swarmintelligence.org/tutorials.php

[4] https://en.wikipedia.org/wiki/Cuckoo_search

[5] Evaluation of text document clustering approach based on particle swarm optimization, Stuti Karol, Veenu Mangat U.I.E.T, Panjab University, Chandigarh, India

[6] Intelligent Information and Database Systems: 4th Asian Conference, ACIIDS 2012, Kaohsiung, Taiwan, March 19-21, 2012, Proceedings, Part II