K-Means Initialization Methods for Improving Clustering by Simulated Annealing

Gabriela Trazzi Perim, Estefhan Dazzi Wandekokem, and Flávio Miguel Varejão

Universidade Federal do Espírito Santo - Departamento de Informática, Av. Fernando Ferrari, s/n, Campus de Goiabeiras, CEP 29060-900, Vitória-ES, Brasil
{gabrielatp,estefhan}@gmail.com, fvarejao@inf.ufes.br

Abstract. Clustering is defined as the task of dividing a data set such that elements within each subset are similar between themselves and are dissimilar to elements belonging to other subsets. This problem can be understood as an optimization problem that looks for the best configuration of the clusters among all possible configurations. K-means is the most popular approximate algorithm applied to the clustering problem, but it is very sensitive to the start solution and can get stuck in local optima. Metaheuristics can also be used to solve the problem. Nevertheless, the direct application of metaheuristics to the clustering problem seems to be effective only on small data sets. This work suggests the use of methods for finding initial solutions to the K-means algorithm in order to initialize Simulated Annealing and search solutions near the global optima.

Keywords: Combinatorial Optimization, Metaheuristics, K-means, Simulated Annealing.

1 Introduction

The fundamental problem of clustering consists on grouping elements that belong to a data set, according to the similarity between them [1]. Given the data set, defined as $X = \{\mathbf{x_1}, ..., \mathbf{x_N}\}$, with N elements, and each element $\mathbf{x_i} = [x_{i1}, ..., x_{id}]^t$ having d attributes, the goal is to find K groups $(C_1, ..., C_K)$, such that elements within each cluster are similar between themselves and are dissimilar to elements belonging to the other clusters.

This problem can be formulated as the combinatorial optimization task of finding the best configuration of partitions among all possible ones, according to a function that assesses the quality of partitions. With this formulation, the problem is known to be NP-Hard.

Partitional algorithms are often used for solving this problem. The most common example of this class of algorithms is K-means [2], which is applied to data with numerical attributes. Simplicity and fast convergence to the final solution are the main features of this algorithm. However, the K-means algorithm is very sensitive to the choice of the initial solution, used as a starting element for the searching process. Furthermore, the algorithm is subject to achieve local optima as a solution, according to Selim and Ismail [3].

H. Geffner et al. (Eds.): IBERAMIA 2008, LNAI 5290, pp. 133–142, 2008.

Another approach for solving the clustering problem is the use of metaheuristics, which combine techniques of local search and general strategies to escape from local optima, broadly searching the solution space. However, Rayward-Smith [4] indicates that the direct application of metaheuristics to the problem seems to be more effective in smaller sets (with a small number of elements and partitions).

Based on the tendency of the K-means algorithm to obtain local optima, this work investigates a new approach to solve the clustering problem, using initialization methods originally defined to K-means, PCA-Part [14] and K-means++ [15], together with the metaheuristic Simulated Annealing. The objective of this combination is to obtain solutions that are closer to the global optima of the problem.

Experiments were performed using a procedure based on the cross-validation technique in order to avoid bias on the results. This procedure uses different subsets of the available data to search the best values of the parameters of Simulated Annealing algorithm (initial temperature, number of iterations without improvement and the temperature decreasing rate) and to estimate the objective function. In addition, a statistical approach was adopted to analyze the results of the experiments. The results were obtained with eight real databases and the proposal approach was generally better than the other combinations of algorithms tried on the experiments.

Section 2 presents the K-means and Simulated Annealing approaches. Section 3 introduces two initialization methods, and proposes the use of these methods together with the metaheuristic Simulated Annealing. Section 4 shows the experimental results, and compares the performance of combinations of the initialization methods with K-means and Simulated Annealing. Finally, Section 5 presents the conclusions and future work.

2 Clustering with K-Means and Metaheuristics

The K-means algorithm performs the search using an iterative procedure which associates the elements to its closest partition, aiming to minimize an objective function. This function is calculated by taking the sum of the square of the Euclidian distance between each element and the centroid of the partition to which it belongs. The centroid is the representative element of the partition and is calculated as the mass center of its elements. The function which evaluates the partitions in this way is called Sum of the Squared Euclidian distances (SSE).

The classical version of K-means, according to Forgy [2], initially chooses a random set of K elements to represent the centroids of the initial solution. Each iteration generates a new solution, associating every element to the closest centroid in the current solution and recalculating the centroids after all elements are associated. This procedure is performed while convergence is not achieved.

Despite the simplicity and rapid convergence of the algorithm, K-means may be the highly sensitive to the choice of the initial solution. If that choice is bad, the algorithm may converge to a local minimum of the objective function.

Initialization methods may be used to improve the chances of a search finding a solution that is close to the global optima. The next section presents two initialization methods for choosing the initial solution of the problem. One of them, PCA-Part, represents a deterministic procedure with good results compared to other methods,

and the other, K-means++, is a stochastic procedure that guarantees that the solution is close to the global optima.

Even with the use of a meticulous initialization method, the K-means algorithm may still not escape from local optima. If the initialization leads to a solution that is also close to local optima, the result of the algorithm could be a local optimum solution.

An alternative approach for solving the problem is to apply metaheuristics, which are heuristic procedures with more general characteristics that have mechanisms to escape from locally optimal solutions.

The algorithm Simulated Annealing [5] is an example of metaheuristics that can be applied to this problem. Klein and Dubes [6] present an implementation of the Simulated Annealing algorithm for solving the clustering problem, and analyzes the appropriate values for the parameters of the algorithm, as well as the appropriate neighborhood function. Selim and Alsultan [7] also suggest an adaptation of Simulated Annealing to the clustering problem, besides conducting a detailed assessment and interpretation of the parameters. Both works were successful with simple databases.

Other works apply different metaheuristics for the problem. For example, Murty and Chowdhury [8] and Hall et al. [9] use genetic algorithms for finding solutions that are closer to the global optima. Tabu Search [10], Particle Swarm [11] and Ant Colony Optimization [12] were also used for the same purpose.

However, as indicated by Rayward-Smith [4], the direct application of metaheuristics to the clustering problem seems to be more effective in small databases.

Hybrid approaches have achieved more promising results in this case. One possible hybrid approach is the use of metaheuristics to find promising solutions in the search space and then use partitional algorithms to explore these solutions and find better ones. The work presented by Babu and Murty [13] proposes the implementation of Simulated Annealing to choose the initial solution of the K-means algorithm.

Another example of a hybrid approach is the application of partitional algorithms for generating the initial solution of some metaheuristic. For instance, Merwe and Engelbrecht [11] shows that the performance of the Particle Swarm algorithm applied to the clustering problem can be enhanced by an initial solution found by K-means.

3 K-Means Initialization Methods with Simulated Annealing

This paper proposes using the methods PCA-Part and K-means++ to choose the initial solution of the Simulated Annealing algorithm applied to the clustering problem in order to increase the chances of finding solutions that are closer to the global optima.

The proposed approach aims to obtain better results, especially in larger data sets of the problem, than those obtained with the classical version of Simulated Annealing (where the original solution is randomly chosen) and also those achieved by the K-means algorithm using random, PCA-Part and K-means++ initialization methods.

The PCA-Part method uses a hierarchical approach based on the PCA (Principal Component Analysis) technique [16]. This method aims to minimize the value of the SSE function for each iteration. Initially, the method considers that the entire data set forms a single partition. Then, the algorithm iteratively selects the partition with the highest SSE and divides it into two, in the direction that minimizes the value of SSE

after the division. This is the same direction that maximizes the difference between the SSE before and after the division. This problem is simplified to find the direction that contributes to the highest value of SSE before the division, which is determined by the direction of the largest eigenvector of the covariance matrix. The process is repeated until K partitions are generated.

The interesting aspect of this method is that, besides it potentially provides better results than those obtained with the traditional random initialization, it also performs a deterministic choice of the start solution, therefore removing the random nature of K-means.

The method K-means++ is based on a random choice of centroids, with a specific probability. Initially, the method selects a random element of the data set to represent the first centroid. The other $K-1$ centroids are chosen iteratively by the selection of elements of the data set with probability proportional to its contribution to the SSE function. Thus, the higher is the contribution of a element to the function, the greater will be the chances of that element to be chosen as centroid. Besides being fast and simple, this method improves the quality of the K-means results, assuring that the SSE of the solution will be proportional to the SSE of the optimal solution by a constant in $O(\log (K))$.

The algorithm used in this work for searching the final solution of the clustering problem was Simulated Annealing. This metaheuristic was chosen because it is a traditional algorithm and it has also been effective in a large number of optimization problems. Moreover, it is a conceptually simple procedure. Indeed, its procedure is equally simple as the K-means procedure.

The Simulated Annealing algorithm is an iterative search method based on the annealing process. The algorithm starts from an initial solution (either randomly or heuristically constructed) and an initial temperature T_0. Then, the procedure takes iterations until it achieves a stopping criterion, that is, in most implementations, the achievement of very small values of temperature, lower than a final temperature T_f.

The algorithm performs, for each value of temperature, a perturbation at the current solution until the thermal equilibrium is achieved. That equilibrium is usually implemented as a fixed number of iterations N_{iter} without improvements in the visited solutions. The algorithm randomly generates a neighbor of the current solution for performing the perturbations. If the objective function evaluation improves, the new solution is accepted. Otherwise, the solution is accepted with a probability that is directly proportional to the temperature, allowing the algorithm to escape from local optima.

Once the thermal equilibrium is reached, the temperature is reduced by a rule of cooling and the algorithm can continue doing perturbations in the solutions until a stopping criterion is achieved. One of the most common cooling rules follows a geometric form wherein the temperature decreases exponentially by a rate r, such that $T = r \times T$, with $0 < r < 1$.

Besides this basic procedure, the implemented version of the Simulated Annealing presents some characteristics that are listed as following:

The representation of the solution of the clustering problem, based on a proposal of Murty and Chowdhury [8], is given by an array of size N, with each position i of the vector representing a element x_i of the data set and having values in the range $1..K$, indicating which partition the element x_i belongs. For example, if the solution of a

specific clustering problem is $C_1 = \{\mathbf{x_1}, \mathbf{x_4}\}$, $C_2 = \{\mathbf{x_3}, \mathbf{x_5}\}$, $C_3 = \{\mathbf{x_6}, \mathbf{x_7}\}$ and $C_4 = \{\mathbf{x_2}\}$, then this solution is represented by the array [1 4 2 1 2 3 3]. This representation makes the calculation of the objective function and the centroids of the clusters easier.

The proposed algorithm, like K-means, uses the SSE criterion as objective function in order to make a consistent comparison between the algorithms.

The neighborhood function follows the proposal of Klein and Dubes [6]. It makes a random choice of an element in the data set to be moved to a random partition that is different of the current one. With this function, the new solution will be in a neighborhood that is close to the current solution.

4 Experiments and Results

The experiments evaluated the use of Simulated Annealing and K-means, both initialized with random, PCA-Part and K-means++ methods, with the goal of finding the best configuration of partitions in eight real data bases. Table 1 shows the characteristics of these databases, all of them being available in public repositories.

In these experiments, the search for the best parameters of Simulated Annealing and the estimated value of SSE were based on the cross-validation technique. The adopted procedure divides randomly the data set in q independent subsets, and takes q iterations. Iteratively one of the q subsets (Y_i) is used as a test set to estimate the outcome of the algorithm, using the best parameters found with the other (q-*1*) subsets, which are forming a training set (T_i). The final estimate of the value of SSE is the average of the results obtained in the q iterations.

Table 1. Data sets descriptions used in the experiments

Data Set	Number of Elements	Number of Attributes	Number of Classes
Iris	150	4	3
Wine	178	13	3
Vehicle	846	18	4
Cloud	1024	10	10
Segmentation	2310	19	7
Spam	4601	57	2
Pen digits	10992	16	10
Letter	20000	16	26

The search for the best parameters is made with the use of a second cross-validation procedure, in which the set T_i is randomly divided into m independent subsets and the combination of values of the parameters that minimizes the function SSE is set to be applied in Y_i. This second cross-validation procedure was used because it is necessary training the algorithm in a subset different of the subset used to estimate the SSE criterion. We assumed that another cross-validation procedure would produce a more general clustering method than the one obtained by simply finding the parameters values that optimize the SSE function in the whole partition. The two-step cross-validation procedure was performed with small values of q and m ($q = 10$ and $m = 3$) due to the exhaustive adjustment adopted.

Since the parameter value space is vast and the search performed by the algorithm is a time consuming task, the procedure evaluates all possible combinations of discrete values for the parameters. These values were chosen based on the analysis of Selim and Alsultan [7] and were equal to: T_0 = (500; 5500, 10500, 15500, 20500), T_f = (0.1), r = (0.85, 0.90, 0.95) and N_{iter} = (1000, 2000, 3000).

The combination of these methods (random, PCA-Part and K-means++) were also applied to K-means for the same subsets Y_i used with Simulated Annealing.

Table 2 presents the experimental results using the combination of different initialization methods (random, PCA-Part and K-means++) and search algorithms (K-means and Simulated Annealing). For each base, three experiments with different values of K were performed, one of them equal to the original number of classes of the data set. The best results of each test are emphasized in bold. It is worth to notice that the SSE objective function monotonically decreases when the value of K increases.

Analyzing the results of Table 2, we may notice that, in 14 of the 24 tests, the K-means initialized with PCA-Part obtained better results than K-means++, indicating that the PCA-Part may be generally better than the initialization method of K-means++ (which has the property of assuring closeness to the global optima).

Another observation is that the best result with the Simulated Annealing was greater or equal to the best result with the K-means in 20 of the 24 tests.

It was also noticed that the best results were obtained by the Simulated Annealing initialized with PCA-Part method, which got the best result in 14 of the 24 tests.

Moreover, Simulated Annealing with PCA-Part and K-means++ methods achieved greater or equal results than the K-means initialized with the K-means++ method in 20 of the 24 tests. This indicates that the metaheuristics initialized with these methods may get even closer to the global optima.

It was also observed that, in 17 of the 24 tests, the Simulated Annealing initialized with PCA-Part was better or equal to the same metaheuristic combined to the initialization method of K-means++, reinforcing the indication that the first combination is better than the second.

We applied statistical methods adapted for algorithm comparison in multiple domains (different databases). These methods show if there are differences between the algorithms with the significance level of $a\%$. The significance level indicates the probability of a random data sample generates the result, assuming that the algorithms are equivalent (null-hypothesis). Whenever the random sample produces the result with a probability that is lower than the desired significance level (generally is used $a = 5\%$), then the null-hypothesis is rejected.

The most appropriate method for comparison of multiple algorithms is the Friedman test [17, 18]. This method verifies if in c different, dependent experiments, $c > 1$, at least two are statistically different. The test makes the ordering of the algorithms for each problem separately, giving a rank to each of them with values of 1 to c. The best algorithm receives the rank 1, the second best receives the rank 2, and so on. In case of ties the algorithms receive the average of ranks that would be

Table 2. Experimental Results – SSE Average

		K-means			Simulated Annealing		
Data Set	K	Random	PCA-Part	K-means++	Random	PCA-Part	K-means++
Iris	2	13.9491	13.9491	**13.8699**	**13.8699**	**13.8699**	**13.8699**
	3	9.96375	6.7914	6.75412	**5.98379**	**5.98379**	**5.98379**
	4	5.00787	4.23722	3.87814	**3.46131**	**3.46131**	**3.46131**
Wine	2	369267	364926	368955	**362683**	**362683**	**362683**
	3	159229	154158	155967	151839	**145690**	150402
	4	93706.8	86061.1	107483	**76841.1**	79477.1	83833.8
Vehicle	3	**463986**	471304	481356	482624	466703	480391
	4	324726	290645	318651	286907	**283006**	294798
	5	242253	221410	234724	222688	**216325**	220228
Cloud	9	834627	504282	569423	678689	**477777**	525079
	10	669664	421224	448449	451459	**392694**	397056
	11	659576	377512	388657	393348	**345548**	347743
Segm.	6	1.44e+06	1.25e+06	1.246e+06	1.36e+06	**1.17e+06**	1.22e+06
	7	1.17e+06	1.11e+06	1.08e+06	1.14e+06	1.07e+06	**1.05e+06**
	8	1.12e+06	976326	969249	943673	**921804**	937324
Spam	2	9.00e+07	9.00e+07	8.32e+07	8.07e+07	8.07e+07	**7.07e+07**
	3	5.39e+07	5.39e+07	3.0754e+07	4.17e+07	4.17e+07	**3.0753e+07**
	4	2.59e+07	2.52e+07	2.04e+07	2.40e+07	2.34e+07	**2.01e+07**
Pen digits	9	5.49e+06	5.32e+06	5.38e+06	5.35e+06	**5.28e+06**	5.33e+06
	10	5.03e+06	5.02e+06	4.988e+06	**4.95e+06**	5.04e+06	4.99e+06
	11	4.91e+06	4.73e+06	4.82e+06	4.70e+06	**4.69e+06**	4.75e+06
Letter	25	62001	**61791.9**	62543.7	65033.3	63663	63532.4
	26	61347.2	**60979.2**	61519.8	63274.1	64721.6	67876.8
	27	60509.6	**60053**	60386.5	62428.4	62173.2	61355

assigned to them. The hypothesis that the algorithms are equal is rejected if the value of Friedman statistic indicates a probability that is lower than the desired significance level.

Whenever the null-hypothesis is rejected, the alternative hypothesis is accepted, indicating that at least two algorithms are statistically different. In this case, the analysis continues to find out which pairs of algorithms are different. The Nemenyi test [19] is used to identify the difference between the algorithms. Every algorithm is compared to each other. It means that there is not an algorithm of reference to which the others should be compared. With this test, two algorithms are considered significantly different if the corresponding average ranks differ by the critical difference $CD = q_a\,[c(c+1)/6n]^{1/2}$, at least, where the critical values q_a are based on the t distribution divided by $2^{(1/2)}$.

In the statistical analysis we have only used the results obtained from each base when K is equal to the number of classes of data, because if more than one test were used on the same data set, even with different values of K, the problems would not represent independent samples, violating the preconditions of the statistical test. Moreover, once the algorithms are executed on the same subsets obtained by the cross-validation division, the experiments are considered dependent, satisfying the required restrictions of the Friedman test.

Table 3. Statistical comparison algorithms over multiple data sets using Friedman Test

	K-means			Simulated Annealing		
Data Set	Random	PCA-Part	K-means++	Random	PCA-Part	K-means++
Iris	6	5	4	2	2	2
Wine	6	4	5	3	1	2
Vehicle	6	3	5	2	1	4
Cloud	6	3	4	5	1	2
Segm.	6	4	3	5	2	1
Spam	5.5	5.5	4	2.5	2.5	1
Pen dig.	5	4	2	1	6	3
Letter	2	1	3	4	5	6
p.m.	5.3	3.7	3.8	3.1	2.6	2.6

Table 3 shows the ranks assigned by the Friedman test and the average of posts (in the last line of the table).

The Friedman test reports a probability of 2.04% that the null-hypothesis is true. This hypothesis may therefore be rejected with 5% significance level. The result of the test indicates that there is at least one pair of statistically different algorithms.

Considering the Nemenyi test, two algorithms are different if their average ranks differ by at least CD = $2.85[(6x7)/(6x8)]^{1/2} = 2.67$. Thus, the test reports that Simulated Annealing initialized with both PCA-Part and K-means++ are significantly better than the random K-means (5.3125 - 2.5625 = 2.75 > 2.67). Concerning the other algorithms, the analysis couldn't conclude whether there is a difference between them.

5 Conclusions and Future Work

We investigated the use of initialization methods, PCA-Part and K-means++, combined to Simulated Annealing to obtain better results for the clustering problem. The objective of this combination is to find initial solutions that are closer to the global optima, guiding the search algorithm to find the best solution according to the SSE objective function.

In order to analyze the performance of these combinations, experiments were performed in eight databases available in public repositories. The experimental evaluation indicated that the proposed approach performs better than K-means and the classical Simulated Annealing algorithm.

Statistical analysis showed that the Simulated Annealing algorithm initialized with the method PCA-Part and the method of K-means++ has a better performance than the random K-means.

As possible future work, we could extend the statistical analysis by conducting experiments with new bases. We could also try out other initialization methods and metaheuristics to verify if better results are obtained.

Simulated Annealing got bad results in the Letter data set (the largest one) in our experiments, presenting a different behavior of what we expected. Thus, another

future work is to analyze the reason of this result and verify if there is some feature of this data set that justifies this behavior.

In addition, the execution time of the algorithms should be analyzed in order to compare the time performance of K-means and Simulated Annealing algorithms. Although this result were not shown here, the experiments showed that K-means had a more efficient time performance than Simulated Annealing because it needs less iterations to reach the final solution.

Acknowledgments. Authors thank the financial support of Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (process 620165/2006-5).

References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM Computing Surveys 31, 264–323 (1999)
2. Forgy, E.W.: Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classifications. Biometrics 21, 768–780 (1965)
3. Selim, S.Z., Ismail, M.A.: K-means-type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality. IEEE Transaction on Pattern Analysis and Machine Intelligence 6, 81–87 (1984)
4. Rayward-Smith, V.J.: Metaheuristics for Clustering in KDD. In: IEEE Congress on Evolutionary Computation, vol. 3, pp. 2380–2387 (2005)
5. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science 220, 671–680 (1983)
6. Klein, R.W., Dubes, R.C.: Experiments in Projection and Clustering by Simulated Annealing. Pattern Recognition 22, 213–220 (1989)
7. Selim, S.Z., Alsultan, K.: A Simulated Annealing Algorithm for the Clustering Problem. Pattern Recognition 24, 1003–1008 (1991)
8. Murty, C.A., Chowdhury, N.: In Search of Optimal Clusters using Genetic Algorithms. Pattern Recognition Letter 17, 825–832 (1996)
9. Hall, L.O., Özyurt, I.B., Bezdek, J.C.: Clustering with a Genetically Optimized Approach. IEEE Transaction on Evolutionary Computation 3, 103–112 (1999)
10. Alsultan, K.: A Tabu Search Approach to the Clustering Problem. Pattern Recognition 28, 1443–1451 (1995)
11. Merwe, D.W., Engelbrecht, A.P.: Data Clustering using Particle Swarm Optimization. In: Congress on Evolutionary Computation, vol. 1, pp. 215–220 (2003)
12. Kanade, P.M., Hall, L.O.: Fuzzy Ants Clustering by Centroid Positioning. In: IEEE International Conference on Fuzzy Systems, vol. 1, pp. 371–376 (2004)
13. Babu, G.P., Murty, M.N.: Simulated Annealing for Optimal Initial Seed Selection in K-means Algorithm. Indian Journal of Pure and Applied Mathematics 3, 85–94 (1994)
14. Su, T., Dy, J.: A Deterministic Method for Initializing K-means Clustering. In: IEEE International Conference on Tools with Artificial Intelligence, pp. 784–786 (2004)
15. Arthur, D., Vassilvitskii, S.: K-means++: The Advantages of Careful Seeding. In: Symposium on Discrete Algorithms (2007)
16. Jolliffe, I.T.: Principal Component Analysis. Springer, New York (1986)

17. Friedman, M.: The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. Journal of the American Statistical Association 32, 675–701 (1937)
18. Friedman, M.: A Comparison of Alternative Tests of Significance for the Problems of m Rankings. Annals of Mathematical Statistics 11, 86–92 (1940)
19. Nemenyi, P.B.: Distribution-free Multiple Comparisons. PhD thesis. Princeton University (1963)