Improving penalized regression-based clustering model in big data

Sarah Ghanim Mahmood Al-kababchee¹, Omar Saber Qasim², Zakariya Yahya Algamal³

¹Department of Mathematics, Education College, University of AL-Hamdaniya
Email: sarahghanim@uohamdaniya.edu.iq

²Department of Mathematics, University of Mosul, Mosul, Iraq
E-mail: omar.saber@uomosul.edu.iq

³Department of Statistics and Informatics, University of Mosul, Mosul, Iraq
E-mail: zakariya.algamal@uomosul.edu.iq
ORCID: 0000-0002-0229-7958

Abstract: Clustering is the main procedure for data mining with a wide application such as gene analysis. Clustering is a method of separates (grouping) previously unclassified data on the basis of its features, and it is an unsupervised learning problem that divides data into groups in such a way that it makes those data in the same group more similar to each other compared to in other groups. Penalized regression-based clustering is an extension of the “Sum Of Norms” clustering model. In this paper, the nature-inspired algorithm is employed to improve the penalized regression-based clustering to better estimation. The real data application on gene expression data results suggests that our proposed improvement can bring significant improvement relative to others.

1. Introduction

One of the problems encountered clustering at scale is complicated works in data [1], pattern recognition process [2], image processing [3], and usual fields of sciences. The main goal of clustering data is to divide it into several clusters according to previously defined characteristics. Based on this clustering, the data are similar to each other to a large extent in the same cluster and not similar to other clusters.

Currently, known the clustering can be classified as a two parts: hierarchical clustering and partitional clustering [1]. One of the most widely known clustering algorithms is the center-based clustering algorithms. An algorithm K-means has widely used in recent years due to its simplicity and efficiency among these algorithms [4]. One of these algorithms is the Bat algorithm we used in our work, the bat algorithm helps us to increase the number of features for data by encoding that features to binary, that method is useful when we have many numbers of features and also it makes our computational less.
The "sum-of-norms" (SON) clustering method or clusterpath method [5-10], consider SON is a convex centroid based method. This method depends on over-parameterization which can control and the number of clusters by using a sum-of-norm regularization and the trade-off between the model fit. Clusterpath (SON) is a convex relaxation of hierarchical clustering and k-means [11]. To find a solution for the convex problem many algorithms have been proposed [5-10], for example, ADMM ("the alternating direction method of multipliers) and AMA (the alternating minimization algorithm") [10]. Particularly difficult to separate the non-convex groups because the son with the convex base produces estimates of biased parameters. Also, some clustering methods have the same difficulty, like k-means and k-means ++[12]. To decreasing this difficulty, it is suggested to use clustering based on the penalized regression which is considered to be an extension to SON[13].

2. Penalized clustering
Clustering has applied in many fields and is an unsupervised application, these include signal processing, social sciences, “marketing, economics, and business, as well as medicine and biology. The task of clustering is to create a cohesive grouping of patterns, data points, or objects. Suppose $D$ is the data set with $n$ number of objects (patterns) and each object is of $m$ dimensional. $D$ can be represented as $D = \{X_1, X_2, ..., X_n\}$, where $D \in \mathbb{R}^{n \times m}$. Then a clustering $C$ can be defined as follows: $C = \{K_1, K_2, ..., K_k\}$, such that the following conditions are satisfied

1- $K_i \neq \emptyset$ for $i = 1, ..., k$.
2- $K_i \cap K_j = \emptyset$ for $i \neq j, j = 1, ..., k$.
3- $\text{sim}(X_i, X_j) > \text{sim}(X_i, Y_j), X_i, X_j \in K_i \text{ and } Y_i \in K_j, i \neq j$.

To evaluate the results of clustering, it is possible to use several similarity measures ("Simple Matching Coefficient (SMC) and cosine similarity [15]) or dissimilarity measures (Minkowski distance [16], Manhattan distance [15] and Euclidean distance" [17]). The Euclidean distance has been used in this paper and defined as follows:

$$dis(X, Y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$$  \hspace{1cm} (1)

where $dis(X, Y)$ is the Euclidean distance between two $m$ dimensional objects $X$ and $Y$.

The similarity "between the objects in the cluster and the objects in the other clusters decreases when the similarity of the objects in the cluster increases. The goal of clustering is to reduce (minimize) similarity between inter-cluster or increase (maximize) similarity within inter-cluster. When the object is a data point, the similarity of the two points can be measured by calculating the distance between the two points. The smaller the sum of the distances between data points and their cluster centers (intra-cluster distance), or the greater the distance between clusters (inter-cluster distance) [18], the better the clustering effect. Therefore, the objective function can be shown as follows$^{19}$:

$$f = \sum_{j=1}^{k} \sum_{e \in K_j} dis(e, o_j)^2$$  \hspace{1cm} (2)

where $dis(e, o_j)$ the Euclidean distance between object $e$ and center of the $j^{th}$ cluster $o_j$ [19].

Let $\{x_1, x_2, ..., x_n\}$ be the data set which should be clustered such that between-group dissimilarity is maximized and the within-group similarity, where $x_i \in \mathbb{R}^m$. Suppose any data point $x_i$ has its has centroid $\mu_i$. The aim is to estimate $\mu_i$’s s.t $\mu_i$’s that their corresponding are equal from the same cluster.
\[
\min \sum_{i=1}^{N} L(x_i - \mu_i) + \lambda \sum_{i \neq j} h(\mu_i - \mu_j)
\]  
(3)

where \(L(.)\) represents a loss function, \(h(.)\) is a fusion penalty and \(\lambda > 0\) is a predefined parameter. This model for squared error (SE) and L1-norm fusion penalty is:

\[
\min \sum_{i=1}^{N} \|x_i - \mu_i\|^2 + \lambda \sum_{i \neq j} \|\mu_i - \mu_j\|_2
\]  
(4)

which is a convex SON clustering model [6].

Especially to get rid of difficulties separating due to producing biased parameter estimates from SON with convex loss function (LF) and convex fusion penalty, PRBC (penalty regression-based clustering), i.e. a group-truncated lasso penalty, was used by (Pan et al. 2013). As follows:

\[
\min \sum_{i=1}^{N} \|x_i - \mu_i\|^2 + \lambda \sum_{i \neq j} \min \{\|\mu_i - \mu_j\|_1, \tau\}
\]  
(5)

where \(\tau > 0\) is a predefined tuning parameter.

3. The proposed method

In big data, many redundant reflected the performance of clustering [20, 21]. As a result, selecting a small subset of related features from a big number of features is an important task for building a predictive clustering algorithm. Looking for the best subset of features is known to be an NP-hard problem where it requires a long time for computing linked with high cost.

In recent years many different algorithms, been suggested to get semioptimal subsets of the solutions, that designed by simulate natural evolution, these algorithms are GA (genetic algorithms), PSO (particle swarm optimization), ACO (ant colony optimization), and many others, [22, 23].

Bat algorithm proposed by Yang [24] is based on the echolocation ability of the microbats that guides them on their foraging behavior. The bat algorithm (BA) used three idealized rules: (i) All bats use echolocation to sense distance, and they also “know the difference between food/prey and back-ground barriers in some magical way. (ii) A bat roams randomly with a velocity \(v_i\) at a position \(x_i\) with a fixed frequency range \([f_{\text{min}}, f_{\text{max}}]\), varying its emission rate \(r \in (0,1)\) and loudness \(A_i\) to search for prey, depending on the proximity of its target. (iii) Although the loudness can vary in many ways, we assume that the loud-ness varies from a large (positive) \(A_i\) to a minimum constant value” \(A_{\text{min}}\) [25].

Bat algorithm starts with the random population in an n-dimensional , where the position of the bat \(i\) denoted by \(x_i\) and velocity is \(v_i\) at the time \(t\).[21, 26] Therefore, the new positions \(x_i^{t+1}\) and new velocities \(v_i^{t+1}\) at time step \(t+1\) can be determined by

\[
v_i^{t+1} = v_i^t + (x_i^t - G_{\text{best}})\delta_i^t,
\]  
(6)

\[
x_i^{t+1} = x_i^t + v_i^{t+1},
\]  
(7)

\[
\delta_i = \delta_{\text{min}} + (\delta_{\text{max}} - \delta_{\text{min}})\theta, r
\]  
(8)

where \(\theta\) is a random number in \([0, 1]\), \(G_{\text{best}}\) is the current global optimal solution, and \(\delta_i^t\) is the pulse frequency emitted by bat \(i\) at the current moment, where \(\delta_{\text{min}}\) the minimum values of pulse frequency and \(\delta_{\text{max}}\) the maximum values of pulse frequency. \(\delta_i^t\) is assigned
randomly at the beginning for each bat which is elected uniformly from $[\delta_{\text{min}}, \delta_{\text{max}}]$ [24, 27-30]. It’s possible to update the corresponding position of the random bat as

$$x_{\text{new}} = x_{\text{old}} + cL^t,$$  \hspace{1cm} (9)

where $x_{\text{old}}$ represents “a random solution chosen from the current best solutions[30], $L^t$ is the loudness, and $\epsilon$ is a random vector that is drawn from $[-1,1]$. The pulse emission rate and the loudness $L$ are updated by controlling the balance between these techniques as follows”:

$$r_{i,t}^{t+1} = r_0^t \times [1 - \exp(-b_1 t)],$$  \hspace{1cm} (10)

$$L_{i,t}^{t+1} = b_2 L_{i,t}^t,$$  \hspace{1cm} (11)

where $b_1$ and $b_2$ are constants.

In feature selection, “the binary bat algorithm (BBA) is used. Here, every bat we can represent by the p-bit binary string, where $p$ are several features in the dataset. To update the position, usually, the transfer function is used to force bat to fly in a binary space [31]. A transfer function in Eq. (7) can be used to build this binary vector, in which the new solution is constrained to only binary values”:

$$x_{i,t}^{t+1} = \begin{cases} 1 \quad \text{if } T(x) > \text{rand} \\ 0 \quad \text{otherwise} \end{cases},$$  \hspace{1cm} (12)

where $\text{rand} \in [0,1]$, $T(x)$ is transfer function for the velocity. Two transfer functions are used (S-shaped family & V-shaped) [32]. Our contribution is that performing feature selection using a binary bat algorithm and then the selected features will be used to perform clustering.

4. Experimental results

Our proposed algorithm, BPRBC, is compared with the PRBC and K-means method using three datasets. Table 1 presents the datasets, the features, and class for a chemical dataset. For each data set, the optimal values of the parameters $\lambda = 0.005$ and $\tau = 0.1$. Also, $\delta_{\text{min}} = 0$, $\delta_{\text{max}} = 2$, and $b_1 = b_2 = 0.9$ in BBA.

**Table 1: Description of the datasets used.**

| Dataset   | #Samples | #features | Class                          |
|-----------|----------|-----------|--------------------------------|
| Chalcone  | 212      | 3657      | 108 / 104 (active / inactive) compounds |
| Hepatitis | 121      | 2559      | 31 / 90 (active / inactive)    |
| H1N1      | 479      | 2322      | 213 / 266 (active / weakly active) |

Tables 2 and 3 compare our proposed algorithm, “BPRBC, with the PRBC and K-means method in terms of purity and computational time in seconds. As it can be observed from Table 3, BPRBC overtakes the standard PRBC. Moreover, it can be noticed that in all datasets, BPRBC obtains the highest purity with fewest selected features results compared with the others. The best results are bolded. For example, in H1N1, the improvement in purity using BPRBC was 6.02% and 11.16% of PRBC and K-means”.

As can be seen from Table 4, in terms of computational efficiency, the BPRBC has less time than PRBC. Consequently, it can be inferred that the BPRBC outperforms the K-means algorithm. In general, BPRBC, sinusoidal map, is the fastest among all the used algorithms on all dataset
Table 2: Purity results for each used algorithm

|          | BPRBC   | PRBC    | K-means |
|----------|---------|---------|---------|
| Chalcone | 0.9714  | 0.9059  | 0.8634  |
| Hepatitis| 0.9508  | 0.8917  | 0.8443  |
| H1N1     | 0.9584  | 0.9007  | 0.8514  |

Table 3: Computational time, in seconds, results for each used algorithm

|          | BPRBC | PRBC | K-means |
|----------|-------|------|---------|
| Chalcone | 57    | 67   | 94      |
| Hepatitis| 42    | 59   | 88      |
| H1N1     | 74    | 84   | 126     |

To further highlight the efficiency of our proposed algorithm, Table 4 shows the average purity for 20 times replications. As can be inferred that the BPRBC outperforms PRBC and K-means algorithms on all datasets. “The p-values (*) from Wilcoxon’s rank-sum test (nonparametric statistical test) with a 5% significance level are adopted. The statistical test is needed to indicate that the BPRBC provides a significant improvement compared to the other algorithms. It can be seen that there is a statistical difference between BPRBC and all the others for all datasets”.

Table 4: Average purity results for each used algorithm depending on 20-time replications

|          | BPRBC       | PRBC        | K-means       |
|----------|-------------|-------------|---------------|
| Chalcone | 0.9702 ± 0.005 | 0.9071 ± 0.011* | 0.8628 ± 0.022* |
| Hepatitis| 0.9514 ± 0.008 | 0.8926 ± 0.012* | 0.8457 ± 0.021* |
| H1N1     | 0.9573 ± 0.008 | 0.9019 ± 0.010* | 0.8532 ± 0.023* |

5. Conclusion

In order to determine the most useful features of clustering, we proposed this algorithm in this paper which is a binary bat algorithm with penalized regression-based clustering. Feature selection plays a fundamental and important role in developing a successful clustering algorithm. Via the results of statistical analysis and experimental results on the three groups of chemical data, the proposed BPRBC compared with the PRBC and K-means leads to a better in terms of performance time computational and purity.

References

1. Evangelou, I.E., et al., Data Mining and Knowledge Discovery in Complex Image Data using Artificial Neural Networks. Proceedings of the Workshop on Complex Reasoning and Geographical Data, Paphos, Cyprus, 2001.
2. Wu, C., et al., “A New Algorithm and Theory for Penalized Regression-based Clustering.pdf”. Journal of Machine Learning Research, 2016. 17: p. 1-25.
3. M.Omran, A.Salman, and A.P.Engelbrecht, Imageclassificationusingparticle swarm optimization. In Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, Singapore, 2002: p. 4.
4. Yan, X., et al., A new approach for data clustering using hybrid artificial bee colony algorithm. Neurocomputing, 2012. 97: p. 241-250.
5. A, P., et al., Clustering by sum of norms: stochastic incremental algorithm, convergence and cluster recovery. International conference on machine learning, 2017: p. 8.
6. K, P., et al., Convex clustering shrinkage. PASCAL workshop on statistics and optimization of clustering workshop, 2005.
7. F, L., O. H, and L L, Clustering using sum-of-norms regularization: with application to particle filter output computation. Statistical signal processing workshop (SSP) 2011 IEEE, 2011(3): p. 201–204.
8. TD, H., et al., Clusterpath an algorithm for clustering using convex fusion penalties. International conference on machine learning, 2011: p. 1.
9. GK, C., et al., Convex clustering: an attractive alternative to hierarchical clustering. PLoS Comput Biol 2015.
10. EC, C. and L K, Splitting methods for convex clustering. J Comput Graph Stat, 2015.
11. J, M., Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, 1967: p. 16.
12. D, A. and V. S, K-means ++: the advantages of careful seeding. In: Proceedings of the eighteenth annual ACMSIAM symposium on discrete algorithms. Society for Industrial and Applied Mathematics, 2007: p. 8.
13. W, P., S. X, and L B, Cluster analysis: unsupervised learning via supervised learning with a non-convex penalty. J Mach Learn Res, 2013. 14: p. 24.
14. Tasoulis, S., N.G. Pavlidis, and T. Roos, Nonlinear dimensionality reduction for clustering. Pattern Recognition, 2020. 107.
15. Xu, R. and D. Wunsch, Survey of clustering algorithms. IEEE Trans Cybern, 2005. 16: p. 33.
16. Gan, G., C. Ma, and J. Wu, Data Clustering Theory, Algorithms, and Applications. 2007: p. 489.
17. Mao, J. and A.K. Jain, A self-organizing network for hyperellipsoidal clustering (HEC). IEEE Transactions on Neural Networks, 1996. 7: p. 13.
18. GONZALEZ, T.F., CLUSTERING TO MINIMIZE THE MAXIMUM INTERCLUSTER DISTANCE. Theoretical Computer Science, 1985. 38: p. 293-306.
19. Chen, J., et al., Quantum-inspired ant lion optimized hybrid k-means for cluster analysis and intrusion detection. Knowledge-Based Systems, 2020. 203.
20. Algamal, Z.Y. and M.H. Lee, A two-stage sparse logistic regression for optimal gene selection in high-dimensional microarray data classification. Advances in Data Analysis and Classification, 2018. 13(3): p. 753-771.
21. Qasim, O.S., N.A. Al-Thanoon, and Z.Y. Algamal, Feature selection based on chaotic binary black hole algorithm for data classification. Chemometrics and Intelligent Laboratory Systems, 2020. 204.
22. Qasim, O.S. and Z.Y. Algamal, Feature selection using particle swarm optimization-based logistic regression model. Chemometrics and Intelligent Laboratory Systems, 2018. 182: p. 41-46.
23. Sayed, G.I., A. Darwish, and A.E.J.J.o.C. Hassani, Binary whale optimization algorithm and binary moth flame optimization with clustering algorithms for clinical breast cancer diagnoses. 2019: p. 1-31.
24. Yang, X.-S., A new metaheuristic bat-inspired algorithm, in Nature inspired cooperative strategies for optimization (NICS0 2010). 2010, Springer. p. 65-74.
25. Yang, X.-S. and M. Karamanoglu, *Swarm intelligence and bio-inspired computation: an overview*, in *Swarm intelligence and bio-inspired computation*. 2013, Elsevier. p. 3-23.

26. Huang, J. and Y.J.M.P.i.E. Ma, *Bat Algorithm Based on an Integration Strategy and Gaussian Distribution*. 2020.

27. Alomari, O.A., et al., *A novel gene selection method using modified MRMR and hybrid bat-inspired algorithm with β-hill climbing*. Applied Intelligence, 2018. 48(11): p. 4429-4447.

28. Rizk-Allah, R.M. and A.E. Hassanien, *New binary bat algorithm for solving 0–1 knapsack problem*. Complex & Intelligent Systems, 2017. 4(1): p. 31-53.

29. Zhao, D. and Y. He, *Chaotic binary bat algorithm for analog test point selection*. Analog Integrated Circuits and Signal Processing, 2015. 84(2): p. 201-214.

30. Qasim, O.S., Z.Y.J.I.J.o.M. Algamal, Engineering, and M. Sciences, *Feature Selection Using Different Transfer Functions for Binary Bat Algorithm*. 2020. 5(4): p. 697-706.

31. Mirjalili, S., S.M. Mirjalili, and X.-S. Yang, *Binary bat algorithm*. Neural Computing and Applications, 2014. 25(3-4): p. 663-681.

32. Mirjalili, S. and A. Lewis, *S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization*. Swarm and Evolutionary Computation, 2013. 9: p. 1-14.