International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18 (2026) pp. 1-10

https://doi.org/10.70917/ijcisim-2026-2361
Article

The Application of Cluster Analysis Algorithms for Student Assessment Data in Educational Quality Management

Yuanyuan Zhang [1,*]

[1] Krirk University, International College, Bangkok, 10220, Thailand
* Correspondence author: bjzyymm@yeah.net

Abstract: The implementation of teaching quality management plays a guiding role in a series of educational tasks, including teaching reform; selecting appropriate evaluation methods is crucial for the effective implementation of teaching quality management. This paper combines student grade data with an ant colony clustering algorithm, which establishes a "global memory bank of historical positions" to guide ants in purposefully and rapidly depositing the data objects they carry, while also effectively avoiding the occurrence of local optima. Finally, the ant colony clustering algorithm was applied to student evaluation and experimentally compared with the K-Means algorithm. The experimental results show that, compared to K-Means, the ant colony clustering algorithm improved the contour coefficient by 0.1009 (48.53%) and reduced the DBI index by 0.6823 (32.84%) in the clustering analysis of student evaluation data, significantly enhancing stability. Three-dimensional visualization further validated the clustering structure characterized by intra-cluster compactness and distinct inter-cluster separation. This method provides data support for precise student evaluation and differentiated instruction, demonstrating promising application prospects.

Keywords: Teaching quality management; Ant colony clustering algorithm; Student evaluations; Shape coefficient

1. Introduction

Student assessment data refers to the vast amounts of data generated by students on a daily basis during the teaching process, including the time taken to complete assignments, click patterns in online courses, fluctuations in test scores, and the frequency of classroom interactions [1–3]. This data holds the key to understanding "how students learn," and analyzing it is crucial for implementing effective educational quality management [4–5]. Traditional evaluation methods analyze and assess students' raw scores. Although these methods are simple, straightforward, and uniformly standardized, they rely on absolute scores and thus have certain limitations that hinder the objective and accurate assessment of students' learning progress [6–8].

Cluster analysis, a mathematical method for classifying objects according to specific criteria, offers a new approach to analyzing student evaluation data. It categorizes student evaluation data based on predetermined conditions, grouping data with similar or related characteristics into a single category, thereby transforming a large volume of seemingly unrelated data into several interconnected categories [9–11]. The application of clustering algorithms helps to understand the distribution of student evaluation data and the interrelationships among the data. Through these analytical results, factors influencing student learning can be identified, allowing teachers to adopt targeted strategies to improve the quality of teaching management [12–15].

The research in this paper focuses on using dynamic data clustering to implement a graded evaluation of student performance, and demonstrates the feasibility of this method through experiments using real-world data. This dynamic clustering approach helps mitigate the unfairness that arises from evaluating students based solely on scores. Subsequently, a novel ant colony clustering algorithm is proposed, which incorporates a "global memory bank of historical positions" mechanism to record the historical locations where ants have deposited data objects. Instead of moving randomly after each load of a data object, an ant first accesses this "memory bank" to search for similar data. This reduces the randomness of ant movement, thereby improving the convergence speed of the ant colony clustering algorithm and the accuracy of the clustering results. Finally, using learning behavior data from 324 students as the experimental dataset, the ant colony clustering algorithm was applied in practice and experimentally compared with the K-Means algorithm.

2. Related Technologies

2.1. K-Means Algorithm

Cluster analysis typically refers to the process of decomposing a dataset containing a large number of data objects into multiple clusters. During this partitioning process, we group data objects based on their similarity or a specific distance metric, placing those with high similarity into the same cluster and those with significant differences or low similarity into different clusters. Clustering aims to group all records into distinct clusters without prior knowledge of the number of classes in the database.

The K-Means algorithm is a classic partition-based clustering algorithm. Its core principle is as follows: given a specific dataset D and the number of clusters K (K is typically specified by the user), the K-Means algorithm uses an iterative process based on a distance function to partition all data points into K clusters. By default, a cluster center represents a class; this cluster center is defined as the mean of all data points within that cluster.

The K-Means algorithm minimizes the sum of squared errors of the distances from all samples in the clustering domain to their respective cluster centers through continuous iteration.

The cost function of the K-Means algorithm:

$$J(k,u) = \sum_{i=1}^{n} \left\| X^{(i)} - \mu_k^{(i)} \right\|^2 \tag{1}$$

Based on an understanding of the algorithm, the K-Means algorithm can be summarized as follows:

(1) Suppose the sample size is n; randomly select k data points to serve as the initial cluster centers $(z_1, z_2, z_3, \ldots, z_k)$ for the clustering process;

(2) Find a cluster center z_v for the data sample x_i such that the cluster center is closest to x_i, and then assign x_i to the specific category u_v identified by the cluster center;

(3) Calculate the new cluster centers for each category after the cluster analysis;

(4) Calculate $D = \sum_{i=1}^{n} \left[\min_{1 \le r \le k} d(x_i, z_r)^2 \right]$;

(5) If the D value converges, return $(z_1, z_2, z_3, \ldots, z_k, U)$ and terminate the algorithm; otherwise, proceed to (2).

When using this algorithm for cluster analysis, a drawback is the need to manually specify the number of clusters k, in advance. If the pre-set value of k is unreasonable, this can directly lead to poor analysis results and reduced effectiveness. Additionally, the algorithm is particularly susceptible to the influence of outliers or extreme values in the sample, which can significantly impact performance. Furthermore, if the dataset is large and complex, different choices of initial cluster centers can result in significantly varying clustering outcomes, ultimately leading to unstable analysis results. However, if the clustering structure of the research subjects is such that members within a cluster are closely related while those across clusters are distinct, this algorithm yields relatively good clustering results.

2.2. Cluster Analysis Based on the Ant Colony Algorithm

2.2.1. Ant Colony Optimization Algorithm

Swarm intelligence-based clustering algorithms originated from research on the sorting of ant eggs by ant colonies. Scientists discovered that ants are capable of piling their dead bodies together and sorting their larvae. Inspired by this, clustering analysis methods based on ant colony algorithms were subsequently developed.

Ant colony-based clustering methods can be categorized into the following four types based on their principles:

(1) Models based on ant foraging behavior, which use pheromones to cluster objects;

(2) Models based on ant self-aggregation behavior, which cluster objects;

(3) Models based on the formation of ant mounds, which are used to cluster data;

(4) Methods that utilize the principles of ant nest classification models and rely on the ants' chemical recognition system to achieve clustering.

Each of the four methods described above is suitable for specific application scenarios.

The basic idea is to randomly distribute the data to be clustered across a two-dimensional plane, and then randomly distribute the virtual ants within this space as well. At the same time, the ants move randomly within this data space. When an ant encounters a data point to be clustered during its movement, it picks up the data and continues to move randomly across the plane while carrying it. If the similarity between the data near the ant's path and the data it is carrying exceeds a predefined threshold, the ant will deposit the data at that location. It then continues to move randomly, repeating the process of picking up and depositing data. Through this iterative process, similar data is ultimately clustered, yielding the clustering results.

The basic process of the ant colony-based clustering algorithm is described as follows:

(1) Initialize each individual ant in the colony. Let N denote the number of ants in the colony, M denote the maximum number of iterations for the entire algorithm, s denote the side length of the local region used for similarity calculations, and specify other parameter values required for the ant colony clustering algorithm;

(2) Project all data objects to be clustered onto a plane within a given range; that is, randomly assign each data object to a point on this two-dimensional plane with coordinates (x, y);

(3) Initially, each ant carries no data objects; now, it randomly selects a data object from the plane;

(4) For i =1, 2, …, M; For j =1, 2, …, N;

Calculate the average similarity of the data objects:

Building upon the ant colony clustering algorithm, this paper introduces a "global history database" mechanism. Specifically, when an ant is not carrying a load, the probability that it will pick up a data object is calculated and denoted as P_p. If the probability of picking up an object P_p exceeds a certain random threshold, and at the same time the data object has not been picked up by any other ant, then this ant picks up the data object and marks itself as loaded.

It then moves randomly to another location within the defined two-dimensional plane; if the object's probability P_p does not meet the condition, the ant will refuse to pick up that data object and will instead randomly select another data object in the area to repeat the process. When an ant is already carrying a data object, it must calculate the probability of dropping it, which we assume to be P_d.

If the probability of dropping the data object P_d exceeds a certain threshold, the ant will drop the data object. At the same time, it will mark its status as unloaded and then select a new data object; if the probability of dropping the data object does not meet the condition, the ant will carry the data object and continue moving randomly to a new location.

(5) If an object is isolated, or if the number of objects in its neighborhood is less than a specified constant, then that object is an isolated point and must be marked; otherwise, a cluster ID must be assigned to that object, and the same ID must be recursively assigned to the objects in its neighborhood as a marker.

2.2.2. Calculation of Average Similarity

Suppose that at a certain time t, if an ant finds a data object labeled o_i at location r, the average similarity between this object o_i and its neighboring object labeled o_j is calculated using the following formula:

$$f\left(O_i\right) = \max\left\{0, \frac{1}{S^2}\sum o_{j \in Neigh_{sxs}}(r)\left[1-\frac{d\left(o_i,o_j\right)}{a\left(1-(v-1)/v_{\max}\right)}\right]\right\} \tag{2}$$

In the equation, a represents the similarity parameter, v represents the ant's movement speed, $V_{\max}$ represents the ant's maximum movement speed, $Neigh_{sxs}(r)$ represents the local square region with side length s surrounding location r, and $d\left(o_i,o_j\right)$ represents the distance between objects o_i and o_j in the attribute space.

Here, the distance between objects is calculated using either Euclidean distance or cosine distance. The specific meaning is expressed based on a linear combination of the two. Euclidean distance is a method for calculating geometric distances in multidimensional space. The definition of Euclidean distance is shown in the following formula:

$$d\left(o_i,o_j\right)=\sqrt{\sum_{k=1}^{m}\left(o_{ik}-o_{jk}\right)^2} \tag{3}$$

In the formula, m represents the number of attributes possessed by objects o_i and o_j.

For the definition of the cosine distance, see the formula:

$$d\left(o_i,o_j\right)=1-\mathrm{sim}\left(o_i,o_j\right) \tag{4}$$

The function $sim(o_i,o_j)$ in Equation (4) is defined as:

$$sim\left(o_i,o_j\right)=\frac{\sum_{k=1}^{m}\left(o_{ik}\bullet o_{jk}\right)}{\sqrt{\sum_{k=1}^{m}\left(o_{ik}\right)^2\bullet\sum_{k=1}^{m}\left(o_{jk}\right)^2}} \tag{5}$$

$sim\left(o_i,o_j\right)$ function is a cosine similarity function, which is defined as the cosine of the angle between two vectors (i.e., the dot product of the two vectors divided by their magnitude). The closer $sim\left(o_i,o_j\right)$ value is to 1, the more similar the two data objects are; conversely, $sim\left(o_i,o_j\right)$ value approaching 0 indicates that the objects are very different. In the definition of Equation (2), a is a parameter that adjusts the degree of similarity among data objects; this value also determines the final number of clusters and the convergence speed of the clustering process. The larger the value of a is set, the greater the similarity between objects, which can result in objects with significant differences being grouped into the same cluster. When the number of clusters is smaller, the algorithm converges more quickly; Conversely, the smaller the value of a, the lower the similarity between data objects. In some extreme cases, this may result in a large category being divided into numerous smaller subcategories. Since the number of clusters increases as a result, the convergence rate slows down.

Therefore, during the actual calculation process, these parameter values must be determined reasonably based on the specific problem through empirical observation or preliminary calculations. Additionally, the ants' movement speed on the plane also affects the final clustering results. If the ants move quickly, they will quickly but roughly divide the data objects into several broad categories; if they move slowly, they can subdivide the data objects more precisely. Therefore, based on differences in the ants' movement speed (denoted as v), there are three distinct speed definitions:

(1) When v is set to a constant; in this case, all ants move at the same speed.

(2) When v is set to a random number, the ants' speeds are within a certain range, specifically a random number between 1 and .

(3) When v is set to a decreasing random number, the ants initially move at a faster speed to achieve rapid clustering; subsequently, the speed values gradually decrease in a random manner, resulting in a more refined clustering outcome.

2.2.3. Calculation of Probability Transformation Functions

The so-called probability transformation function is actually a function of $f\left(o_i\right)$; this function converts the calculation of the average similarity of data objects into the calculation of the probability of picking up or putting down an object. The principle behind this transformation is as follows: the lower the average similarity between a data object and its neighborhood, the lower the probability that the data object belongs to that neighborhood; therefore, the probability of picking up the object is higher, and the probability of putting it down is lower. The reverse is also true. Based on this principle, the symmetric sigmoid function is selected as the probability transformation function. The probability P_p that a randomly moving ant without a load picks up an object is defined by the following formula:

$$P_p=1-Sigmoid\left(f\left(o_i\right)\right) \tag{6}$$

Let P_d denote the probability that a randomly moving ant carrying a data object drops the object; it is defined as:

$$P_d=Sigmoid\left(f\left(o_i\right)\right) \tag{7}$$

In particular, the $Sigmoid(f(o_i))$ formula is defined as follows:

$$Sigmoid(x) = \frac{1-e^{-cx}}{1+e^{-cx}} \tag{8}$$

The sigmoid function takes the form of a natural exponential. As can be seen from the formula, the larger the parameter c is, the faster the curve saturates, which in turn causes the algorithm to converge more quickly.

3. A Student Evaluation Model Based on the Ant Colony Clustering Algorithm

Drawing on the principles of the ant colony clustering algorithm, this paper proposes a student evaluation model based on the ant colony clustering algorithm. The results are shown in Figure 1.

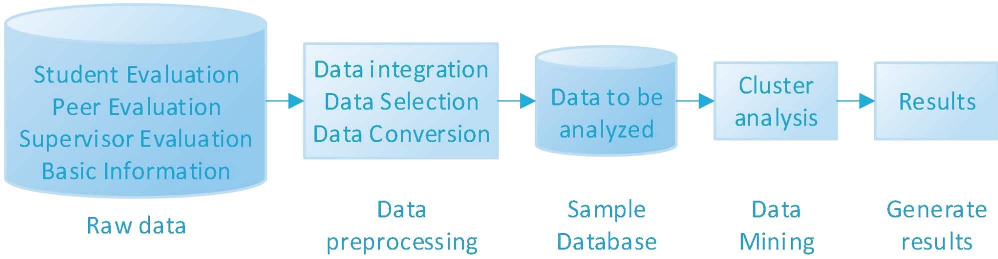


Figure 1. Student evaluation Model based on ant colony clustering algorithm

3.1. Raw data

The raw data primarily consists of evaluations of teachers by supervisors, evaluations by students, evaluations by peers, and basic teacher information. After understanding the clustering task and the content of the raw data, we identify useful features within the data and reduce the data volume as much as possible while preserving the data's original form, so that the data's patterns and underlying characteristics become more apparent. Choosing the wrong method for processing the raw data will not only fail to yield the correct prototypes but also result in incorrect clustering.

3.2. Data Preprocessing

Data preprocessing is a critical step in cluster analysis; the quality of the student evaluation results directly depends on whether the experimental data has been properly handled. Appropriate data preprocessing helps improve the accuracy and performance of the subsequent clustering process. Data preprocessing is primarily divided into three categories: data integration, data selection, and data transformation.

(1) Data integration primarily involves combining data from multiple sources into a unified data repository, processing duplicate data, and removing redundant attributes irrelevant to the current mining task, thereby improving the accuracy and speed of the mining process. Student and faculty data required for educational evaluation are often stored in disparate databases. This data is not always accurate, and data obtained through practical applications frequently contains inevitable issues such as incompleteness, inconsistency, inaccuracy, and duplication. When importing data into a database, it must be cleaned to remove noise and ensure the data is valid, complete, and consistent.

(2) Data selection primarily involves identifying the data sets required for analysis under the guidance of relevant domain and expert knowledge, narrowing the scope of processing, and improving the effectiveness and quality of cluster analysis.

(3) Data transformation primarily involves operations such as smoothing, aggregation, data generalization, normalization, and attribute construction. The purpose is to convert the data into a form suitable for clustering. Since sample data may not be suitable for direct cluster analysis, it must be transformed before use.

To improve the accuracy of the ant colony clustering algorithm, the parameter data to be clustered can be normalized to the interval [0,1]. Let $\max_A$ be the maximum value of a certain attribute A in the data, and $\min_A$ be the minimum value of that attribute. Using the following formula, the value v of attribute A for a given data point is mapped to a value v' in the interval [0,1], as follows:

$$v' = \frac{v - \min_A}{\max_A - \min_A} \tag{9}$$

By applying the ant colony clustering algorithm to the raw data, the data preprocessing step yields the cluster centers and the number of clusters. At the same time, this process also removes outliers from the raw data; these points will not be included in the subsequent data processing steps.

3.3. Sample Database

After data preprocessing—including data cleaning, noise removal, and normalization—the sample database contains data that has been specially processed and stored in a uniform format. This data can be used directly for cluster analysis using clustering tools.

3.4. Student Feedback

Student evaluation is at the core of the model. During the student evaluation phase, the primary task is to select an appropriate evaluation algorithm to analyze the selected data, thereby completing the data clustering task or achieving the objectives of student evaluation. There are many methods for student evaluation, and clustering is one of the most common techniques. Clustering techniques aim to identify commonalities and differences within a dataset and group tuples with common characteristics into corresponding clusters.

This paper proposes the Ant Colony Clustering algorithm to analyze sample data and uncover patterns relevant to students. For example: the classroom behavioral characteristics of each student, and the classroom learning characteristics of each student group (based on different grades, learning behaviors, etc.). As data collection becomes more refined and accumulates over time, the model can perform even deeper levels of data analysis.

3.5. Generated Results

The results of the cluster analysis are presented in an easily understandable format, such as graphs and charts. On the one hand, these results enable teachers and administrators to analyze data more intuitively, thereby identifying the key factors that truly influence student learning behavior during instruction and providing a reference for schools to allocate teaching duties in a scientific and reasonable manner. On the other hand, based on the selected analytical data, corresponding decision-making rules are derived and presented to decision-makers for their reference, thereby enabling efficient and precise management of educational quality.

4. Experimental Design and Analysis of Results

4.1. Data Source

This paper uses learning behavior data from 324 students in a specific major at the School of Science, H University, during the fall semester of 2024 as the experimental dataset. The evaluation metrics include summative grades and formative grades (covering classroom performance, graded assignments, and midterm exams). The dataset is complete with no missing values and requires no preprocessing. In accordance with the syllabus for the “Mathematical Modeling” course, students’ scores for each evaluation metric are calculated using a weighted average of 60% for summative grades and 40% for formative grades. Prior to inputting the data into the ant colony clustering algorithm, it was standardized using z-scores to eliminate differences in measurement scales and improve the accuracy and stability of the clustering.

4.2. Algorithm Parameter Settings

The experiment employed K-Means and Ant Colony Clustering algorithms for cluster analysis, comparing their performance and validating the results. K-Means algorithm settings: initial number of clusters K=3, convergence tolerance 1e-1; Ant Colony Clustering algorithm settings: initial number of clusters K=3, target number of ants per colony 100, inertia weight ω =0.6788, individual learning factor c_1 =1.526, and population learning factor c_2 =1.385. Both algorithms were set to cluster into 3 clusters, each run independently 10 times with a maximum of 100 iterations to ensure reliable results and stable convergence.

The C-index and DBI index were selected to evaluate the clustering performance of the K-Means and Ant Colony clustering algorithms. The C-index measures the cohesion of a sample with its cluster and the separation from its nearest neighbor clusters; its range is [-1, 1], and a value closer to 1 indicates better clustering performance. The DBI index is used to comprehensively measure both intra-cluster compactness and inter-cluster separation; a smaller value indicates a clearer clustering structure and better classification performance. Each student's category is determined based on their maximum membership score within each cluster, thereby enabling a tiered evaluation based on the clustering results. By conducting a quantitative analysis of the clustering results using the aforementioned metrics, the effectiveness of the proposed algorithm is validated, providing a scientific basis for student evaluation and classification.

4.3. Analysis of Experimental Results

4.3.1. Choosing the number of clusters K

The K-means clustering algorithm requires the number of clusters, K, to be specified in advance; however, when dealing with high-dimensional datasets, it is difficult to determine the most appropriate value of K simply by visual inspection. Therefore, before performing clustering with the K-means algorithm, we can first use a hierarchical clustering algorithm to perform "pre-clustering," which helps determine the appropriate value of K and the initial cluster centers. Based on the above method, we first performed hierarchical clustering on the grade data of the 324 students. More specifically, we used Ward's hierarchical clustering method to determine the appropriate number of clusters. The similarity metric used was Euclidean distance. The results of the hierarchical clustering are shown in Figures 2 and 3.

Based on the results of the hierarchical clustering and the corresponding ESS curve, it can be observed that the ESS curve decreases significantly when the number of clusters ranges from 3 to 6, indicating that dividing the student population into 3 to 6 subgroups is a reasonable approach. If the number of subgroups is too small, the clustering results tend to group high-performing students into one category and low-performing students into another. Such results are rather uninformative and do not facilitate the analysis of students' learning characteristics. Conversely, if the number of subgroups is too large, the sample size within each cluster becomes too small, making it difficult to identify common characteristics among students within the same group. As for which cluster number within the 3–6 range is optimal, this must be determined by performing the clustering calculation and selecting the optimal number of clusters based on the results.

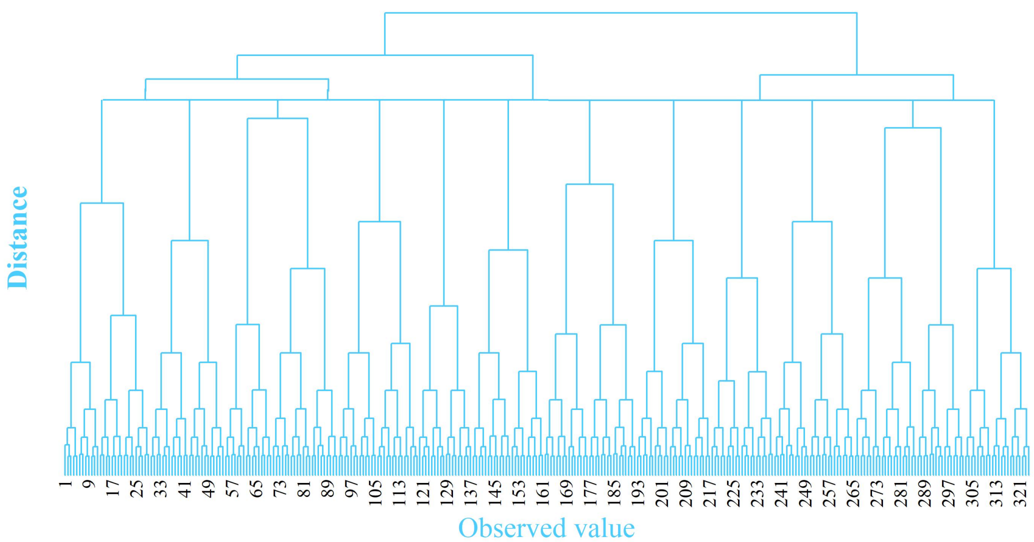


Figure 2. Hierarchical clustering results of the grades of 324 students

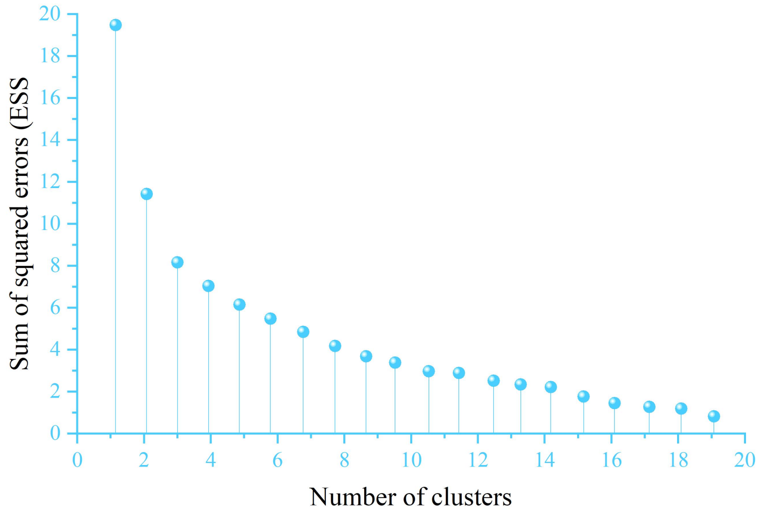


Figure 3. ESS Curve of hierarchical clustering

4.3.2. Comparison of Performance Metrics

After completing the multi-level clustering experiments, this paper systematically compared the clustering performance of the K-Means and Ant Colony clustering algorithms based on two metrics: the silhouette coefficient and the DBI index. The relevant results are shown in Table 1. The experimental results indicate that, under identical parameter settings, the Ant Colony Optimization (ACO) algorithm generally demonstrates superior clustering performance in student evaluations: its C-coefficient (0.3088) is consistently higher than that of K-Means (0.2079), indicating that the clustering results exhibit stronger intra-class consistency and inter-class separability; simultaneously, the DBI index is lower overall (1.3955), further validating the compactness and clarity of its clustering structure.

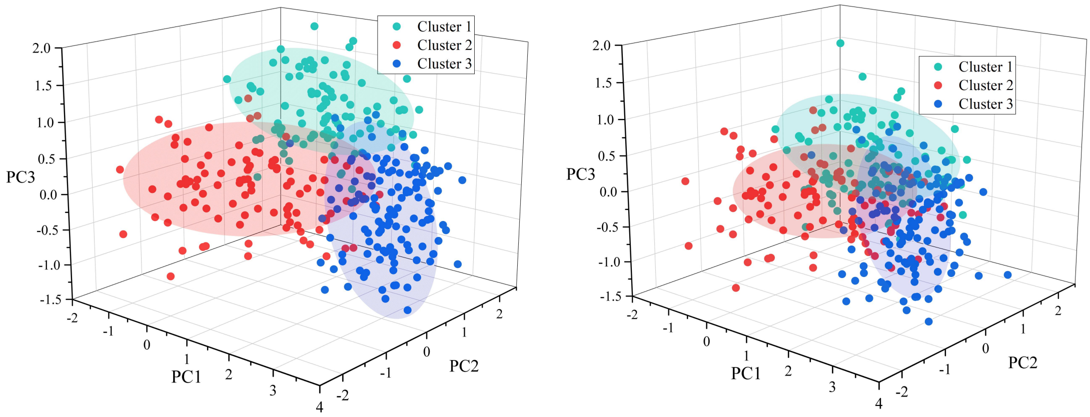
Table 1. Results of Clustering Evaluation Indicators

N	Contour coefficient		DBI Index	
	K-Means	**Ant colony clustering**	**K-Means**	**Ant colony clustering**
1	0.1834	0.2111	1.7997	1.3977
2	0.1637	0.326	2.0114	1.381
3	0.2477	0.2956	1.5421	1.4137
4	0.2315	0.3284	2.3138	1.3029
5	0.2352	0.4017	2.3879	1.4765
6	0.2289	0.2997	1.9063	1.4069
7	0.2469	0.3614	2.1114	1.4192
8	0.1303	0.3263	2.2329	1.4473
9	0.2086	0.2752	2.1908	1.4004
10	0.2029	0.2622	2.2812	1.3094
Ave	0.2079	0.3088	2.0778	1.3955

In addition, the results of multiple K-Means clustering runs showed significant fluctuations across various metrics, indicating a certain degree of instability; in contrast, the distribution of clustering evaluation metrics for the ant colony clustering algorithm was more concentrated across multiple rounds of experiments, with a smaller standard deviation, suggesting that it exhibits greater robustness in finding the global optimal solution. These results further validate the effectiveness of swarm intelligence algorithms in clustering, particularly in avoiding local optima and reducing sensitivity to initial cluster centers.

4.3.3. Visualization of Results

To provide a more intuitive demonstration of the clustering performance of the K-Means and Ant Colony clustering algorithms on student learning behavior data, this paper conducts a three-dimensional visualization analysis of the results with the optimal clustering evaluation metrics from the experiments. First, Principal Component Analysis (PCA) is used to perform dimensionality reduction on the original high-dimensional data, mapping the data into a space defined by the first three principal components; Subsequently, a cluster distribution map of student sample points was plotted in the principal component space to observe the structural relationships and boundary clarity of clusters under different algorithms. The visualization results are shown in Figure 4. The three-dimensional visualization results indicate that, under the Ant Colony Clustering algorithm, student sample points exhibit a distinct cluster-like distribution in the principal component space, with cluster centers highly aligned with the core positions of various student groups. In contrast, the distribution of sample points under the K-Means algorithm exhibits significant overlap, with blurred cluster boundaries. This visually validates the advantage of the student performance evaluation model based on the ant colony clustering algorithm in terms of the rationality of the clustering structure.



(a) Results of ant colony clustering (b) K-Means clustering results

Figure 4.·Clustering Effects of the two algorithms on student learning behavior data e

In summary, the ant colony clustering algorithm demonstrates superior clustering performance and greater stability when applied to student grade data, effectively improving clustering accuracy and the reliability of results. This method provides a theoretical foundation for student evaluation based on clustering analysis algorithms, as well as a data foundation and practical reference for the development of personalized teaching and tiered intervention strategies.

5. Conclusion

Student evaluation is an effective means of improving teachers' instructional skills and educational quality. For schools, establishing a scientific student evaluation system serves as an effective mechanism for educational quality management, as well as a primary method for collecting, processing, and analyzing teaching data. Addressing the issues of insufficient accuracy and robustness in traditional student evaluation methods when handling ambiguous, multidimensional data, this paper constructs a student evaluation model based on the ant colony clustering algorithm, guided by the principles of student evaluation system design. By introducing a "global memory bank of historical positions" mechanism, this method effectively overcomes the shortcomings of K-Means clustering—such as sensitivity to initial cluster centers and a tendency to get stuck in local optima—significantly enhancing the stability and accuracy of the clustering process.

Experimental results indicate that the ant colony clustering algorithm outperforms the traditional K-Means clustering algorithm across clustering evaluation metrics such as the contour coefficient and DBI index, while exhibiting lower performance fluctuations and greater robustness; Combined with Principal Component Analysis (PCA) for dimensionality reduction and 3D visualization, the rationality and validity of the clustering structure were further validated.

In summary, the ant colony clustering algorithm provides reliable technical support for student evaluation and the optimal allocation of educational resources. Future research will further enrich the evaluation indicator system and expand the model's applicability and promotional value across diverse educational environments.

References

1. Smithson, J., Birks, M., Harrison, G., Nair, C. S., & Hitchins, M. (2015). Benchmarking for the effective use of student evaluation data. Quality Assurance in Education, 23(1), 20-29.

2. Fisher, C., Berliner, D., Filby, N., Marliave, R., Cahen, L., & Dishaw, M. (2015). Teaching behaviors, academic learning time, and student achievement: An overview. The Journal of Classroom Interaction, 50(1), 6-24.
3. Ghani, A. S. A., Rahim, A. F. A., Yusoff, M. S. B., & Hadie, S. N. H. (2021). Effective learning behavior in problem-based learning: a scoping review. Medical science educator, 31(3), 1199-1211.
4. Kurdi, M. M., Al-Khafagi, H., & Elzein, I. (2018, November). Mining Educational Data to Analyze Students' Behavior and Performance. In 2018 JCCO Joint International Conference on ICT in Education and Training, International Conference on Computing in Arabic, and International Conference on Geocomputing (JCCO: TICET-ICCA-GECO) (pp. 1-5). IEEE.
5. Wasif, M., Waheed, H., Aljohani, N. R., & Hassan, S. U. (2019). Understanding student learning behavior and predicting their performance. In Cognitive computing in technology-enhanced learning (pp. 1-28). IGI Global.
6. Firestone, W. A., & Donaldson, M. L. (2019). Teacher evaluation as data use: what recent research suggests. Educational Assessment, Evaluation and Accountability, 31(3), 289-314.
7. Aggarwal, C. C. (2018). An introduction to cluster analysis. In Data clustering (pp. 1-28). chapman and hall/crc.
8. Jaeger, A., & Banks, D. (2023). Cluster analysis: A modern statistical review. Wiley Interdisciplinary Reviews: Computational Statistics, 15(3), e1597.
9. Dalmaijer, E. S., Nord, C. L., & Astle, D. E. (2022). Statistical power for cluster analysis. BMC bioinformatics, 23(1), 205.
10. Brusco, M. J., Singh, R., Cradit, J. D., & Steinley, D. (2017). Cluster analysis in empirical OM research: survey and recommendations. International Journal of Operations & Production Management, 37(3), 300-320.
11. Köhn, H. F., & Hubert, L. J. (2014). Hierarchical cluster analysis. Wiley StatsRef: statistics reference online, 1-13.
12. Battaglia, O. R., Di Paola, B., & Fazio, C. (2016). A new approach to investigate students' behavior by using cluster analysis as an unsupervised methodology in the field of education. Applied Mathematics, 7(15), 1649-1673.
13. Nen-Fu, H., Hsu, I., Chia-An, L., Hsiang-Chun, C., Jian-Wei, T., & Tung-Te, F. (2018, September). The Clustering Analysis System Based on Students' Motivation and Learning Behavior. In 2018 Learning With MOOCS (LWMOOCS) (pp. 117-119). IEEE.
14. Labrović, J. A., Petrović, N., Anđelković, J., & Meršnik, M. (2025). Patterns of behavioral engagement in an online English language course: cluster analysis. Journal of Computing in Higher Education, 37(1), 1-26.
15. Wilson, D., Wright, J., & Summers, L. (2021). Mapping patterns of student engagement using cluster analysis. Journal for STEM Education Research, 4(2), 217-239.