Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2021, Article ID 8899649, 14 pages
https://doi.org/10.1155/2021/8899649

Research Article

On Characterization of Norm-Referenced Achievement Grading Schemes toward Explainability and Selectability

Thepparit Banditwattanawong[1] **and Masawee Masdisornchote**[2]

[1]*Department of Computer Science, Faculty of Science, Kasetsart University, Bangkok 10900, Thailand*
[2]*School of Information Technology, Sripatum University, Bangkok 10900, Thailand*

Correspondence should be addressed to Thepparit Banditwattanawong; debharit@hotmail.com

Received 7 August 2020; Revised 26 December 2020; Accepted 8 February 2021; Published 20 February 2021

Academic Editor: Christian Dawson

Grading is the process of interpreting learning competence to inform learners and instructors of the current learning ability levels and necessary improvement. For norm-referenced grading, the instructors use a conventionally statistical method, z score. It is difficult for such a method to achieve explainable grade discrimination to resolve dispute between learners and instructors. To solve such difficulty, this paper proposes a simple and efficient algorithm for explainable norm-referenced grading. Moreover, the rise of artificial intelligence nowadays makes machine learning techniques attractive to the norm-referenced grading in general. This paper also investigates two popular clustering methods, K-means and partitioning around medoids. The experiment relied on the data sets of various score distributions and a metric, namely, Davies–Bouldin index. The comparative evaluation reveals that our algorithm overall outperforms the other three methods and is appropriate for all kinds of data sets in almost all cases. Our findings however lead to a practically useful guideline for the selection of appropriate grading methods including both clustering methods and z score.

1. Introduction

In both formal and informal education, grading is the process of interpreting learning competence to inform learners and instructors of current learning ability levels and necessary improvement. There are basically two types of nonbinary grading systems [1]: criterion-referenced grading and norm-referenced grading. The former normally calculates the percentage of a learning score and maps it to the predefined percent range of a specific grade. This grading system is suitable for an examination that covers all content topics of learning and thus requires long exam-taking as well as answer-checking times. In contrast, large classes and/or large courses widely use the norm-referenced grading system to meet exam-taking time constraints and to save exam-answer-checking resources. Such a system compares the score of each individual to relative criteria defined based on all individuals' scores to determine a proper grade. The criteria are set by a conventionally statistical means either without or with conditions (e.g., a class's grade point average (GPA) must be kept below 3.25).

This paper focuses on the unconditionally norm-referenced grading. The type of problem that the paper targets is data clustering where its difficulty is that the reasons behind cluster boundaries must be explainable as the first priority. A concrete problem is norm-referenced grading while its difficulty is that how to make learners whose scores are contiguously ranked accept their different grades (i.e., their scores fell in different cluster boundaries) with no doubt. To our experiences, this classical problem has long made graders seriously reluctant to resolve dispute with learners. Let us consider the following example to comprehend such a situation: given a simplified series of ranked scores . . ., 84, 80, 78, . . ., performing the norm-referenced grading on such a score series by using a traditional method may result in grades . . ., A, B, B, . . ., respectively. The learner who scores 80 can make an objection to why he or she receives B rather than A. It is not only difficult for grader to explain the entire

steps of the traditional method (which is complicated) but also difficult for the learner to understand. Our algorithm provides a simple and clear-cut justification based on the widest score gaps: "because 80 is closer to 78 than to 84 so 80 should be assigned the same performance level as 78 rather than 84."

The rise of artificial intelligence nowadays makes machine learning techniques attractive to the norm-referenced grading. We therefore investigate an opportunity to exclusively adopt four methods from the realm of statistical and machine learning: our novel algorithm, a conventionally statistical method, and two unsupervised machine learning techniques, namely, K-means and Partitioning around medoids (PAM) (aka K-medoids). We selected the unsupervised learning techniques since the norm-referenced grading cannot have a training data set. In particular, we selected K-means and PAM as they are the only well-known clustering algorithms that allow us to specify the number of output clusters to represent the desired number of grades (as specified by an employed grading policy). Therefore, both K-means and PAM are naturally applicable to the norm-referenced grading. The grading results of each approach will be measured and compared based on the practical data sets of various distribution characteristics.

The main contributions of this paper are a simple and efficient grading algorithm and a novel insight into the performance of statistical method, machine learning methods, and our algorithm in unconditionally norm-referenced grading. To the best of our knowledge, we also demonstrate for the first time the applicability of K-means and PAM clustering techniques for norm-referenced grading. The merit of this paper would help worldwide graders with the selection of the right grading method to meet their objectives.

The rest of this paper is organized as follows. Section 2 explores previously existing research studies. Section 3 explains the z score grading method. Section 4 reviews machine learning techniques, which includes K-means and PAM, applicable to norm-referenced grading. Section 5 explains our proposed grading algorithm. Section 6 justifies a grading performance metric in terms of clustering quality. Section 7 experiments our algorithm, z score, K-means, and PAM methods based on normal and asymmetric distribution data sets. Section 8 discusses the main findings. Section 9 draws the conclusion.

2. Related Work

As for applying a machine learning clustering technique to learners' achievement, Arora and Badal [2] analyzed the competency of students by using K-means. The competency is attributed by 10-subject marks. The centroid of each cluster was mapped to one of the grade symbols A to G. The resulting grade of each cluster was the competency indicator of students belonging to such a cluster. Academic planners could use such an indicator to take appropriate action to remedy the students. Similarly, Borgavakar and Shrivastava [3] clustered GPAs and internal class assessments (e.g., class test marks, lab performance, assignment, quiz, and attendance) separately by using K-means. Therefore, each student's competency was associated with several clusters, which were used to create a set of rules for classifying the student. Any weak students were identified before the final exam to reduce the ratio of fail students. Research by Parveen et al. [4] employed K-means to create 9 groups of GPAs: exceptional, excellent, superior, very good, above average, good, high pass, pass, and fail. Students whose GPAs belonged to the exceptional and the fail groups were called gifted and dunce, respectively. The gifted students were enhanced of their knowledge, whereas the dunce students were remedied through differentiated instruction. Research by Shankar et al. [5] clustered students from different countries based on their attributes: average grade, the number of participated events, the number of active days, and the number of attended chapters. An optimal k value of K-means was determined by means of the Silhouette index resulting in $k = 3$. Among the 3 clusters, the most compact cluster (i.e., a cluster with the least value of within-cluster sum of square) was further analyzed for correlation between the average grade and the other attributes. Xi [6] utilized K-means to cluster students' test scores into 4 classes, excellent, good, moderate, and underachiever, to take the appropriate self-development and teaching strategy for treatment. Research by Iqbal et al. [7] explored several machine learning techniques for early grade prediction to allow instructors to improve students' competency in early stages. In such work, Restricted Boltzmann Machine was found to be most accurate for students' grade prediction. K-means was also used to cluster students based on technical course and nontechnical course performance.

Regarding an automated grading and scoring approach, Ramen and Joachims [8] proposed a peer grading method to enable student evaluation at scale by having students assess each other. Since students are not trained in grading, the method enlisted probabilistic models and ordinal peer feedback to solve a rank aggregation problem. Bai and Chen [9] proposed a method to automatically construct grade membership functions, lenient-type grades, strict-type grades, and normal-type grades, to perform fuzzy reasoning to infer students' scores.

This paper significantly extends our immature work [10] with a full-fledged algorithm, a newly practical data set, a newly experimented machine learning method, a set of new findings, and a novel guideline for method selection.

3. Conventionally Statistical Grading

A conventionally statistical grading method relies on z scores and t scores [1]. z score is a measure of how many standard deviations below or above the population mean a raw score is. z score (z) is technically defined in (1) as the signed fractional number of standard deviations (σ) by which the value of an observation or a data point x is above the mean value (μ) of what is being observed or measured.

$$Z = \frac{x - \mu}{\sigma}. \tag{1}$$

Observed values above the mean have positive z scores, otherwise, negative z scores.

The t score converts individual scores into standard forms and is much like z score when the sample size is above 30. In psychometrics, t score (t) is a z score shifted and scaled to have a mean of 50 and a standard deviation of 10 as in (2).

$$t = 10 * Z + 50. \quad (2)$$

The statistical grading method begins by converting raw scores to z scores. The z scores are further converted to t scores to simplify interpretation because t scores normally range from 0 to 100, unlike z scores that can be negative real numbers. The t scores are then sorted and a range between maximum and minimum t scores is divided by the desired number of grades to obtain an identical score interval. The interval is used to define the t score ranges of all grades. In this way, raw scores can be mapped to z scores, the z scores to t scores, the t scores to t score intervals, and the t score intervals to resulting grades, respectively.

4. Machine Learning-Based Grading

This section explains how to apply K-means and PAM clustering algorithms to the norm-referenced grading, which is natural to unsupervised learning rather than supervised one. K-means and PAM were selected since both allow specifying the number of clusters in advance to match the number of eligible grades known a priori.

4.1. K-Means. K-means [11] is an unsupervised machine learning technique for partitioning n objects into k clusters. K-means begins by randomizing k centroids, one for each cluster. Assign every object to a cluster whose centroid is nearest to the object. Recalculate the means of all assigned objects within each cluster to serve as k new centroids aka barycenters of the clusters. Iterate the object assigned to the clusters and the centroid recalculation until no more object moves between clusters. In other words, the K-means algorithm aims at minimizing an objective function $\sum_{j=1}^{k} \sum_{i=1}^{n_j} |x_i - c_j|$, where n_j is the number of objects in cluster j, $x_i = <x_{i1}, x_{i2}, \ldots, x_{im}>$ is an object in cluster j whose centroid is c_j, x_{i1} to x_{im} are the features of x_i, and $|x_i - c_j|$ is Euclidean distance. Also, note that the initial centroid randomization can result in different final clusters.

When applying the K-means algorithm to higher educational grading, k is set to the number of eligible grades. Graders must decide such a number in advance.

4.2. Partitioning around Medoids. Unlike K-means representing each cluster with the mean value of objects within clusters, PAM [12] represents each cluster by one of the objects nearest to the cluster's center. PAM proceeds in two phases. In the first phase, build, select k objects nearest to the center of all other unselected objects. Such k objects called medoids are selected one by one. In the second phase, swap, assign all unselected objects to their nearest medoids to obtain k initial clusters. For each cluster, calculate average dissimilarity (i.e., average distance) between a medoid and the other objects. Then, for such a cluster, search whether any object if it became a new medoid minimizes the average dissimilarity. If it does, select such an object as a new medoid. Once all clusters have been searched and if at least one medoid has changed, repeat the second phase; otherwise, PAM ends.

Similar to applying K-means, PAM requires that k be set to the number of eligible grade symbols beforehand.

5. Proposed Grading Algorithm

This section proposes a statistical algorithm for norm-referenced unconditional grading. The algorithm works step by step as defined in Algorithm 1.

The algorithm is explained as follows. In line 1, sort(S) initially ranks the scores of learners within a group from the best down to the worst. In line 2, countEligibleGrades(GS) counts the number of eligible grades. In line 3, calculateAllScoreGaps(S) sequentially goes through the score ranked list to straightforwardly determine a gap between every two contiguous scores (i.e., a score difference). Line 4 sorts the gaps in a descending order. In line 5, selectWidestGaps(SG, cnt—1) selects a set of maximum gaps that equal the number of eligible grades minus one. For instance, four eligible grades require four score ranges; thus, selectWidestGaps(SG, cnt—1) function returns the first three maximum gaps. In case that some gaps are identical, the gaps of scores that are closest to the middle of the score rank will be returned by the function. In line 6, defineScoreRangesFromGaps(SG) creates a series of score ranges, each of which is associated with each eligible grade. For instance, the score range of grade B is 76 to 82 points. Finally, grades(S, R) in line 7 completely assigns proper grades to all scores based on the defined score ranges. In this way, our algorithm is simple while its performance will be proved in Section 7.

As for the cost effectiveness of the proposed algorithm, we analyze its computational complexity as follows. Let n be the number of scores to be graded (i.e., $|S|$). In the worst case, sort(S) in line 1 finishes in $n\log_2 n$, countEligibleGrades(GS) takes $|GS|$, calculateAllScoreGaps(S) takes n to do all subtractions between every consecutive scores, descendingSort(SG) takes $n/2\log_2 n/2$, selectWidestGaps(SG, cnt—1) takes $|GS|$—1, defineScoreRangesFromGaps(SG) takes $|GS|$, and grades(S, R) takes n. Therefore, the algorithm takes at most $n\log_2 n + |GS| + n + n/2\log_2 n/2 + (|GS|—1) + |GS| + n$. Suppose that n is much greater than $|GS|$, thus our algorithm $= O(n\log_2 n)$, which is relatively tractable.

Remark that our algorithm gets only two input parameters, the learners' scores and the eligible grades while the local variables of the algorithm are used for temporary value assignment rather than as controlling parameters. Also, all of the called functions in our algorithm perform straightforward tasks as implied by their names without any tuning parameters. Therefore, our algorithm keeps users away from the parameter tuning burden.

```
Input
    S: vector of learners' scores
    GS: set of ranked eligible grade symbols
Output
    G: vector of learners' grades
Local variable
    cnt: number of eligible grades
    SG: vector of score gaps
    R: vector of score ranges
Begin
(1)  S ← sort(S);
(2)  cnt ← countEligibleGrades(GS);
(3)  SG ← calculateAllScoreGaps(S);
(4)  SG ← descendingSort(SG);
(5)  SG ← selectWidestGaps(SG, cnt – 1);
(6)  R ← defineScoreRangesFromGaps(SG);
(7)  G ← grades(S, R);
End
```

Algorithm 1: Proposed algorithm.

6. Grading Performance Measurement

In this paper, the performance of each grading method is represented with clustering quality. The quality of clustering results can be measured by using a well-known metric namely Davies–Bouldin index (DBI). We employed DBI instead of another related metric, Silhouette, because DBI is much computationally less complex; thus, it is highly readable by practical graders. Let us denote by δ_j the mean intracluster distance of the n_j points (each of which is expressed as x_i) belonging to cluster C_j to their barycenter c_j: $\delta_j = (1/n_j)(\sum_{i=1}^{n_j} |x_i - c_j|)$. Let us also denote a distance between barycenters $c_{j'}$ and c_j of clusters $C_{j'}$ and C_j by $\Delta_{jj'} = |c_{j'} - c_j|$. DBI is figured out by using (3) [13]. The lower DBI, the better quality of clustering results (i.e., low DBI clusters have low intracluster distances and high intercluster distances).

$$\mathrm{DBI} = \frac{1}{k}\sum_{j=1}^{k} \max_{\forall j' \in \{1,..,k\} \wedge j' \neq j} \left(\frac{\delta_j + \delta_{j'}}{\Delta_{jj'}}\right). \tag{3}$$

The underlying reason for using DBI as the grading performance metric in norm-referenced grading is intuitive as follows. Learners with much similar achievement should receive the same grade (i.e., equivalent to low intracluster distances), and different grades must be able to discriminate achievements between the groups of learners as much clearly as possible (i.e., equivalent to high intercluster distances). DBI value will be low (i.e., better grading performance result) if clusters are compact and far away from one another.

7. Evaluation

We evaluated our algorithm, z score method, K-means, and PAM in norm-referenced unconditional grading. Experimental configuration and data sets' characteristics are initially described. Then, grading results along with performance metrics are provided.

7.1. Experimental Configuration. A grading policy that evaluated the scores into 5 eligible grades, A, B, C, D, and F, without any class GPA constraint was engaged. The grading policy was implemented in 4 ways by using our algorithm, z score, K-means, and PAM methods. The number of clusters was predefined to 5 (i.e., the 5 eligible grades) in K-means and PAM. Each method had its performance measured in DBI metric as if the grades represented distinct clusters.

The six data sets of accumulative term scores were used to ensure fair comparison among the grading methods. We characterized the data sets through data distribution in order to verify their coverage of all possible distribution patterns (i.e., the representativeness of various case studies). In particular, the data distribution patterns that were employed included normal distribution (ND data set in Table 1) and positively and negatively skewed distributions (SD+ and SD− data sets in Tables 2 and 3). The algorithm's effectiveness was also double-checked by using two additional data sets, slightly positively and negatively skewed distributions (RD+ and RD− data sets in Tables 4 and 5). Last but not least, the other rare data set with an exclusively wide score gap (WD data set in Table 6) was also exploited. The scores relied on a scale of 0.0 to 100.0 points. A one-dimensional vector was used to represent each data set as shown in Tables 1–6 so that readers could dive deep into the scores to judge the effectiveness of each applied method. Every data set is also described in the term of statistic along with its distribution pattern.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\left((1/2)((x-\mu)/\sigma)^2\right)}. \tag{4}$$

The first data set, namely, ND, has a normal distribution. Table 1 shows the raw scores of ND. Mean and median are 63. Mode is unavailable as every score has the same frequency of 1. σ is 13.9.

Table 1: Sorted scores of ND data set.

Record#	Score
1	88
2	86
3	84
4	79
5	78
6	77
7	76
8	75
9	74
10	73
11	72
12	67
13	66
14	65
15	64
16	63
17	62
18	61
19	60
20	59
21	54
22	53
23	52
24	51
25	50
26	49
27	48
28	47
29	42
30	40
31	38

Table 2: Sorted scores of SD + data set.

Record#	Score
1	92
2	90
3	89
4	86
5	77
6	74
7	73
8	73
9	73
10	65
11	62
12	61
13	60
14	54
15	53
16	53
17	53
18	52
19	52
20	52
21	52
22	52
23	51
24	51
25	51
26	51
27	50
28	50
29	46
30	46
31	45

To comprehend the characteristics of ND, Figure 1 projects its normal distribution. The horizontal axis represents z score. The curve was computed with (4) where x represents a score. The area under the curve represents a distribution value [1].

The second and the third data sets have positively and negatively skewed distributions namely SD+ and SD−, respectively. Positively skewed distribution is an asymmetric bell shape skewed to the left probably caused by overly difficult exam questions from the viewpoint of learners. Table 2 shows the raw scores of SD + set. Mode, median, mean, and σ are 52, 60.9, 53, and 14.236 respectively. Figure 2 depicts the normal distribution of SD + set. Its skewness is heavy and equals 1.006.

Negatively skewed distribution is an asymmetric bell shape skewed to the right probably caused by too easy exam questions from the viewpoint of learners. Table 3 shows the raw scores of SD−. Mode, median, mean, and σ equal 87, 82, 73.5, and 16.929, respectively. Figure 3 depicts the normal distribution of SD−. The skewness is as heavily as −1.078.

These 3 data sets contain the same number of raw scores and were realistically synthesized to clarify the extreme behaviors of the four studied methods.

The fourth data set RD− was collected from a group of real 61 anonymized learners taking the same undergrad course in the academic year 2019. Unlike SD+ and SD− that are heavily skewed, RD− (and RD+) represents imperfectly normal distributions (i.e., slightly skewed). RD− in Table 4 has the slightly negative skew of −0.138 as shown in Figure 4. Mode, median, mean, and σ equal 66.7, 56.6, 57.9, and 12.136 respectively.

The fifth data set, RD+, was the real term scores of the other group of 100 anonymized learners from another anonymized university. Opposite to RD−, RD + has the slightly positive skew of 0.155. The characteristics of RD + are shown in Table 5 and Figure 5. Mode, median, mean, and σ equal 82.5, 66.4, 65.7, and 9.662, respectively.

The last data set, WD, consists of the broad range of scores with a relatively wide gap. Such a score pattern exists in the group of learners with a learning competency divide. As a result, some enclosed grade ought to be skipped. The characteristics of WD are shown in Table 5. A significant gap lies between the scores 79 and 30 as depicted in Figure 6. Mode, median, mean, and σ equal 87, 82, 62.3, and 31.975, respectively. WD has the moderately negative skew of −0.450.

7.2. Grading Result. We graded ND data set by using the proposed algorithm, z score, K-means, and PAM methods and reported their results, respectively, in angle brackets:

> < our-algorithm grade, z score grade, K-means grade, PAM grade >

Table 3: Sorted scores of SD− data set.

Record#	Score
1	94
2	93
3	87
4	87
5	87
6	87
7	86
8	86
9	86
10	85
11	85
12	85
13	84
14	84
15	83
16	82
17	77
18	75
19	74
20	73
21	72
22	65
23	64
24	63
25	62
26	61
27	52
28	50
29	38
30	36
31	34

Table 4: Sorted scores of RD− data set.

Record#	Score
1	80.8
2	80.2
3	78.7
4	76.8
5	76.1
6	75.2
7	75.1
8	72.5
9	72.1
10	71.6
11	70.8
12	70.6
13	69.1
14	68.7
15	68
16	67.6
17	66.7
18	66.7
19	65.8
20	63.5
21	61.6
22	61.5
23	61.4
24	60.7

Table 4: Continued.

Record#	Score
25	60.5
26	59.2
27	58.7
28	58.5
29	57.8
30	57.4
31	56.6
32	55.7
33	55.5
34	55.5
35	55.2
36	55.2
37	55.1
38	54.7
39	53.9
40	52.6
41	52.5
42	51.7
43	51.3
44	51
45	50.7
46	50
47	48.8
48	48.7
49	48.6
50	46.7
51	46.4
52	46.2
53	45
54	44.9
55	44.6
56	44.5
57	43.5
58	42
59	35.7
60	28.4
61	28

shown in Table 7 resulting in an Nx4 matrix where N rows equal a number of scores. Our algorithm delivered exactly the same results as those of K-means. Both methods' DBIs equaled 0.330. z score method yielded the equivalent DBI of 0.443. It might be questionable from student viewpoint why graders using z score gave learners who scored 78 and 79 the same grades A as that of 84 and 47 mark holder the same grade F as that of 42 marks. These are simply because 78 and 79 fell in the same z score interval of A while 47 fell in the z score interval of F. PAM also yielded the DBI of 0.330 despite too many grades A.

We also graded SD+ data set with our algorithm, z score, K-means, and PAM methods as shown in Table 8. Our algorithm delivered the same results as K-means and PAM. Their DBIs were 0.222. z score method gave the equivalent DBI of 0.575. There were many grades F when using z score method.

Next, we graded SD− data set in Table 9. Our algorithm delivered the equivalent DBI of 0.299. The DBIs of z score, K-means, and PAM methods were equally 0.233.

Table 5: Sorted scores of RD + data set.

Record#	Score
1	89.47
2	87.1
3	82.73
4	82.53
5	82.53
6	82.17
7	80.7
8	80.5
9	79.97
10	79.43
11	79.3
12	78.9
13	78.47
14	78.27
15	77.87
16	77.87
17	75.73
18	74.57
19	73.3
20	73.2
21	73.1
22	72.83
23	72.63
24	72.1
25	71.83
26	71.77
27	70.8
28	70.4
29	70.23
30	70.2
31	70.2
32	69.43
33	69.17
34	69.17
35	69.1
36	68.77
37	68.6
383	68.27
9	67.87
40	67.77
41	67.63
42	67.63
43	67.57
44	67.33
45	67.1
46	67
47	66.77
48	66.73
49	66.4
50	66.37
51	66.37
52	66.1
53	65.87
54	65.8
55	64.77
56	64.73
57	64.73
58	64.57
59	64.57
60	64.3

Table 5: Continued.

Record#	Score
61	64.17
62	64.13
63	63.93
64	63.9
65	63.57
66	63
67	62.83
68	60.63
69	60.33
70	59.83
71	58.93
72	58.87
73	58.53
74	58.47
75	58.27
76	57.53
77	57
78	56.77
79	55
80	54.8
81	54.57
82	54.5
83	54.5
84	54.43
85	54.37
86	53.8
87	53.73
88	53.37
89	53.37
90	52.87
91	52.47
92	52.1
93	52
94	51.97
95	51.8
96	50.9
97	50.7
98	50.2
99	50.1
100	45

In practice, there is no perfectly normal distribution with respect to learners' achievement. Now experimental results based on data sets having slightly skewed distributions are described. We graded RD data set with our algorithm, z score, K-means, and PAM methods as shown in Table 10. The gap columns show differences between every two consecutive scores (i.e., the results of calculateAllScoreGaps() function in Algorithm (1) to be utilized by our algorithm where 4 widest gaps (indicated by the bold numbers) were used as grading steps.

All four methods produced different grading results. Particularly, our algorithm and K-means assigned A for the same group of learners whereas z score and K-means methods gave F to the same group of learners. Our algorithm had the DBI of 0.375 whereas K-means, PAM, and z score method gave the equivalent DBIs of 0.469, 0.474, and 0.492,

Table 6: Sorted scores of WD data set.

Record#	Score
1	98
2	97
3	93
4	92
5	91
6	90
7	89
8	87
9	87
10	86
11	85
12	85
13	84
14	83
15	83
16	82
17	81
18	81
19	79
20	30
21	28
22	27
23	26
24	25
25	23
26	22
27	21
28	21
29	20
30	18
31	17

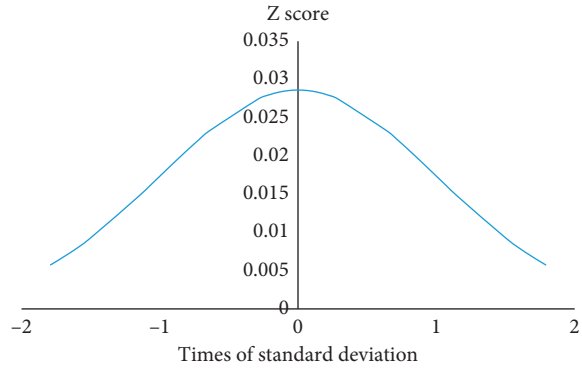

Figure 1: Distribution of ND data set.

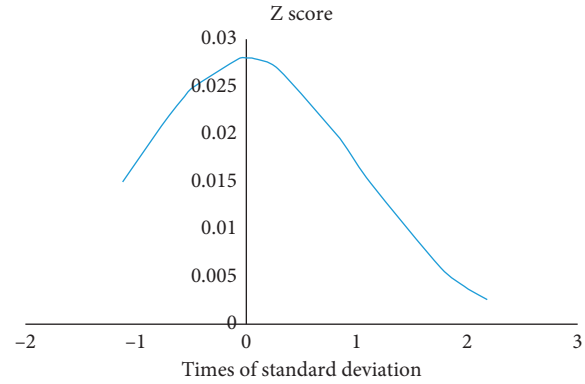

Figure 2: Distribution of SD + data set.

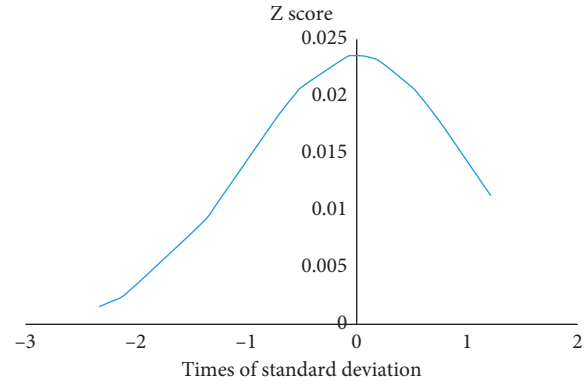

Figure 3: Distribution of SD−.

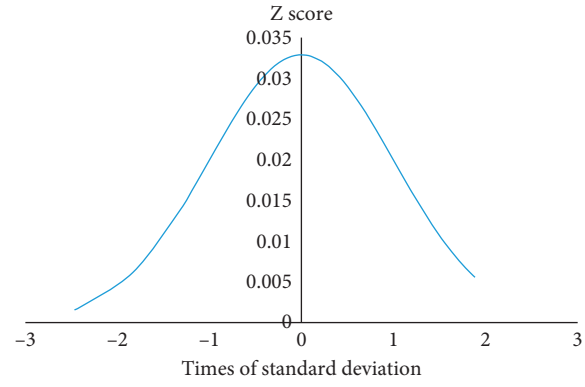

Figure 4: Distribution of RD−.

respectively. Therefore, our algorithm delivered the best grading results for RD−-. Our algorithm accomplished the lowest DBI partly because grade D has only one member score, comparable to the smallest possible cluster, which DBI favors.

We graded RD + data set as shown in Table 11. With this large data set, the grading results of all methods are totally different. Our algorithm, z score method, K-means, and PAM methods yielded DBIs of 0.345, 0.529, 0.486, and 0.487, respectively, meaning that our algorithm defeated the others.

WD data set was graded as shown in Table 12. Our algorithm, z score method, K-means, and PAM yielded DBIs of 0.403, 0.452, 0.449, and 0.449, respectively. Although our algorithm outperformed the others in terms of DBI, recall that WD data set had the exceptional pattern of so significant gap that assigning 5 grades completely may not be plausible. As shown in Table 12, only z score method is capable of automatically skipping grades C and D.

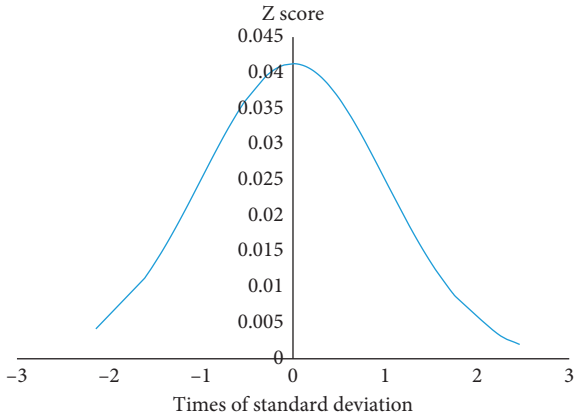

Figure 5: Distribution of RD+.

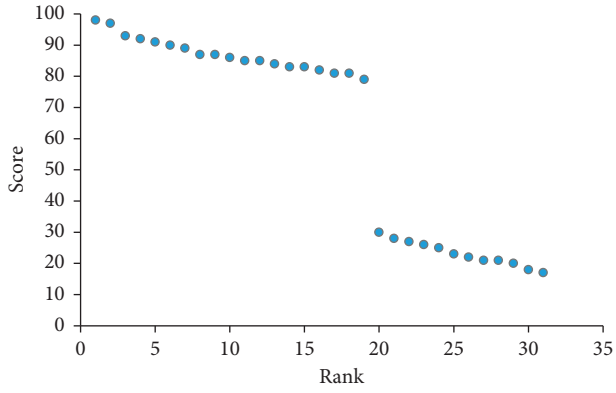

Figure 6: Divide in WD data set.

Table 7: Grading results of ND.

Score	Grade
88	<A, A, A, A>
86	<A, A, A, A>
84	<A, A, A, A>
79	<B, A, B, A>
78	<B, A, B, A>
77	<B, B, B, A>
76	<B, B, B, A>
75	<B, B, B, A>
74	<B, B, B, A>
73	<B, B, B, A>
72	<B, B, B, A>
67	<C, C, C, A>
66	<C, C, C, A>
65	<C, C, C, A>
64	<C, C, C, A>
63	<D, D, D, A>
62	<D, D, D, A>
61	<C, C, C, A>
60	<C, C, C, A>
59	<C, C, C, A>
54	<C, C, C, B>
53	<C, C, C, B>
52	<D, D, D, B>
51	<D, D, D, B>
50	<D, D, D, B>
49	<D, D, D, C>
48	<D, D, D, D>
47	<D, F, D, D>
42	<F, F, F, F>
40	<F, F, F, F>
38	<F, F, F, F>

8. Result Analysis, Finding, and Discussion

Figure 7 comparatively projects all aforementioned DBIs with respect to each grading method and data set. They can be analyzed as follows. Our algorithm has DBIs' $\mu = 0.329$ and $\sigma = 0.058$. z score has DBIs' $\mu = 0.454$ and $\sigma = 0.109$. K-means' DBIs have $\mu = 0.365$ and $\sigma = 0.109$. PAM′ DBIs have $\mu = 0.366$ and $\sigma = 0.110$. The overall performance of each method is revealed in Figure 8. As the lower DBI the better clustering quality, the heights of stacks show that our algorithm performs best due to the lowest overall DBI whereas K-means and PAM produce underneath performance results by 10.90% and 11.21% higher DBIs than ours, respectively. z score method performs worst, 38.03% greater DBI than ours. These relative performance differences show the practical significance of our algorithm.

We also conducted paired (Student's) t-test to evaluate whether the means of our algorithm's DBI are statistically significantly different from those of the other methods. Particularly, paired t-test was employed to compare DBI means produced by our algorithm with those of z score, K-means, and PAM methods for the 6 data sets. We used the standard significance level of 0.05 and the hypothesized mean difference of 0 (i.e., null hypothesis value indicating no DBI difference between methods) to figure out p value for one-tailed t-test. A smaller p value means that there is stronger evidence in favor of an alternative hypothesis (i.e., there is DBI difference between methods). Firstly, DBI difference between our algorithm and z score had the p value of 0.040, which was less than 0.050. Therefore, our algorithm outperformed z score with statistical significance. Secondly, DBI difference between our algorithm and K-means had the p value of 0.144. Lastly, DBI difference between our algorithm and PAM had the p value of 0.141. Therefore, our algorithm outperformed K-means and PAM without statistical significance. Note that, unlike the practical significance, the statistical significance only provides evidence that performance differences exist since it is a mathematical definition that does not know anything about our subject area.

Table 8: Grading results of SD+.

Score	Grade
92	<A, A, A, A>
90	<A, A, A, A>
89	<A, A, A, A>
86	<A, A, A, A>
77	<B, B, B, B>
74	<B, B, B, B>
73	<B, C, B, B>
73	<B, C, B, B>
73	<B, C, B, B>
65	<C, C, C, C>
61	<C, D, C, C>
61	<C, D, C, C>
60	<C, D, C, C>
54	<D, F, D, D>
53	<D, F, D, D>
53	<D, F, D, D>
53	<D, F, D, D>
52	<D, F, D, D>
52	<D, F, D, D>
52	<D, F, D, D>
52	<D, F, D, D>
52	<D, F, D, D>
51	<D, F, D, D>
51	<D, F, D, D>
51	<D, F, D, D>
51	<D, F, D, D>
50	<D, F, D, D>
50	<D, F, D, D>
46	<F, F, F, F>
46	<F, F, F, F>
45	<F, F, F, F>

Table 9: Grading results of SD−.

Score	Grade
94	<A, A, A, A>
93	<A, A, A, A>
87	<B, A, A, A>
87	<B, A, A, A>
87	<B, A, A, A>
87	<B, A, A, A>
86	<B, A, A, A>
86	<B, A, A, A>
86	<B, A, A, A>
85	<B, A, A, A>
85	<B, A, A, A>
85	<B, A, A, A>
84	<B, A, A, A>
84	<B, A, A, A>
83	<B, A, A, A>
82	<B, A, A, A>
77	<B, B, B, B>
75	<B, B, B, B>
74	<B, B, B, B>
73	<B, B, B, B>
72	<B, B, B, B>
65	<C, C, C, C>
64	<C, C, C, C>
63	<C, C, C, C>
62	<C, C, C, C>
61	<C, C, C, C>
52	<D, D, D, D>
50	<D, D, D, D>
38	<F, F, F, F>
36	<F, F, F, F>
34	<F, F, F, F>

Our algorithm and K-means lead to fairly similar grading results based on normal and heavily-positively skewed distributions. Furthermore, by examining Tables 7–12, PAM produced the most A and the least F by average.

The behavior of our proposed algorithm can be discussed in terms of the definition of (3) as follows. The algorithm performed clustering effectively in almost all cases of data sets (i.e., ND, SD+, RD−, RD+, and WD) because Algorithm 1 always selects the maximum score gaps to draw cluster boundaries, that is, maximum $\Delta_{jj'}$. Although the algorithm does not deal with the minimization of δ_j, it usually has less impact on DBI than $\Delta_{jj'}$ since δ_j takes part in the summation (thus requiring the minimization of the other term $\delta_{j'}$), whereas $\Delta_{jj'}$ is the sole divider in (3). Nevertheless, in an exceptional case, merely maximizing $\Delta_{jj'}$ is not enough as substantiated by our algorithm that performed worst when clustering SD− data set.

Key findings based on the result analysis are provided as follows. In general, Figure 7 reveals that the absolute degree rather than the positive or negative polar of skewness has more impacts on the methods' grading performance: the greater the absolute skewness, the lower the grading performance. This is because the greater absolute skewness implies more dispersed or dissimilar scores.

Table 10: Grading results of RD−.

Score	Gap	Grade
80.8	—	<A,A,A,A>
80.2	0.6	<A,A,A,A>
78.7	1.5	<A,A,A,A>
76.8	1.9	<A,A,A,A>
76.1	0.7	<A,A,A,A>
75.2	0.9	<A,A,A,A>
75.1	0.1	<A,A,A,A>
72.5	**2.6**	<B,A,B,A>
72.1	0.4	<B,A,B,A>
71.6	0.5	<B,A,B,A>
70.8	0.8	<B,A,B,A>
70.6	0.2	<B,A,B,A>
69.1	1.5	<B,B,B,A>
68.7	0.4	<B,B,B,A>
68	0.7	<B,B,B,A>
67.6	0.4	<B,B,B,A>
66.7	0.9	<B,B,B,A>
66.7	0	<B,B,B,A>
65.8	0.9	<B,B,B,A>
63.5	**2.3**	<C,B,B,A>
61.6	1.9	<C,B,C,A>
61.5	0.1	<C,B,C,A>
61.4	0.1	<C,B,C,A>
60.7	0.7	<C,B,C,A>

Table 10: Continued.

Score	Gap	Grade
60.5	0.2	<C,B,C,A>
59.2	1.3	<C,C,C,A>
58.7	0.5	<C,C,C,A>
58.5	0.2	<C,C,C,A>
57.8	0.7	<C,C,C,A>
57.4	0.4	<C,C,C,A>
56.6	0.8	<C,C,C,A>
55.7	0.9	<C,C,C,A>
55.5	0.2	<C,C,C,A>
55.5	0	<C,C,C,A>
55.2	0.3	<C,C,C,A>
55.2	0	<C,C,C,A>
55.1	0.1	<C,C,C,A>
54.7	0.4	<C,C,C,A>
53.9	0.8	<C,C,C,A>
52.6	1.3	<C,C,C,A>
52.5	0.1	<C,C,C,A>
51.7	0.8	<C,C,D,A>
51.3	0.4	<C,C,D,A>
51	0.3	<C,C,D,A>
50.7	0.3	<C,C,D,A>
50	0.7	<C,C,D,A>
48.8	1.2	<C,C,D,A>
48.7	0.1	<C,C,D,A>
48.6	0.1	<C,D,D,A>
46.7	1.9	<C,D,D,A>
46.4	0.3	<C,D,D,A>
46.2	0.2	<C,D,D,A>
45	1.2	<C,D,D,A>
44.9	0.1	<C,D,D,A>
44.6	0.3	<C,D,D,A>
44.5	0.1	<C,D,D,A>
43.5	1	<C,D,D,A>
42	1.5	<C,D,D,B>
35.7	**6.3**	<D,F,F,C>
28.4	**7.3**	<F,F,F,F>
28	0.4	<F,F,F,F>

Considering the nature of each method in conjunction with the above grading results leads to a guideline in Table 13 for appropriate method selection.

As we had not experimented our algorithm against data sets from other application domains, we did not claim the other applications of our algorithm besides that of the norm-referenced grading. However, the potential applications of our algorithm might include resource-consumer clustering problems in real life where their practical requirements of cluster-boundary explainability are the first priority: why two contiguously ranked data points (i.e., consumer profiles) belong to different clusters (i.e., different resource allocation levels) needs to be straightforwardly acceptable by data point owners. Some concrete applications can include the nation-wide selection of government loan applicants. Otherwise, serious arguments or even protests might occur between not only data-clustering processor and data owners but also discriminated data owners themselves. The main characteristic of our algorithm meets such requirements by providing a simple and clear-cut answer based on the widest gap

Table 11: Grading results of RD+.

Score	Gap	Grade
89.47	—	<A,A,A,A>
87.1	2.37	<A,A,A,A>
82.73	**4.37**	<B,A,A,A>
82.53	0.2	<B,A,A,A>
82.53	0	<B,A,A,A>
82.17	0.36	<B,A,A,A>
80.7	1.47	<B,A,A,A>
80.5	0.2	<B,B,A,A>
79.97	0.53	<B,B,A,A>
79.43	0.54	<B,B,A,A>
79.3	0.13	<B,B,A,A>
78.9	0.4	<B,B,A,A>
78.47	0.43	<B,B,A,A>
78.27	0.2	<B,B,A,A>
77.87	0.4	<B,B,A,A>
77.87	0	<B,B,A,A>
75.73	**2.14**	<C,B,B,A>
74.57	1.16	<C,B,B,A>
73.3	1.27	<C,B,B,A>
73.2	0.1	<C,B,B,A>
73.1	0.1	<C,B,B,A>
72.83	0.27	<C,B,B,A>
72.63	0.2	<C,B,B,A>
72.1	0.53	<C,B,B,A>
71.83	0.27	<C,B,B,A>
71.77	0.06	<C,B,B,A>
70.8	0.97	<C,C,B,A>
70.4	0.4	<C,C,B,A>
70.23	0.17	<C,C,B,A>
70.2	0.03	<C,C,B,A>
70.2	0	<C,C,B,A>
69.43	0.77	<C,C,B,A>
69.17	0.26	<C,C,B,A>
69.17	0	<C,C,B,A>
69.1	0.07	<C,C,B,A>
68.77	0.33	<C,C,B,A>
68.6	0.17	<C,C,B,A>
68.27	0.33	<C,C,C,A>
67.87	0.4	<C,C,C,A>
67.77	0.1	<C,C,C,A>
67.63	0.14	<C,C,C,A>
67.63	0	<C,C,C,A>
67.57	0.06	<C,C,C,A>
67.33	0.24	<C,C,C,A>
67.1	0.23	<C,C,C,A>
67	0.1	<C,C,C,A>
66.77	0.23	<C,C,C,A>
66.73	0.04	<C,C,C,A>
66.4	0.33	<C,C,C,A>
66.37	0.03	<C,C,C,A>
66.37	0	<C,C,C,A>
66.1	0.27	<C,C,C,A>
65.87	0.23	<C,C,C,A>
65.8	0.07	<C,C,C,A>
64.77	1.03	<C,C,C,A>
64.73	0.04	<C,C,C,A>
64.73	0	<C,C,C,A>
64.57	0.16	<C,C,C,A>
64.57	0	<C,C,C,A>
64.3	0.27	<C,C,C,A>

Table 11: Continued.

Score	Gap	Grade
64.17	0.13	<C,C,C,A>
64.13	0.04	<C,C,C,A>
63.93	0.2	<C,C,C,A>
63.9	0.03	<C,C,C,A>
63.57	0.33	<C,C,C,A>
63	0.57	<C,C,C,A>
62.83	0.17	<C,C,C,A>
60.63	**2.2**	<D,D,D,A>
60.33	0.3	<D,D,D,A>
59.83	0.5	<D,D,D,A>
58.93	0.9	<D,D,D,A>
58.87	0.06	<D,D,D,A>
58.53	0.34	<D,D,D,A>
58.47	0.06	<D,D,D,A>
58.27	0.2	<D,D,D,A>
57.53	0.74	<D,D,D,A>
57	0.53	<D,D,D,A>
56.77	0.23	<D,D,D,A>
55	1.77	<D,D,F,A>
54.8	0.2	<D,D,F,A>
54.57	0.23	<D,D,F,A>
54.5	0.07	<D,D,F,A>
54.5	0	<D,D,F,A>
54.43	0.07	<D,D,F,A>
54.37	0.06	<D,D,F,A>
53.8	0.57	<D,F,F,A>
53.73	0.07	<D,F,F,A>
53.37	0.36	<D,F,F,A>
53.37	0	<D,F,F,A>
52.87	0.5	<D,F,F,B>
52.47	0.4	<D,F,F,B>
52.1	0.37	<D,F,F,C>
52	0.1	<D,F,F,C>
51.97	0.03	<D,F,F,C>
51.8	0.17	<D,F,F,C>
50.9	0.9	<D,F,F,D>
50.7	0.2	<D,F,F,D>
50.2	0.5	<D,F,F,D>
50.1	0.1	<D,F,F,D>
45	**5.1**	<F,F,F,F>

between cluster boundaries; the other algorithms require that data owners completely understand the complicated algorithms to get answers.

Last but not least, to have an unbiased view, we point out the limitation of the proposed algorithm as follows. Although our algorithm can justify grade changes over evaluated scores through obvious score dissimilarity, the score ranges of the grades might be relatively different unlike z score. For instance, our algorithm might yield only a few learners receiving grade B and more receiving grade C. This can be negatively translated as unfair chances to receive both grades. Furthermore, unlike z score, our algorithm cannot skip any eligible grade if no one deserves such a grade (i.e., criterion based). The example lies in Table 12. However, this drawback holds only if some sense of criterion-referenced grading is introduced instead of pure norm-referenced grading.

Table 12: Grading results of WD.

Score	Grade
98	<A, A, A, A>
97	<A, A, A, A>
93	<B, A, B, A>
92	<B, A, B, A>
91	<B, A, B, A>
90	<B, A, B, A>
89	<B, A, B, A>
87	<B, A, B, A>
87	<B, A, B, A>
86	<B, A, C, A>
85	<B, A, C, A>
85	<B, A, C, A>
84	<B, A, C, A>
83	<B, A, C, A>
83	<B, A, C, A>
82	<B, A, C, A>
81	<B, B, C, B>
81	<B, B, C, B>
79	<C, B, C, C>
30	<D, F, D, D>
28	<F, F, D, D>
27	<F, F, D, D>
26	<F, F, D, D>
25	<F, F, D, D>
23	<F, F, F, D>
22	<F, F, F, D>
21	<F, F, F, D>
21	<F, F, F, D>
20	<F, F, F, D>
18	<F, F, F, F>
17	<F, F, F, F>

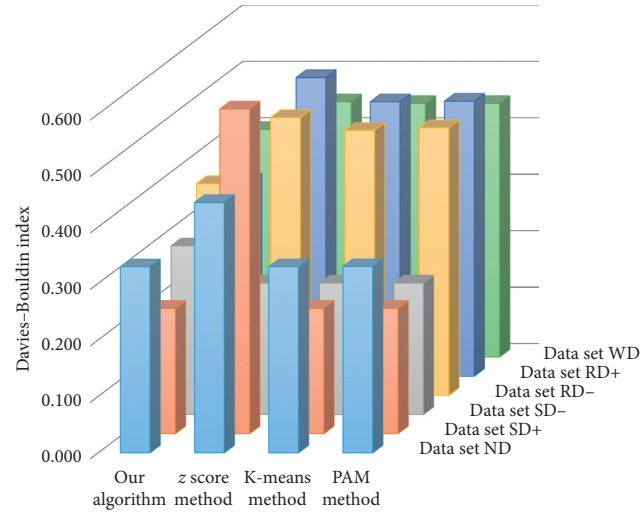

Figure 7: Performance of each method given each data set.

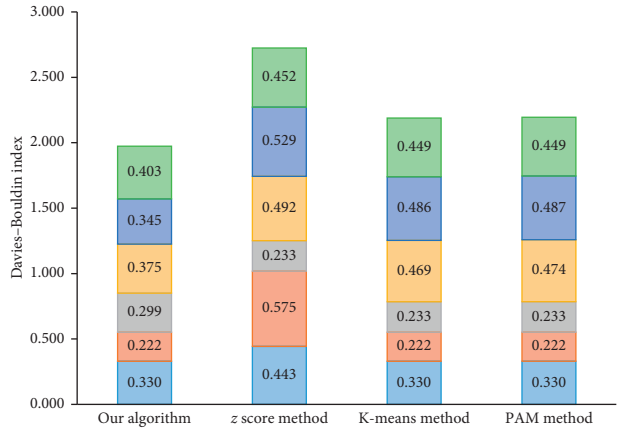

Figure 8: Overall performance of each method.

Table 13: Grading scheme selection guideline.

Method	Characteristic	Suitability
K-means	It prioritizes intracluster similarity, that is, score similarity within each learner group.	(i) This method is suitable when the same grade is always supposed to be held by learners with closely similar abilities. (ii) As indicated in Figure 7, K-means is also suitable for heavily skewed distribution like SD+ and SD− data sets.
PAM	PAM that produced the most A and the least F by average implies that the group GPA of learners tends to be high when grading with PAM.	PAM is also suitable for heavily skewed distribution like SD+ and SD− data sets as indicated in Figure 7.
Our algorithm	(i) In contrast with K-means, our algorithm prioritizes intercluster dissimilarity, that is, gaps between scores at the borders of different groups. (ii) Our algorithm is friendly to not only the heavily skewed distribution (i.e., SD+ and SD−) but also normal (i.e., ND) and slightly-to-moderately skewed distributions (i.e., RD−, RD+, and WD).	(i) This method is of a good choice when different grades are supposed to distinguish learning ability divides. (ii) Our algorithm is generally appropriate for all kinds of data distributions. The reason is that our algorithm's strategy is the determination of score gaps, which draw the clear-cut boundaries of clusters.
z score	(i) z score method disregards the notion of cluster (dis) similarity by engaging the even ranges of the best and the worst scores within each learner group. (ii) z score method is not good at dealing with norm-referenced grading in general mainly because its operation is blind to inherent raw-score gaps.	(i) This method should be used when all grades are supposed to encompass an equal score range. Let us consider Table 10. Grade C produced by our algorithm ranges from 42 to 63.5 points which is relatively wider than the score ranges of the other grades. This situation is avoided in z score's results. In other words, z score tries to equalize score ranges across all grades. (ii) Unlike the other methods, z score method is recommended for grading a score set that holds some wide divide (i.e., WD) because z score method allows skippable grades.

9. Conclusions

This paper provides the comprehension of four unconditionally norm-referenced grading methods: our new algorithm, z score, K-means, and PAM. We conducted the experiments with multiple data sets of various distribution characteristics based on DBI performance metric. Overall, our algorithm outperforms the other methods. K-means method is ranked second followed by PAM. z score is the worst but appropriate for some case. In fact, our algorithm is so simple that it is implementable by using a spreadsheet tool. We plan to conduct more experiments with constraints and apply our algorithm to other domains as well.

Data Availability

The data used to support the findings of the study are included within the article.

Disclosure

The preliminary version of this paper was published under the title "Norm-Referenced Achievement Grading: Methods and Comparison" in the Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2020.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was financially supported by the Department of Computer Science, Faculty of Science, Kasetsart University, Thailand.

References

[1] S. Wadhwa, *Handbook of Measurement and Testing*, Ivy Publishing House, Delhi, India, 2008.

[2] R. K. Arora and D. Badal, "Evaluating student's performance using k-means clustering," *International Journal of Computer Science And Technology*, vol. 4, pp. 553–557, 2013.

[3] S. P. Borgavakar and A. Shrivastava, "Evaluating student's performance using *K*-means clustering," *International Journal of Engineering Research & Technology*, vol. 6, pp. 114–116, 2017.

[4] Z. Parveen, A. Alphones, A. Alphones, and S. Naz, "Extending the student's performance via *K*-means and blended learning," *International Journal of Engineering and Applied Computer Science*, vol. 2, no. 4, pp. 133–136, 2017.

[5] S. Shankar, B. D. Sarkar, S. Sabitha, and D. Mehrotra, "Performance analysis of student learning metric using K-mean clustering approach," in *Proceedings of the 6th International Conference on Cloud System and Big Data Engineering*, Noida, India, January 2016.

[6] S. Xi, "A new student achievement evaluation method based on k-means clustering algorithm," in *Proceedings of the 2nd International Conference on Education Reform and Modern Management*, Hong Kong, China, April 2015.

[7] Z. Iqbal, A. Qayyum, S. Latif, and J. Qadir, "Early student grade prediction: an empirical study," in *Proceedings of the 2nd International Conference on Advancements*, Changsha, China, July 2019.

[8] K. Ramen and T. Joachims, "Methods for ordinal peer grading," in *Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, August 2014.

[9] S. M. Bai and S. M. Chen, "Automatically constructing grade membership functions for students' evaluation for fuzzy grading systems," in *Proceedings of 2006 World Automation Congress International Congress*, Budapest, Hungary, July 2006.

[10] T. Banditwattanawong and M. Masdisornchote, "Norm-referenced achievement grading: methods and comparison," in *Proceedings of Advances in Intelligent Systems and Computing 6th International Conference on Advanced Intelligent Systems and Informatics*, Cairo, Egypt, 2020.

[11] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers, Burlington, MA, USA, 2016.

[12] L. Kaufmann and P. Rousseeuw, *Data Analysis Based on the L1-Norm and Related Methods*, Springer, Berlin, Germany, 1987.

[13] B. Desgraupes, *Clustering Indices*, University of Paris, Paris, France, 2017.