Comparing Clustering Methods in Group-level Test Collusion Detection

Luyao Peng
Awada Tech LLC.
luyaopeng.cn@gmail.com

ABSTRACT

Test collusion occurs when examinees have prior access to a subset of items (compromised items) administered on a live test form. We compared three types of test collusion detection methods: ddClone, Spectral Clustering and Generalized Binomial Test (GBT) clustering method on real data. Results show that GBT clustering has higher power and higher type I error rates, ddClone has moderate power and type I error rates, Spectral Cluster has relatively higher power and the lowest type I error rates for two sets of real data.

Keywords

Bayesian Inference, Spectral Clustering, Test Collusion, Generalized Binomial Test Clustering

1. INTRODUCTION

As testing programs have moved into computer-based testing and digital communication devices have become ubiquitous, examinees can more easily access and widely share test content, resulting in test collusion [1]. Test collusion becomes a more serious problem for online classes and take-home exams since there exists no universally applicable method for proctoring online and take-home exams. It is therefore hardly feasible to stop students from illegally working together.

Because compromised items are more likely to be shared within a group of examinees, there are studies that explore methods to detect group-level test collusions, where the groups could refer to a group of examinees, test-prep centers, or schools that the examinees belong to.

Those studies include answer similarity indices (e.g. [12],[13],[16], [7]), clustering analysis based on those answer similarity indices (e.g. [5],[15],[2]) and machine learning clustering techniques (e.g. [6], [14]). However, the hybrid model utilizing both answer similarity indices and non-parametric techniques on test collusion detection has still been underrepresented in the test security literature.

L. Peng. Comparing clustering methods in group-level test collusion detection. In B. Paaßen and C. D. Epp, editors, *Proceedings of the 17th International Conference on Educational Data Mining*, pages 893–897, Atlanta, Georgia, USA, July 2024. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.12729989

The main goal of this article is (1) apply hybrid method ddClone [12] (a hybrid model utilizes both answer similarity index and non-parametric answer similarity matrix to infer group-level test collisions) to test collusion problems; (2) compare three types of clustering methods in detecting group-level test collusion: GBT clustering, Spectral Clustering (SC), and ddClone.

2. THREE TEST COLLUSION DETECTION METHODS

2.1 GBT Clustering

2.1.1 GBT Index

[13] proposed Generalized Binomial Test (GBT) Index to detect test collusion among pairwise students based on the probability of each pairwise matching response. Let P_j be the probability of matching response on item j assuming a dichotomous IRT model for the response data, $x_{j,copier/source} =$ 0 or 1 indicates an incorrect or correct answer from the copier or the source , P_j is defined as:

$$\begin{aligned} P_j = &[P(x_{j,copier} = 1|\theta, b_j)P(x_{j,source} = 1|\theta, b_j)] \\ &+ [P(x_{j,copier} = 0|\theta, b_j)P(x_{j,source} = 0|\theta, b_j)]. \end{aligned} \quad (1)$$

The joint probability of observing m pairwise matching responses across J items between a copier and a source given the abilities and the item difficulties is:

$$f_J(x_{\text{copiers, source}} = m|\theta, b_j) = \sum \prod_{j=1}^{J} P_j^t (1 - P_j)^{1-t}. \quad (2)$$

f_J is the GBT index, it shows the likelihood of obtaining m matching responses for a pair of copier and source across J items. GBT index will be compared with the Bonferonni-adjusted p-value (0.05/the number of pairs of students) to determine if the pair of copier and source has an unusually high number of matching responses.

2.1.2 Nearest-neighbor Clustering

[15] applied nearest-neighbor clustering method to the pairwise GBT index matrix to detect a group-level test collusion:

[15] first compute the pairwise similarity indices matrix (e.g., M4 index [7], GBT index [13]). Then, let $T_{k'}$ and T_k denote

two sets of clusters containing examinees $\{x_i\}, i \in 1 : n_{T_{k'}}$ and $\{y_i\}, i \in 1 : n_{T_k}$, respectively. If the GBT index for at least one pair of x_i and x_j is below a prespecified threshold, δ (0.05/the number of pairs of students), the clusters T_k and $T_{k'}$ will be grouped together to become one large cluster. [1]

2.2 Spectral Clustering

SC is a powerful clustering technique that can be particularly effective for clustering complex data structures. The reason we chose to include this method in our comparison is that our response data is binary, and SC is especially useful and efficient for clustering binary data.

2.2.1 Jaccard Score Matrix

First, the method computes the similarity matrix $S = \{s_{i,k}\}$ for the binary data using the Jaccard score measure, which effectively captures the similarity between binary vectors from the copier and the source. The Jaccard score is computed by first defining 4 quantities:

- a = the number of attributes that equal 1 for both objects i and k;
- b = the number of attributes that equal 0 for object i but 1 for object k;
- c = the number of attributes that equal 1 for object i but 0 for object k;
- d = the number of attributes that equal 0 for both objects i and k;

The Jaccard score is:

$$s_{i,k} = JaccardScore_{i,k} = \frac{a+d}{a+b+c+d}, \tag{3}$$

where $a+d$ gives the number of pairs with common responses and $a+b+c+d$ gives the total number of pairs with both common and different responses. [2]

Use n-nearest neighbor to transform S into a sparse adjacency matrix $A = \{a_{i,k}\}$ with n clusters. We determine the value of n by using validation data.

2.2.2 Normalized Laplacian

Then, SC calculates the Laplacian of the similarity graph S. The normalized Laplacian will be used in this study:

$$L_{sym} = I - D^{-1/2} S D^{-1/2}, \tag{4}$$

where I is the identity matrix, D is a diagonal degree matrix $D = \{d_{ii}\} = \left\{\sum_k^K = a_{i,k}\right\}$, where d_{ii} is the sum of the similarities of node i to all other nodes in A matrix.

[1] For example, cluster T_k contains 3 examinees, denoted as $T_k = \{x_1, x_2, x_3\}$, $T_{k'}$ contains 2 examinees, denoted as $T_{k'} = \{y_1, y_2\}$, if the M4 index between any members in T_k and $T_{k'}$ exceeds a clustering threshold, cluster T_k and $T_{k'}$ will be joined together.

[2] For example, source={0,1,1}, copier={1,1,1}, $a + d = 2$, $a + b + c + d$=3.

2.2.3 Eigenvalue Decomposition and Clustering

SC then performs the eigenvalue decomposition on the Laplacian matrix L_{sym} to reduce the data to a lower-dimensional space. Finally, a traditional clustering algorithm, like k-means, is used to cluster the data points in the reduced space defined by the eigenvectors.

The advantage of spectral clustering is that it does not make strong assumptions on the form of the clusters. As opposed to k-means, SC can solve very general problems [14].

2.3 ddClone, distance-based Bayesian Model

ddClone [11] originally aims to enhance the detection of clonal cell clusters in cancers by statistically integrating data from both single cell and bulk tumor sequencing.

In this study, we apply ddClone to test collusion detection scenario because the ddClone model leverages the strength of parametric likelihood model based on the observed number of pairwise matching responses and the non-parametric Bayesian prior informed by the Jaccard Distance of pairwise response vectors. We expect this hybrid model to offer improved inference in clustering group-level test collusion.

2.3.1 The Likelihood Function of Pairwise Matching Responses

We modified Eq (2) to have the same probability of a matching response if the current copier is assigned to cluster k. According to the Binomial theorem, the modified Eq (2) is equivalent to the term with $\binom{J}{m}$ in the expansion of $(P_k + Q_k)^J$, and becomes:

$$\begin{aligned} f_J(x_{\text{copiers, source}} = m | P_k) &= \sum \prod_{j=1}^{J} P_k^t (1-P_k)^{1-t} \\ &= \binom{J}{m} P_k^m (1-P_k)^{J-m}, \end{aligned} \tag{5}$$

where P_k is the probability parameter of the binomial distribution.

The purpose of this modification is to identify collusion at group level directly since the probability parameter is now group-specific rather than item-specific as in Eq (2).

Then, the prior of P_k is assumed to follow a Beta distribution if the assigned cluster of the copier is cluster k :

$$P_k | c_k \overset{iid}{\sim} Beta(1,1), k = 1, \ldots, K. \tag{6}$$

2.3.2 The Latent Collusion Cluster of Examinees: $c_{(i)}$

To estimate P_k, we define the latent cluster c of the ith copier as $c_{(i)}$ with the following prior distribution:

$$c_{(i)} \overset{iid}{\sim} Categorical(\pi_1, \ldots, \pi_K), i = 1, \ldots, N, \tag{7}$$

where $\pi_k, k \in 1 : K$, is modeled by a non-parametric prior probability using the distance-dependent Chinese Restaurant Process (ddCRP) [11].

2.3.3 Cluster Probability Model: ddCRP

In traditional Chinese Restaurant Process (CRP), customers enter a Chinese restaurant and opt to sit at a table where the probability of joining a table is proportional to the number of customers already sitting at the table. Customer may also choose to sit at a new table with probability proportional to parameter α.

In the case of test collusions, customers represent students and tables represent collusion clusters. Let $c_{(i)}$ denote the cluster the copier i is assigned to, the probability of copier i in cluster k is proportional to a function of the Jaccard distance (answer similarity) between the copier i and the source k (the source k can also be interpreted as the cluster k).

Let $s_{i,k}$ denote the Jaccard Distance between copier i and source k, the probability of $c_{(i)}$ is:

$$\pi(c_{(i)} = k|S, \alpha) = \begin{cases} f(s_{i,k}) & \text{for } i \neq k \\ \alpha & \text{for } i = k \end{cases} \quad (8)$$

where $f(s_{i,k}) = exp(-s_{i,k}/a)$ and $S = \{s_{i,k}\} = JaccardDistance(i,k)$, for $i, k = 1, \ldots, N$, in Eq (3).

We set up a threshold of the $f(s_{i,k})$ to 0.9 to determine which sources are the candidate clusters to the current copier (including the copier him/herself). The sources with the distances to the copier that are above the threshold will be used to compute the cluster probability π in Eq (8); the sources with distances below the threshold will be ignored.

2.3.4 Inference and Clustering

We define c_k as the kth cluster, $x_{1:N}$ as the observed matching response vector across N copiers. Given $P_{1:K}$, the probability of having a matching response for each cluster, the joint conditional likelihood for N copiers is factored as:

$$L(x_{1:N}|P_{1:K}) = \prod_{k=1}^{K} \prod_{copier \in c_k} p(x_{copier,source}|P_k), \quad (9)$$

where $p(x_{copier,source}|P_k)$ is the same as Eq (5).

Having the conditional likelihood, we need to find out the posterior distribution of the cluster identity for each copier. Let $c_{(1:N)}$ be the cluster assignments for all examinees. $c_{(i)}$ is the cluster of the ith copier, $c_{(-i)}$ is the copiers' cluster assignments other than the ith copier, define $\lambda = \{a, \alpha\}$ be the collection of the hyperparameters in the ddCRP model, the full posterior conditional distribution of the cluster for the ith copier is:

$$p(c_{(i)}|x_{1:N}, c_{(-i)}, \lambda) \propto \pi(c_i|\lambda, D)p(x_{1:N}|c_{(i)}, c_{(-i)}, \lambda), \quad (10)$$

where $\pi(c_i|\lambda, S)$ is the same as Equation 8, $p(x_{1:N}|c_{(i)}, c_{(-i)}, \lambda)$ is factored as:

$$\int_{P_1} \cdots \int_{P_K} L(x_{1:N}|P_{1:K}) \prod_{k=1}^{K} \pi(P_k|c_k) dP_1 \ldots dP_K, \quad (11)$$

where $L(x_{1:N}|P_{1:K})$ is equal to Eq (9) and $\pi(P_k|c_k)$ is equal to Ep (6).

To infer the cluster identity of each examinee, we use Gibbs sampler to draw samples from the posterior distribution of $c_{(i)}$ in Eq (10). We initialize the sampler such that all examinees are in their own groups.

After burn-in Markov chain and Monte Carlo (MCMC) samples, each student is assigned clusters with the number of samples times. To cluster examinees into groups, we first compute the posterior similarity matrix and then maximize the PEAR index to compute a point estimate [3] as implemented in the R package mcclust.

3. REAL DATA

3.1 Data

We applied the 3 methods to the common credentialing dataset [3]. The data come from a single year of testing for a computer-based licensure program that tests continuously. The identity of the program is confidential.

This licensure program administers multiple equated forms, Form 1 and Form 2. Each form contains 170 scored items and is paired with one of three different 10-item pretest sets, for a total test length of 180 items. Between Forms 1 and 2, there are 87 common items and 83 scored items that are unique to the form.

Dataset Form 1 contains 1636 examinees, 46 of whom had been flagged by the test vendor for illegally obtaining live test content prior to the exam (though other types of misconduct were possible as well)[4]. Form 2 contains 1644 examinees, 48 of whom had been flagged [1].

Both forms included binary responses to 170 items. The dataset also provided grouping variables such as schools ID, center ID, and the flagging information for aberrant students.

Table 1 shows the center ID with the number of flagged examinees greater than 1 for Form 1 and Form 2. Due to the computational complexity of ddClone, we apply 3 methods on selected groups on Form 2: $\{2305\}, \{2305, 5856\}$, $\{2305, 5856, 2331\}$ on Form 2 (Form 1 is used as validation data).

3.2 Evaluation Metrics

[3]Data is obtained upon request to the Testing and Evaluation Services of The University of Wisconsin-Madison, http://www.testing.wisc.edu

[4]Candidates were flagged through a combination of statistical analysis and a careful investigative process which brought in other pieces of information. While all examinees flagged are believed to have engaged in test fraud, it is certainly possible that there are other examinees who ought to have been flagged, but were not

Table 1: center id with at least 2 flagged examinees

Form1		Form2	
cent_id	flagged	cent_id	flagged
1	2	81	2
37	2	2305	6
2305	6	2331	2
		5204	2
		5856	3

We take the most assigned cluster from each of the 3 methods as the collusion cluster. We compute the following two evaluation measures for the students in the collusion cluster:

1. The power is computed as the number of true positive students who are clustered in the biggest cluster divided by the total number of flagged examinees in the selected center/centers.

2. The type I error is computed as the number of true negative students who are clustered in the biggest cluster divided by the total number of naive examinees in the considered center/centers.

3.3 Implementation

We used Form 1 data as the validation data to determine the tuning parameters: the threshold of the $f(s_{i,k})$ matrix in Eq (8), the number of clusters prespecified in the Spectral Clustering method, the threshold of GBT index δ in GBT clustering method.

We implemented ddClone using Python library NumPyro [9], a lightweight probabilistic programming library supports a number of inference algorithms, with a particular focus on MCMC algorithms. Convergence of ddClone model is assessed in a standard fashion using the approach proposed in [4]. We run four chains with diffuse initializations and verify that they converge to the same mean and variances (using the criterion $\hat{R} < 1.1$).

We implemented SC using scikit-learn package [8] using prespecified Jaccard distance matrix as the affinity matrix and $cluster_qr$ as the label assignment method; we also implemented GBT clustering using R [10].

3.4 Results

We applied ddClone, GBT clustering and SC method on Form 2 data. We cluster examinees using a threshold of 0.9 for ddClone and Bonferoni corrected 0.0005 threshold for the GBT clustering, the number of clusters prespecified in spectral clustering are 22, 35, 41 for data with different selected 'cent_id'.

Table 2 shows the powers and type I errors for 3 methods for data of different selected centers. GBT clustering has the highest power across different centers, however, its type I errors are also high; ddClone performs in the middle compared to the other methods in that it has moderate powers and lower type I errors compared to GBT clustering; SC performs the best due to its good power and very low type I error rates.

Table 2: Powers and Type I Error Comparisons for Form2 Data

	Center	2305	2305, 5856	2305, 2331, 5856
ddClone	hit rate	0.5	0.444	0.636
	false alarm rate	0.136	0.125	0.184
GBT	hit rate	0.833	0.889	0.909
	false alarm rate	0.409	0.375	0.368
SC	hit rate	0.667	0.533	0.555
	false alarm rate	0.053	0.038	0.032

We can visualize the clustering results by using the pairwise same-answer similarity matrix to create a heatmap (each cell is the number of matching responses between a pair of students, the darker the more common responses) and reordering the cells by putting the students in the biggest cluster in front of the students outside the biggest cluster on both X and Y axis. Hence, we can identify if the clustered students really have high number of observed common responses. The heatmap should have a black diagonal because each student has the most common responses with himself/herself.

Figure (1) shows the heatmaps of 3 methods for 2 sets of centers from Form 2 test. Since all the clustered students are put in the front, it is obvious that the upper left corners have darker colors (higher amount of common responses among those clustered students).

For both data (Center 2305, Centers 2305 and 5856), GBT has larger dark areas due to its high power, but it also included some light cells due to its high type I errors, while ddClone and Spectral Clustering miss some dark cells. Spectral Clustering performs relatively better for Centers 2305 and 5856 compared to ddClone.

4. CONCLUSIONS

This study compared 3 clustering methods in detecting group-level test collusion. The results show that GBT has higher power and higher type I error rates; ddClone has moderate power and type I error rates; Spectral Clustering has relatively higher power compared to the ddClone and the lowest type I errors across all data sets.

To implement these methods in detecting test fraud at group level, SC is an efficient method for large data and is flexible for different type of data such as polynomial responses, but the method is not suitable if the data contains too many different fraud clusters; GBT clustering has high power but also high type I errors, so the detection results should be judged via other analysis and manual check; ddClone performs relatively worse and it also takes a long time to sample the posterior clusters for each student, therefore, it is not suitable for large data.

This study has 3 limitations: 1, more clustering methods can be included for a more comprehensive comparison, such as SC using other distance matrices for binary data; 2, even though ddClone performs moderately, it is worth exploring

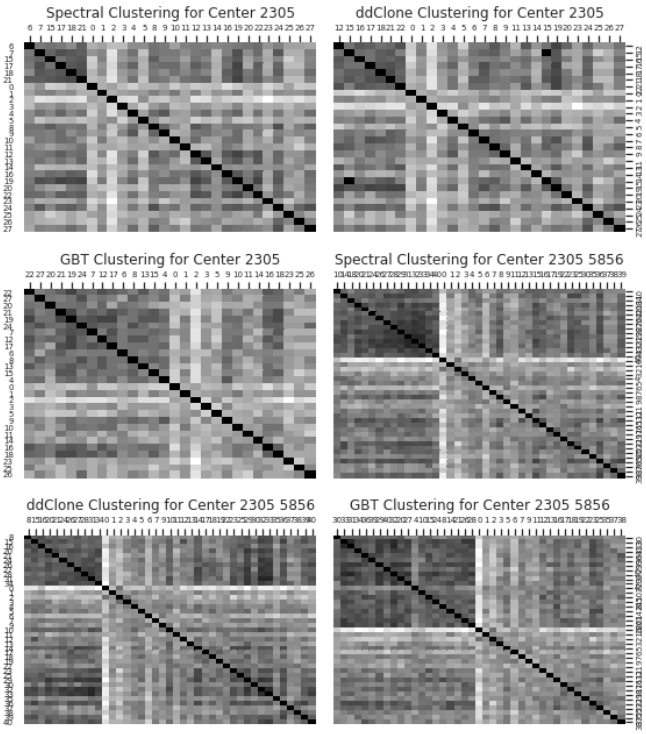

Figure 1: Clustering Heatmaps for center (2305) and (2305, 5856)

other matching options instead of matching responses for each question in the future study; 3, more evaluation criteria can be used such as F1 accuracy.

5. ACKNOWLEDGMENTS

This study is sponsored by Shanghai Pujiang Program, through Grant No. 22PJ1421800.

6. REFERENCES

[1] G. J. Cizek and J. A. Wollack. *Handbook of quantitative methods for detecting cheating on tests*. Taylor & Francis, 2016.

[2] C. Eckerly. Answer similarity analysis at the group level. *Applied Psychological Measurement*, 45(5):299–314, 2021.

[3] A. Fritsch and M. A. Fritsch. Package 'mcclust'. 2009.

[4] A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, pages 457–472, 1992.

[5] B. A. Jacob and S. D. Levitt. Rotten apples: An investigation of the prevalence and predictors of teacher cheating. *The Quarterly Journal of Economics*, 118(3):843–877, 2003.

[6] J. Langerbein, T. Massing, J. Klenke, N. Reckmann, M. Striewe, M. Goedicke, and C. Hanck. A data mining approach for detecting collusion in unproctored online exams. *arXiv preprint arXiv:2302.07014*, 2023.

[7] D. D. Maynes. Detecting potential collusion among individual examinees using similarity analysis. In *Handbook of quantitative methods for detecting cheating on tests*, pages 47–69. Routledge, 2016.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[9] D. Phan, N. Pradhan, and M. Jankowiak. Composable effects for flexible and accelerated probabilistic programming in numpyro. *arXiv preprint arXiv:1912.11554*, 2019.

[10] RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, PBC., Boston, MA, 2020.

[11] S. Salehi, A. Steif, A. Roth, S. Aparicio, A. Bouchard-Côté, and S. P. Shah. ddclone: joint statistical inference of clonal populations from single cell and bulk tumour sequencing data. *Genome biology*, 18:1–18, 2017.

[12] L. S. Sotaridona and R. R. Meijer. Two new statistics to detect answer copying. *Journal of Educational Measurement*, 40(1):53–69, 2003.

[13] W. J. van der Linden and L. Sotaridona. Detecting answer copying when the regular response process follows a known response model. *Journal of Educational and Behavioral Statistics*, 31(3):283–304, 2006.

[14] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.

[15] J. A. Wollack and D. D. Maynes. Detection of test collusion using cluster analysis. In *Handbook of quantitative methods for detecting cheating on tests*, pages 124–150. Routledge, 2016.

[16] C. Zopluoglu. Copydetect: An r package for computing statistical indices to detect answer copying on multiple-choice examinations. *Applied psychological measurement*, 37(1):93–95, 2013.