Proceedings of the 90th Annual International Meeting of the Psychometric Society, Minneapolis, United States (2025).

Identifying Attribute Hierarchies in Cognitive Diagnosis from Data: A Graph-Theoretic Approach

Hyunjoo Kim* and Hans Friedrich Köhn

Quantitative Psychology, Department of Psychology, Champaign, 61820, IL, USA
*Corresponding author. Email: hyunjoo5@illinois.edu

Abstract

Attributes (cognitive skills) in Cognitive Diagnosis (CD) are not isolated but interdependent in their effect on test performance. One way to represent these dependencies is to assume that attributes are hierarchically organized such that mastery of one or more attributes serves as a prerequisite for mastering others. Imposing such hierarchies constrains the latent attribute space by excluding impermissible combinations of attributes. In practice, however, attribute hierarchies are rarely known a priori and therefore must be estimated from data.

To date, only a limited number of studies in the CD literature have examined the identification of attribute hierarchies directly from item-response data. The present study investigates a two-step, graph-theoretic framework for uncovering attribute hierarchies: First, graph-clustering algorithms are applied to item-response data to detect item structures. Second, a directed acyclic graph (DAG) structure learning procedure is used to infer attribute hierarchies assumed to be reflected in these item structures. Simulation results indicate that the proposed framework shows promise as a viable exploratory approach for identifying attribute hierarchies.

Keywords: Cognitive diagnosis, Attribute hierarchy, Graph-clustering, Directed acyclic graph

1. Introduction

Cognitive Diagnosis (CD) in educational assessment is designed to evaluate examinees' mastery of skills — also referred to as "attributes" — that are required for instructional content. CD provides fine-grained diagnostic information about both mastered and yet-to-be mastered cognitive skills. In CD models, attributes are typically represented as binary variables, and combinations of these attributes form attribute profiles that define distinct proficiency classes, written as $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_K)'$. The complete set $\mathcal{L}$ of all possible attribute profiles is commonly referred to as the latent attribute space (Tatsuoka, 2009).

Analogous to examinees, whose proficiencies are characterized by attribute profiles, the items $j = 1, 2, ..., J$ in a CD assessment are also characterized by K-dimensional binary vectors, $\mathbf{q}_j$, which specify the attributes required for a correct response. Specifically, $q_{jk} = 1$ if a correct answer requires mastery of the kth attribute, and 0 otherwise. These item-attribute profiles, called q-vectors, are collected into the Q-matrix of a CD assessment, $\mathbf{Q} = \{q_{jk}\}_{(J \times K)}$ (Tatsuoka, 1985).

In many assessment contexts, attributes are assumed to be interdependent rather than independent. One way to model such dependencies is through prerequisite relations, whereby mastery of one attribute is required before another can be acquired, thereby inducing hierarchical structures among attributes (Leighton & Gierl, 2007; Leighton et al., 2004; Tatsuoka, 2009). Imposing these hierarchies constrains the latent attribute space $\mathcal{L}$ by excluding attribute profiles that violate the prerequisite

relations. Ignoring hierarchical structures may therefore lead to inaccurate classifications of examinees into latent classes that are theoretically impermissible.

Incorporating attribute hierarchies into CD assessments was introduced by Leighton et al. (2004) through the Attribute Hierarchy Method (AHM), which uses a reduced Q-matrix containing only permissible q-vectors consistent with the constrained latent attribute space $\mathcal{L}$. Building on this framework, Templin and Bradshaw (2014) proposed the Hierarchical Diagnostic Classification Model (HDCM), which integrates the AHM with the Log-linear Cognitive Diagnosis Model (LCDM; Henson et al., 2009). The HDCM provides a statistical framework for evaluating the presence of attribute hierarchies — primarily linear ones — by fixing the parameters associated with impermissible latent classes to zero, thereby reducing the number of parameters estimated relative to the full LCDM.

To date, the identification of attribute hierarchies in CD models has been largely confirmatory, relying on a priori specification. However, in practical applications, hierarchical relations among attributes are often unknown. In response, several data-driven methods have recently been proposed to identify attribute hierarchies from observed item responses. These include variants of a regularized Expectation-Maximization (EM) algorithm for latent variable selection (Gu & Xu, 2019; C. Ma et al., 2023; Wang & Lu, 2021) and Bayesian estimation methods using the Metropolis-Hastings algorithm with Gibbs sampling (Chen & Wang, 2023). Despite their contributions, these approaches exhibit notable limitations, including sensitivity to the choice of penalty terms (C. Ma et al., 2023) and heavy parameterization that may impede convergence (Chen & Wang, 2023).

As an alternative, the present study investigates graph-theoretic approaches that first detect item structures from item responses and then extract underlying attribute hierarchies, under the assumption that item structures reflect attribute structures. Tatsuoka (2009) suggested that hierarchical relations among items may point at an underlying hierarchy of latent attributes, which can be formalized using directed acyclic graphs (DAGs). This idea is consistent with Gagné (1965), who argued that attribute hierarchies underlying test items should be reflected in the hierarchical organization of the items themselves. Building on these theoretical foundations, the present study offers a preliminary examination of whether this mechanism is empirically recoverable using graph-theoretic approaches that operate directly on item response data, without relying on parametric statistical models. Rather than proposing a fully developed method, our goal is to investigate the viability of this idea under controlled and relatively simple simulation conditions, providing a proof-of-concept basis for more comprehensive investigations in future work.

From a graph-theoretic point of view, items can be represented as nodes in a graph, with pairwise item proximities defining edge weights. Several item proximity measures have been proposed, including the phi correlation coefficient, the tetrachoric correlation coefficient, and Kendall's tau. Building on Hubert (1974), who reviewed and extended the applications of graph theory to clustering, a variety of graph-based algorithms have been developed for use in assessment contexts. These include hierarchical clustering of item proximity matrices (Hubert, 1973), Boolean analysis (Schrepp, 2003), and the k-mode algorithm (de Chiusole et al., 2017) for extracting knowledge structures among items. In addition, spectral clustering has been applied to item graphs to identify groups of items with similar properties (Chen et al., 2017), and network models that estimate potential causal relations among items have also been proposed (Letina et al., 2019; van Borkulo et al., 2014).

Despite these advances, recovering hierarchical item structures from noisy item-response data remains challenging, especially when the number of items is large. Kim (n.d.) provided preliminary simulation evidence that graph-theoretic item-structure recovery can be sensitive to response noise and item-pool size. Moreover, prior work has largely focused on identifying item hierarchies, with limited investigation into how such hierarchies correspond to underlying attribute hierarchies from a CD perspective.

To address these issues, the present study investigates a stepwise, data-driven framework that

integrates graph-clustering methods with DAG-based structure learning algorithms to infer attribute hierarchies from item-response data. This framework is grounded in the assumption that latent attribute structures can be identified from item structures revealed, for example, through clustering. Specifically, items grouped within the same cluster are assumed to share highly similar — or even identical — item-attribute vectors (i.e., q-vectors) that drive the observed item responses. From each cluster, a single item is selected as a representative. The dependent relations among these representative items are then modeled using a heuristic Bayesian network learning algorithm, producing a DAG. The hierarchical structure encoded in the resulting DAG reflects the hierarchy among items and thus, by assumption, the hierarchy among the underlying item-attribute vectors. Because these vectors are partially ordered, they, in turn, imply the corresponding hierarchy among the individual attributes.

In summary, the objectives of this study are to:

1. detect item partitions using graph-clustering algorithms applied to item proximity matrices, resulting in clusters of items that share similar item-attribute vectors.
2. examine prerequisite relations among the selected items from the identified item clusters by applying DAG-based structure learning algorithms to infer attribute hierarchies.

The remainder of this paper is organized as follows. Section 2 introduces attribute hierarchies and their connection to lattice theory, followed by an overview of graph-theoretic methodologies — focusing on graph-clustering algorithms — used to detect item hierarchies from proximity matrices. Section 3 presents simulation studies illustrating the application of these methods in conjunction with a DAG structure learning algorithm. Finally, Section 4 provides a general discussion of the potential of graph-theoretic approaches and outlines directions for future research.

2. Background

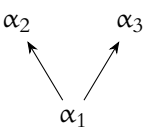
2.1 Attribute Hierarchies

As an example of attribute hierarchies, consider three attributes, α_1, α_2, and α_3, underlying the items of a CD assessment. If these attributes do not have a hierarchical structure, then the latent attribute space $\mathcal{L}$ contains all possible attribute profiles (characterizing distinct proficiency classes) with $2^3 = 8$ elements:

Table 1. Possible attribute profiles of three attributes without a hierarchical structure

No.	$(\alpha_1\ \alpha_2\ \alpha_3)$
1	(0 0 0)
2	(1 0 0)
3	(0 1 0)
4	(0 0 1)
5	(1 1 0)
6	(1 0 1)
7	(0 1 1)
8	(1 1 1)

Now suppose a divergent attribute hierarchy is imposed on α_1, α_2, and α_3:



Here, α_1 is a prerequisite for α_2 and α_3. Under this hierarchical structure, the latent attribute space $\mathcal{L}$ is restricted to only five proficiency classes: (000), (100), (110), (101) and (111). The profiles (010), (001), and (011) are no longer permissible because they violate the prerequisite relations.

As a theoretical framework for estimating attribute hierarchies from observed item responses in CD, Köhn and Chiu (2019, 2021) proposed a general framework grounded in lattice theory. Key features of this framework include:

(1) Attributes $\alpha_k \in \{0, 1\}$ are treated as Boolean variables, and the K-dimensional attribute profiles $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_K)'$ are Boolean vectors.
(2) A partial order $\preceq$ is defined on pairs of binary attribute vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$ such that $\boldsymbol{\alpha} \preceq \boldsymbol{\alpha}^*$ if and only if $\alpha_k \leq \alpha_k^* \forall k$.
(3) The latent space $\mathcal{L}$, together with the prerequisite relation $\preceq$, forms a lattice if every finite subset has both an infimum and a supremum. It is called a complete lattice if it has universal bounds — specifically, the all-zero vector as its least element and the all-one vector as its largest element.

Figure 1 presents Hasse diagrams of the (un-)constrained attribute latent space $\mathcal{L}$: The left panel shows the unconstrained lattice with no attribute hierarchy imposed, comprising $2^3 = 8$ possible proficiency classes, whereas the right panel displays the constrained structure induced by the divergent attribute hierarchy.

When no hierarchy is imposed, the attributes α_1, α_2, and α_3 are independent, and the single-attribute profiles, (100) = $\mathbf{e}_1$, (010) = $\mathbf{e}_2$, and (001) = $\mathbf{e}_3$ (highlighted with grey boxes) serve as the "basic attribute vectors" Köhn and Chiu (2019, 2021). In this structure: (1) all possible proficiency classes can be generated through the Boolean operations $\wedge$ and $\vee$ applied to the basic vectors; (2) proficiency classes are vertically ordered, with edges indicating the partial order $\preceq$; (3) because the order is transitive, any relation between proficiency classes can be deduced by tracing edges upward; and (4) the infimum and supremum of every subset of $\mathcal{L}$ also belong to $\mathcal{L}$.

In contrast, when the divergent hierarchy is imposed, the Hasse diagram of $\mathcal{L}$ takes the form shown in the right panel of Figure 1. The three basic attribute vectors (100), (110), and (101) (again shown in grey) — fundamental building blocks used to define the latent attribute space $\mathcal{L}$ — are derived from the tree structure of the divergent hierarchy. The constrained attribute space can be reconstructed from these attribute vectors (e.g., (110) $\vee$ (101) = (111)); (000) and (111) are part of any lattice by definition. Köhn and Chiu (2019, 2021) proved that the latent attribute space $\mathcal{L}$ for any hierarchical attribute structure can be reconstructed as a set of linear combinations of its basic attribute vectors.

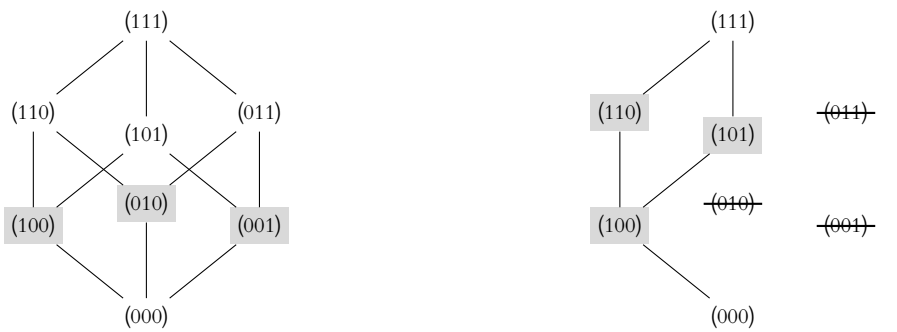


Figure 1. Hasse diagrams illustrating a Boolean lattice structure without a hierarchy (left) and with a divergent hierarchy (right)

2.2 Graph-clustering Algorithms

Graph-clustering algorithms—also commonly referred to as graph-partitioning problems or community detection algorithms, depending on the specific clustering method—seek to identify groups of nodes that are more strongly associated with one another than with the rest of the graph.

In the present study, three such algorithms — spectral clustering, the Louvain method, and affinity propagation algorithm — are used to evaluate the extent to which graphs constructed from item proximities capture item structures and potentially reflect underlying attribute hierarchies. Item graphs are represented as weighted complete graphs. A complete graph G, constructed from a proximity matrix, consists of a node set V, an edge set E, and an edge-weight set W, denoted as $G(V, E, W)$. The proximity values between pairs of items correspond to the elements of W, with each entry in position (j, j') defining the weight of the edge between nodes j and j'. The following subsections provide brief overviews of the three graph-clustering algorithms used in the simulation studies.

2.2.1 Spectral Clustering

Spectral clustering is a graph-based technique for partitioning data points into clusters using the eigenvectors of a Laplacian matrix derived from a similarity matrix of data points (Ng et al., 2002; Shi & Malik, 2000; von Luxburg, 2007). It has been applied in assessment contexts to explore the structure of large item pools (Chen et al., 2017) and to analyze relations between latent classes of students and items (Chen & Li, 2019).

The algorithm begins by constructing a (weighted) adjacency matrix $\mathbf{A}$, typically based on pairwise similarities or local neighborhood relations among nodes. For example, $\mathbf{A}$ may correspond to a fully connected graph or a K-nearest-neighbor (KNN) graph that connects node to its K^* most similar neighbors.

Next, the Laplacian matrix $\mathbf{L}$ is computed from $\mathbf{A}$. Define the diagonal degree matrix $\mathbf{D}$, with diagonal entries given by the row sums of $\mathbf{A}$, that is, $D_{jj} = \sum_{j'=1}^{J} A_{jj'}$, where $j = 1, 2, \ldots, J$. The specific form of the Laplacian matrix depends on the application. Two common choices are the unnormalized and normalized Laplacian. The former emphasizes the raw structure and connectivity of the graph ($\mathbf{L} = \mathbf{D} - \mathbf{A}$), whereas the normalized Laplacian focuses on relative relations and is more robust to variations in node degrees ($\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$).

The eigenvalues of $\mathbf{L}$ are then ordered in increasing order, and the eigenvectors corresponding to the M smallest eigenvalues are extracted, where M is the prespecified number of clusters. When M is not known, the eigengap heuristic can be applied to both unnormalized and normalized Laplacians. This approach selects M based on a large gap between λ_M and λ_{M+1} relative to the gaps among the smaller eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_{M-1}$. The eigenvectors corresponding to the M smallest eigenvalues form a matrix $\mathbf{X}_{(J \times M)} = [\mathbf{x}_1, \ldots, \mathbf{x}_M]$, whose rows represent the data in a reduced-dimensional space. A standard clustering algorithm, such as k-means, is then applied to the rows of $\mathbf{X}$ to partition the data into M clusters. For very large graphs, approximations may be used to reduce the computational cost of finding eigenvectors.

2.2.2 Louvain Method

The Louvain method partitions a weighted, undirected graph by maximizing a modularity function Q, a measure of how much more densely connected nodes are within clusters than between clusters (Blondel et al., 2008). Modularity ranges from -1 to 1, and is maximized when edges within clusters carry substantially higher weights than edges across clusters. For a given partition $\{c_i\}$, modularity is defined as:

$$Q(\gamma) = \frac{1}{2m} \sum_i \sum_j \left[W_{ij} - \gamma \frac{k_i k_j}{2m} \right] \delta(c_i, c_j),$$

where W_{ij} is the edge weight between nodes i and j, $k_i = \sum_j W_{ij}$, and $k_j = \sum_i W_{ij}$ are the weighted degrees of nodes i and j, respectively, $m = \frac{1}{2} \sum_i \sum_j W_{ij}$ is the total edge weight in the graph, and $\delta(c_i, c_j) = 1$ if nodes i and j belong to the same cluster and 0 otherwise. The term $\frac{k_i k_j}{2m}$ is the expected edge weight between nodes i and j under a null model in which edges are placed at random subject to preserving node degrees (the configuration model). Thus, $Q(\gamma)$ measures the extent to which the observed within-cluster connectivity exceeds what would be expected under this degree-preserving random graph.

The resolution parameter γ enters directly into the definition of $Q(\gamma)$ by scaling the null-model term. Different values of γ therefore define different modularity functions and, consequently, different optimization problems. Larger values of γ increase the contribution of the null model, making it more difficult for nodes to be grouped together and thus favoring smaller and more numerous clusters, whereas smaller values favor larger clusters. The Louvain algorithm maximizes $Q(\gamma)$ using a greedy two-phase procedure.

(1) Local Optimization: Each node is initially assigned to its own cluster, and nodes are repeatedly moved to neighboring clusters whenever such a move increases $Q(\gamma)$.

(2) Cluster Aggregation: The resulting clusters are collapsed into super-nodes, yielding a smaller graph on which the same procedure is repeated.

Iteration continues until no further increase in $Q(\gamma)$ is possible.

The value of γ is not determined by the algorithm and must be specified by the user. In practice, $\gamma = 1$ is often used as a default, but this choice is conventional rather than theoretically required. A more informative approach is to evaluate the resulting partition across a range of candidate values of γ and examine how the community structure changes across scales. Accordingly, γ should be interpreted as a scale parameter, and its selection constitutes a model-specification decision rather than an intrinsic property of the data. Because the method operates directly on weighted graphs, it can be applied to item proximity matrices when these are interpreted as weighted similarity networks.

2.2.3 Affinity Propagation

Affinity propagation is an exemplar-based clustering method that operates on a similarity matrix $S = \{s(i, j)\}$, where $s(i, j)$ quantifies the similarity between objects i and j. A natural choice is the negative squared Euclidean distance, $s(i, j) = -\|x_i - x_j\|^2$, so that larger values indicate greater similarity (Frey & Dueck, 2007). Historically, the underlying clustering rationale is closely related to classical facility-location and p-median formulations, in which a subset of observed objects is selected as exemplars and each remaining object is assigned to one exemplar so as to optimize overall within-cluster fit (Brusco & Köhn, 2008, 2009; Hanjoul & Peeters, 1985; Hansen & Mladenovic, 1997; Köhn et al., 2010; Kuehn & Hamburger, 1963; Maranzana, 1964). Unlike standard p-median formulations, however, the number of exemplars is not specified directly. Instead, it is controlled indirectly through the diagonal entries $s(j, j)$, called *preferences*, which are user-specified values governing the propensity of object j to be selected as an exemplar. Larger preference values generally produce more exemplars and hence more clusters, whereas smaller preference values produce fewer. In practice, preferences are often set uniformly across objects, for example, by assigning each diagonal entry to a common summary of the observed proximities, such as the average similarity of an object to all other objects or, more commonly, a global constant such as the median of the off-diagonal similarities.

$$r(i,j) \leftarrow s(i,j) - \max_{j' \neq j}\{a(i,j') + s(i,j')\},$$

so that $r(i,j)$ measures the relative support for assigning object i to candidate exemplar j rather than to competing candidates. Given the updated responsibilities, availabilities are updated as:

$$a(i,j) \leftarrow \min\left\{0, r(j,j) + \sum_{i' \notin \{i,j\}} \max\{0, r(i',j)\}\right\},$$

where $i \neq j$, with $a(j,j) \leftarrow \sum_{i' \neq j} \max\big(0, r(i',j)\big)$. Thus, $a(i,j)$ reflects the current support for candidate exemplar j being available to serve as the exemplar for object i, given the positive support that j receives from the other objects.

The apparent circularity between $r(i,j)$ and $a(i,j)$ is resolved algorithmically: the recursion is initialized, typically with $a(i,j) = 0$, and the two sets of quantities are then updated alternately until convergence. At convergence, object j is selected as an exemplar when $a(j,j) + r(j,j) > 0$, and each object i is assigned to the exemplar j that maximizes $a(i,j) + r(i,j)$. In the present study, affinity propagation is applied to an item proximity matrix to identify exemplar items and corresponding item clusters that may reflect underlying attribute structure.

2.3 Attribute Hierarchies as Directed Acyclic Graphs and DAG Structure Learning

A Directed Acyclic Graph (DAG) consists of nodes and directed edges with no directed cycles. DAG structure learning is the process of discovering dependencies (directed edges) among variables (i.e., nodes) from observational data. It is an NP-hard problem requiring specialized techniques, broadly categorized into constraint-based and score-based approaches. Constraint-based methods use statistical tests of independence to determine the edges between nodes, such as the PC algorithm (Spirtes et al., 2010), whereas score-based methods use algorithms such as Hill-Climbing (Tsamardinos et al., 2006) to find a graph that maximizes a predefined score through a greedy iterative local search.

Estimating a DAG structure of attribute hierarchies has been investigated in previous studies (Chen & Wang, 2023; Gu & Xu, 2023; C. Ma et al., 2023), where directed edges correspond to direct dependencies or precedence relations among attributes. In our simulation study, a DAG structure learning procedure was applied to estimate the attribute hierarchies implied by the item partitions identified by the graph-clustering algorithms in Section 2.2. Each detected item cluster was treated as representing a distinct permissible q-vector in the hierarchy.

3. Computational Experiments

The purpose of the computational experiments reported in the following section was to provide an initial exploration of whether the rationale underlying the proposed procedure is viable: Do item partitions uncovered from item-response data reflect hierarchical relations among underlying attributes? Can these hierarchical relations be detected by combining graph-clustering algorithms with DAG structure learning methods? To this end, synthetic data were generated using items having basic q-vectors that embodied three types of attribute hierarchies. These q-vectors also implied a known clustering of items, providing a basis for evaluating partition recovery. The three graph-clustering methods introduced in the previous section were applied to produce item partitions, and their performance was assessed in terms of their ability to recover the clusters implied by the q-vectors as item responses were increasingly contaminated with perturbations. Item-response data conforming to the Deterministic Input, Noisy "And" gate (DINA) model (Haertel, 1989; Junker & Sijtsma, 2001; Macready & Dayton, 1977) were generated using the `GDINA` R package (W. Ma &

de la Torre, 2020). Let Y_{ij} denote the observed binary response of examinee i to item j. The item response function (IRF) of the DINA model is:

$$P(Y_{ij} = 1|\boldsymbol{\alpha}_i) = (1 - s_j)^{\eta_{ij}} g_j^{1-\eta_{ij}},$$

where $\boldsymbol{\alpha}_i = (\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{iK})'$ is the K-dimensional binary attribute profile of examinee i, and $\eta_{ij} = \prod_{k=1}^{K} \alpha_{ik}^{q_{jk}}$ is the conjunction parameter, indicating whether examinee i has mastered all attributes required for item j. The model includes two item parameters: slipping, $s_j = P(Y_{ij} = 0|\eta_{ij} = 1)$, the probability of an incorrect response despite mastery of all required attributes; and guessing, $g_j = P(Y_{ij} = 1|\eta_{ij} = 0)$, the probability of a correct response despite non-mastery. The monotonicity constraint $0 < g_j < 1 - s_j < 1$ ensures that guessing never yields a higher probability of success than full mastery. Item-response datasets of size $N \times J$ conforming to the DINA model were generated under the following conditions:

- **Sample size** $N = 1000$.
- **Number of attributes** $K = 5$.
- **Test length** $J = 30$. The basic attribute vectors of each hierarchical structure were replicated 6 times, respectively, resulting in "true" Q-matrices of size 30×5. Thus, Items $1, 6, 11, \ldots, 26$, Items $2, 7, 12, \ldots, 27$, Items $3, 8, 13, \ldots, 28$, Items $4, 9, 14, \ldots, 29$, and Items $5, 10, 15, \ldots 30$ share identical q-vectors.
- **Slipping and guessing parameters** $sg = 0.1$, and 0.3. These parameters introduce low and high levels of perturbation error into the datasets, simulating realistic noise.
- **Attribute hierarchies**: Linear, convergent, and divergent structures were imposed (Chen & Wang, 2023; Leighton et al., 2004). Figure 2 shows the graphical structures of attribute hierarchies.

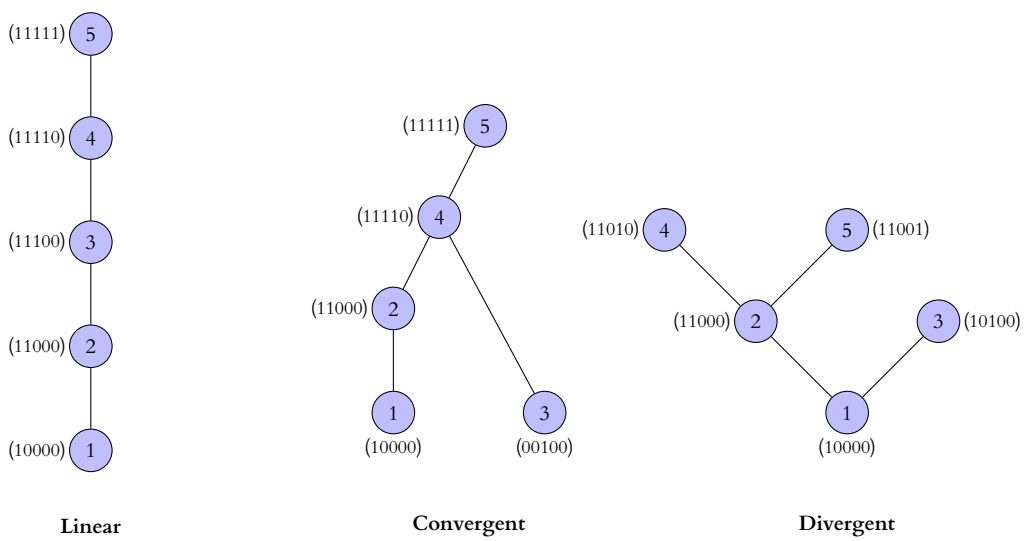


Figure 2. Structures of the attribute hierarchies with corresponding basic attribute vectors

The generated item-response data were then transformed into $J \times J$ proximity matrices using tetrachoric correlation coefficients. For each level of perturbation error and each attribute hierarchy, ten datasets were generated. The resulting inter-item proximity matrices were then subjected to the three graph-clustering algorithms.

3.1 Identification of Item Clusters Using Graph-Clustering Algorithms

Spectral clustering, the Louvain method, and the affinity propagation algorithm were applied to the item-proximity matrices. The performance of each algorithm was evaluated by comparing the identified clusters with the known true clusters using the Adjusted Rand Index (ARI; Hubert & Arabie, 1985), as presented in Table 2 under the low perturbation level (i.e. $sg = 0.1$). Results under the high perturbation level ($sg = 0.3$) are presented in Table 4.

For spectral clustering, the item-proximity matrices — complete weighted graphs in graph-theoretic jargon — were dichotomized using a K-nearest neighbor (KNN) procedure proposed by Brito et al. (1997) (that was also used by Chen et al., 2017) retaining the K^* neighbors with the highest weights (correlations) for each node (item). For each dataset, K^* was determined by Brito et al.'s (1997) algorithm applied consistently across attribute hierarchies: starting from the smallest integer greater than or equal to $\log(J)$ and incrementing by one until the resulting symmetric adjacency matrix $\mathbf{A}$, which retains only each node's K^* most similar neighbors, becomes a connected graph. Based on this transformed graph, the Laplacian matrix was constructed, and k-means clustering was applied to the eigenvectors of the Laplacian associated with the five smallest eigenvalues. Both normalized and unnormalized Laplacians were examined, but no meaningful differences were observed, likely because the node degrees showed little variation in these graphs. The spectral clustering algorithm showed relatively poor performance in recovering item partitions across all hierarchy types under the low perturbation level compared to the other two graph-clustering algorithms.

For the Louvain method, it was run ten times at each value of γ, ranging from $\gamma = 0.5$ to 1.5 in increments of 0.25, for each of the ten datasets. Notice that the Louvain method is non-deterministic due to its greedy nature of making locally optimal choices at each iterative step, so results from a single run may vary. The resolution parameter $\gamma = 1.25$ was selected across all hierarchy types based on two criteria: (i) maintaining modularity scores as high as those obtained for other choices of γ. and (ii) detecting a number of clusters consistent with the simulation design. The reported ARI values in Table 2 represent the mean across ten runs for each dataset, with variance values in parentheses. Under the low perturbation level, the Louvain method achieved high ARI values for the linear and divergent hierarchies, with little variance across the ten runs within the same dataset. As the perturbation level increased, the within-dataset variance of ARI increased, and the overall clustering performance degraded.

Affinity propagation was implemented using the default setting (i.e., the median of the similarity values across all item pairs) so that the chance of being chosen as an exemplar was equal across every point. Affinity propagation performed perfectly (i.e., ARI = 1.00) across all attribute hierarchy types. The five exemplar items selected by the algorithm possessed q-vectors corresponding to the five basic attribute vectors. As the level of perturbation error increased, however, clustering performance decreased.

Figure 3 presents an example of item partitions across attribute hierarchies for one of the ten datasets obtained from each of the three algorithms at the perturbation level $sg = 0.1$. Clustering performance varied across hierarchy types: Item partitions associated with divergent attribute hierarchies were recovered most accurately under both high and low perturbation levels, while structures generated under convergent attribute hierarchies showed the weakest performance.

3.2 Identification of Attribute Hierarchies Using a DAG Structure Learning Algorithm

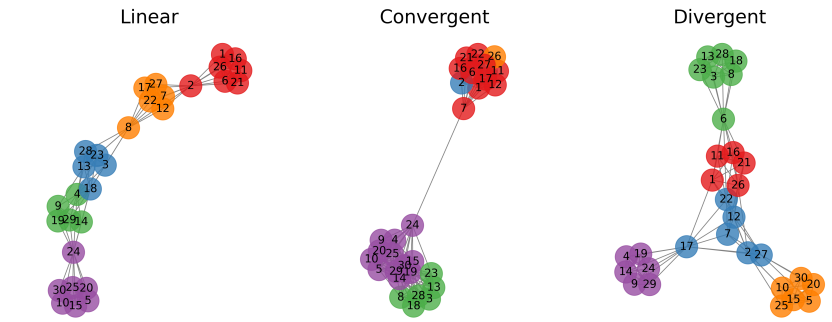
Based on the item partitions identified by the graph-clustering algorithms, a DAG structure learning procedure was applied to estimate attribute hierarchies reflected in those partitions. Each detected item cluster was assumed to represent one permissible attribute vector in the hierarchy. Within a lattice-theoretic perspective on attribute hierarchies, the hierarchy among the basic attribute vectors is equivalent to a DAG representation of the attribute hierarchy (Köhn & Chiu, 2019, 2021).

Table 2. Adjusted Rand Index (ARI) for each graph-clustering algorithm across ten datasets under the low perturbation level (sg = 0.1).

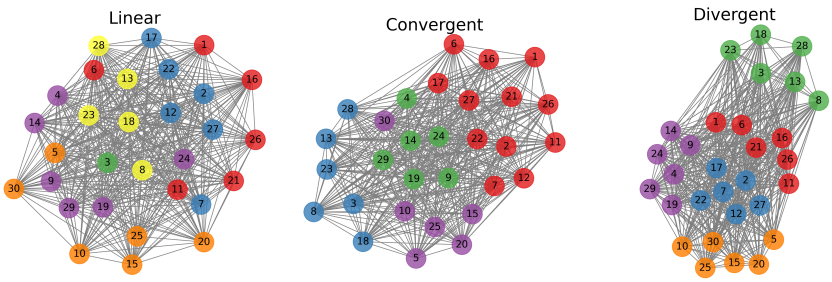
Algorithm	Dataset #	Linear	Convergent	Divergent
Spectral Clustering	1	0.75	0.60	0.75
	2	0.75	0.55	0.91
	3	0.83	0.52	0.77
	4	0.83	0.56	0.75
	5	0.83	0.58	0.83
	6	0.91	0.83	0.83
	7	0.91	0.83	0.83
	8	0.91	0.56	0.77
	9	0.83	0.46	0.75
	10	0.75	0.83	0.83
Louvain Method	1	0.96 (0.00)	0.76 (0.00)	1.00 (0.00)
	2	0.92 (0.00)	0.72 (0.00)	1.00 (0.00)
	3	1.00 (0.00)	0.70 (0.14)	1.00 (0.00)
	4	1.00 (0.00)	0.73 (0.14)	1.00 (0.00)
	5	0.96 (0.10)	0.68 (0.00)	1.00 (0.00)
	6	0.98 (0.26)	1.00 (0.00)	1.00 (0.00)
	7	0.90 (0.00)	0.94 (0.14)	1.00 (0.00)
	8	0.96 (0.00)	0.68 (0.00)	1.00 (0.00)
	9	0.90 (0.10)	0.72 (0.00)	1.00 (0.00)
	10	1.00 (0.00)	0.88 (0.00)	1.00 (0.00)
Affinity Propagation	1	1.00	1.00	1.00
	2	1.00	1.00	1.00
	3	1.00	1.00	1.00
	4	1.00	1.00	1.00
	5	1.00	1.00	1.00
	6	1.00	1.00	1.00
	7	1.00	1.00	1.00
	8	1.00	1.00	1.00
	9	1.00	1.00	1.00
	10	1.00	1.00	1.00

To apply the DAG structure learning algorithm, one item was selected from each identified cluster, and the response patterns for these items were combined to form a reduced dataset of size $N \times K$. In this reduced matrix, each column represents a cluster representative, assumed to correspond to a specific attribute vector. The DAG structure learning algorithm was then applied to this reduced item-response matrix to estimate attribute hierarchies based on the dependent relations implied by the response patterns of the selected items.

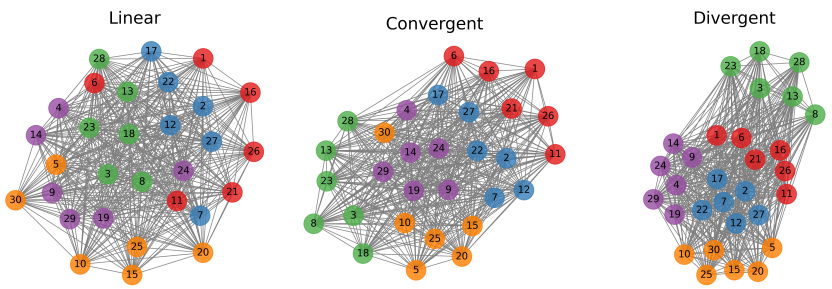
In our simulation study, item partitions derived from the affinity propagation algorithm were used to estimate the underlying attribute hierarchies, as this algorithm demonstrated strong and consistent performance across perturbation levels in identifying item groups sharing the same q-vectors, while requiring the least amount of pre-specification among the methods considered. In the design with K = 5, each item cluster contained six items, yielding 6^5 possible combinations of selecting one item from each cluster. For each combination, a score-based structure learning



(a) Spectral Clustering



(b) Louvain Method



(c) Affinity Propagation

Figure 3. An example of item partitions identified by spectral clustering, the Louvain method, and affinity propagation under the low perturbation level (sg = 0.1).

procedure (the hill-climbing algorithm) was applied to the corresponding subset of item-response data to estimate a DAG representing the underlying attribute hierarchy. Among all estimated DAGs, the one with the lowest BIC value was selected and compared with the true attribute hierarchy using the Structural Hamming Distance (SHD), defined as the number of edge differences between two adjacency matrices.

In short, for each of the ten generated datasets, the estimated DAG with the lowest BIC value was obtained using affinity propagation for clustering followed by the hill-climbing algorithm for structure learning. Table 3 presents the true adjacency matrix corresponding to the attribute hierarchies shown in Figure 2 alongside the averaged adjacency matrix across the ten optimally selected DAGs.

The accuracy of the estimated DAGs varied across hierarchy types. The mean (variance) values of the Structural Hamming Distance between the true and learned DAGs across ten datasets are

Table 3. True (top row) and estimated (bottom row) adjacency matrices for each attribute hierarchy under affinity propagation for clustering and the hill-climbing algorithm for structure learning, applied under the low perturbation level

. Rows (top to bottom) and columns (left to right) of each adjacency matrix correspond to Attributes 1–5 shown in Figure 2. Values in the estimated adjacency matrices represent edge inclusion proportions across ten optimally selected DAGs (one per dataset).

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 0.5 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0.2 \\ 0 & 0 & 0 & 1 & 0.3 \\ 0 & 0 & 0 & 0 & 0.9 \\ 0 & 0 & 0 & 0.1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.7 & 1 & 0.5 \\ 0 & 0 & 0 & 0 & 0.1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0.9 & 0.1 & 0 \\ 0 & 0 & 0.4 & 1 & 1 \\ 0 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Linear **Convergent** **Divergent**

2.2 (0.92), 3.3 (0.68), and 1.1 (0.99) for the linear, convergent, and divergent structures, respectively. Most discrepancies resulted from additional edges connecting attributes that were not adjacent in the true hierarchy but were still reachable (i.e., became identical through transitive reduction; Chen and Wang, 2023; Gu and Xu, 2023; C. Ma et al., 2023), indicating local over-connectivity, not structural misidentification. In a few cases, however, extra edges were also estimated among unrelated attributes, and several attribute pairs exhibited reversed directional relations.

4. Discussion

This study introduces a framework for estimating attribute hierarchies from item partitions constructed directly from item-response data, without reliance on parametric assumptions. By applying graph-clustering algorithms to item proximity matrices and subsequently using DAG structure learning on cluster-based item subsets, the proposed two-step, graph-theoretic method provides an exploratory approach for recovering attribute hierarchies within the CD framework. Results from preliminary computational experimentation suggested that affinity propagation was particularly effective in identifying item groups that shared the same latent attribute vectors, while spectral clustering and the Louvain method exhibited greater sensitivity to the type of attribute hierarchy.

When the clustering step successfully identified item structures corresponding to basic attribute vectors, the subsequent DAG learning step was able to recover key features of the underlying attribute hierarchies. At the same time, the DAG-based results revealed notable limitations. The accuracy of the estimated DAGs varied across hierarchy types, with discrepancies often arising from spurious edges or reversed directions between pairs of basic attribute vectors. These findings suggest that, although the proposed framework is suitable for exploratory analysis, the degree of faithfulness of the recovered hierarchies depends critically on both the quality of the item structures obtained from clustering and the performance of the structure learning algorithm.

Future work will extend simulation conditions to include more complex hierarchies with larger numbers of attributes and non-basic attribute vectors, as well as more heterogeneous Q-matrix designs with a larger number of replicated datasets. Alternative DAG structure learning algorithms also warrant investigation. In particular, alternative score-based and constraint-based structure learning procedures should be explored to improve the precision and stability of the estimated attribute hierarchies. In addition, note that the procedure developed in this study is at an experimental stage.

Therefore, we caution that the exploratory DAGs produced by the methods presented here be treated as hypotheses concerning the structural relations among items rather than as established prerequisite relations between them and their underlying attribute vectors. For example, reporting the Completed Partially Directed Acyclic Graph (CPDAG) rather than a single arbitrarily oriented DAG is advisable when full identifiability cannot be established, and leveraging asymmetries in item response patterns to resolve ambiguous orientations before drawing substantive conclusions can be an additional step.

Ultimately, the goal is to develop practical graph-theoretic tools that use item-response data directly to identify plausible attribute hierarchies grounded in item structures, thereby guiding more informed confirmatory modeling and substantive interpretation in cognitive diagnosis.

Acknowledgement

This research was supported by grants from University of Kansas ATLAS (Accessible Teaching, Learning, and Assessment Systems).

Competing Interests None.

References

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of communities in large networks. *J. Stat. Mech., P10008*.

Brito, M., Chavez, E., Quiroz, A., & Yukich, J. (1997). Connectivity of the mutual k-nearest-neighbor graph in clustering and outlier detection. *Statistics & Probability Letters*, *35*, 33–42.

Brusco, M. J., & Köhn, H. F. (2008). Optimal partitioning of a data set based on the *p*-median model. *Psychometrika*, *73*, 89–105.

Brusco, M. J., & Köhn, H. F. (2009). Exemplar-based clustering via simulated annealing: A comparison to affinity propagation and vertex substitution. *Psychometrika*, *74*, 457–475.

Chen, Y., & Li, X. (2019). Exploratory data analysis for cognitive diagnosis: Stochastic co-blockmodel and spectral co-clustering. In M. von Davier & Y.-S. Lee (Eds.), *Handbook of diagnostic classification models* (pp. 287–306). Springer.

Chen, Y., Li, X., Liu, J., Xu, G., & Ying, Z. (2017). Exploratory item classification via spectral graph clustering. *Applied Psychological Measurement*, *41(8)*, 579–599.

Chen, Y., & Wang, S. (2023). Bayesian estimation of attribute hierarchy for cognitive diagnosis models. *Journal of Educational and Behavioral Statistics*, *48*, 810–841.

de Chiusole, D., Stefanutti, L., & Spoto, A. (2017). A class of k-modes algorithms for extracting knowledge structures from data. *Behavior Research Methods*, *49*, 1212–1226.

Frey, B., & Dueck, D. (2007). Clustering by passing messages between data points. *Science*, *315*, 972–976.

Gagné, R. M. (1965). *The conditions of learning*. New York: Holt, Rinehart, Winston.

Gu, Y., & Xu, G. (2019). Learning attribute patterns in high-dimensional structured latent attribute models. *Journal of Machine Learning Research*, *20*, 1–58.

Gu, Y., & Xu, G. (2023). Identifiability of hierarchical latent attribute models. *Statistica Sinica*, *33(4)*, 2561–2591.

Haertel, E. H. (1989). Using restricted latent class models to map the skill structure of achievement items. *Journal of Educational Measurement*, *26*, 333–352.

Hanjoul, P., & Peeters, D. (1985). A comparison of two dual-based procedures for solving the *p*-median problem. *European Journal of Operational Research*, *20*, 378–396.

Hansen, P., & Mladenovic, N. (1997). Variable neighborhood search for the *p*-median. *Location Science*, *5*, 207–226.

Henson, R. A., Templin, J., & Wilse, J. T. (2009). Defining a family of cognitive diagnosis models using log-linear models with latent variables. *Psychometrika*, *74*, 191–210.

Hubert, L. (1973). Min and max hierarchical clustering using asymmetricc similarity measures. *Psychometrika*, *38(1)*, 63–72.

Hubert, L. (1974). Some applications of graph theory to clustering. *Psychometrika*, *39(3)*, 283–309.

Junker, B. W., & Sijtsma, K. (2001). Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, *25*, 258–272.

Kim, H. (n.d.). *Learning attribute hierarchies in cognitive diagnosis from data: A graph-theoretic approach* [Unpublished final report: ATLAS Project, University of Kansas].

Köhn, H. F., & Chiu, C. Y. (2019). Attribute hierarchy models in cognitive diagnosis: Identifiability of the latent attribute space and conditions for completeness of the qmatrix. *Journal of Classification*, *36*, 541–565.

Köhn, H. F., & Chiu, C. Y. (2021). A unified theory of the completeness of q-matrices for the dina model. *Journal of Classfication*, *38*, 500–518.

Köhn, H. F., Steinley, D., & Brusco, M. J. (2010). The *p*-median model as a tool for clustering psychological data. *Psychological Methods*, *15*, 87–95.

Kuehn, A. A., & Hamburger, M. J. (1963). A heuristic program for locating warehouses. *Management Science*, *9*, 643–666.
Leighton, J. P., & Gierl, M. J. (2007). *Cognitive diagnostic assessment for education: Theory and applications*. Cambridge, UK: Cambridge University Press.
Leighton, J. P., Gierl, M. J., & Hunka, S. (2004). The attribute hierarchy model: An approach for integrating cognitive theory with assessment practice. *Journal of Educational Measurement*, *41*, 205–236.
Letina, S., Blanken, T. F., Deserno, M. K., & Borsboom, D. (2019). Expanding network analysis tools in psychological networks: Minimal spanning trees, participation coefficients, and motif analysis applied to a network of 26 psychological attributes. *Complexity*, *9424605*, 205–236.
Ma, C., Ouyang, J., & Xu, G. (2023). Learning latent and hierarchical structures in cognitive diagnosis models. *Psychometrika*, *88*, 175–207.
Ma, W., & de la Torre, J. (2020). Gdina: An r package for cognitive diagnosis modeling. *Journal of Statistical Software*, *93(14)*, 1–26.
Macready, G. B., & Dayton, C. M. (1977). The use of probabilistic models in the assessment of mastery. *Journal of Educational Statistics*, *33*, 379–416.
Maranzana, F. E. (1964). On the location of supply points to minimize transportation costs. *Operations Research Quarterly*, *12*, 138–139.
Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In T. Dietterich, S. Becker, & Z. Ghahramai (Eds.), *Advances in neural information processing systems 14* (pp. 849–856, Vol. 2). Cambridge, MA: MIT Press.
Schrepp, M. (2003). A method for the analysis of hierarchical dependencies between items of a questionnaire. *Methods of Psychological Research*, *19*, 43–79.
Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*, 888–905.
Spirtes, P., Glymour, C., Scheines, R., & Tillman, R. (2010). Automated search for causal relations: Theory and practice. In R. Dechter, H. Geffner, & J. Y. Halpern (Eds.), *Heuristics, probability, and causality: A tribute to judea pearl* (pp. 467–506). College Publications.
Tatsuoka, K. K. (1985). A probabilistic model for diagnosing misconception in the pattern classification approach. *Journal of Educational and Behavioral Statistics*, *12*, 55–73.
Tatsuoka, K. K. (2009). *Cognitive assessment. an introduction to the rule space method*. New York: Routledge/Taylor & Francis.
Templin, J., & Bradshaw, L. (2014). Hierarchical diagnostic classification models: A family of models for estimating and testing attribute hierarchies. *Psychometrika*, *79(2)*, 317–339.
Tsamardinos, I., Brown, L. E., & Aliferis, C. F. (2006). The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, *65*, 31–78.
van Borkulo, C. D., Borsboom, D., Epskamp, S., Blanken, T., Boschloo, L., Schoevers, R. A., & Waldorp, L. J. (2014). A new method for constructing networks from binary data. *Scientific Reports*, *4*, 5918.
von Luxburg, U. (2007). A tutorial on spectral clustering. *Stat Comput*, *17*, 395–416.
Wang, C., & Lu, J. (2021). Learning attribute hierarchies from data: Two exploratory approaches. *Journal of Educational and Behavioral Statistics*, *46*, 58–84.

Appendix

Table 4. Adjusted Rand Index (ARI) for each graph-clustering algorithm across ten datasets under the high perturbation level (sg = 0.3).

Algorithm	Dataset #	Linear	Convergent	Divergent
Spectral Clustering	1	0.75	0.79	1.00
	2	0.69	1.00	0.91
	3	0.53	1.00	1.00
	4	0.91	0.91	0.91
	5	0.91	1.00	1.00
	6	0.83	1.00	1.00
	7	0.73	0.91	0.91
	8	1.00	1.00	1.00
	9	0.53	1.00	0.84
	10	1.00	0.91	1.00
Louvain method	1	0.87 (0.14)	0.59 (0.10)	0.93 (0.33)
	2	0.78 (0.24)	0.78 (0.28)	0.91 (0.00)
	3	0.58 (0.24)	0.76 (0.00)	0.99 (0.17)
	4	0.82 (0.24)	0.76 (0.00)	0.65 (0.10)
	5	0.75 (0.10)	0.97 (0.30)	0.84 (0.00)
	6	0.55 (0.00)	0.67 (0.33)	1.00 (0.00)
	7	0.56 (0.26)	0.75 (0.10)	0.91 (0.00)
	8	0.58 (0.00)	0.83 (0.00)	1.00 (0.00)
	9	0.63 (0.24)	1.00 (0.00)	0.84 (0.00)
	10	0.78 (0.22)	0.84 (0.00)	0.78 (0.20)
Affinity Propagation	1	0.75	0.76	1.00
	2	0.78	0.91	1.00
	3	0.66	0.91	0.84
	4	0.73	0.92	0.77
	5	0.92	0.73	0.93
	6	0.87	0.91	1.00
	7	0.43	0.83	0.91
	8	0.78	0.84	0.91
	9	0.66	0.91	0.91
	10	0.69	0.87	0.91