

Clustering Students and Inferring Skill Set Profiles with Skill Hierarchies

Alan Mishler

Department of Statistics and Data Science
Carnegie Mellon University
Pittsburgh, PA 15213
amishler@stat.cmu.edu

Rebecca Nugent

Department of Statistics and Data Science
Carnegie Mellon University
Pittsburgh, PA 15213
rnugent@stat.cmu.edu

ABSTRACT

Cognitive diagnosis models (CDMs) are a popular tool for assessing students’ mastery of sets of skills [5]. Given a set of K skills tested on an assessment, students are classified into one of 2^K latent skill set profiles that represent whether they have mastered each skill or not. Traditional approaches to estimating these profiles are computationally intensive and become infeasible on large datasets [5]. Instead, proxy skill estimates can be generated from the observed responses and then clustered, and these clusters can be assigned to different profiles [1]. Building on the work of [7], we consider how to optimally perform this clustering when not all 2^K profiles are possible, e.g. because of hierarchical relationships among the skills, and when not all possible profiles are present in the population. We compare hierarchical clustering and several k-means variants, including semisupervised clustering using simulated student responses. The empty k-means algorithm of [7] paired with a novel method for generating starting centers yields the best overall performance.

Keywords

Cognitive diagnosis models, hierarchical skills, clustering, capability scores

1. INTRODUCTION

Cognitive diagnosis models (CDMs) provide a means of estimating which skills students have and have not mastered. Given N students, K skills, and J items, let \mathbf{Y} be an $N \times J$ binary item response matrix, where Y_{ij} represents whether student i got item j correct or not. (While missing values can exist, they are outside the scope of this work.) CDMs also utilize a binary $J \times K$ Q -matrix, with each entry $q_{j,k}$ indicating whether item j requires skill k . The Q -matrix is constructed by domain experts and is typically assumed to be correct. The goal is the estimation of a latent binary vector $\alpha_i = (\alpha_{i1}, \dots, \alpha_{iK})$ for $i = 1, 2, \dots, N$, representing each student’s mastery of the skills.

CDMs are commonly estimated using maximum likelihood methods via the Expectation Maximization (EM) algorithm, or in a Bayesian setting using Markov Chain Monte Carlo (MCMC) simulation. However, these approaches are computationally intensive and may be infeasible given large numbers of students, items, or skills, or for particularly complex models [5].

As an alternative, [1] suggest using clustering methods such as k-means or hierarchical agglomerative clustering (HC) to group students into latent classes. The item response matrix and the Q -matrix are used to compute a K -dimensional vector for each student, representing performance on items requiring each of the K skills. Those vectors are then clustered to estimate the 2^K profiles. Under certain conditions, [1] show that HC is a consistent procedure, correctly partitioning students with probability converging to 1. They also show that k-means performs well in simulations.

Both HC and k-means require choosing the number of clusters, and the theoretical consistency of HC requires that each of the 2^K profiles is sampled with probability greater than 0. Since the number of profiles is exponential in K , it is likely or even inevitable that not all 2^K profiles will be present in the sample. [7] present a variant of k-means called *empty k-means* in which the number of clusters is not specified in advance and anywhere between 1 and 2^K clusters are returned. Their method outperforms HC and conventional k-means when fewer than 2^K profiles are present.

Typically there is no way to know which profiles are present in a population, but in some cases it is possible to exclude some profiles. If skills are hierarchically arranged, then students cannot master a “child” skill without mastering its “parent” skill. For example, if addition is prerequisite to multiplication, then a profile $(0, 1)$ representing mastery of addition without mastery of multiplication is not possible, but profiles $(0, 0)$, $(1, 0)$, and $(1, 1)$ are possible. [8] estimates CDMs with hierarchical skill relationships by precluding profiles that are not possible under the hierarchy and then estimating model parameters using maximum marginal likelihood. Although this approach improves model fit and item parameter estimation in several simulations [8], it is expected that it would become computationally intractable as the number of skills, items, and/or students grew.

We extend previous work by investigating the use of clustering methods in cases when not all profiles are present

and there is information about which profiles are absent, for example because of hierarchical skill relationships. These clustering approaches are computationally scalable and are agnostic to the data-generating mechanism, making them appealing in a broad range of settings.

2. METHOD

We use simulation to compare several clustering methods when fewer than 2^K profiles are present, potentially given some skill hierarchy. In this section, we describe the derivation of the capability scores, the hierarchies we examined, and the clustering methods we used.

2.1 Sum and capability scores

Recall the response vectors \mathbf{Y}_i and \mathbf{Q} -matrix defined in Section 1. For clustering, [1] define, for each student, a vector $\mathbf{W}_i = (W_{i1}, W_{i2}, \dots, W_{iK})$, where $W_{ik} = \sum_{j=1}^J Y_{ij} q_{jk}$ is the *sum score* for that student for skill k . Each sum score represents the number of items a student got right out of items that require skill k . The vectors \mathbf{W}_i therefore lie in a hyperrectangle touching the origin, with the length of side k equal to the number of items that require skill k .

Here, we follow [7] and divide each W_{ik} by k to produce *capability scores*, which are constrained to lie in the k -dimensional unit hypercube. This is intuitively appealing, since the scores serve as a proxy for the latent skill profiles, which lie at the vertices of this hypercube.

2.2 Skill hierarchies

In the CDM literature, it is typical to assume that all 2^K profiles are possible, which is tantamount to assuming that skills can be mastered in any order. This assumption is often not realistic, however, as some skills may need to be mastered before others [6]. Skill hierarchies can be represented by Directed Acyclic Graphs (DAGs), with edges indicating prerequisite relationships. Although any DAG is possible in principle, we compare four common types of skill hierarchies: *linear*, *convergent*, *divergent*, and *unstructured*. Figure 1, taken from Figure 1 in [6], shows examples of these four types for a set of six skills. We also examine the case of a null hierarchy, i.e. when the skills can be learned in any order. Let L_h be the number of possible profiles under a hierarchy h . With six skills, $L_{\text{null}} = 2^6 = 64$. The other hierarchies admit far fewer profiles: $L_{\text{linear}} = 7$, $L_{\text{convergent}} = 12$, $L_{\text{divergent}} = 16$, and $L_{\text{unstructured}} = 33$.

2.3 Clustering methods

2.3.1 Hierarchical agglomerative clustering (HC)

This method starts with each data point in its own cluster and iteratively merges the closest clusters, according to a specified distance function, until all the data points are in a single cluster. The resulting dendrogram can then be cut to produce a specified number of clusters between 1 and N . We use complete linkage as the distance function and cut the dendrogram at the point representing the largest merge distance. If this produces more than L_h clusters, then we instead cut the dendrogram so as to produce L_h clusters.

2.3.2 k -means

We use traditional k -means, with the number of clusters set to L_h ([2]) via `kmeans()` in *R* with 5 random restarts. This

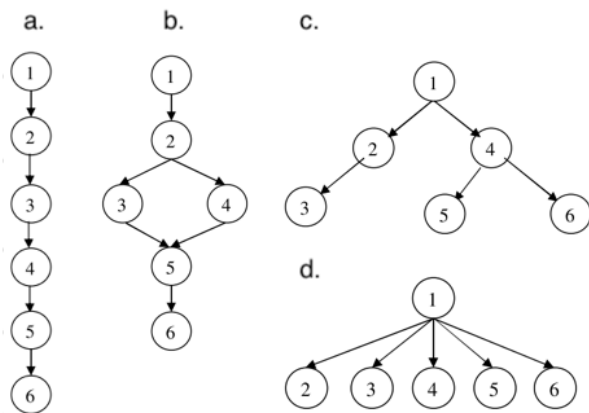


Figure 1: Four types of skill hierarchies: (a) linear, (b) convergent, (c) divergent, and (d) unstructured.

algorithm generates an error when clusters are empty. As such, we use only random starting center observations.

2.3.3 Empty k -means

The algorithm in [7], set to allow up to L_h clusters. We select starting centers by (1) random sampling, (2) the rescaling method described in [7], and (3) generating “pseudocenters” from an assumed model. The pseudocenters rely on the intuition that if the true data-generating model were known, then it could be used to select appropriate starting centers, for example by finding the expected capability score vector under each possible profile. As an approximation to this, the pseudocenters are generated by simulating data under a particular model for each possible profile and then taking the mean of each profile’s capability scores. The model used to generate the pseudocenters may be different from the model used to generate the to-be-clustered data; we consider the issue of model misspecification below.

2.3.4 Semisupervised clustering

As with the pseudocenters, pseudodata is generated under a model for each possible profile; then semisupervised clustering is performed using the pseudodata and the real data. We use the `lcqvqe()` function from the `conclust` package in *R* [3]. The LCVQE algorithm is a k -means variant that clusters labeled and unlabeled data points while attempting to preserve the clustering implied by the labels. This technique initially produces L_h clusters, but some of those clusters may end up containing only the labeled data, in which case they are discarded and a smaller number of clusters of the (originally) unlabeled data remain.

3. SIMULATION RESULTS

3.1 Framework

The numbers of skills K , items J , and students N were 6, 30, and 250, respectively. We sampled a single \mathbf{Q} matrix of 9 single-skill items (30%), 18 two-skill items (60%), and 3 three-skill items (10%). For each skill hierarchy h , subsets of profiles of size 3, 4, \dots , L_h were selected. For the linear, convergent, and divergent hierarchies, the subsets consisted of rows 1-3, 1-4, \dots , 1- L_h from the matrices in [8] repre-

senting the possible profiles. For the unstructured and null hierarchies, the subsets were sampled at random.

Item responses were generated from two CDMs, a DINA and a NIDA [5]. The item response functions are:

$$P(y_{ij} = 1 | \eta_{ij}, s_j, g_j) = (1 - s_j)^{\eta_{ij}} g_j^{1 - \eta_{ij}} \quad (\text{DINA})$$

$$P(y_{ij} = 1 | \alpha_i, s_k, g_k) = \prod_{k=1}^K [(1 - s_k)^{\alpha_{ik}} g_k^{1 - \alpha_{ik}}]^{q_{jk}} \quad (\text{NIDA})$$

where α_i is the skill set profile, $\eta_{ij} = \prod_{k=1}^K \alpha_{ik}^{q_{jk}}$ indicates whether student i has mastered all the skills necessary for item j , and s_j, s_k, g_j, g_k are slip and guess parameters indexed on items and skills, respectively. The slip parameters represent the probability of responding incorrectly even with sufficient skills; the guess parameters represent the probability of guessing correctly without those skills. The slip and guess parameters were sampled from a Uniform(0, 0.30) and a Uniform(0, 0.15) distribution, respectively. Item responses were converted to capability scores for clustering. Clustering performance was assessed via the Adjusted Rand Index (ARI), a measure of agreement between two partitions [4]. (Here, the second partition consisted of the true profile labels). The expected ARI for a pair of random partitions is 0, while identical partitions yield the maximum ARI of 1.

3.2 Results

Figures 2 and 3 shows ARIs for data generated from the DINA and NIDA, respectively. The x-axes show the proportion of profiles present out of the number L_h that are possible under each hierarchy. The four “emptyK” methods refer to empty k-means with four different sets of starting centers. The “DINA” and “NIDA” suffixes refer to the models used to generate pseudodata; these models were independent of the models used to generate the “real” data.

The DINA and NIDA data show similar results. Empty k-means with pseudocenters performed the best in nearly every case. Surprisingly, this method performed well even under model misspecification, e.g. when a NIDA was used with an underlying DINA. The hierarchical clustering method performed moderately well with small numbers of profiles, but performance dropped off quickly as that number increased. These patterns held even though both k-means and HC were constrained to a maximum of L_h clusters, a more favorable scenario than the traditional 2^K clusters.

Performance did not increase consistently in the proportion of profiles, even though the clustering methods were all set to allow up to L_h clusters, which might be expected to lead to too many clusters for small profile subsets. This may reflect the fact that more profiles make the score space more crowded, which makes it difficult to distinguish clusters.

The ARI fluctuations in the null and unstructured hierarchies are likely due to the particular profiles that were sampled. The methods appear to be closely correlated in these cases, suggesting that some subsets are harder to cluster than others, probably due to skill estimate attenuation [7].

Semisupervised clustering performed moderately well, but again, the correctly specified model did not necessarily out-

perform the misspecified model.

4. DISCUSSION AND FUTURE WORK

Previous work has suggested using k-means [1] or empty k-means [7] to partition students into classes based on their latent skill set profiles, while [8] used a likelihood-based approach to estimate CDMs in the presence of skill hierarchies. We synthesized and extended these lines of research, applying a range of clustering approaches when skill hierarchies render some profiles impossible and when not all possible profiles are present in a sample. A novel approach that combines the empty k-means algorithm with starting centers generated from pseudodata yielded the best performance, even when the generating model was misspecified.

In future work, we intend to investigate further how robust the empty k-means method with pseudocenters is to model misspecification. Both the DINA and the NIDA are conjunctive CDMs, so it is possible that performance would degrade if they were paired with disjunctive models. We plan to investigate the fluctuations in the null and unstructured hierarchy results to understand what kinds of profile subsets are easiest to distinguish. Finally, we intend to explore allowing for misspecification of the skill hierarchy, for example by translating it into a set of soft rather than hard constraints; and we will consider ways to infer the skill hierarchy when it is unknown or only partially known.

The pseudodata-based approaches have an advantage in that they yield not just partitions but profile labels, since the final clusters are associated with existing, labeled data points. With the other methods, labels must be derived by, for example, assigning clusters to the nearest licit vertex of the capability score space. We think this is a promising approach that merits further investigation.

5. REFERENCES

- [1] C.-Y. Chiu and J. Douglas. Cluster analysis for cognitive diagnosis: Theory and applications. *Psychometrika*, 74(4):633–665, 2009.
- [2] J. A. Hartigan and M. A. Wong. A K-Means Clustering Algorithm. *Applied Statistics*, 28(1):100–108, 1979.
- [3] T. K. Hiep and N. M. Duc. *conclust: Pairwise Constraints Clustering*, 2016. R package version 1.1.
- [4] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.
- [5] B. W. Junker and K. Sijtsma. Cognitive Assessment Models with Few Assumptions, and Connections with Nonparametric Item Response Theory. *Applied Psychological Measurement*, 25(3):258–272, 2001.
- [6] J. P. Leighton, M. J. Gierl, and S. M. Hunka. The attribute hierarchy method for cognitive assessment: A variation on Tatsuoka’s rule-space approach. *Journal of Educational Measurement*, 41(3):205–237, 2004.
- [7] R. Nugent, N. Dean, and E. Ayers. Skill set profile clustering: the empty K-means algorithm with automatic specification of starting cluster centers. *Proceedings of the 3rd International Conference on Educational Data Mining*, pages 151–160, 2010.
- [8] Y.-L. Su. Cognitive diagnostic analysis using hierarchically structured skills. *ProQuest Dissertations and Theses*, 2013.

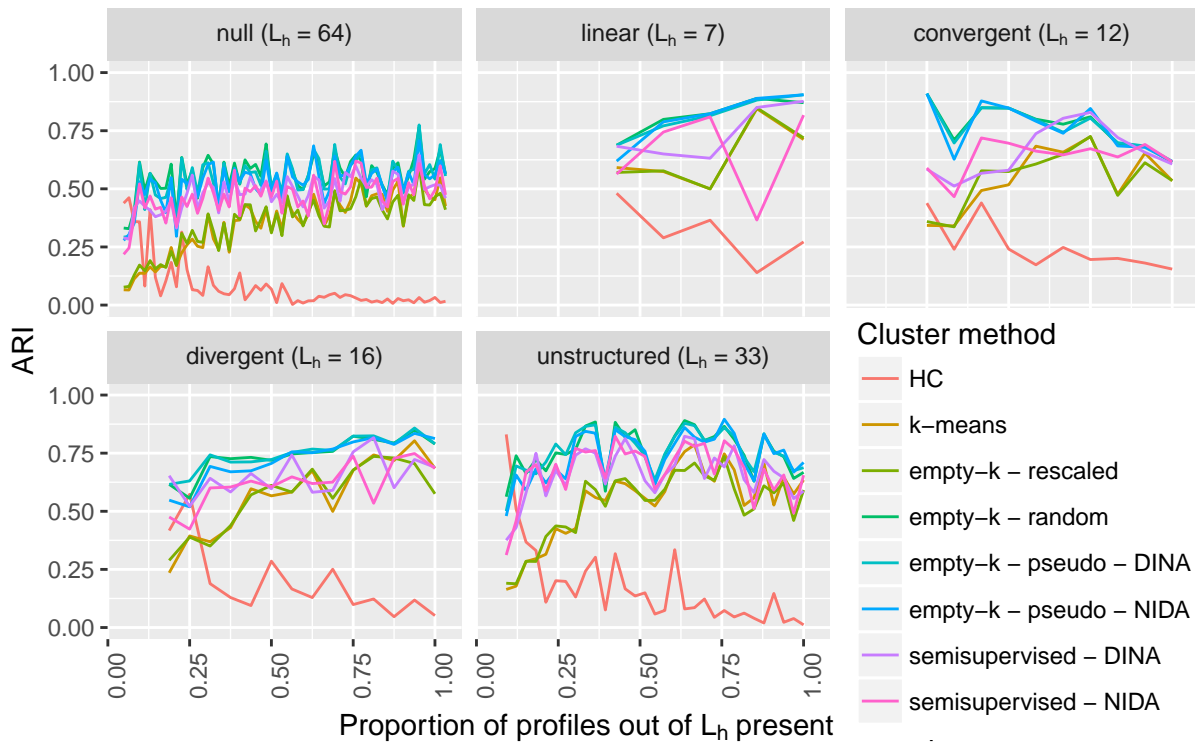


Figure 2: ARIs for five hierarchies and four clustering methods, for DINA-generated data

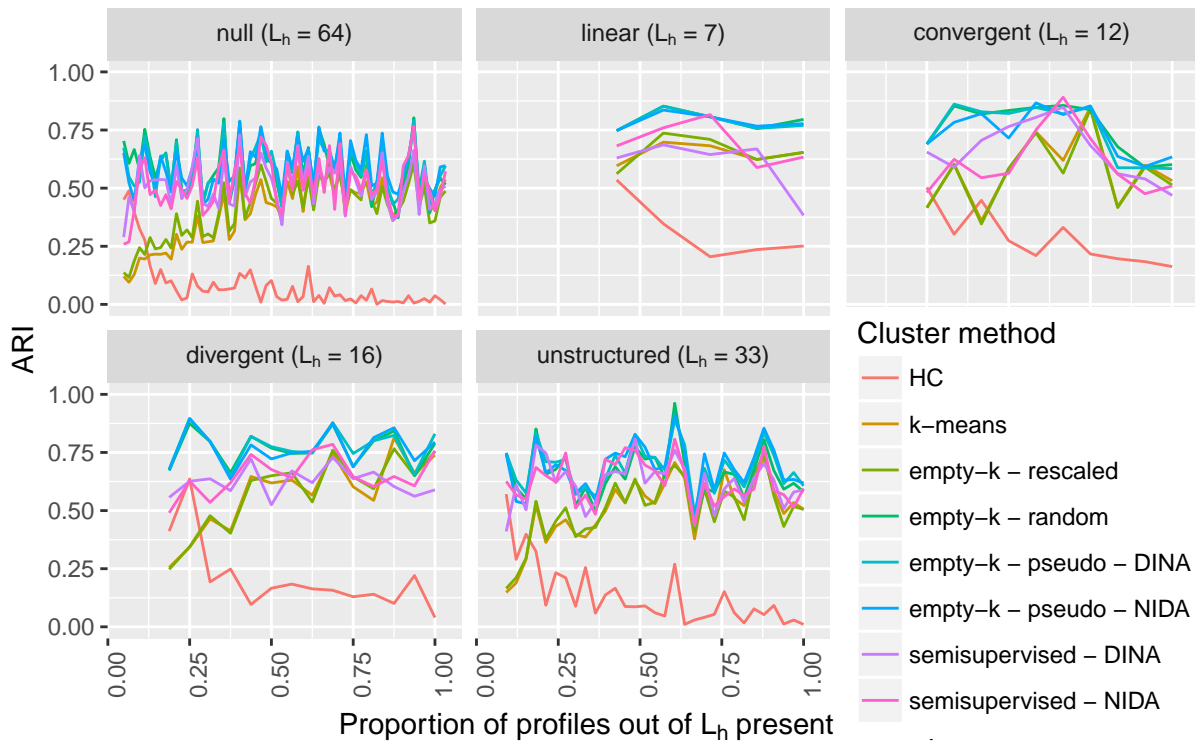


Figure 3: ARIs for five hierarchies and four clustering methods, for NIDA-generated data