Skill Set Profile Clustering: The Empty K-Means Algorithm with Automatic Specification of Starting Cluster Centers

Rebecca Nugent[1], Nema Dean[2], and Elizabeth Ayers[3]
rnugent@stat.cmu.edu, nema@stats.gla.ac.uk, eayers@berkeley.edu
[1]Department of Statistics, Carnegie Mellon University
[2]Department of Statistics, University of Glasgow
[3]Graduate School of Education, University of California, Berkeley

Abstract. While students' skill set profiles can be estimated with formal cognitive diagnosis models [8], their computational complexity makes simpler proxy skill estimates attractive [1, 4, 6]. These estimates can be clustered to generate groups of similar students. Often hierarchical agglomerative clustering or k-means clustering is utilized, requiring, for K skills, the specification of 2^K clusters. The number of skill set profiles/clusters can quickly become computationally intractable. Moreover, not all profiles may be present in the population. We present a flexible version of k-means that allows for empty clusters. We also specify a method to determine efficient starting centers based on the Q-matrix. Combining the two substantially improves the clustering results and allows for analysis of data sets previously thought impossible.

1 Introduction

A common objective in educational research is the identification of students' current skill set profiles. That is, which skills do they have? Which skills do they not have? Which skills are they in the process of learning? A variety of cognitive diagnosis models (e.g. DINA, NIDA, RUM) estimate these latent profiles using information from a student item response matrix and an expert-elicited assignment matrix of the skills required for each item [8, 10]. However, even simple models become difficult to estimate and computationally infeasible as the number of skills, items, and students grow [8].

Recent work has proposed using computationally simpler skill set estimates, e.g. capability scores and sum scores, as proxies for the cognitive diagnosis model estimates [1, 2, 4, 6]. These estimates are then clustered using common methods such as k-means and hierarchical linkage clustering to generate groups of students with similar skill set profiles. A common assumption is that all possible (combinations of complete/zero skill mastery) profiles exist in the population, a restriction that prevents us from being able to work with small samples or large numbers of skills. In addition, both capability scores and sum scores suffer from a strong dependency on a conjunctive assumption, namely that to answer an item correctly, the student must have completely mastered all necessary skills. This assumption effectively (and possibly erroneously) attenuates the individual skill set estimates in the presence of multiple skill items and relies heavily on the presence of large numbers of single skill items for reasonable estimates (which in our view is in most cases impractical).

In this paper, we propose a flexible version of k-means that utilizes more appropriate starting centers given the conjunctive assumption (conditioning on the items themselves) and allows for empty clusters, removing the restriction that each possible skill set profile will have a corresponding cluster. In our work thus far, this version outperforms both traditional k-means and hierarchical clustering in almost all situations. In Section 2, we review two skill mastery estimates and hierarchical agglomerative and k-means clustering. Details of our more flexible version of k-means are provided in Section 3; selection of sensible starting values and an illustrative example follow in Section 4. Further simulation results are in Section 5. In Section 6, we finish with concluding remarks and other possible applications.

2 Skill Set Profile Clustering

In general, the goal of cognitive diagnosis models (CDMs) is to estimate the true skill set profile for each student. Given K skills, the true skill set profile for student i is denoted α_i where $\alpha_{ik} \in \{0, 1\}$ for $k = 1, 2, ..., K$. A student that has mastered skills 1, 3 but not skill 2 would have the profile $\alpha_i = \{1, 0, 1\}$. There are 2^K possible skill set profiles for K skills; this collection of profiles is the set of corners of a K-dimensional unit hyper-cube. For example, if $K = 2$, the four possible profiles are: $\{0, 0\}, \{1, 0\}, \{0, 1\}, \{1, 1\}$.

Estimation of the α_i is done using a student response matrix Y and an item-skill dependency matrix Q. Student responses are assembled in a $N \times J$ matrix Y where y_{ij} indicates both if student i attempted item j and whether or not they answered it correctly. N is the total number of students, J the number of items. If student i did not answer item j, then $y_{ij} = NA$ (i.e. the indicator $I_{y_{ij} \neq NA} = 0$). If student i attempted item j ($I_{y_{ij} \neq NA} = 1$), then $y_{ij} = 1$ if they answered correctly (0 if not). The Q-matrix, also referred to as a skill coding or transfer model [3, 11], is a $J \times K$ matrix where $q_{jk} = 1$ if item j requires skill k and 0 if not. The Q-matrix is usually an expert-elicited assignment matrix (here assumed to be known/correct).

2.1 Skill Mastery Estimates

Here we briefly describe two proxy estimates for the CDM estimates, $\hat{\alpha}_i$: sum scores and the capability matrix. Both estimates are easily derived from the response matrix Y and the transfer model Q and have been shown to give comparable results to CDMs [2].

First, we present the *sum score* method of [4, 6]. Here W_i is defined as the vector of sum scores where, for $k = 1, 2, ..., K$,

$$W_{ik} = \sum_{j=1}^{J} y_{ij} q_{jk}.$$

The W_{ik} are simply the number of items student i answered correctly for each skill k, assuming that all students answered all items. When an item requires more than one skill, i.e., a *multiple skill item*, it contributes to more than one W_{ik}. The W_i map the students into a K-dimensional hyper-rectangle where the range of the kth dimension is $[0, J_k]$ and J_k is the total number of items that require skill k.

In [1, 2], we define an $N \times K$ *capability matrix* B, where B_{ik} is the proportion of correctly answered items involving skill k that student i attempted. That is,

$$B_{ik} = \frac{\sum_{j=1}^{J} I_{y_{ij} \neq NA} \cdot y_{ij} \cdot q_{jk}}{\sum_{j=1}^{J} I_{y_{ij} \neq NA} \cdot q_{jk}}.$$

The vector B_i estimates student i's skill set knowledge and maps student i into the same K-dimensional unit hypercube as defined by the true α_i. For each B_{ik}, zero indicates no skill mastery, one is complete mastery, and values in between are less certain. This skill knowledge estimate accounts for the number of items in which the skill appears as well as for items not answered. If $B_{ik} = NA$, we could impute an uninformative value (e.g., 0.5, mean, median). The examples presented here do not have any missing values.

In this paper, we use the capability matrix as our skill mastery estimate; however, the presented work could easily incorporate the sum score. (Comments are made where appropriate to indicate any needed changes for the use of sum scores.) In addition, estimates derived from the CDMs could similarly be analyzed.

Regardless of estimate choice, similarly to [4, 6], we find groups of students with similar skill set profiles by clustering the B_i vectors. The algorithm returns a set of cluster centers and a cluster assignment vector. The cluster center represents the skill set profile for that subset of students. Note that cluster centers are not restricted to be in the neighborhood of a hypercube corner (although they could be assigned to one). Returning cluster centers rather than their closest corners gives more conservative estimates of skill mastery (rather than zero/complete mastery). Briefly we describe two commonly used clustering methods.

2.2 Hierarchical Agglomerative Clustering

Hierarchical agglomerative clustering (HC) "links up" groups in order of closeness to form a dendrogram from which a cluster solution can be extracted [5]. The user-defined distance measure is most commonly Euclidean distance. Briefly, all observations begin as their own group. The distances between all pairs of groups are found (initially just the distance between all pairs of observations). The closest two groups are merged; the inter-group distances are then updated. We alternate the merging and updating operations until we have one group containing all observations. All merging steps are represented in a tree structure where two groups are linked at the height equal to their inter-group distance at the time of merging. The algorithm requires *a priori* the definition of the distance between two groups containing multiple observations. Here we use the complete linkage method. Complete linkage defines the distance between two groups as the largest distance between all pairs of observations, one per pair from each group, e.g., for Euclidean distance, $d(C_k, C_l) = \max_{i \in C_k, j \in C_l} ||(\underline{x}_i - \underline{x}_j)||$. It tends to partition the data into spherical shapes.

Once constructed, we extract G clusters by cutting the tree at the height corresponding to G branches; any cluster solution with $G = 1, 2, ..., N$ is possible. In [4], extraction of

$G = 2^K$ clusters is suggested. This choice may not always be wise. First, if not all skill set profiles are present in the population, we may split some profile clusters incorrectly into two or more clusters. Moreover, if $N < 2^K$ (a reasonable scenario for many end-of-year assessment exams), we will be unable to extract the desired number of skill set profiles. [4] has shown that in the presence of single skill items, hierarchical clustering will find the correct clusters under some long test theory conditions (as $N, J \to \infty$). However, this again relies on the assumption that all possible profiles are present.

2.3 K-means

K-means is a popular iterative algorithm for data $X = \{\underline{x}_1, \underline{x}_2, ..., \underline{x}_n\} \in R^K$ [9]. It uses squared Euclidean distance as a dissimilarity measure and tries to minimize within-cluster distance and maximize between-cluster distance. For a given number of clusters G, k-means searches for cluster centers m_g and assignments A that minimize the criterion $WC = \sum_{g=1}^{G} \sum_{A(i)=g} ||\underline{x}_i - m_g||^2$. The algorithm alternates between optimizing the cluster centers for the current assignment (by the current cluster means) and optimizing the cluster assignment for a given set of cluster centers (by assigning to the closest current center) until convergence (i.e. cluster assignments do not change). It tends to find roughly equal-sized, spherical clusters and requires the number of clusters G and a starting set of cluster centers. A common method for initializing k-means is to choose a random set of G observations as the starting set of centers. In this application, the suggested number of clusters is 2^K, the number of possible skill set profiles for K skills [4]. However, similarly to hierarchical clustering, if we are missing representatives from one or more skill set profiles in our population, forcing 2^K clusters may split some clusters into sub-clusters unnecessarily.

3 Empty K-Means

A traditional problem in k-means is the choice of G. A common approach is to create an "elbow graph" that plots the WC criterion against a range of proposed numbers of clusters. As increasing G almost always corresponds to a decrease in the criterion (depending on the set of starting centers), we subjectively identify the number of clusters that corresponds to the end of the large decreases in the WC value as our choice for G.

However, in this application (and others), we may have a natural number of clusters. While it may seem that we should just search for the 2^K different profiles, this number is likely just an upper bound. All profiles might not be present in the population. Moreover, without careful prior examination of the data, we will not know which profiles might be missing. Ideally, we would like a flexible approach that searches for 2^K *possible* clusters but is not forced to find them.

We modify the k-means algorithm to allow for empty clusters (or absent skill set profiles) in the following way:

1. Set the 2^K starting cluster centers m_g appropriately in the K-dim hyper-cube (Sec. 4).
2. Create the cluster assignment vector A by assigning each B_i to the closest m_g.
3. For all g, if no B_i is assigned to m_g, i.e. $\sum_i I_{A(i)=g} = 0$, then m_g remains the same. Else, $m_g = \frac{1}{n_g} \sum_{A(i)=g} B_i$.
4. Alternate between 2) and 3) until the cluster assignment vector A does not change.

This algorithm continues to minimize the WC criterion with each step; the empty clusters make no contribution to the criterion value. We discuss the choice of appropriate starting centers in Section 4.

Our k-means variation allows for empty clusters or fewer clusters than originally requested. This flexibility removes the constraint that there be one cluster per skill set profile. Early work in this area has relied heavily on small examples with $K = 2, 3, 4$ skills. With the advent of online tutoring systems and end-of-year assessment exams, the number of skills has grown considerably. It is not uncommon to be interested in $K = 10, 15, 20$, etc. For $K = 10$, say, we would have $2^{10} = 1024$ different skill set profiles. In practice, it would be extremely uncommon to see a sample with all 1024 different subgroups. Moreover, the large number of profiles computationally prohibits clustering of samples where $N < 2^K$. Our k-means variation allows for the identification of the clusters/profiles that we do have; any computational constraints (e.g. memory, storage) are limited and are a characteristic of the operating system/platform and not of the algorithm.

4 Choosing Starting Centers

It is well-known than k-means can be dependent on the set of starting centers [9]. Given our goal of identification of the true skill set profiles in the population, a natural set of starting centers might be the hypercube corners $\alpha_i = \{\alpha_{i1}, \alpha_{i2}, ..., \alpha_{iK}\}$ where $\alpha_{ik} \in \{0, 1\}$. If students map closely to their profile corners, k-means should locate the groups affiliated with the corners very quickly.

However, even if all profiles are present, the students may not be near their profile corner due to attenuation of our skill estimates in the presence of multiple skill items. Below are two possible Q matrices for $J = 24$ items. In Q_1, items 1-8 only require skill 1, items 9-16 only skill 2, and items 17-24 only skill 3 (all single skill items). In Q_2, the first 12 items are single skill; the remaining items require multiple skills. If a student's true skill set profile is $\{0, 1, 0\}$, (s)he should miss items 1-8, 17-24 in Q_1 but receive a B_{i2} of 1. In Q_2, (s)he should miss items 1-4, 9-24 which correspondingly drops B_{i2} from $\frac{13}{13}$ to $\frac{4}{13}$. Similarly, a student with profile $\{1, 0, 1\}$ will have $B_{i1} = B_{i3} = 1$ for Q_1 but see a drop in capability from $\frac{13}{13}$ to $\frac{7}{13}$ using Q_2. (Analogous drops are seen in sum scores.) These attenuated estimates are not reflective of the true profiles.

$$(Q_1)^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$(Q_2)^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

4.1 Generating Response Data

To illustrate, we generate response data for $N = 250$ students for both Q-matrices from the deterministic inputs, noisy "and" gate (DINA) model, a common educational research conjunctive cognitive diagnosis model [8]. The DINA item response form is

$$P(y_{ij} = 1 \mid \eta_{ij}, s_j, g_j) = (1 - s_j)^{\eta_{ij}} g_j^{1-\eta_{ij}}$$

where α_i is the true skill set profile and $\eta_{ij} = \prod_{k=1}^{K} \alpha_{ik}^{q_{jk}}$ indicates whether student i has all skills needed for item j. The slip parameter s_j is $P(y_{ij} = 0 \mid \eta_{ij} = 1)$; the guess parameter g_j is $P(y_{ij} = 1 \mid \eta_{ij} = 0)$. Similar to the capability matrix (and the sum score), if student i is missing any skills required for item j, $P(y_{ij} = 1)$ decreases due to the conjunctive assumption. Prior to simulating the y_{ij}, we fix the skills to be of equal medium difficulty with an inter-skill correlation of either 0 or 0.25 and generate true skill set profiles α_i for each student. (In our work thus far, only a perfect inter-skill correlation has a non-negligible effect on the results.) These choices spread students among the 2^K true skill set profiles. We randomly draw our slip and guess parameters ($s_j \sim \text{Unif}(0, 0.30)$; $g_j \sim \text{Unif}(0, 0.15)$). Given the true skill set profiles and slip/guess parameters, we then generate the student response matrix Y and estimate their corresponding capabilities.

Figure 1a below contains the capabilities estimated from the Q_1 matrix, numbered by their true profile (slightly jittered for visualization purposes). The absence of multiple skills allows the mapping of the students to (near) their profile corners. Figure 1b contains the capabilities estimated via the Q_2 matrix, also jittered, numbered by the true profile. The presence of multiple skills has pulled the non-$\{1, 1, 1\}$ profiles toward the profile $\{0, 0, 0\}$. Using the hypercube corners as the starting centers for empty k-means in the second data set will make it more difficult to find the true groups. In fact, if there are no students within a corner's octant (0.5 as the cutoff), that profile will not be found. When multiple skill items are included, the hypercube corners are no longer representative of the true profiles. We would expect their locations to be attenuated as well. Given the Q matrix, we map the true skill set profiles to their corresponding rescaled locations in the hypercube.

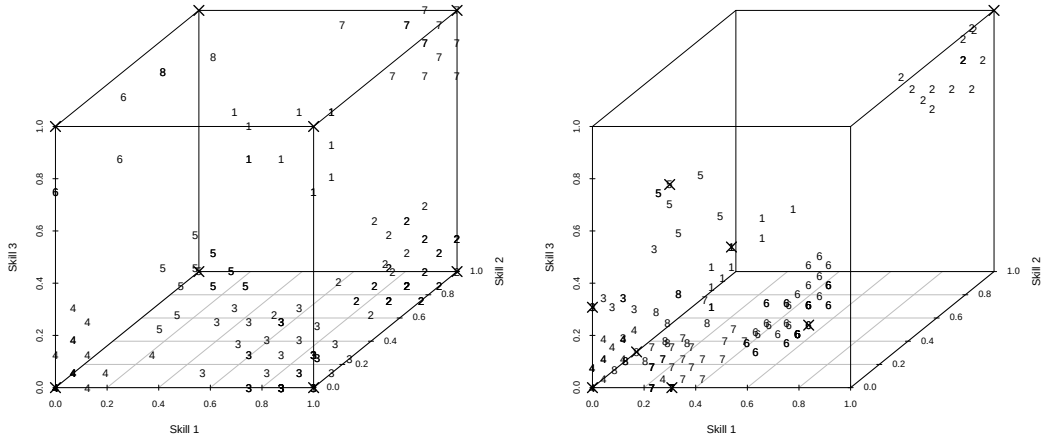

Figure 1: a) B_i for Q_1; b) B_i for Q_2; Starting centers indicated with X's

4.2 Rescaling the Starting Centers

Let α_p be the possible true skill set profiles where $p = 1, 2, ..., 2^K$ (e.g. $\{0,0\}, \{1,0\}, \{0,1\}, \{1,1\}$ for $K = 2$). Let $A_{pj} = \prod_{k=1}^{K} \alpha_{pk}^{q_{jk}}$. Then A_{pj} indicates whether or not a student with true skill set profile p has all the skills necessary to answer item j. If yes, $A_{pj} = 1$, 0 otherwise. Our starting centers C_p^* are then, for $k = 1, 2, ..K$,

$$C_{pk}^* = \frac{\sum_{j=1}^{J} I_{q_{jk}=1} \cdot A_{pj}}{\sum_{j=1}^{J} q_{jk}}.$$

The numerator is counting the number of items with skill k that the skill set profile p could answer. The denominator is the number of items requiring skill k. (Note that $\sum_{j=1}^{J} q_{jk} = J_k$. If we were using sum scores, we would not scale C_{pk}^* by the denominator.) If the Q matrix contains only single skill items, the starting centers return to the hypercube corners α_i.

In our example, the starting centers for Q_2 would be, (as indicated by X's in Figure 1b):
$(0,0,0); (4/13,0,0); (0,4/13,0); (0,0,4/13); (7/13,7/13,0); (7/13,0,7/13); (0,7/13,7/13); (1,1,1)$.
These values are representative of the true profile locations given the Q matrix if all students answered items according to their true profiles. They are derived with respect to the conjunctive assumption made by the capability matrix (and the sum score). In practice, we would expect students to slip up or make some lucky guesses; however, setting the starting centers to these rescaled profile locations will allow the empty k-means (or even traditional k-means) to easily find the groups. With respect to missing profiles, we still use the full set of C_p^* as our starting centers and allow the algorithm to discard the unnecessary ones.

Note that A_{pj} is similar in form to η_{ij} in the DINA model. Although they serve a similar function, our approach is not unique to clustering DINA-generated data. The capability score (and the sum score) are reasonable estimates for any conjunctive CDM. As we will see in Section 5, we can similarly rescale the centers for use with other CDMs.

4.3 Performance

After calculating the corresponding B matrix, we cluster the students using hierarchical clustering (complete linkage) and traditional k-means, both asking for $2^3 = 8$ clusters. We then re-cluster with traditional k-means and the empty k-means variation using the rescaled starting centers. (Note that the symmetry of the rescaled starting centers is a direct result of the balanced Q matrix; an unbalanced Q matrix will give asymmetric starting centers.)

To gauge performance, we calculate percent correct as the correct classification rate based on the best one-to-one mapping of clusters to true skill set profiles. We also quantify the clusters' agreement to the true profiles using the Adjusted Rand Index (ARI), a common measure of agreement between two partitions [7]. Under random partitioning, the expected ARI value is zero. Larger values indicate better agreement; the maximum value is one.

Table 1: Comparing Clustering Methods with the True Skill Set Profiles via % Correct, ARIs

	HC: Complete (2^3)	k-means (2^3, random)	k-means (2^3, rescaled)	k-means ($\leq 2^3$, rescaled)
% Correct	0.940	0.847	0.973	0.980
ARI	0.952	0.745	0.947	0.971

All methods performed well; the rescaled starting centers resulted in the highest percents correct and ARIs. Our k-means variation (correctly) found 8 clusters. In order to assess the performance when not all possible skill set profiles are present, we then removed the three smallest profiles $\{(0,0,1);(0,1,1);(1,0,1)\}$ (which is the most favorable situation for the other methods) and re-clustered.

Table 2: Comparing Clustering Methods with a Subset of the True Skill Set Profiles via % Correct, ARIs

	HC: Complete (2^3)	k-means (2^3, random)	k-means (2^3, rescaled)	k-means ($\leq 2^3$, rescaled)
% Correct	0.756	0.732	—	0.984
ARI	0.759	0.678	—	0.940

Again, all methods performed fairly well. Random starting centers for k-means showed a decrease in performance when clustering a subset of the profiles. Traditional k-means returned an error when using the rescaled starting centers since the initial cluster assignment returned empty clusters (as expected). Our k-means variation, however, found five clusters and had almost perfect agreement with the true skill set profiles. Even if we knew the true number of clusters (5), it is not a guarantee of superior performance. The five-cluster complete linkage solution was 93.5% percent correct with an ARI of 0.937. The traditional k-means (5 random centers) was 80.5% correct with an ARI of 0.679. Even when using only the five rescaled starting centers corresponding to the present profiles, the traditional k-means performance was comparable (97.6%, ARI = 0.946) to using our k-means variation which used the rescaled centers but required only an upper bound on the number of clusters.

5 Simulations

We explore the performance of our approach using two conjunctive CDMs while varying N, J, and K. For each simulation, the Q-matrix is randomly generated with a parameter dictating the percentage of single skill questions. We initially cluster all generated students and then remove a random number of profiles and re-cluster (the notation "—" corresponds to errors in standard k-means). We first simulate from the DINA model (Section 4.1).

Table 3: Performance with DINA-generated Responses: % Correct (ARIs)

K	N	J	Q % S/% M	Profiles Removed	HC: Complete (2^K)	k-means (2^K, random)	k-means (2^K, rescaled)	k-means ($\leq 2^K$, rescaled)
3	200	50	54/46		0.980 (0.978)	0.840 (0.776)	0.975 (0.981)	0.975 (0.981)
				(4 removed)	*0.640 (0.589)*	*0.620 (0.554)*	—	*0.900 (0.933)*
3	200	50	16/84		0.615 (0.416)	0.660 (0.352)	0.800 (0.778)	0.820 (0.814)
				(1 removed)	*0.775 (0.515)*	*0.639 (0.443)*	—	*0.835 (0.838)*
8	500	60	45/55		0.410 (0.197)	0.414 (0.166)	—	0.764 (0.655)
				(6 removed)	*0.393 (0.173)*	*0.380 (0.129)*	—	*0.236 (0.659)*
8	500	60	5/95		0.332 (0.058)	0.356 (0.074)	—	0.482 (0.199)
				(8 removed)	*0.328 (0.048)*	*0.343 (0.057)*	—	*0.468 (0.159)*
4	30	40	80/20		0.800 (0.581)	0.700 (0.390)	—	1.000 (1.000)
				(2 removed)	*0.741 (0.447)*	*0.741 (0.544)*	—	*1.000 (1.000)*

In all cases, the k-means variation with attenuated starting centers outperforms the other methods (via ARIs). We also noted in our simulations (not all presented here) that increasing the percentage of multiple skill items decreases the other methods' performance while our k-means variation remains fairly steady. Moreover, in "classroom" size data sets, this variation identified the profiles present while other methods unnecessarily split the clusters.

We also present results using responses generated from the noisy input, deterministic output "and" gate (NIDA) model, another common conjunctive CDM. The item response form is

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

where s_k, g_k are slip, guess parameters indexed on skill (rather than item); see [8] for further details. Responses are similarly generated; the results are comparable.

Table 4: Performance with NIDA-generated Responses: % Correct (ARIs)

K	N	J	Q % S/% M	Profiles Removed	HC: Complete (2^K)	k-means (2^K, random)	k-means (2^K, rescaled)	k-means ($\leq 2^K$, rescaled)
3	200	50	36/64		0.935 (0.941)	0.705 (0.622)	0.965 (0.942)	0.965 (0.942)
				(4 removed)	*0.604 (0.432)*	*0.549 (0.342)*	—	*0.549 (0.408)*
3	200	50	18/82		0.760 (0.552)	0.830 (0.688)	0.895 (0.787)	0.895 (0.787)
				(3 removed)	*0.838 (0.808)*	*0.738 (0.649)*	—	*0.900 (0.922)*
8	500	60	63/37		0.450 (0.225)	0.420 (0.163)	—	0.734 (0.663)
				(7 removed)	*0.429 (0.212)*	*0.404 (0.155)*	—	*0.753 (0.680)*
4	30	40	54/46		0.700 (0.357)	0.633 (0.321)	—	0.867 (0.661)
				(2 removed)	*0.615 (0.250)*	*0.538 (0.202)*	—	*0.846 (0.600)*

6 Conclusions

The modified k-means algorithm presented here, called "empty k-means", allows a more flexible approach to clustering for use in applications such as skill set profile clustering where the true number of clusters is not known, but may be bounded. It allows the user to specify a maximum number of possible clusters which removes the need to make a subjective decision on the number of clusters. We define our attenuated starting cluster centers by the Q-matrix (giving us the hypercube corners in the case of all single skill items). As seen in the simulated results, in cases where all natural clusters were present, such starting values gave superior clustering results (compared with both k-means with random starts and hierarchical clustering). In cases where some natural clusters were not present, the empty k-means algorithm with the defined starting values again had superior performance, while commonly traditional k-means would report an error due to empty clusters. Other applications might fit this framework as well. For example, compositional data on the simplex would have natural cluster centers on the corners of hyper-triangle. Empty k-means could also be used to investigate both the validity of theorized cluster centers and the believed number of clusters. Further exploration of this approach is ongoing.

References

[1] Ayers, E, Nugent, R, Dean, N. "Skill Set Profile Clustering Based on Student Capability Vectors Computed from Online Tutoring Data". *Educational Data Mining 2008: 1st International Conference on Educational Data Mining, Proceedings* (refereed). R.S.J.d. Baker, T. Barnes, and J.E. Beck (Eds), Montreal, Quebec, Canada, June 20-21, 2008. p.210-217.

[2] Ayers, E, Nugent, R, Dean, N. (2009) "A Comparison of Student Skill Knowledge Estimates". *Educational Data Mining 2009: 2nd International Conference on Educational Data Mining, Proceedings*. Barnes, T., Desmarais, M., Romero, C., and Ventura, S. (Eds), Cordoba, Spain, p.101-110.

[3] Barnes, T.M. (2003). *The Q-matrix Method of Fault-tolerant Teaching in Knowledge Assessment and Data Mining*. Ph.D. Dissertation, Department of Computer Science, NCSU.

[4] Chiu, C, Douglas, J. A, Li, X. (2009) Cluster Analysis for Cognitive Diagnosis: Theory and Applications. *Psychometrika*, 74 (4), p.633-665.

[5] Hartigan, J.A. *Clustering Algorithms*. Wiley. 1975.

[6] Henson, J., Templin, R., and Douglas, J. (2007) Using efficient model based sum-scores for conducting skill diagnoses. *Journal of Education Measurement*, 44, p.361-376.

[7] Hubert, L. and Arabie, P. (1985) Comparing partitions. *Journal of Classification*, 2, p.193-218.

[8] Junker, B.W., Sijtsma K. (2001) Cognitive Assessment Models with Few Assumptions and Connections with Nonparametric Item Response Theory. *Applied Psych Measurement*, 25, p.258-272.

[9] Steinley, D. (2008). Stability analysis in K-means clustering. *British Journal of Mathematical and Statistical Psychology*, 61, p.255-273.

[10] Tatsuoka, K.K. (1983). Rule Space: An Approach for Dealing with Misconceptions Based on Item Response Theory. *Journal of Educational Measurement*. 20 (4), p.345-354.