

# Conditional Subspace Clustering of Skill Mastery: Identifying Skills that Separate Students

Rebecca Nugent<sup>1</sup>, Elizabeth Ayers<sup>1</sup>, and Nema Dean<sup>2</sup>  
{rnugent, eayers}@stat.cmu.edu, {nema}@stats.gla.ac.uk

<sup>1</sup>Department of Statistics, Carnegie Mellon University

<sup>2</sup>Department of Statistics, University of Glasgow

**Abstract.** In educational research, a fundamental goal is identifying which skills students have mastered, which skills they have not, and which skills they are in the process of mastering. As the number of examinees, items, and skills increases, the estimation of even simple cognitive diagnosis models becomes difficult. We adopt a faster, simpler approach: cluster a *capability matrix* estimating each student’s individual skill knowledge to generate skill set profile clusters of students. We complement this approach with the introduction of an automatic subspace clustering method that first identifies skills on which students are well-separated prior to clustering smaller subspaces. This method also allows teachers to dictate the size and separation of the clusters, if need be, for practical reasons. We demonstrate the feasibility and scalability of our method on several simulated datasets and illustrate the difficulties inherent in real data using a subset of online mathematics tutor data.

## 1 Introduction

One of the most important classroom objectives in educational research is identifying students’ current stage of skill mastery (complete/partial/none). A variety of cognitive diagnosis models address this problem using information from a student response matrix and an expert-elicited assignment matrix of the skills required for each item [9, 12]. However, even simple models become more difficult to estimate as the numbers of skills, items, and students grow [9]. Faster methods that scale well with large datasets and provide immediate feedback in the classroom are needed. In addition, these methods also need to be able to incorporate practical information from and be interpreted by classroom teachers.

In previous work [1], we introduced a *capability matrix* showing for each skill the proportion correct on all items tried by each student involving that skill (extending the sum-score work of [3,7]) and applied two standard clustering methods to identify students with similar skill set profiles. This approach gives faster, comparable results to common cognitive diagnosis models, scales well to large datasets, and adds flexibility in skill mastery assignment (allowing for partial mastery). However, the use of clustering algorithms usually requires assumptions about the number, size, and shape of the clusters which may be unknown. Moreover, standard techniques do not allow for easy incorporation of user-specified separation and size thresholds.

In this paper, we complement our previous work by proposing an alternative approach, an automatic conditional subspace clustering algorithm that takes advantage of obvious group

separation in one or more dimensions (skills). Users do not need to specify a number of clusters nor a particular cluster shape. The method only requires a separation threshold (i.e. how far apart groups of students should be before they would be considered different) and a size threshold (i.e. what size would warrant the implementation of an additional strategy).

After describing the use of the capability matrix (Section 2), we introduce an algorithm in Section 3 that identifies skills with clearly separated groups of students (if any) and correspondingly partitions the feature space. In Sections 4, 5, we demonstrate the approach on simulated data from a common cognitive diagnosis model as well as data from the Assistance Project [6], an ongoing IES funded online mathematics tutor development research project. Finally we conclude with comments on current and future work in Section 6.

## 2 Skill Set Profile Clustering

After estimating the students' skill knowledge via the capability matrix (or other appropriate estimate), we use clustering methods to partition the students into similar skill set profiles. In recent cognitive diagnosis clustering work, hierarchical clustering, k-means, and model-based clustering have all been utilized. We do not detail the methods here (see e.g.[4,5] ) but instead briefly define and highlight strengths/weaknesses. Also, this paper's focus is the description of an automatic conditional subspace clustering algorithm; detailed comparisons of estimates' and algorithms' performances are elsewhere [submitted].

### 2.1 The Capability Matrix

The capability matrix is constructed using an item-skill dependency matrix  $Q$  and a student response matrix  $Y$ . The  $Q$ -matrix, also referred to as a transfer model or skill coding [2, 12], is a  $J \times K$  matrix where  $q_{jk} = 1$  if item  $j$  requires skill  $k$  and 0 if it does not,  $J$  is the total number of items, and  $K$  is the total number of skills. The  $Q$ -matrix is usually an expert-elicited assignment matrix. This paper assumes the given  $Q$ -matrix is known and correct. Student responses are assembled in a  $N \times J$  response matrix  $Y$  where  $y_{ij}$  indicates both if student  $i$  attempted item  $j$  and whether or not they answered it correctly and  $N$  is the total number of students. If student  $i$  did not answer item  $j$ , then  $y_{ij} = NA$  (i.e.  $I_{y_{ij} \neq NA} = 0$ ). If student  $i$  attempted item  $j$  ( $I_{y_{ij} \neq NA} = 1$ ), then  $y_{ij} = 1$  if they answered correctly (0 if not).

In [1], 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,

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

where  $y_{ij}$  and  $q_{jk}$  are the corresponding entries from the response matrix  $Y$  and  $Q$ -matrix. The vector  $B_i$  estimates student  $i$ 's skill set knowledge and then maps student  $i$  into a  $K$ -dimensional hypercube. For each dimension, zero indicates no skill mastery, one is complete mastery, and values in between are less certain. The  $2^K$  hypercube corners correspond to the true skill set profiles  $C_i = \{C_{i1}, C_{i2}, \dots, C_{iK}\}$ ,  $C_{ik} \in \{0, 1\}$ . This skill knowledge estimate accounts for the number of items in which the skill appears as well as for missing data.

If  $B_{ik} = NA$ , we impute an uninformative value (e.g., 0.5, mean, median). Exploring this choice is ongoing. Here we assume the data are complete or correctly imputed. Similarly to [3,7], we find groups of students with similar skill set profiles by clustering the  $B_i$ .

## 2.2 Hierarchical Agglomerative Clustering

Hierarchical agglomerative clustering (HC) “links up” groups in order of closeness to form a tree structure (dendrogram) from which a cluster solution can be extracted. 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. The results are represented in a tree structure where two groups are linked at the height equal to their inter-group distance. The algorithm requires *a priori* how to define the distance between two groups. Here we use the common complete linkage method. Complete linkage defines the distance between two groups as the largest distance between a pair of observations, one from each group, i.e.  $d(C_k, C_l) = \max_{i \in C_k, j \in C_l} \|\underline{x}_i - \underline{x}_j\|^2$ . 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  $G = 1, 2, \dots, N$  is possible. In [3], 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.

## 2.3 K-means

K-means is a popular iterative descent algorithm for data  $X = \{\underline{x}_1, \underline{x}_2, \dots, \underline{x}_n\} \in R^K$ . 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  $\min_A \sum_{g=1}^G \sum_{A(i)=g} \|\underline{x}_i - \bar{\underline{x}}_g\|^2$ . 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 compact, 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 our hyper-cube, another natural set of starting cluster centers could be the  $2^K$  skill set profiles at the corners. If students mapped closely to their profile corners, k-means should easily locate the nearby groups. Again,  $G = 2^K$  has been suggested [3]. However, again 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. In [1], this issue was addressed by allowing k-means to have empty clusters.

## 2.4 Model-Based Clustering

Model-based clustering (MBC) [4, 10] is a parametric statistical approach that assumes: the data  $X = \{\underline{x}_1, \underline{x}_2, \dots, \underline{x}_n\}$ ,  $\underline{x}_i \in R^K$  are an independently and identically distributed sample from an unknown population density  $p(\underline{x})$ ; each population group  $g$  is represented by a (often Gaussian) density  $p_g(\underline{x})$ ; and  $p(\underline{x})$  is a weighted mixture of these density components, i.e.  $p(\underline{x}) = \sum_{g=1}^G \pi_g \cdot p_g(\underline{x}; \theta_g)$  where  $\sum \pi_g = 1$ ,  $0 < \pi_g \leq 1$  for  $g = 1, 2, \dots, G$ , and  $\theta_g = (\mu_g, \Sigma_g)$  for Gaussian components. The method chooses the number of components  $G$  by maximizing the Bayesian Information Criterion (BIC) and estimates the means and variances  $(\mu_g, \Sigma_g)$  via maximum likelihood. While it may assume Gaussian components, its flexibility on their shape, volume, and orientation allows student groups of varying shapes and sizes. MBC also often fits overlapping components in an effort to improve fit; users are not able to specify cluster separation information and are also required to give a range of possible numbers of clusters. If multiple students map to the same hypercube location, MBC may overfit the data by using spikes with near singular covariance in these locations. To alleviate this concern (and improve visualization), we jitter the  $B_i$  a small amount (0.01). The effect on our results is minimal.

In all three cases, the algorithm returns a set of cluster centers and an assignment vector mapping each  $B_i$  to a cluster. A 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 (vs. 0/1).

As a small illustrative example, we use a subset of 26 items requiring three skills from the Assistent System online mathematics tutor [6]. The  $Q$ -matrix is unbalanced; Skill 1 (Evaluating Functions) appears in eight items (six single, two triple), Skill 2 (Multiplication) in 20 items (18 single, two triple), and Skill 3 (Unit Conversion) in two items (both triple). Overall, 551 students answered at least one item. Figure 1 shows the corresponding 3-D cube, each corner one of eight true skill set profiles. Since Unit Conversion appears in only two items,  $B_{iUC} \in \{0, \frac{1}{2}, 1\}$ ; students are mapped to three well-separated planes.

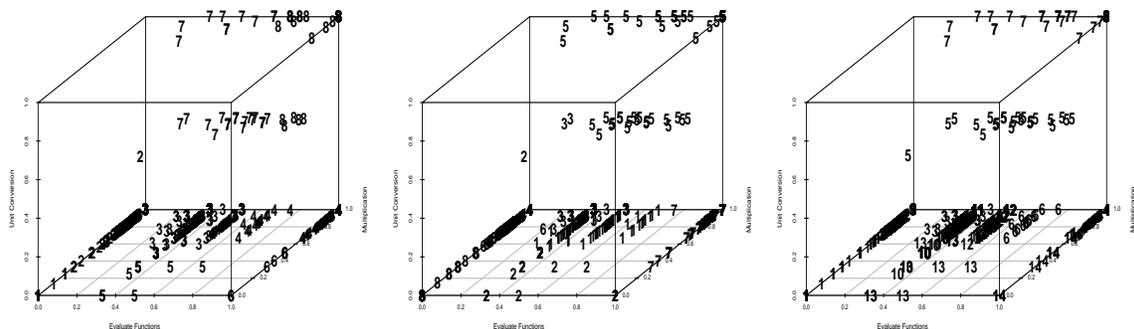


Figure 1: Cluster Assignments: a) HC, Complete  $G=8$ ; b) K-means  $G=8$ ; c) MBC  $G=14$

Figures 1a-c) show the clusters found by HC (complete), k-means, and MBC respectively. We set  $G=2^K=8$  for both HC and k-means; MBC searched over  $G=1$  to 25, choosing 14. Only MBC separated the students in the three Unit Conversion planes ( $B_{iUC}=0$ : 1-4, 6, 9-14;  $B_{iUC}=0.5$ : 5;  $B_{iUC}=1$ : 7, 8). Both HC and k-means combined students with (arguably very) different Unit Conversion capability across planes into clusters. In contrast, MBC assigns one cluster to the students with  $B_{iUC}=0.5$  and two clusters to those with  $B_{iUC}=1.0$  (the corner cluster contains multiple students). In all three solutions, the  $B_{iUC}=0$  students are split among several clusters defined by their  $B_{EF}$  and  $B_M$  capabilities. In the HC and k-means results, these clusters include one to three students with  $B_{iUC}=0.5$ .

Updating the clusters with new items, skills, etc requires minimal computational time; for example, MBC required  $\approx 21$  seconds. Classroom teachers can quickly see the changes in the students' skill knowledge over time. However, none of the three solutions seems the obvious winner. In addition, the user was only able to dictate the number of clusters (and somewhat restrict shape); no guarantees were made about their separation and size.

### 3 Conditional Subspace Clustering or “Valley-Hunting”

In general, final clusters are chosen according to a criterion or measure of closeness. Often the user has to define the number of clusters in advance which could be useful to a teacher with fixed resources. For example, he/she might ask for three groups of students clustered on their skill knowledge. However, three clusters may not represent the class well. There may be more or fewer unique skill set profiles. Moreover, the three clusters might be very similar or very different sizes (which both may be impractical). A more useful definition of a cluster might be a well-separated group of students larger than some size threshold.

While any skill's marginal distribution will always have a finite number of unique values, the marginal distribution of some skills may show very well-separated groups of students. We can take advantage of these skills by partitioning the hypercube along their marginal separations. This subsetting alone may be enough to divide students into appropriate clusters. However, it may be the case that there is multivariate cluster structure not detectable by examining the marginal distributions. As such, we advocate using this algorithm either alone or as a dimension reduction tool for other clustering methods. That is, we could first use the marginal distributions to select skills with obvious group structure and then cluster (if needed) the resulting subspaces. Reducing the dimensionality prior to clustering can greatly improve efficiency and/or results [10]. While the Figure 1 hyperplane separation is clear, it could be very difficult to identify obvious separations in a higher dimensional hypercube with noisier marginal distributions. A method to automatically find candidate skills for partitioning (and alert teachers to skills that separate the class) is more desirable.

Akin to the nonparametric clustering notion that a density's mode corresponds to a group in the population [5] and the discretization of continuous variables, we condition on a skill if its marginal distribution contains one or more “significant valleys”, a non-trivial area of low density between two high density areas. This decision is made by investigating the marginal distribution's contours. Scanning from zero to one, the low density area must be

preceded by a descent and followed by an ascent, both of gradient larger than a specified depth threshold (cluster size), and must be wider than a specified width threshold (cluster separation). There are at least two ways in which low density areas might occur. A skill only occurs in a few items and so has few possible  $B_{ik}$  values, or the  $B_{ik}$  might be centered around only a few values. If one or more significant valleys are found, we partition the hypercube at the minimum density point of each significant valley. (Other choices could be made, e.g. the halfway point between the two peaks.) In practice, we initially search for significant valleys in all skills' marginal distributions to select skills for partitioning (if any). The resulting subspaces consisting of dimensions (skills) without obvious separations are then clustered if desired; the results can be combined into one final clustering solution.

Let  $\tau_d, \tau_w$  be the respective depth and width separation thresholds (user-specified). These thresholds can be constant or differ over skills ( $\tau_{dk}, \tau_{wk}$ ). For computational ease, we use histograms to represent each skill's marginal distribution. The user may also choose a histogram bin width. The automatic subspace partitioning algorithm is as follows:

For each skill  $k$ :

Calculate the probability histogram for the given bin width. Let  $\lambda_i =$  height of Bin  $i$ .

Define the gradient  $\gamma_{i,i+1}$  as the difference in the percent of students in bins  $i, i + 1$ .

Let  $\gamma_N = \lambda_i - \lambda_j$  be the total descent gradient from a peak (Bin  $i$ ) to a valley (Bin  $j$ ).

Let  $\gamma_P = \lambda_i - \lambda_j$  be the total ascent gradient from a valley (Bin  $i$ ) to a peak (Bin  $j$ ).

Let  $L_m$  be the location of the mode preceding the current valley (scan's startpoint).

Let  $L_v$  be the location of the lowest height of the current valley.

Initialize  $L_m = L_v =$  Bin 1.

**1)** Scan  $\gamma_{i,i+1}$  until  $\gamma_{i,i+1} < 0$ .

If no such gradient exists, there are no remaining valleys.

**2)** Else, scan  $\gamma_{i,i+1}$  until  $\gamma_{i,i+1} \geq 0$  (end of valley) or out of bins; compute  $\gamma_N$ .

If  $|\gamma_N| > \tau_d$ , have found a "significant" descent. Set  $L_v =$  Bin  $i + 1$ .

**3)** Scan  $\gamma_{i,i+1}$  until  $\gamma_{i,i+1} < 0$  (end of peak) or out of bins; compute  $\gamma_P$ .

If  $|\gamma_P| > \tau_d$ , we have found a "significant" ascent. Find valley width  $w$ .

If  $w > \tau_w$ , significant valley; store mode locations. Else, do not store.

In either case, set  $L_m = L_v =$  Bin  $i + 1$ . Scan for next valley (**return to 1**).

Else, have not found significant ascent.

Scan  $\gamma_{i,i+1}$  until  $\gamma_{i,i+1} \geq 0$  (end of next valley) or out of bins.

If  $\lambda_{i+1} < \lambda_{L_v}$ , current valley is lower than valley at  $L_v$ .

Set  $L_v =$  Bin  $i + 1$ . (**return to 3**)

Else, current valley is higher than valley at  $L_v$ ; have "hiccup mode".

(**return to 3**)

Else, have not found a significant descent.

Scan  $\gamma_{i,i+1}$  until  $\gamma_{i,i+1} < 0$  (end of next peak) or out of bins.

If  $\lambda_{i+1} > \lambda_{L_m}$ , current peak is higher than peak at  $L_m$ .

Set  $L_m =$  Bin  $i + 1$ . Scan for next valley (**return to 1**).

Else, current peak is lower than peak at  $L_m$ ; have "hiccup mode". (**return to 2**)

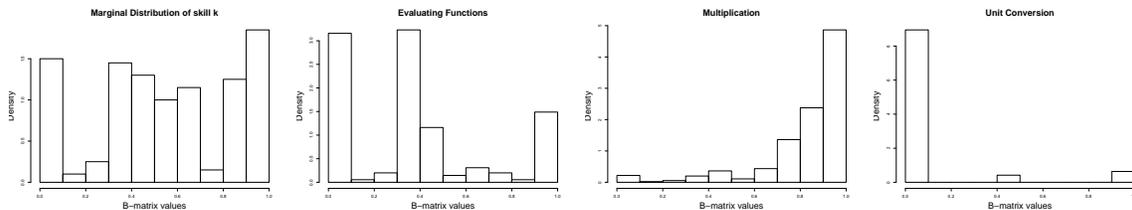


Figure 2: Marginal Skill Distributions: Illustrative Example, Three Assistent Skills

The spirit of our algorithm is similar to mode-hunting [e.g. 11] excepting that we only want to identify modes that are separated by a valley of substantial depth and width. In a sense, we are “valley-hunting”. For example, if while searching for a descent of substantial depth we find a “hiccup mode” where the marginal distribution slightly increases and then continues to decrease, the algorithm does not view that small valley to be important. (A “hiccup mode” might similarly be found when searching for a substantial ascent.) Figure 2a contains an example marginal distribution of Skill  $K$ , a histogram with bin width = 0.10. For example, say a teacher will only adapt classroom strategies for groups of students who are at least 10% of the class and whose capability values are separated by at least 20%. Given  $\tau_d = 0.1, \tau_w = 0.2$ , we start at Bin 1 and immediately find a descent of 0.14 ( $1.5 \cdot 0.10 - 0.1 \cdot 0.10$ ). We know that there is at least one bin in the preceding mode with at least 10% of the students (our depth threshold). We continue scanning to find a total ascent of 0.135 ( $1.45 \cdot 0.10 - 0.1 \cdot 0.10$ ) at Bin 4, evidence that the next mode also has at least 10% of the students. As both gradients exceed  $\tau_d$ , we check that the valley is wide enough by measuring the distance between the two modes (0.0, 0.3). Since  $0.3 > 0.2 = \tau_w$ , both modes are separated by at least 20% capability, and we have identified a “significant valley”. Continuing to scan, we find another descent and valley at Bin 6. In this case, the descent is not large enough yet to indicate a well-separated group (Bin 7 is a “hiccup mode”). A large enough descent is eventually found between Bin 4 and Bin 8, followed by a significant ascent. The next significant valley is then from Bin 4 to Bin 10. We partition the skill at Bin 2 (0.15) and Bin 8 (0.75) to create three groups of students of size at least 10% of the class separated by at least 20% capability on Skill  $K$ . If our thresholds were  $\tau_d = .045, \tau_w = 0.10$ , four groups would have been found (cutpoints: 0.15, 0.55, 0.75).

Figure 2 also includes the three Assistent skill marginal distributions. While Unit Conversion (Figure 2d) has three well-separated peaks, given reasonable depth/size thresholds, our algorithm would not partition this skill since two non-zero bin counts are very small (i.e. modes of trivial mass). We also would likely not partition the skewed Multiplication distribution. Given  $\tau_d=0.1, \tau_w=0.2$ , we do partition Evaluate Functions at 0.15, 0.75 for three groups of students and cluster the three subsequent two-dimensional subspaces. Figure 3 shows the methods’ respective results. There is less cross-plane clustering in both HC and k-means without partitioning on Unit Conversion (Figures 3a,b). MBC again chose 14 with similar results; however, the subspace clustering (including both finding the partitions and clustering the subspaces) took  $\approx 6$  seconds (vs. 21) for computational savings of 71%.

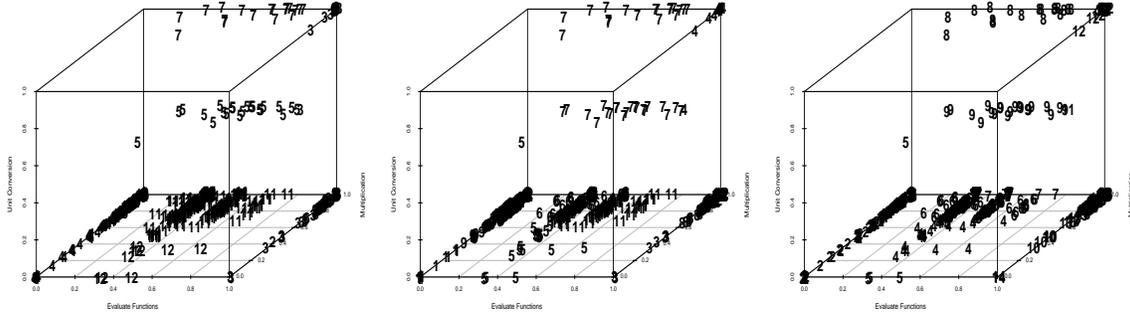


Figure 3: Cluster Assignments: a) HC, Complete  $G=3 \cdot 2^2$ ; b) K-means  $G=3 \cdot 2^2$ ; c) MBC  $G=14$

## 4 Recovering the True Skill Set Profiles

In this section, we simulate data from the DINA model, a common educational research model, to compare the methods' ability to recover the students' true skill set profiles. The deterministic inputs, noisy “and” gate model (DINA) is a conjunctive cognitive diagnosis model used to estimate student skill knowledge [9]. The DINA model item response form is  $P(y_{ij} = 1 | \eta_{ij}, s_j, g_j) = (1 - s_j)^{\eta_{ij}} g_j^{1-\eta_{ij}}$  where  $\alpha_{ik} = I_i(\text{Student } i \text{ has skill } k)$  and  $\eta_{ij} = \prod_{k=1}^K \alpha_{ik}^{q_{jk}}$  indicates if student  $i$  has all skills needed for item  $j$ ;  $s_j = P(y_{ij}=0 | \eta_{ij}=1)$  is the slip parameter; and  $g_j = P(y_{ij}=1 | \eta_{ij}=0)$  is the guess parameter. If student  $i$  is missing any of the required skills 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  $C_i$  for each student. In our work thus far, only a perfect inter-skill correlation has a non-negligible effect on the results. These parameter choices evenly spread students among the  $2^K$  natural 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 generate the student response matrix  $Y$ . Then, using a fixed  $Q$  matrix, we calculate and cluster the corresponding  $B$  matrix.

For both HC and k-means, we use  $G = 2^K$ ; in MBC, we search from 1 to  $G > 2^K$ ; in conditional subspace clustering, we initially use  $\tau_d=0.1$ ,  $\tau_w=0.2$ . To gauge performance, we calculate their agreement to the true profiles using the Adjusted Rand Index (ARI), a common measure of agreement between two partitions [5]. Under random partitioning,  $E[\text{ARI}] = 0$ , and the maximum value is one. Larger values indicate better agreement.

We present selected simulations for  $K = 3, 7, 10$  for varying  $J, N$ . Table 1 shows the ARIs comparing the clustering results to the true profiles. In the Cond (MBC) column, the first ARI corresponds to the partitioning alone, the second to the clustering of the partitioned subspaces (with MBC). We also vary the  $Q$ -matrix design to include only single skill items, only multiple skill items, or both. In addition, the  $Q$ -matrix was balanced (bal) or unbalanced (unbal). If balanced, all single skill and skill combination items occur the same number of times. Unbalanced refers to uneven representation of or missing skills (miss).

Table 1: Comparing Clustering Methods with the True Generating Skill Set Profiles

K	J	N	Q Matrix Design	HAC	K-means	MBC	Cond (MBC)	Selected Skills
3	30	250	Single	1.000	1.000	0.970	1.000	3
3	30	250	Both, bal	0.792	0.615	0.939	0.531 (0.402)	2
3	30	250	Both, unbal, uneven	0.541	0.625	0.703	0.241 (0.641)	1
3	30	250	Both, unbal, miss	0.582	0.578	0.707	0.249 (0.713)	2
3	30	250	Multiple, bal	0.414	0.419	0.416	0.222 (0.495)	1
3	30	250	Multiple, unbal, uneven	0.350	0.504	0.515	—	0
3	30	250	Multiple, unbal, miss	0.235	0.242	0.194	—	0
7	40	300	Single	0.746	0.553	0.987	0.982	7
7	40	300	Both, unbal, miss	0.333	0.308	0.386	0.290	3
10	100	2500	Single	0.876	0.786	0.062	0.958	10

Excepting the multiple unbalanced design, the subspace algorithm selected one or more skills for partitioning (in most cases, all skills were correctly selected). In almost all simulations, MBC was comparable to or better than HC and k-means for true skill set profile recovery. The partitioning method coupled with using MBC on the reduced subspaces gave comparable or better results in all cases excepting the balanced single and multiple skill design. In addition, subspace partitioning/MBC was always faster than MBC alone.

Table 2: Comparison of Depth, Width Thresholds

$\tau_d$	$\tau_w$	Cond (MBC)	Selected Skills
0.1	0.2	0.249 (0.713)	1
0.1	0.1	0.249 (0.713)	1
0.05	0.2	0.569 (0.510)	2
0.05	0.1	0.569 (0.510)	2
0.025	0.1	0.629 (0.694)	3

In addition, for the fourth  $K=3, J=30$  Q matrix design, we vary the depth and width thresholds. Smaller values of  $\tau_d, \tau_w$  will find narrower, shallower separations; in addition, smaller isolated clusters will be found. In this particular example, we found that as we decreased the depth threshold, more skills were (correctly) selected, and the performance of the partitioning by itself improved. While the parameters are designed to be user-specified, we are currently exploring their behavior in order to make good default suggestions.

## 5 Thirteen Skill Assistent Example

Finally, we briefly look at a higher dimensional Assistent example with  $K=13$  skills,  $N=344$  students, and  $J=135$  items. This data set included multiple skill items and a large amount of missing response data. HC and k-means are not appropriate choices; finding  $2^{13}=8192$  clusters is unreasonable (without, say, allowing for empty clusters as in []); MBC will largely depend on choosing an appropriate search range. The conditional subspace clustering algorithm however searches the space for obvious separation and partitions 9 of the 13 skills for a total of 221 subspaces (1 sec). All subspaces contained  $\leq 13$  students and so could likely be used alone or as subspaces for further clustering if needed.

## 6 Conclusions

We presented a conditional subspace clustering algorithm for use with the capability matrix or an alternative skill knowledge estimate. The method selects skills that separate students well and reduces dimensionality for subsequent clustering. Our work so far shows that for most Q-matrix designs, the recovery of true skill set profiles is comparable or better than other clustering methods while also including skill selection. Since the true profiles in the Assistent examples are unknown, we cannot judge their recovery. However, visual inspection indicates that the partitions seem sensible. Moreover, skills showing large student separation can be identified. Ongoing work includes designing a weighted capability matrix to better account for multiple skill items. To our knowledge, most work in this area has not adequately addressed the need for analyzing high-dimensional Q-matrices. The approach presented, while allowing for real time estimation of student skill set profiles, can handle large numbers of skills as well as incorporate practical user specifications.

## 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] 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.
- [3] Chiu, C (2008). *Cluster Analysis for Cognitive Diagnosis: Theory and Applications*. Ph. D. Dissertation, Educational Psychology, University of Illinois at Urbana Champaign.
- [4] Fraley, C. and Raftery, A. Mclust: Software for model-based cluster analysis. *Journal of Classification*, 1999, 16, 297-306.
- [5] Hartigan, J.A. *Clustering Algorithms*. Wiley. 1975.
- [6] Heffernan, N.T., Koedinger, K.R. and Junker, B.W. *Using Web-Based Cognitive Assessment Systems for Predicting Student Performance on State Exams*. Research proposal to the Institute of Educational Statistics, US Department of Education. Department of Computer Science at Worcester Polytechnic Institute, Worcester County, Massachusetts, 2001.
- [7] Henson, J., Templin, R., and Douglas, J. Using efficient model based sum-scores for conducting skill diagnoses. *Journal of Education Measurement*, 2007, 44, 361-376.
- [8] Hubert, L. and Arabie, P. Comparing partitions. *Journal of Classification*, 1985, 2, 193-218.
- [9] Junker, B.W. and Sijtsma K. Cognitive Assessment Models with Few Assumptions and Connections with Nonparametric Item Response Theory. *Applied Psych Measurement*, 2001, 25, 258-272.
- [10] Raftery, A and Dean, N. *Variable Selection for Model-Based Clustering*. *Journal of the American Statistical Association*, Vol. 101, No. 473 (March 2006), pp. 168-178.
- [11] Silverman, B. W. (1981) Using kernel density estimates to investigate multimodality. *J. Royal. Statistical Society Series B*. 43: 97-99.
- [12] Tatsuoka, K.K. (1983). Rule Space: An Approach for Dealing with Misconceptions Based on Item Response Theory. *Journal of Educational Measurement*. 1983, Vol. 20, No. 4, 345-354.