Kappa Learning: A New Item-Similarity Method for Clustering Educational Items from Response Data

Tanya Nazaretsky
Weizmann Institute of Science
Rehovot, Israel
tanya.nazaretsky@weizmann.ac.il

Sara Hershkovitz
The Center for Educational Technology
Tel Aviv-Yafo, Israel
sarah@cet.ac.il

Giora Alexandron
Weizmann Institute of Science
Rehovot, Israel
giora.alexandron@weizmann.ac.il

ABSTRACT

Sequencing items in adaptive learning systems typically relies on a large pool of interactive question items that are analyzed into a hierarchy of skills, also known as Knowledge Components (KCs). Educational data mining techniques can be used to analyze students response data in order to optimize the mapping of items to KCs, with similarity-based clustering as one of the two main approaches for this type of analysis. However, current similarity-based methods make the implicit assumption that students' performance on items that belong to the same KC should be similar. This assumption holds if the latent trait (mastery of the underlying skill) is relatively fixed during students' activity, as in the context of *testing*, which is the primary context in which these methods were developed and applied. However, in adaptive learning systems that aim for *learning*, and address subject matters such as K-6 Math that consist of multiple sub-skills, this assumption *does not* hold. In this paper we propose a new item-similarity measure, termed *Kappa Learning* (KL), which aims to address this gap. KL identifies similarity between items under the assumption of *learning*, namely, that learners' mastery of the underlying skills changes as they progress through the items. We evaluate KL on data from a K-6 Math Intelligent Tutoring System, with experts' tagging as ground truth, and on simulated data. Our results show that clustering that is based on KL outperforms clustering that is based on commonly used similarity measures (Cohen's Kappa, Yule, and Pearson), and that KL is also superior in the task of discovering the number of KCs.

Keywords

Intelligent Tutoring Systems, Adaptive Learning, Clustering Educational Items, Similarity Measurement

1. INTRODUCTION

Mastery learning [4] is based on the assumption that the domain knowledge can be analyzed into a hierarchy of component skills, with prerequisites between them [9, 10]. This structure can be used to sequence learning in an Intelligent Tutoring System (ITS) so that students master prerequisite skills before moving to skills that depend upon them [10]. *Cognitive model* is a formal representation of this structure that is encoded into the ITS. It is typically generated in a process that relies on domain experts, learning scientists, and programmers [6].

A significant part of this process is the mapping of question items into the skills that underlie them (skills are also referred to as Knowledge Components, abbreviated KCs[1]; in this paper we use the two terms interchangeably). Q-matrix is a standard representation used in Psychometrics to specify the relationships between individual test items and target skills [28]. Generating item-to-skill mapping requires a significant human-labor and expertise [13]. In addition, evidence shows that experts' mapping of items into skills can be significantly inconsistent with students' learning process [20]. Thus, methods that identify the skills underlying each item, or assist human experts in doing so, can optimize the process by increasing its accuracy and reducing human labor [14, 19].

Constructing Q-Matrix from response data is an active research topic. Barnes [3] "mined" students' data to create concept models that can be used to direct learning paths. Examples within the Psychometrics literature include [11, 21, 28]. A Matrix Factorization-based method for Q-matrix construction was proposed in [12], and was later used for enhancing expert-based Q-Matrices [14]. Learning Factor Analysis (LFA) [6] is a combinatorial search algorithm for optimizing the cognitive model while controlling for model complexity. In [22] it was demonstrated that using LFA to refine the human-generated cognitive model of an ITS improves learning gains. Performance Factor Analysis (PFA) [24] reconfigured LFA to enable predictions for individual students with individual skills (LFA assumes all students accumulate learning at the same rate), and also addresses the multiple KCs problem (standard Knowledge Tracing [10] assumes that each item requires one KC; examples of extensions that address multiple skills include [15, 32]). A different approach for 'human-in-the-loop' Student Model Discovery (finding the item-to-skill assignment that best describes students' behavior) was proposed in [27].

In general, there are two approaches for mapping items into

Tanya Nazaretsky, Sara Hershkovitz and Giora Alexandron "Kappa Learning: A New Item-Similarity Method for Clustering Educational Items from Response Data" In: *Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019)*, Collin F. Lynch, Agathe Merceron, Michel Desmarais, & Roger Nkambou (eds.) 2019, pp. 129 - 138

[1] in the Psychometrics literature, skills are also referred to as *latent factors* or *constructs*

skills: *Model-based*, and *Similarity-based* [26]. Model-based methods reduce the dimensionality of the problem and try to infer the latent factors (=skills or KCs) that underlie the items. The methods mentioned above fall into this category. Similarity-based methods are based on the assumption that students will tend to have similar performance on items that require the same skill, thus seek to identify the similarity between pairs of items. Examples of methods that are based on item-similarity measures include [2, 26]. The method that we propose in this paper falls into this category as well.

The first phase of item similarity-based methods consists of computing a similarity measure for each pair of items. This measure can be then used to cluster items, which is naturally interpreted as associating the items of a cluster with a single KC. In [26], different measures of item similarity (Pearson, Cohen's Kappa, Yule, Jaccard, and Sokal) were evaluated on real and simulated data. A different method for identifying the similarity between pairs of items, which is based on Fisher‘s Exact Test of independence, was proposed in [2] and was applied to data from an Introductory Physics MOOC. In addition to correct/incorrect information, 'item-similarity' can be based also on other behavioral characteristics, such as response-times [5, 26].

The item-similarity methods used in educational data mining for clustering items make the implicit assumption that the latent trait (mastery of the specific skill) is *fixed* during the learning activity that generated the responses (so students' responses to items that belong to the same KC should be highly correlated). This assumption may be reasonable in the context of *testing* (summative assessment), which is expected to occur *after* the learning process. (In [26], the authors explicitly refer to this shortcoming of the item-similarity measures and mention that by using these methods "we mostly ignore the issue of learning", p. 17.) However, this assumption *does not* hold in the context of *learning*. In such cases, the correlation between the items might not be a good indication of their similarity (e.g., students will tend to fail on the first items of each KC, and succeed on later ones).

The goal of this research is to address this gap by providing a measure that can capture similarity in the context of *learning*. For that, we propose a new item-similarity measure termed *Kappa Learning* (KL). The main assumption behind KL is that students' performance on items belonging to the same KC can be increasing, but not decreasing. As we use dichotomous scoring (correct/incorrect on first attempt), we expect that the performance of student s on KC k would take the form of a 'step' function, which moves from 0 to 1 when s masters k (*guess* or *slip* may occur, and introduce noise). To quantify that, KL extends the notion of 'agreement' in *Cohen's Kappa* [8].

We first make the assumption that the items are administered to the students in the same order (defined by the instructional designers), but we later explain how our formula naturally generalizes to random or adaptive ordering. We note that we do not assume that all items belonging to the same KC will be presented to students one after the other, or that all the students attempt all the items. On the contrary, we assume that students can skip items, and that items from different KCs may interleave (as in the data that we analyze), which makes the clustering non-trivial. We then compare a clustering that is based on KL to clustering that is based on the similarity measures evaluated in [26], and show that KL significantly outperforms them (Jaccard and Sokal, which achieved the lowest results in [26], and also on our data, are omitted from the analysis).

The rest of this paper is organized as follows. In Section 2, we present Cohen's Kappa and our new measure, Kappa Learning. Section 3 describes the clustering method. The details of the empirical setting and the data are provided in Section 4. In Sections 5 and 6 we evaluate the performance of Kappa Learning against standard similarity measures on real and simulated data, respectively. Finally, in Section 7 we discuss the results and suggest directions for future research.

2. COHEN'S KAPPA AND KAPPA LEARNING

2.1 Cohen's Kappa

Cohen's Kappa (sometimes abbreviated as Kappa and denoted S_k) is a measure of inter-rater agreement for nominal scales [8].

$$S_k = \frac{P_o - P_e}{1 - P_e} \tag{1}$$

where:
P_o is an *observed* level of agreement
P_e is an *expected* level of agreement

The observed level of agreement is the proportion of the cases the raters agree upon. The expected level of agreement is the proportion of agreement that is expected by chance.

We consider items as raters, learners as subjects to classify, and learners' responses as classification results. We interpret learner's correct/incorrect answer to an item (encoded as 1/0) as the rater's (=item) attempt to identify if the learner has mastered the KC underlying the item. Let us consider a contingency table summarizing learners' responses to two different items: Q_1 and Q_2. Assume n learners answered both items. The number of learners in each cell is defined as follows (Table 1):

- a - number of learners answered both Q_1 and Q_2 correctly
- b - number of learners answered Q_1 incorrectly and Q_2 correctly
- c - number of learners answered Q_1 correctly and Q_2 incorrectly
- d - number of learners answered both Q_1 and Q_2 incorrectly
- n - total number of learners ($n = a + b + c + d$)

The number of cases the raters agree upon (the learner gave the same answer to both items) is equivalent to $a + d$. Intuitively, if two different items belong to the same skill, and

Table 1: A contingency table for Q_1 and Q_2.

	Q_1 correct	Q_1 incorrect	
Q_2 correct	a	b	a + b
Q_2 incorrect	c	d	c + d
	a + c	b + d	n

learner's mastery of that skill is *fixed* during the learning activity, the learner is expected to answer both items either correctly or incorrectly, depending on whether the skill is mastered or not. So, it follows that:

$$P_o = \frac{a+d}{n}$$

The items are independent in the sense that each item independently 'rates' if a learner belongs to a category of learners knowing a particular KC. So we could compute the level of agreement that is expected by chance as a sum of products of marginal probabilities (Table 1).

$$P_e = \frac{(a+b)(a+c)+(b+d)(c+d)}{n^2}$$

By doing substitution of P_o and P_e into Equation 1 and some straightforward simplification we get:

$$S_k = \frac{2(ad-bc)}{(a+b)(b+d)+(a+c)(c+d)}$$

2.2 Kappa Learning: Adjusting Kappa to Accommodate Learning

To accommodate for learning, we give a different interpretation to the notion of 'agreement' in Cohen's Kappa formula, taking into account possible improvement of learner's skill, or in other words, learning.

We make the following assumptions on the process:

1. The items are presented to the learners in a fixed order[2].
2. The items belong to k KCs (k>1); Each item belongs to one KC; Items belonging to different KCs may interleave (which makes the clustering non-trivial)
3. Learner's success on items belonging to the same KC behaves like a 'step' function: Before mastering the skill of KC k, the student fails on items of k; once mastering the skill underlying k, the student succeeds on future items of k (*guess* and *slip* may occur; we assume no 'forgetting').

For a pair of items Q_1, Q_2, where Q_1 is presented to the learners *before* Q_2, we define the values in the contingency table (Table 1) as follows:

- a - number of learners who got both items correct, namely mastered the required skills before getting to Q_1. This is a case of agreement.
- b - number of learners who got the first item incorrect and the second item correct, namely, mastered the required skill after getting to Q_1, but before getting to Q_2. **This is a case of agreement, and is where our measure differs from Cohen's Kappa**
- c - number of learners who got the first item correct and the second item incorrect. This is the only case interpreted as *disagreement.*
- d - number of learners answered both Q_1 and Q_2 incorrectly. This is a case of agreement.
- n - total number of learners ($n = a+b+c+d$)

Based on these, we define P_o and P_e as follows:

$$P_o = \frac{a+b+d}{n}$$

$$P_e = \frac{(a+b)(b+d)+(a+b)(a+c)+(b+d)(c+d)}{n^2}$$

By doing substitution of P_o and P_e into Equation 1 and some straightforward simplification we get:

$$S_{kl} = \frac{(ad-bc)}{(a+c)(c+d)} \qquad (2)$$

We call this measure Kappa Learning and denote it S_{kl}. The values of both Kappa and Kappa Learning range between -1 and $+1$, where 0 means independence, and $+1$ means perfect agreement. In Kappa it is achieved when both c and b are equal to 0. In Kappa Learning, perfect agreement is achieved when c equals 0.

3. METHOD

3.1 Similarity Measures

To evaluate the performance of Kappa Learning (denoted S_{kl}), we compare it to the following similarity measures:

- S_k: Cohen's Kappa inter-rater agreement
- S_p: Pearson product-moment correlation coefficient
- S_y: Yule coefficient of association

Cohen's Kappa (see also in Subsection 2.1) coefficient is defined as:

$$S_k = \frac{2(ad-bc)}{(a+b)(b+d)+(a+c)(c+d)} \qquad (3)$$

Pearson product-moment coefficient is a measure of linear correlation between two variables. When applied to dichotomous data, the Pearson correlation coefficient returns the phi (ϕ) coefficient. So, in terms of a, b, c and d (Table 1) the value of S_p is computed as follows:

$$S_p = \frac{(ad-bc)}{\sqrt{(a+c)(a+b)(b+d)(c+d)}} \qquad (4)$$

Yule coefficient of association is a measure of colligation between two binary variables and it is commonly used for

[2] We later explain how this assumption can be removed

analyzing scores in Item Response Theory (IRT). It is the number of pairs in agreement (ad) minus the number in disagreement (cb) over the total number of paired observations and it is defined as:

$$S_y = \frac{(ad - bc)}{(ad + bc)} \tag{5}$$

All three measures range from minus unity to unity, where 1 indicates perfect agreement, -1 indicates perfect disagreement and 0 indicates no relationship [30]. A thorough evaluation of these measures as means for clustering items in an interactive learning environment was done by Řihák and Pelánek [26] (they analyzed the most appropriate measures among the 76 measures analyzed in [7]).

We follow a similar methodology to the one proposed in [26], described below, and demonstrate that Kappa Learning outperforms the other measures.

3.2 Process

Our process has two main steps: 1) Cluster the items based on the four similarity measures (Kappa Learning, and the three reference measures). 2) Compare the goodness-of-fit of the clusterings computed in step 1.

Step 1. Computing the clustering includes the following sub-steps (per similarity measure):

1. From students' performance data, we compute *user-based* item similarity matrix, denoted $M1$. $M1[i, j]$ contains the result of the relevant similarity measure for items q_i and q_j.
2. Compute *item-based* distance matrix from the user-based similarity matrix $M1$. The rationale is that for a pair (q_i, q_j), if q_i and q_j are similar (i.e., belong to the same KC), they should have a similar distance to a third item q_k (whether it is in the same KC or not). This incorporates more information into the similarity between the items, which should improve the accuracy of the clustering [26]. We denote the *item-based* distance matrix with $M2$. Two standard metrics are used for computing $M2$: Pearson and Euclidean.
3. Two clustering algorithms are applied on $M2$: K-Means and Ward's Hierarchical [17]. The number of clusters is derived from the hierarchal Knowledge Tree defined by content experts (see Subsection 4.2).

Step 2. Per clustering, we use Adjusted Rand Index (ARI) [16, 25] to measure the goodness-of-fit against ground truth – experts' mapping of the items into Knowledge Components.

ARI is a common measure for comparing the similarity between two clusterings. In ARI, a similarity is interpreted as the number of pairs of items on which the clusterings 'agree', adjusted for the amount of agreement 'by chance'.

To be concrete, assume C is a dataset which contains m items, with two clusterings of C into k clusters, denoted C_1 and C_2. For a pair of items (i_1, i_2), C_1 and C_2 'agree' on (i_1, i_2) iff i_1 and i_2 are either assigned to the same cluster, or to different clusters, in both C_1 and C_2.

To evaluate the level of agreement between C_1 and C_2, we define a contingency table with the values a, b, c, and d, as follows:

- a - number of pairs (i_1, i_2) where i_1 and i_2 are assigned to the *same* cluster in C_1 and in C_2. This is a case of agreement.
- b - number of pairs (i_1, i_2) where i_1 and i_2 are assigned to the *same* cluster in C_1 and to *different* clusters in C_2. This is a case of disagreement.
- c - number of pairs (i_1, i_2) where i_1 and i_2 are assigned to *different* clusters in C_1 and to the *same* cluster in C_2. This is a case of disagreement.
- d - number of pairs (i_1, i_2) where i_1 and i_2 are assigned to *different* cluster in C_1 and in C_2. This is a case of agreement.
- n - total number of pairs ($n = a + b + c + d = \frac{m(m-1)}{2}$)

Using this definition of a, b, c, and d, we can construct a contingency table similar to Table 1 for pairs of items, and compute Cohen's Kappa based on this table, which is equivalent to Adjusted Rand Index [31].

4. EMPIRICAL SETTINGS

4.1 The Learning Environment

We use data from an ITS that teaches Fractions for 4th grade. The students progress through the ITS on their own pace, in a linear order defined by the content experts. The subject matter knowledge that the ITS covers is modeled by a Knowledge Graph, which is described in Subsection 4.2 (since it is hierarchical, hereafter we use the term Knowledge Tree).

The content of the ITS includes 550 items, instructional materials such as videos, and on-line labs that students can use to explore the various concepts. These are arranged in 112 learning units. Each of the learning units contains a collection of items and learning materials and is designed to take approximately $5 - 15$ minutes.

The course is divided into two parts. Part A contains 57 learning units which include 337 items, and Part B contains 55 learning units which include 213 items. Concepts are first introduced and explained and are later re-visited. This means that items which require a certain skill can appear in different locations.

4.2 Knowledge Tree and Content Mapping

The course was designed according to a Knowledge Tree (KT) that models the hierarchy of skills that students should master (under the root topic "Fractions for 4th grade"). The KT was developed by the content experts who built the course. The first level, termed 'subject', includes 8 topics. Some of these topics have a second level, termed 'sub-subject'. On this level of the tree (sub-subject + subjects that do not have second level) there are 19 topics. The division of the first two levels is curricular (e.g., 'adding fractions' as the first level, with 'adding fractions with a common denominator' and 'adding fractions with a different denominator' as its children).

In addition to these two levels, there is a third level, termed 'goals', which is orthogonal to the classification into subjects and sub-subjects, and refers to the cognitive type of the task (inspired by Bloom's Taxonomy [1]). Since the 'goal' level is orthogonal to the division into subjects/sub-subjects (see Figure 1), it can be interpreted as refining the categories under 'subject' (hereafter denoted *first level + goals*), or as refining the 'sub-subject' (hereafter denoted *second level + goals*).

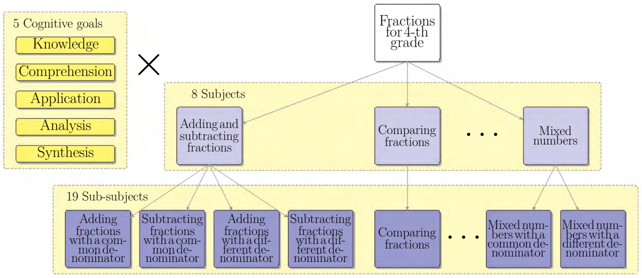
5 Cognitive goals
Knowledge
Comprehension
Application
Analysis
Synthesis
×
Fractions for 4-th grade
8 Subjects
Adding and subtracting fractions
Comparing fractions
. . .
Mixed numbers
19 Sub-subjects
Adding fractions with a common denominator
Subtracting fractions with a common denominator
Adding fractions with a different denominator
Subtracting fractions with a different denominator
Comparing fractions
. . .
Mixed numbers with a common denominator
Mixed numbers with a different denominator

Figure 1: Content expert's Knowledge tree for the topic "Fractions for 4th grade". The refining of subjects/sub-subjects into 'first level + goal'/'second level + goal' is computed as a Cartesian product of goals layer with subjects/sub-subjects layers correspondingly.

The experts tagged each item with the 'subject', 'sub-subject', and 'goal' it belongs to. In most cases, each item is mapped into one category on each level. In the few cases were an item was mapped into more than one category, we assume that each unique combination of subjects/sub-subjects is actually a new knowledge component (similar to the rationale of [15]). For example, if item i is marked as belonging to subjects 1 and 2, we create a new artificial subject for this combination of subjects. We removed from the data artificial combinations containing only one item, and the few items (< 5) that belong to these combinations.

4.3 Knowledge Components

We interpret Knowledge Component (KC) as a group of items that deal with the same concept (i.e., require the same skill) [3]. We examine classifications of items into Knowledge Components that are based on different levels of granularity with respect to the Knowledge Tree. For example, 'first level' is a classification that is based only on the first split of the tree ('subject'). Table 2 presents the number of KCs defined by each level of the KT.

4.4 Data

The data include the responses of 594 4th grade students, who used the ITS for a few hours a week during regular class hours, for a period of 2 months. (We remove the data of students who attempted less than 50 items, and the few who had less than 25% success on first attempt, as we assume they were mainly 'gaming the system'). On average, students spent about 12 hours in the ITS.

Table 2: Number of Knowledge Components by the level of Knowledge Tree.

Level of Knowledge Tree	Number of Knowledge Components
First	14
First with Goals	42
Second	32
Second with Goals	62

Students' performance is operationalized as *correct on first attempt*. From the log files, we build a 0/1 *student* × *item* response matrix, denoted RM. $RM[i, j]$==1 iff students i solved item j correctly on first attempt.

5. RESULTS ON REAL DATA

5.1 Computing the Similarity Matrix

We compute the similarity matrix for each of the four measures, as described in Section 3. This yields four similarity matrices.

To cluster the items based on these matrices, we use three clustering algorithms:

- Ward's Hierarchical clustering using Pearson correlation Distance
- Ward's Hierarchical clustering using Euclidean Distance
- K-means clustering using Euclidean Distance

As noted before, the number of clusters is defined according to the number of Knowledge Components of the Knowledge Tree (Table 2). Goodness-of-fit of clustering is evaluated by measuring its similarity to the ground truth labeling, using Adjusted Rand Index (ARI).

5.2 Results of Hierarchical Clustering

Table 3 demonstrates the results of the Hierarchical Clustering on the entire course, based on the four similarity measures, using Pearson Distance (which outperforms Euclidean Distance in all combinations; thus we omit the results for Euclidean Distance). As can be seen, clustering that is based on Kappa Learning outperforms the other measures in all the combinations.

5.3 Results of K-Means Clustering

In addition to the comparison that is based on Hierarchical Clustering, we make a comparison that is based on K-Means Clustering. Since K-Means is non-deterministic (depends on random assignment of initial cluster centers), we run the algorithm 100 times for each combination, each time computing the Adjusted Rand Index against ground truth. The distribution of the Adjusted Rand Index for each combination are presented in Figures 2 and 3. As can be seen, Kappa Learning outperforms the other similarity measures.

[3] We use the term KC in two ways – as *skill*, and as a *set of items* that require a certain skill

Table 3: Adjusted Rand Index for different similarity measures, using Hierarchical Clustering and Pearson Distance, for number of KCs that is based on different levels of the Knowledge Tree.

	First	First with Goals	Second	Second with Goals
Kappa Learning	**0.26**	**0.21**	**0.26**	**0.36**
Kappa	0.16	0.17	0.18	0.27
Yule	0.15	0.19	0.21	0.29
Pearson	0.16	0.18	0.21	0.30

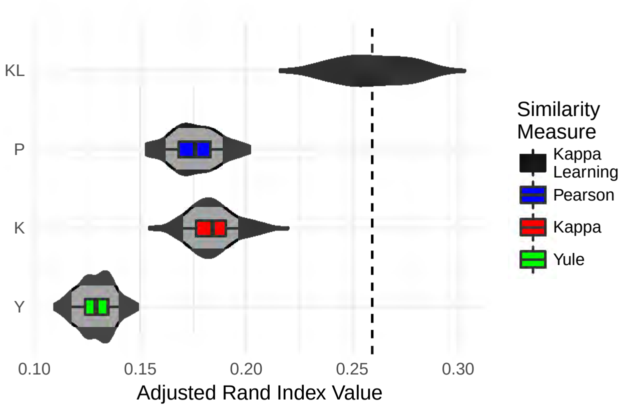

Figure 2: Results of K-means clustering for the entire course and number of KCs as defined by the second level of the Knowledge Tree. The vertical dashed line goes through the mean of the distribution of the ARI results for Kappa Learning.

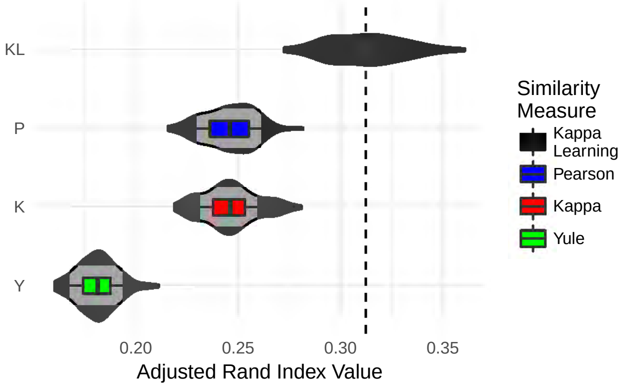

Figure 3: Results of K-means clustering for the entire course and number of KCs as defined by the second level of the Knowledge Tree + Goals. The vertical dashed line goes through the mean of the distribution of the ARI results for Kappa Learning.

5.4 Finding optimal number of clusters

While the clustering algorithms described above take the number of clusters as input, in many real-life scenarios it is unknown in advance and needs to be discovered from the data. Finding an optimal number of clusters is a fundamental problem in clustering analysis that is typically ill-posed [16], as there is no rigorous definition of a cluster, and the practical considerations are domain and application-specific. For example, in our model, we consider number of clusters that is based on different resolution of experts' hierarchical Knowledge Graph (Subsection 4.2).

The 'goodness' of the resulting clustering is usually measured by cluster cohesion and cluster separation. One of the measures for cluster cohesion or compactness is Within Cluster Sum of Squares (WSS), W_k. For any clustering of a set S into k clusters $S = \{S_1, S_2, \dots, S_k\}$, WSS is defined as

$$W_k = \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{x,y\in S_i} [dist(x,y)]^2 \tag{6}$$

where $dist(x, y)$ is a measure for distance between two items of a set.

In our case the value of W_k depends on the method used for evaluating the item's similarity matrix based on student's performance matrix, the method for measuring the distance between items of similarity matrix and the clustering algorithm used. Within Clusters Sum of Squares is commonly used to find an optimal number of clusters using the 'elbow' heuristic, however, in our case, there is no clear 'elbow' in the graph. Another common method for estimating an optimal number of clusters using WSS measure is the Gap statistic method.

5.4.1 Gap Statistic

The main idea of Gap statistic is comparing the goodness of clustering applied to a specific dataset with the goodness of clustering obtained when applied on a uniformly distributed data with no clustering structure at all (so-called 1-cluster data) [29]. The GAP_k measure used in Gap statistic is the difference between an expected value of $\log(W_k)$ computed for clustering of 1-cluster random data into k clusters and $\log(W_k)$ value obtained from clustering of input dataset into the same number of clusters k. The random data is generated from a uniform distribution over the same range as the input dataset. The Gap statistic method receives $K.max$ – the maximal number of clusters to consider, a clustering algorithm, a distance measure, and an input dataset. For each k from 1 to $K.max$, it computes GAP_k value and searches for the value of k that maximizes the Gap value.

For the four similarity matrices obtained (Section 3) we compute Gap statistic using Ward's Hierarchical Clustering, Pearson distance and $K.max = 70$. Then we apply two different methods for computing the optimal number of clusters: First SE Max (first local maximum of Gap value within one standard error) and First Max (first local maximum of Gap value) [29].

Running Gap statistic on Kappa Learning dataset produces 19 as an optimal number of clusters (see Figure 4), which is similar to the number of Knowledge Components at the Second level of the Knowledge Tree (see Figure 1; the second level of the Knowledge Tree is denoted 'sub-subjects').

The optimal number of clusters based on Yule similarity measure is 14 (with First SE Max) or 15 (with First Max),

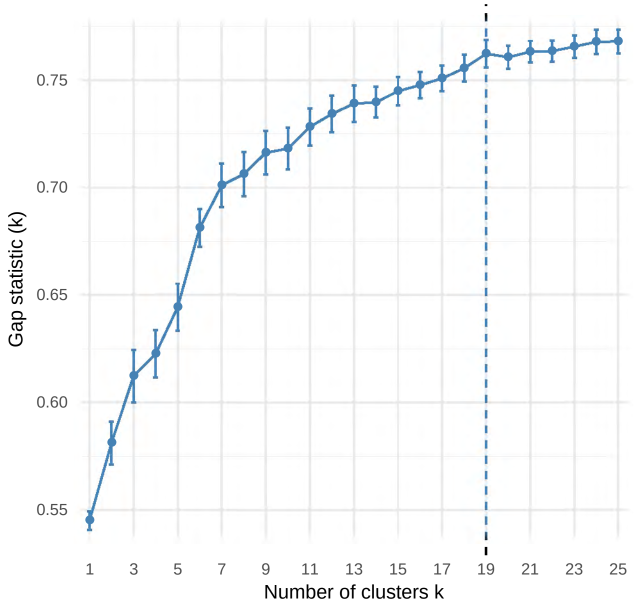

Figure 4: The values of GAP statistic for k in a range from 1 to 25 computed based on Kappa Learning similarity matrix. The vertical dashed line indicates the optimal number of clusters as predicted by both firstMax and firstSEMax methods.

which is also quite close to the ground truth. For the two other methods (Cohen's Kappa, Pearson), Gap statistic does not produce meaningful results (Table 4).

Table 4: Optimal number of clusters by GAP statistic.

	First Max Method	First SE Max Method
Kappa Learning	**19**	**19**
Kappa	1	1
Yule	15	14
Pearson	1	1

6. SIMULATION STUDY

In addition to evaluating our method on data from a real learning environment (Subsection 4.1), we conduct a simulation study.

6.1 Data Generation

Our simulation model makes the following assumptions:

1. Each item belongs to one of K knowledge components (KCs); the items are uniformly distributed among these KCs. Each KC has an individual difficulty level (drawn from a probability distribution defined below).
2. The order of appearance of KCs is predefined. We assume that the topic is first presented and explained to the learners, so the majority ($\approx 60\%$, chosen empirically based on the data) of the items that belong to it appear one after the other. The rest of the items that belong to the KC are presented to the learner on a later stage and interleaved between items from other KCs.
3. Students learn as they interact with the items; Learners have individual learning rate (drawn from a probability distribution defined below).

6.1.1 Hidden Markov Model and Bayesian Knowledge Tracing

Bayesian Knowledge Tracing (BKT) [10] is a popular approach to model skill acquisition in ITSs. It models a student knowledge as a latent binary variable of a Hidden Markov Model. Learning is modeled as a transition from 'not mastered' to 'mastered' state. The standard BKT model uses the same four parameters for all the students and items. Several studies extended the basic BKT model with individualized parameters for student ability and item difficulty (e.g., [18, 23, 33]). We use the model introduced in [33] as the underlying model for the data generation process.

6.1.2 Individualized Bayesian Knowledge Tracing

We apply Individualized BKT approach with parameter splitting [33] to model a learning process. Namely, we construct individual HMM per student and KC. All items of the same KC are assumed to have the same difficulty. The model assumes students learn as they practice more. On each opportunity to solve an item that belongs to a knowledge component, the probability that the student masters the skill underlying the item's KC increases.

Let us define:

- L - number of learners
- K - number of Knowledge Components
- N - total number items (questions)

For each KC k and each student l we generate the following parameters:

- $P(L_0)$ - the probability that a student initially knows a particular KC. In this model we assume the students have no initial knowledge.
- $P(T)_l^k$ - the probability of learning for student l and skill k.
- $P(S)$ - the probability of slip, meaning making an incorrect attempt when applying a known skill. We assume $P(S) = 0.1$ (not individualized; determined by an educational expert).
- $P(G)$ - the probability of random guess, meaning making a correct attempt when applying an unknown skill. We assume $P(G) = 0.2$ (not individualized; determined by an educational expert).

As proposed in [33], the value of the parameters $P(T)_l^k$ is combined from two components: a per-skill component and a per-student component. So, we generate for each skill and each student a pair of parameters $(P(T)_l, P(T)^k)$. Each of the above parameters is generated from a uniform distribution $\mathcal{U}(0,1)$. Then for each student l and KC k the parameters are combined as follows:

$$P(T)_l^k = \sigma(l(P(T)_l) + l(P(T)^k)) \tag{7}$$

where:

$$\sigma(x) = 1/(1 + exp^{-x}) \tag{8}$$

and

$$l(x) = \log{(x/1 - x)} \tag{9}$$

Where $\sigma(x)$ and $l(x)$ are the sigmoid and logit functions, respectively.

The performance matrix for each student and knowledge concept is generated using R's HMM package, and the data is combined into a $L \times N$ student's performance matrix. For all Knowledge Components containing more than 6 items, the first 6 items are placed one after another, modeling the introduction of the concept to the learners. The rest of the items are shuffled randomly between future KCs.

6.1.3 Model Parameters

In the experiment reported below the basic setting is 1000 learners, 20 Knowledge Components, 200 items. The parameters are chosen in a way that approximates the multivariate distribution of the real data with respect to the average number of items per Knowledge Component and the mean performance of students, as illustrated in Table 5.

Table 5: Comparison of simulation model to empirical data.

	Average number of Questions per KC	Average performance of Students
Empirical data	≈9	67%
Simulation model	10	64%

To evaluate the clustering that is based on each of the four measures, we follow the same process as described in Section 3. Since the results depend on the simulated data, we repeat the process 700 times, each time starting with generating a new performance matrix.

6.2 Results on Simulated Data

The results are presented in Table 6 (right column) and Figure 5. As can be seen, Kappa Learning outperforms all other measures in its ability to reproduce the original clusters. Table 6 also presents the results of each measure on the real data (left column), for reference.

To verify the statistical significance of the results, we conduct a t-test for the results of Kappa Learning vs. the three other measures (Yule, Cohen's Kappa, and Pearson). For all combinations, the p-value is less than 0.01.

7. DISCUSSION

The results show that Kappa Learning - the new similarity measure that we propose, which is based on adjusting Cohen's Kappa to 'learning', can improve the clustering of educational items into Knowledge Components, compared to the state-of-the-art (the measures that are reported in [26] as producing the best results). We ascribe this to the fact that Kappa Learning explicitly models similarity under the assumption that students' skill can grow during the activity (= learning), while the conventional measures are based on the assumption that students' skill is fixed.

Table 6: Comparison of Adjusted Rand Index values for different similarity measures.

	Real data, Second level with Goals	Simulation Model (averaged over 700 runs)
Kappa Learning	**0.36**	**0.40**
Kappa	0.27	0.35
Yule	0.29	0.31
Pearson	0.30	0.37

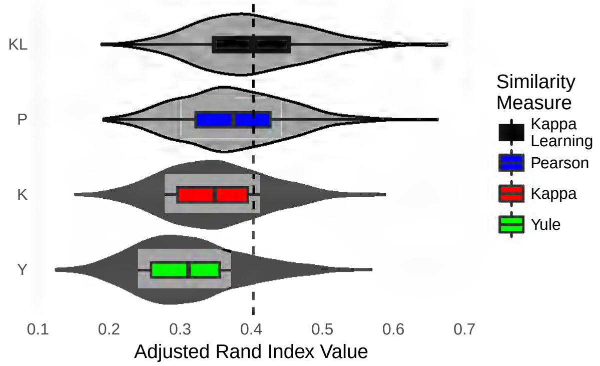

Figure 5: Distribution of Adjusted Rand Index values for different similarity measures (KL - Kappa Learning, K - Kappa, Y - Yule, P - Pearson). The vertical dashed line goes through the mean of the distribution of ARI values for Kappa Learning.

On real data, with different combinations for the number of clusters (Table 2), the improvement with Hierarchical Clustering was in the range of $10 - 60\%$ (Table 3), comparing to the conventional measures (Kappa, Yule, and Pearson). On simulated data that follow the 'mastery' assumption, and allow items of different Knowledge Components to interleave (which makes the task more difficult; if all the items of a certain KC are presented together, the clustering is almost trivial), the improvement with Hierarchical Clustering was in the range of $10 - 20\%$ (Table 6).

In real-life scenarios, the number of clusters, which the clustering algorithms that we use take as input, is typically unknown, and it is necessary to extract it from the data. On the task of finding an optimal number of clusters, Gap statistic on clustering that is based on Kappa Learning yielded a number of clusters (19) that is similar to the number of clusters in the ground truth (according to the second level of the Knowledge Graph. See Table 2). Among the other measures, Gap statistic on Yule-based clustering also produced results that are reasonably close to the ground truth. For

Kappa and Pearson, Gap statistic did not yield meaningful results.

Overall, Kappa Learning was superior with respect to all the combinations that were evaluated: Various interpretations of the ground truth (deciding on the KCs according to different levels of Knowledge Graph); clustering algorithm – K-Means and Hierarchical Clustering; real and simulated data; and in reproducing the number of clusters with Gap statistic. Thus, we conclude that in the context of learning in structured domains (such as K-6 Math), Kappa Learning provides a significant improvement to the task of clustering that is based on item similarity, compared to conventional item-similarity measures.

7.1 Generalizing to Random Order of Items

In Subsection 2.2, the definition of Kappa Learning was based on the assumption that the items are presented to the learners in a fixed order. We now explain how this assumption can be removed. Let us assume that the items administered to the learners in a random order, meaning that different learners may see the items in a different order. In this case, for each learner and for each pair of items we construct the contingency table (similar to Table 1) by computing the values of a, b, c, d as follows:

- a - number of learners who got both items correct
- b - number of learners who got the first item presented to them (among Q_1 and Q_2) correct, and the second incorrect
- c - number of learners who got the first item presented to them (among Q_1 and Q_2) incorrect, and the second correct
- d - number of learners who got both items incorrect

7.2 Future Work

This work provides a few directions for future research. On the next step, we intend to work with the developers of the ITS on using the results of Kappa Learning to refine and optimize the pedagogic design of the ITS (cognitive modeling, but also questions such as which KCs require more content, are too difficult, too easy, etc.).

Algorithmic directions include studying additional ways to insert the notion of 'learning' into existing item-to-skill detection methods, and additional sources of information such as domain experts or analysis of the body of the items (text, images, mathematical symbols, etc.).

In terms of use cases, it would be interesting to evaluate Kappa Learning on data from a variety of learning environments (e.g., MOOCs) and subject matters, and in particular, on domains in which knowledge is less structured (e.g., reading comprehension).

7.3 Summary and Conclusions

This paper presents a new method for measuring the similarity between educational items, termed Kappa Learning. The novelty of this method, compared to previous measures of similarity between educational items, lies in the fact that it explicitly captures the notion of 'learning', namely, change of the latent trait (student's mastery of the concept). This is done by extending the notion of 'agreement' within Cohen's Kappa basic formula.

Our results show that clustering that is based on Kappa Learning outperforms clustering that is based on conventional methods (Cohen's Kappa, Yule, Pearson), on real data from K-6 Math ITS that teaches multiple concepts, and on generated data that simulates learning of multiple, interleaved concepts. Thus, we believe that Kappa Learning is more suitable than existing measures for computing similarity between items in the context of learning in structured domains.

8. ACKNOWLEDGMENTS

This research was supported by The Willner Family Leadership Institute for the Weizmann Institute of Science, Iancovici-Fallmann Memorial Fund, established by Ruth and Henry Yancovich, and by Ullmann Family Foundation. The authors would like to thank The Center for Educational Technology for giving us access to the data, and to Hagar Rubinek and Yael Weisblatt for their help in conducting this research. The authors would like to thank Ido Roll for useful comments.

9. REFERENCES

[1] L. W. Anderson, D. R. Krathwohl, P. W. Airasian, K. A. Cruikshank, R. E. Mayer, P. R. Pintrich, J. Raths, and M. C. Wittrock. A taxonomy for learning, teaching, and assessing: A revision of bloom's taxonomy of educational objectives, abridged edition. *White Plains, NY: Longman*, 2001.

[2] T. Balint, R. Teodorescu, K. Colvin, Y.-J. Choi, and D. E. Pritchard. Identifying characteristics of pairs of questions that students answer similarly. In *Physics Education Research Conference 2015*, pages 55–58, 2015.

[3] T. Barnes. The q-matrix method: Mining student response data for knowledge. In *American Association for Artificial Intelligence 2005 Educational Data Mining Workshop*, pages 1–8, 2005.

[4] B. S. Bloom. Learning for mastery. *Instruction and Curriculum*, 1968.

[5] P. Boroš, J. Nižnan, R. Pelánek, and J. Řihák. Automatic detection of concepts from problem solving times. In H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *Artificial Intelligence in Education*, pages 595–598, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[6] H. Cen, K. Koedinger, and B. Junker. Learning factors analysis – a general method for cognitive model evaluation and improvement. In M. Ikeda, K. D. Ashley, and T.-W. Chan, editors, *Intelligent Tutoring Systems*, pages 164–175, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[7] S.-s. Choi and S.-h. Cha. A survey of binary similarity and distance measures. *Journal of Systemics, Cybernetics and Informatics*, pages 43–48, 2010.

[8] J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[9] A. T. Corbett and J. R. Anderson. Student modeling and mastery learning in a computer-based programming tutor. In C. Frasson, G. Gauthier, and G. I. McCalla, editors, *Intelligent Tutoring Systems*, pages 413–420, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg.

[10] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.

[11] J. de la Torre. An empirically based method of q-matrix validation for the dina model: Development and applications. *Journal of Educational Measurement*, 45(4):343–362, 2008.

[12] M. C. Desmarais, B. Beheshti, and R. Naceur. Item to skills mapping: Deriving a conjunctive q-matrix from data. In S. A. Cerri, W. J. Clancey, G. Papadourakis, and K. Panourgia, editors, *Intelligent Tutoring Systems*, pages 454–463, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[13] M. C. Desmarais and M. Gagnon. Bayesian student models based on item to item knowledge structures. In W. Nejdl and K. Tochtermann, editors, *Innovative Approaches for Learning and Knowledge Sharing*, pages 111–124, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[14] M. C. Desmarais and R. Naceur. A matrix factorization method for mapping items to skills and for enhancing expert-based q-matrices. In H. C. Lane, K. Yacef, J. Mostow, and P. Pavlik, editors, *Artificial Intelligence in Education*, pages 441–450, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[15] Y. Huang, J. P. González-Brenes, and P. Brusilovsky. General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge. In *Proceedings of the 7th International Conference on Educational Data Mining, EDM 2014*, pages 84–91, 2014.

[16] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[17] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[18] M. Khajah, R. Wing, R. Lindsey, and M. Mozer. Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. In *Proceedings of the 7th International Conference on Educational Data Mining, EDM 2014*, pages 99–106, 2014.

[19] K. R. Koedinger, E. A. McLaughlin, and J. C. Stamper. Automated student model improvement. In *Proceedings of the 5th International Conference on Educational Data Mining, EDM 2012*, pages 17–24, 2012.

[20] S. Lee, Z. Chen, D. Pritchard, A. Kimn, and A. Paul. Factor analysis reveals student thinking using the mechanics reasoning inventory. In *Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale*, L@S '17, pages 197–200, 2017.

[21] J. Liu, G. Xu, and Z. Ying. Data-driven learning of q-matrix. *Applied Psychological Measurement*, 36(7):548–564, 2012.

[22] R. Liu and K. R. Koedinger. Closing the loop: Automated data-driven cognitive model discoveries lead to improved instruction and learning gains. *Journal of Educational Data Mining*, 9(1):25–41, 2017.

[23] Z. A. Pardos and N. T. Heffernan. Kt-idem: introducing item difficulty to the knowledge tracing model. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 243–254. Springer, 2011.

[24] P. I. Pavlik, H. Cen, and K. R. Koedinger. Performance factors analysis –a new alternative to knowledge tracing. In *Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems That Care: From Knowledge Representation to Affective Modelling*, pages 531–538. IOS Press, 2009.

[25] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[26] J. Rihak and R. Pelanek. Measuring Similarity of Educational Items Using Data on Learners' Performance. In *Proceedings of the 10th International Conference on Educational Data Mining, EDM 2017*, pages 16–23, 2017.

[27] J. C. Stamper and K. R. Koedinger. Human-machine student model discovery and improvement using datashop. In G. Biswas, S. Bull, J. Kay, and A. Mitrovic, editors, *Artificial Intelligence in Education*, pages 353–360, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[28] K. K. Tatsuoka. Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 20(4):345–354, 1983.

[29] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[30] M. J. Warrens. On association coefficients for 2× 2 tables and properties that do not depend on the marginal distributions. *Psychometrika*, 73(4):777, 2008.

[31] M. J. Warrens. On the equivalence of cohen's kappa and the hubert-arabie adjusted rand index. *Journal of Classification*, 25(2):177–183, 2008.

[32] Y. Xu and J. Mostow. Comparison of methods to trace multiple subskills: Is lr-dbn best? In *Proceedings of the 5th International Conference on Educational Data Mining, EDM 2012*, pages 41–48, 2012.

[33] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon. Individualized bayesian knowledge tracing models. In *International Conference on Artificial Intelligence in Education*, pages 171–180, 2013.