Journal of Universal Computer Science, vol. 21, no. 1 (2015), 93-109
submitted: 16/5/14, accepted: 1/10/14, appeared: 1/1/15 © J.UCS

Towards a Learning-Aware Application Guided by Hierarchical Classification of Learner Profiles

Behnam Taraghi, Anna Saranti, Martin Ebner
(Graz University of Technology
Graz, Austria
{b.taraghi, martin.ebner}@tugraz.at, s0473056@sbox.tugraz.at)

Vinzent Müller, Arndt Großmann
(UnlockYourBrain GmbH
Berlin, Germany
{vinzent.mueller, arndt}@unlockyourbrain.com)

Abstract: Learner profiling is a methodology that draws a parallel from user profiling. Implicit feedback is often used in recommender systems to create and adapt user profiles. In this work the implicit feedback is based on the learner's answering behaviour in the Android application *UnlockYourBrain*, which poses different basic mathematical questions to the learners. We introduce an analytical approach to model the learners' profile according to the learner's answering behaviour. Furthermore, similar learner's profiles are grouped together to construct a learning behaviour cluster. The choice of hierarchical clustering as a means of classification of learners' profiles derives from the observations of learners behaviour. This in turn reflects the similarities and subtle differences of learner behaviour, which are further analysed in more detail. Building awareness about the learner's behaviour is the first and necessary step for future learning-aware applications.

Key Words: Learner Profiling, Hierarchical clustering, Dimension Reduction, Learning Analytics, Markov chain

Category: L.2.2, M.0, M.4, M.5

1 Introduction

The objective of learning applications is to support learners to attain their learning goals efficiently. Learning analytics helps to improve reaching these objectives by analysing the implicit gathered data from the learning applications [Ebner and Schön 2013]. The aim is to better understand the learners' behaviour and optimize the learning process as well as the application itself [Siemens and Baker 2012]. One approach to reach these objectives is that the learning applications profile the learners according to their learning goals and their relevant learning characteristics. During application usage the learner may exhibit improvement or decline in his learning efficiency. In order to further support the learner as much as possible, the learners' profile must be adapted according to the evolution of the observed learning behaviour.

User profiling is a methodology that became necessary in different domains outside e-learning. In most user profiling applications the user profiles are created based on personal information and learning goals asked directly from users [Linton and Schaefer 2000] [Wei and Yan 2009]. In contrast to this explicit feedback mechanism, there is implicit feedback that is gathered through monitoring the users' interaction with the system. The gathered data is used to better understand the users' preferences, grasp their characteristics and adapt the application correspondingly to support the learners in reaching their learning goals. To offer a personalized support for each individual learner the application must be able to differentiate the learners while taking their similarities also into account. The novelty of the chosen clustering method is that it can provide insights about the small deviations in the learning behaviour of learners classified in a coherent cluster.

A feedback extractor with fusion capability that combines multiple feedback features to infer user preferences is proposed in [Li and Chang 2005]. The user preferences and the levels of expertise are collected by a user profiler to build user profiles. Collaborative filtering is applied on user profiles to provide personalized information to the user. Chen et al. [Chen et al. 2007] apply association rule mining to create the learner profiles in order to discover common learning misconceptions of learners. Jeon et al. [Jeon et al. 2008] describe an adaptive user profiling mechanism for personalized information retrieval. They also apply the collaborative filtering method to deal with user profiles that are frequently changed. Rebaï et al. [Rebaïet al. 2013] propose a semi-supervised learning based adaptive method for learning the user profiles and identifying irrelevant profile elements. The method includes a classification of profile elements and a co-training algorithm. [Mihaescu 2011] uses linear regression for modelling the quantity of accumulated knowledge in relationship with the performed activity in e-learning environments. [Yathongchai et al. 2013] introduce a learner classification that bases on learning behaviour and performance. They applied K-means clustering to analyse the learning behaviours of each learner and a decision tree classifier to generate the learner classification model based on the learning behaviours and student's performance. [Romero et al. 2008] compare different data mining methods and techniques for classifying students within a Moodle[1] environment. Their classifier bases on usage data and the final marks obtained in courses.

In this work we focus on a dataset provided by the Android application *UnlockYourBrain*[2]. The application covers basic mathematical problems (addition, subtraction, multiplication and division). The user has to answer a question correctly in order to unlock the smartphone screen. Two up to five answering options are provided for each posed question. The same question can be provided

[1] `http://moodle.org/` (visited on 22/05/2014)
[2] `http://www.unlockyourbrain.com/en/` (visited on 22/05/2014)

with a different number of answering options. The user is allowed to skip the application any time. The application gathers the answers to the posed questions as implicit data, which are interpreted as the result of the user's learning behaviour.

In our previous works [Taraghi et al. 2014a] [Taraghi et al. 2014b] , we were engaged in analysing the one digit multiplication problem to improve basic mathematics education for primary schools [Schön et al. 2012] . Considering our experience from these works, we improve the dataset preprocessing and define a confidence level for reliable statistical results. We enhance the question difficulty classifier by defining more features. Furthermore we model the learners' profile according to their answering behaviour using Markov chains. The created learner profiles are then clustered using the hierarchical clustering algorithm [Murphy 2012]. The clusters of similar learners (according to learning behaviour) are further analysed. Comparison of neighbouring leaf clusters shows the fine differences in the learning behaviour of the learners they contain. On the other hand, the parent cluster in which they are nested is characterized by their similarities which in turn distinguish and differentiate them from other clusters.

One has to recognize the influence of several factors playing role in the learning process [Soussa 2006]. Considering metadata such as age, gender, learner goals could enhance the research and improve the clustering algorithm. Unfortunately, the used application, does not provide this information. Nevertheless, the observed learning behaviour can be thought of as the result of several known and unknown factors that can depend on each other. These influences can be further explored by a more complex and sophisticated application.

[Section 2] of this publication introduces the methodologies that are used. The used dataset and the preprocessing steps performed before the actual analysis takes place are described in [Section 3]. [Section 4] explains the questions classification process according to difficulty levels. [Section 5] goes through the derived learner profiling and classification. Finally, we present the results so far as well as means to address future challenges.

2 Methodology

This section introduces the main mathematical concepts that were used in our application components. The minimum sample size describes the minimum number of question occurrences in the whole data set that are necessary to obtain a reliable statistical analysis. The Markov chain is used to represent the learner's profile during the use of the application. Two classification algorithms are used for clustering of the question's difficulties and learner profiles respectively; the K-Means and Hierarchical clustering algorithms.

2.1 Minimum Sample Size

The confidence interval (also called margin of error) is the interval in which the values of a probability distribution are expected to be. The confidence level, given in percentage, indicates the reliability that the values of a given probability distribution lie within the confidence interval. A confidence level of 95% with a confidence interval of 2% means that one can be sure with a probability of 95% that the actual probability values lie within ±2% of their calculated values.

For a given confidence level (corresponding Z score) and a confidence interval C_{int} (margin of error), the minimum sample size N_{min} [Rahme and Joseph 1998] is calculated as follows:

$$N_{min} = \frac{Z^2 * p * (1-p)}{C_{int}^2} \ . \tag{1}$$

The calculated minimum sample size in equation (1) is corrected for a given finite population size N_{ps} by:

$$N_{min} = \frac{N_{min}}{1 + \frac{N_{min}-1}{N_{ps}}} \ . \tag{2}$$

2.2 K-Means Classification Algorithm

The K-Means algorithm classifies a set of data samples x_i into K different clusters [Bishop 2006]. Each cluster is mainly characterized by its mean point μ_k (centroid). All samples that are closer to a specific centroid rather than to all the other centroids belong to this cluster. In this work the distance to the centroids is computed by the Euclidean distance. The algorithm works iteratively; in the first iteration the K centroids are guessed (sometimes chosen randomly from the samples of the dataset). After classifying each data sample to a cluster, its centroid is recomputed as the mean of all samples assigned to it. This process is repeated as long as the cumulative distance (3) over all samples and over all centroids converges to a local minimum.

$$J = \sum_i \sum_k r_{ik} \parallel x_i - \mu_k \parallel^2 \tag{3}$$

where:

$$r_{ik} = \begin{cases} 1 & \text{if } argmin_j \parallel x_i - \mu_j \parallel \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

At the end of the algorithm each data point is assigned to a specific cluster (hard classification). To find the optimal number of clusters the algorithm runs for different values of K parameter combined with a stopping criterion to avoid over-fitting. Detailed description of the implementation of the algorithm in our case is provided by [Taraghi et al. 2014a].

2.3 Markov Chain

A finite discrete Markov chain [Cover and Thomas 2006] of order one is a sequence of random variables $X_1, X_2, X_3, ..., X_n$ for which the following Markov property holds:

$$P(X_{n+1} = x_{n+1}|X_n = x_n, ..., X_1 = x_1) = P(X_{n+1} = x_{n+1}|X_n = x_n) . \quad (5)$$

The Markov chain of first order is characterized as memoryless, meaning that the future state is conditionally independent from all past states given that the current state is observed. Considering a Markov chain of order k, the probability of the next state depends on the k previous states. A Markov chain of order k is described formally as follows:

$$\begin{aligned} P(X_{n+1} = x_{n+1}|X_n = x_n, ..., X_1 = x_1) = \\ P(X_{n+1} = x_{n+1}|X_n = x_n, ..., X_{n-k+1} = x_{n-k+1}) . \end{aligned} \quad (6)$$

The Markov model is represented as a matrix P of all stochastic transition probabilities between the states. Hence, for n states, the matrix P is of size $n * n$. Each row in the matrix represents the stochastic transition probabilities from one state to all the other states. As a result the sum of probabilities within a row is always 1.0.

2.4 Hierarchical Clustering

In contrary to the K-Means algorithm, hierarchical clustering classify a set of data samples into a hierarchy of clusters that are nested within each other. The objects that are close to each other (according to a linkage metric) are merged together to build a new cluster. The main two approaches of hierarchical cluster creation are bottom-up (agglomerative clustering) and top-down (divisive clustering) [Murphy 2012].

In the agglomerative approach each data sample is considered as a one-member object leaf of the tree. The most similar pair of clusters are merged to one parent cluster at each step. This bottom-up process is continued until the root cluster in the hierarchy is reached, containing all the data divided into subclusters.

In contrast to the bottom-up approach, divisive clustering is performed in a reverse order. Beginning with the root, the whole data samples are considered as one cluster. The root cluster is split in to subclusters until the leaf clusters of the hierarchy are created.

The result of hierarchical clustering is a tree-like structuring of the clusters. Starting with the root cluster, the distance or height difference between branches in the generated tree represent the dissimilarity between the subclusters that are

Table 1: The size and composition of the cleaned dataset.

	Addition	Subtraction	Multiplication	Division
#Users	107603	47340	48114	48785
#Questions (distinct)	397914	161645	8633	52008
#Questions (total)	8169868	2125309	1773888	1796888

merged. Different dissimilarity measures can be used for this purpose. One of them, the Ward's method bases on the incremental sum of squares; that is the increase of sum of squares of distances between all objects of the clusters and the cluster centroids when the clusters are merged. The merging cost of combining the clusters used as distance measure is calculated as follows:

$$d(A, B) = \sqrt{\frac{2n_A n_B}{n_A + n_B}} \parallel \bar{c}_A - \bar{c}_B \parallel^2 \ . \tag{7}$$

whereas n_A and n_B are the number of objects within some clusters A and B, $\bar{c}_A$ and $\bar{c}_B$ are the centroids of the clusters A and B respectively and $\parallel \bar{c}_A - \bar{c}_B \parallel$ is their Euclidean distance.

3 Dataset Description

The dataset was provided by the Android application *UnlockYourBrain*. The application poses one basic mathematical question (addition, subtraction, multiplication and division) each time the user tries to unlock the screen of the smartphone. The user can select an answer from a set of provided options. The number of answering options varies for each question between two and five. In case the answer is correct, the screen is unlocked, otherwise the user can continue trying to select the correct answer from the remaining options. The user can skip the application at the very beginning or at any further step.

The raw dataset contained inconsistencies due to missing records. Furthermore, there were cases where the user skipped the question without selecting any answering option. It is assumed that when a user answers a question always correctly (without ever trying any of the several false options), then this question is already mastered. Therefore, the number of repeatedly correct answered questions would compromise the statistical analysis. In all the above described cases, the dataset was properly cleaned. The size of the cleaned dataset is given in [Tab. 1].

The fact that some questions in the cleaned dataset appeared rarely indicated that a question must have been posed sufficiently many times to be considered in the forthcoming analysis. The minimum number of question occurrences, known as minimum sample size of the data, can be calculated by equation (1). After

Table 2: The size and composition of the reduced final dataset. The minimum sample sizes are based on a confidence level of 95% and a confidence interval (margin error) of 2%.

	Addition	Subtraction	Multiplication	Division
Minimum sample size	2400	2398	2398	2398
#Users	102722	38708	46357	47558
#Questions (distinct)	667	155	268	204
#Questions (total)	4228439	611312	1191450	1086256

trying several values, we decided on a confidence level of 95% with an error rate of 2%. The minimum numbers of questions that fulfil the requirements above are corrected for each mathematical operation according to the equation (2). Hence for addition it is 2400 and for the other three operations 2398. This led to a reduction of our cleaned dataset. The size of the final dataset that is used for the analysis can be taken from [Tab. 2]. For instance within the set of *addition* problems, only the questions that were posed at least 2400 times (the calculated minimum sample size) were considered for the analysis. Consequently the dataset was reduced from 397914 to 667 distinct questions. Summing up the frequencies of occurrence of each individual distinct question, the total number of questions that were analysed, was reduced from 8169868 to 4228439.

4 Question Classification

The overall answering behaviour of the users to a specific question is an indicator of the relative difficulty of that question. The following subsections describe the possible answering types of the users, the derived difficulty levels, and the classification algorithm that is used to cluster the questions in different difficulty levels.

4.1 Answering Types

As mentioned before, the questions are posed to the users with varying number of answering options. Depending on the number of options, the user has a limited number of answering type possibilities. [Tab. 3] shows the user's answering possibilities with regard to the posed answering options. The rows represent different answer possibilities (types) for each provided number of options. For instance, given a question with two answering options, the user can only choose one answer that will be either correct (R) of false (W). Considering a question with three answering options, the user can either answer correctly (R) or make a mistake in the first round. In the later case two answering options remain.

Table 3: Answer types for a question in regard to different numbers of answering options.

#Options	Answer types							
2	R		W					
3	R	WR	W		WW			
4	R	WR	W	WWR	WW		WWW	
5	R	WR	W	WWR	WW	WWWR	WWW	WWWW
Dimension	1	2	3	4	5	6	7	8

The user goes on with the second round and chooses either the correct (WR) or wrong answer (WW). The user may skip the application without selecting any option in the second round. In that case the answering type of this question remains wrong (W). The answer types for questions with four or five answering options can be defined accordingly.

Every question can be posed with varying numbers of options each time. Therefore, before classifying a question, the probabilities of the same answer types are summed up. In total there are eight different answering possibilities that will define the dimensions used in the classification algorithm.

4.2 Classification of Questions

The classification of questions is based on their answer types. Each question is represented as a point in an eight-dimensional feature space, where each dimension represents the probability of occurrence of one answer type. The questions of each mathematical operation were considered separately. The classification algorithm that was used is K-Means (see [Section 2.2]) which computed 13 clusters for addition, 10 for subtraction, and 11 for multiplication as well as division operations. Each cluster gathers questions that are considered to have similar level of difficulty. [Fig. 1] depicts the computed clusters for the addition operation according to three of the eight features (R, W, WR). Each point in the figure represents one posed *addition* question.

We then sort the clusters according to the difficulty level of the questions they contain. As an example, the centroid's coordinates in each dimension are plotted in [Fig. 2] for the *multiplication* problems. The centroids represent each cluster individually. It can be seen that clusters 11, 2, and 7 contain the most difficult questions whereas 1, 4, and 10 contain the easiest. It can be observed that the more easy a cluster is characterized the less probable it is that a user answers to the questions in this cluster incorrectly.

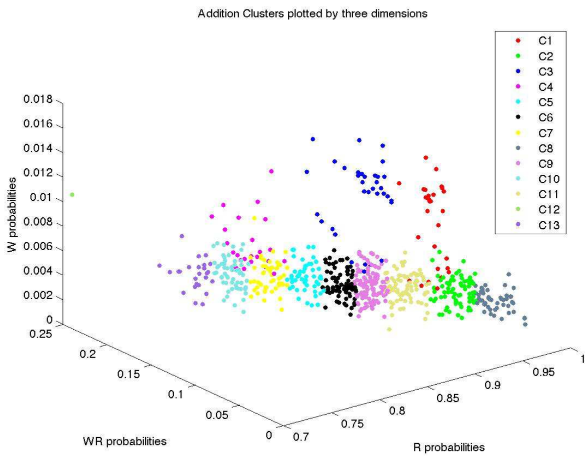

Figure 1: The classifier computed thirteen difficulty levels within the set of *addition* problems. The clusters are plotted on three out of eight dimensions: R, WR and W.

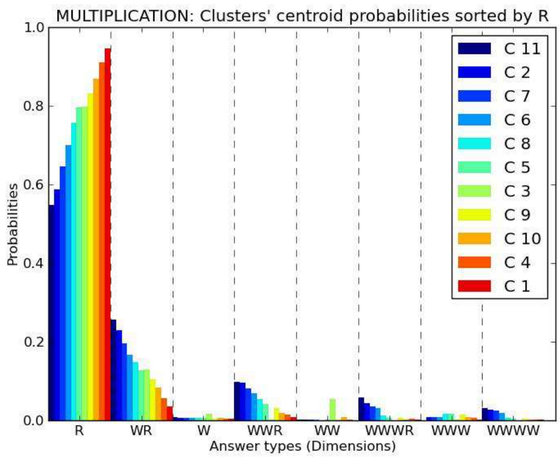

Figure 2: Probability distribution of eleven clusters' centroids of multiplication problems, sorted in ascending order for the probabilities in answer type R. While the probabilities for the sorted list of clusters increase in dimension R, they decrease analogously in dimensions WR, WWR, WWWR and WWWW.

5 Learner Profiling and Classification

5.1 Learner Profiling

The question clustering according to difficulty was designed to support the adaptation of the application according to the user's learning profit. There are several works that deal with the impact of the order of questions appearance in a test or questionnaire [Weinstein and Roediger 2012], [Coniam 1993], [Gray 2004], [Carraso et al. 2013], [Doerner and Calhoun 2009], [Perreault 1975]. Many of them put emphasis on the psychological effects of the question sequence. Without intention to neglect those facts, we nevertheless concentrate more on the evolution of the learning process as it unfolds over time. Assuming the sequence of the posed questions has an influence on the learning process of the users, one goal of the application will be to pose the questions in a sequence that will effectively advance the learning progress. Therefore, a Markov chain representation, where the probability of answering a question of a specific difficulty level can be assumed to be in relation to the previous answers, characterizes the learning effort of the user.

Such an application cannot remain static over time as the new answering behaviours will affect the number and content of question's clusters. Furthermore, it cannot be assumed that each user learns in the same manner. An adaptive users learning profile can be constructed by using the initial answering behaviours as a starting point.

The use of the application can be seen as a sequence of alternating question - answer type pairs. The model of the application use was produced by a Markov chain. Both the question clusters and the answer types are states of this Markov chain. Transitions are only allowed between question clusters states and answer type states. The Markov chain of order $k = 1$ is memoryless. Each transition from a question cluster to an answer type has the probability of the user's answering any question within this specific question cluster as defined by the answer type (see [Section 4.1]). These transition probabilities are computed over the course of the application use and are continuously updated. The transition probability from a specific answer type to a question cluster (probability of the next posed question) is defined by the application. [Fig. 3] presents all possible transitions for order $k = 1$ along with two explanatory examples. For order $k > 1$ the transition probabilities are formed given the probabilities of occurrence of the last $k - 1$ previous state transitions. [Fig. 4] shows all possible transitions for order $k = 2$ along with two explanatory examples.

Given n question clusters and 8 answer types, the Markov chain model contains $n + 8$ states. The number of all possible transitions depends on the order k of the Markov chain. Equation (8) expresses the number of possible transitions in the introduced Markov chain model for each order k.

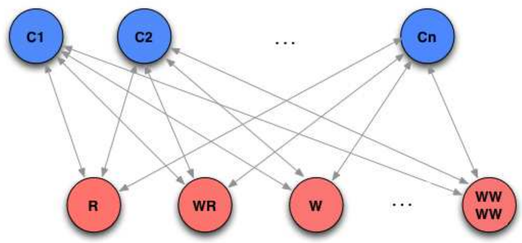

Figure 3: Markov chain model of a user profile for $k = 1$. The states in blue denote the created question clusters. The states in red represent the eight answer types. The links between states are all possible transitions during the use of the application. For instance, $C1 \rightarrow R$ represents the transition probability that the user answers correctly in the first round (R) to the posed questions that are classified in the cluster $C1$. $WR \rightarrow C2$ represents the transition probability for the case that the user solves questions classified in the cluster $C2$, after having answered to the previous posed question correctly in the second round (WR).

$$T_k = \begin{cases} 2(8n)^{\frac{k+1}{2}} & \text{if k is odd} \\ (n+8)(8n)^{\frac{k}{2}} & \text{if k is even} \end{cases} \tag{8}$$

As it can be seen, the number of possible transitions T_k is exponential in the order k. For high order k the model tends to become very large and the transition matrix sparse. The non-existing transition probabilities for a learner are set to zero. The actual number of existing transitions for each learner in our dataset is still very low. In other words, an individual learner experiences only a small fraction number of all possible transitions.

5.2 Classification of Learner Profiles

As described in [Section 5.1] the learner profiles are created using a Markov chain model that bases on the learners answers to the posed questions. The transition probabilities built within the Markov chain model characterize the learners' behaviour. Each transition represents a different dimension or feature. The transition probabilities are the learner profile feature values that are used for classification of the learners into different clusters. Considering the equation (8), the model becomes exponentially large and the transition matrix sparse for higher orders k. Consequently, the computation time as well as the required memory for the clustering algorithm is increased correspondingly. As an example, the Markov chain model of the *addition* problem contains 227136 features for each of the 86786 sample learners for order $k = 4$. This leads to a sparse

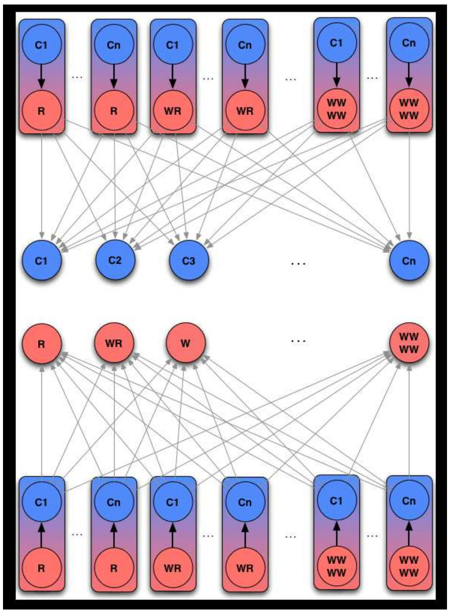

Figure 4: Markov chain model of a user profile for $k = 2$. The states in blue denote the created question clusters, whereas the states in red are the eight answer types. The links between states are all possible transitions in this model. The rectangles wrap the last state transitions, namely the state transitions for the previous step $k = 1$. For instance, $(C1 \rightarrow R) \rightarrow C2$ represents the transition probability that the learner solves questions classified in the cluster $C2$, after having answered correctly a previously posed question from cluster $C1$ in the first round (R). $(WR \rightarrow C1) \rightarrow R$ represents the transition probability that the user answers correctly in the first round (R) to a posed question belonging in the cluster $C1$, given that the previous posed question has been answered correctly in the second round (WR).

probability matrix of size $86786 * 227136$ representing the Markov chain models of all learners.

To overcome this problem, nonlinear dimensionality reduction techniques were applied to reduce the number of features. We compared different dimensionality reduction techniques that best suit our dataset for the purpose of classification [Van der Maaten 2008]. The most appropriate technique for our dataset was Multidimensional Scaling (MDS) [Van der Maaten 2007]. A detailed comparative review on dimensionality reduction techniques can be found in [Van der Maaten et al. 2008].

[Fig. 5] visualizes all learners according to their profiles after applying MDS

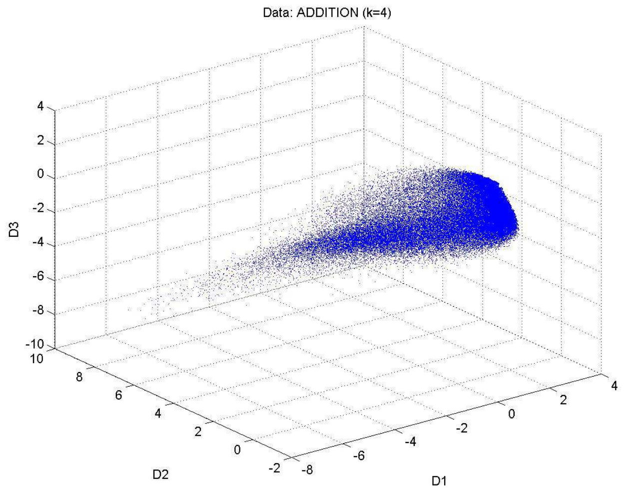

Figure 5: Learners profiles (i.e. Markov chains of order $k = 4$ for the *addition* problem) as points in a nonlinearly reduced space. The initial 227136 dimensions are reduced to five dimensions. The dimensions $D1$, $D2$ and $D3$ represent the first three of the five dimensions after the application of the MDS algorithm.

for the *addition* problem and for order $k = 4$. Each point represents exactly one learner. The axes $D1$, $D2$ and $D3$ represent the first three dimensions after dimensionality reduction. In the *addition* problem, the initial 227136 dimensions are reduced to five dimensions.

For classification of learner profiles into different clusters the agglomerative hierarchical clustering (see [Section 2.4]) has been applied. The Ward's method (see equation 7) has been used to measure the dissimilarity distance between clusters.

[Fig. 6] depicts the dendrogram of the top five clusters and their nested subclusters. [Fig. 7] depicts the five top hierarchical clusters of the data points shown in earlier [Fig. 5]. The hierarchical clustering corresponds also to the dendrogram of the [Fig. 6].

5.3 Analysis of Hierarchical Learner Profiles

From the first observations of the hierarchical learner profiles one can attempt to interpret the results of the classification. The similarities between the question - answer pairs between two users makes them members of the same cluster. We started with the two users that have the lowest non zero distance and compared their $k = 4$ sequences of the *addition* problem. One common sequence was

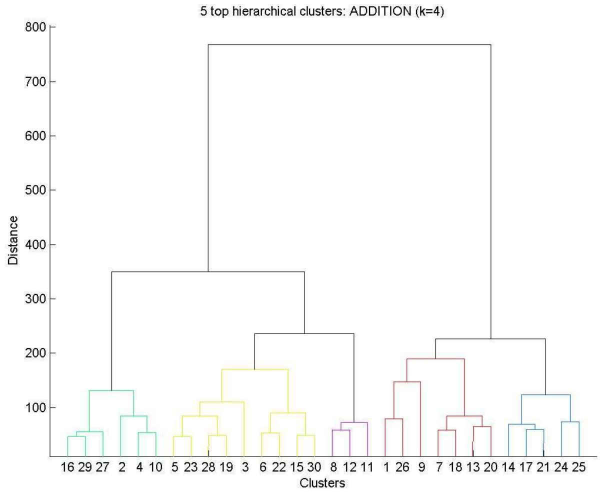

Figure 6: Dendrogram of the top five hierarchical learner profile clusters for the *addition* problem and Markov chains of order $k = 4$. Each cluster contains many nested subclusters. The leaves in the figure (each identified by a cluster ID on the horizontal "Clusters" axis) have themselves further nested clusters, which had to be cropped for improved visibility. The vertical axis represents the distance between the clusters, which is a measure of dissimilarity between them.

$(R \rightarrow C13 \rightarrow WR \rightarrow C10) \rightarrow R$. These transitions represent the following application use scenario: both users begin with some correctly answered question. When confronted with another question from cluster $C13$ they both answer it correctly in the second round. Afterwards, the program chooses one question from cluster $C10$, which is answered by both learners correctly in the first round. As we move on to user profiles with higher distance, the probabilities of the similar transitions of length $k = 4$ became more different. Furthermore, distant clusters have also more different transition sequences.

A detailed and quantitative evaluation that will also take into account the relative distances between the questions clusters (and not only the sorted answer types) will be the first addressed part of future work. The results will provide qualitative and interpretable declarations about the similarities and differences between the learning behaviour of the learners. This is the first step for creating learning-aware applications whereas the other parts include (among others) decision making, support and enhancement of the learners.

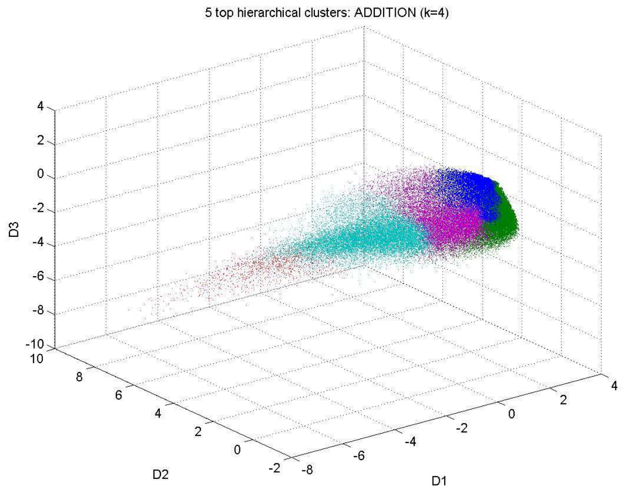

Figure 7: The top five clusters of the hierarchically classified user profiles. Each point represents exactly one learner's Markov chain model reduced to the first three main dimensions $D1$, $D2$ and $D3$. The order of the Markov chain is $k = 4$ and the addressed arithmetical problem is *addition*.

6 Conclusion

A user profiling mechanism applied to a dataset that deals with basic mathematical problems (addition, subtraction, multiplication and division) was proposed. The user profiles are based on a Markov chain model that contains transition probabilities between question clusters and answer types. The questions were classified to different clusters according to their difficulty levels by the K-Means algorithm. User profiles were classified to detect similarities and and differences between the learning process of the users, which can be further used for the improvement and individualization of the learning environment. The balance between learner personalization and detection of learning behaviour similarities can be attained by hierarchical classification algorithms that support a desired and tunable granularity.

The fact that a learning application needs only implicit feedback to enhance the learners efficiency is one of the most important efforts of our proposal. This approach already has several challenges and open research questions. The clustering of very large number of users is computationally time consuming. The large number of defined features describing each learner makes the clustering procedure tedious, hence it cannot be addressed without dimensionality reduction. The presented work consists a proof of concept that needs to be integrated in the original application. The design goals will define the degree of personalization and influence of the learning procedure itself. The evaluation of the potential benefits or drawbacks in the learning process as well as the revision and adaptation of the learning goals can only be fully explored in an integrated, learning-aware environment.

Acknowledgement

We would like to express our gratitude to the team of *UnlockYourBrain* GmbH that provided us with the dataset and made this work possible.

References

[Bishop 2006] Bishop, C.: Pattern Recognition and Machine Learning. Springer Science and Business Media, LLC, New York, USA, 424-430 (2006)

[Carraso et al. 2013] Carrasco A., Lucas M.E., Marín C., Rubio A., Sánchez G.: Does Content Order Affect Performance on Multiple Choice Exams? International Technology, Education and Development Conference, 716-719 (2013)

[Chen et al. 2007] Chen, C., Hsieh, Y., Hsu, S.: Mining learner profile utilizing association rule for web-based learning diagnosis, Journal of Expert Systems with Applications. 33, 1, 6-22 (2007)

[Coniam 1993] Coniam D.: Does the Ordering of Questions in a Test Affect Student Performance? Educational Research Journal, 8, 74-78 (1993)

[Cover and Thomas 2006] Cover, T. M., Thomas, J. A.: Elements of Information Theory 2nd Edition (Wiley Series in Telecommunications and Signal Processing). Wiley-Interscience. ISBN: 0471241954 (2006)

[Doerner and Calhoun 2009] Doerner, W. M. and Calhoun, J. P., The Impact of the Order of Test Questions in Introductory Economics. Available at SSRN: http://ssrn.com/abstract=1321906 or http://dx.doi.org/10.2139/ssrn.1321906 (2009)

[Ebner and Schön 2013] Ebner, M., Schön, M.: Why Learning Analytics in Primary Education Matters!,Bulletin of the Technical Committee on Learning Technology, Karagiannidis, C. and Graf, S (Ed.), 15, 2, 14-17 (2013)

[Gray 2004] Gray K. E.: The Effect of Question Order on Student Responses to Multiple Choice Physics Questions. Master Thesis, Kansas State University (2004)

[Jeon et al. 2008] Jeon, H., Kim, T., Choi, J,: Adaptive User Profiling for Personalized Information Retrieval. In third International Conference on Convergence and Hybrid Information Technology, 2, 836-841 (2008)

[Li and Chang 2005] Li, X., Chang, S. K.: A Personalized E-Learning System Based on User Profile Constructed Using Information Fusion. In proceedings of the 11th International Conference on Distributed Multimedia Systems (DMS'05), Banff, Canada, 109-114 (2005)

[Linton and Schaefer 2000] Linton, F., Schaefer, H. P.: Recommender systems for learning: Building user and expert models through long-term observation of application use. User Modeling and User-Adapted Interaction, 10, 181-207 (2000)

[Mihaescu 2011] Mihaescu, M. C.: Classification of learners using linear regression. Federated Conference on Computer Science and Information Systems (FedCSTS), 717-721 (2011)

[Murphy 2012] Murphy, K.P.: "Machine Learning a probabilistic prospective"; The MIT Press, Cambridge, Massachusetts, London, England (2012)

[Perreault 1975] Perreault, W., D., JR.: Controlling Order-Effect Bias. In The Public Opinion Quarterly, 39, 4, 544-551 (1975)

[Rahme and Joseph 1998] Rahme, E., Joseph, L.: Exact sample size determination for binomial experiments. Journal of Statistical Planning and Inference. 66, 83-93 (1998)

[Rebaïet al. 2013] Rebaï, R. Z., Ghorbel, L., Zayani, C. A., Amous, I.: An Adaptive Method for User Profile Learning. In B. Catania, G. Guerrini and J. Pokorn (eds.), ADBIS, Springer, 126-134 (2013)

[Romero et al. 2008] Romero, C., Ventura, S.,Hervas, C., Gonzales, P.: Data mining algorithms to classify students. International Conference on Educational Data Mining (EDM), Montreal Canada, 8-17 (2008).

[Schön et al. 2012] Schön, M., Ebner, M., Kothmeier, G.: It's Just About Learning the Multiplication Table, In Proceedings of the 2nd International Conference on Learning Analytics and Knowledge (LAK '12), Simon Buckingham Shum, Dragan Gasevic, and Rebecca Ferguson (Eds.). ACM, New York, NY, USA, 73-81. DOI=10.1145/2330601.2330624 (2012)

[Siemens and Baker 2012] Siemens, G., Baker, R. S. J.: Learning Analytics and Educational Data Mining: Towards Communication and Collaboration. LAK '12 Proceedings of the 2nd International Conference on Learning Analytics and Knowledge, 252-254 (2012)

[Soussa 2006] Soussa, D. A.: How the Brain Learns Mathematics, Corwin Press, London, United Kingdom (2008)

[Taraghi et al. 2014a] Taraghi, B., Ebner, M., Saranti, A., Schön, M.: On Using Markov Chain to Evidence the Learning Structures and Difficulty Levels of One Digit Multiplication. In proceedings of the 4th International Conference on Learning Analytics and Knowledge, Indianapolis, USA, 68-72 (2014)

[Taraghi et al. 2014b] Taraghi, B., Saranti, A., Ebner, M., Schön, M.: Markov Chain and Classification of Difficulty Levels Enhances the Learning Path in One Digit Multiplication. In: Learning and Collaboration Technologies. Designing and Developing Novel Learning Experiences. LNCS, Springer International Publishing, 322-333 (2014)

[Van der Maaten 2007] Van der Maaten, L.J.P.: An Introduction to Dimensionality Reduction Using Matlab; TiCC, Tilburg University: Tilburg, The Netherlands (2007).

[Van der Maaten 2008] Van der Maaten, L.J.P.: DrtoolboxMatlab Toolbox for Dimensionality Reduction, 0.7b; TiCC, Tilburg University: Tilburg, The Netherlands (2008).

[Van der Maaten et al. 2008] Van der Maaten, L.J.P., Postma, E.O., Herik van den, H.J.: Dimensionality Reduction: A Comparative Review; TiCC, Tilburg University: Tilburg, The Netherlands (2008).

[Wei and Yan 2009] Wei, X., Yan J.: Learner Profile Design for Personalized E-Learning Systems. In proceedings of the international Conference on Computational Intelligence and Software Engineering, 1-4. (2009)

[Weinstein and Roediger 2012] Weinstein, Y., Roediger H. L. 3rd.: The effect of question order on evaluations of test performance: how does the bias evolve?. In Memory & Cognition, 40, 5, 727-735 (2012)

[Yathongchai et al. 2013] Yathongchai, C., Angskun, T., Yathongchai, W., Angskun, J.: Learner classification based on learning behavior and performance. Open Systems (ICOS), 2013 IEEE Conference, 66-70 (2013)