The Role of Cluster Analysis within Learning Analytics

Isabelle Mayerhofer, Seminar Learning Analytics, [redacted]

February 2021

Abstract

This paper gives an overview of the application of the unsupervised data mining technique of cluster analysis within Learning Analytics (LA). Special focus lies on the K-Means algorithm. Initially a broad overview of the field of LA and clustering in general is given, followed by a detailed description of the K-Means algorithm. K-Means is then critically compared to two other clustering algorithms: the Expectation Maximization (EM) algorithm and Spectral Clustering while presenting three pedagogical applications of cluster analysis within LA. The paper ends with a summary and critical outlook considering ethical aspects as well as promising prospects of applying clustering techniques within LA.

1 Introduction

With Covid-19 keeping students out of the classroom, today's pandemic-ridden world more than ever experiences the need of functioning computer-aided learning technologies. This is where the relatively young research area of *Learning Analytics* (LA in the following), which developed around 2011, finds application. It aims at optimizing teaching and learning via mining data from computer-aided educational settings and using the gained information as input for various pedagogical applications in online as well as offline and also blended-learning contexts, see [ID20], [GD12]. According to the *LA reference model*, see [CDST12], there are four main dimensions within LA: *data and environments*, *stakeholders*, *objectives* and *methods*. The dimension of data and environment refers to data from educational settings. Stakeholders imply people and entities involved, usually teachers and students, but also institutions and e-learning software. Objectives, which will come up in this paper, are e.g. monitoring and analysis, prediction, tutoring, assessment and adaptation. Statistics, information visualisation and data mining belong to the methods of LA. Focus of this paper lies within the LA method of data mining, also known as *knowledge discovery*. It follows a typical process, see [Clo12] and [HKP11], which largely goes hand in hand with the "iterative cycle" [CDST12] of the LA process, see ibid.: initially, potentially relevant data is collected from an educational setting and preprocessed. In case the data derives from different sources it is then integrated. Data that seems relevant for meeting a specified objective is then selected and transformed to a necessary format, e.g. into csv or tabular format. Now the data is ready for the actual data mining step, where a variety of algorithms can be applied in order to gain knowledge from the relevant sources. Among these data mining algorithms is clustering, the focus of this paper. In a nutshell, clustering methods enable the formation of groups aka clusters of similiar items, given a data set. The information gained from applying the data mining algorithm needs to be analyzed, interpreted and presented in an adequate way, processable for the human eye and brain and especially accessible for lay people such as educators and students. The latter point is crucial in order to offer "insightful action" [Clo12] for pedagogical purposes. According to the LA reference model a post-processing phase leads to further improvement of the LA process via using gained information from the described previous steps as input for a new iteration of the LA process, see [CDST12]. In this paper the concept of clustering and its role within LA is presented with special focus on the *K-Means* algorithm. While an initial general overview of clustering methods is given, it is followed by a detailed description of K-Means including its weaknesses. K-Means will then be compared to two other clustering algorithms which find application when K-Means tends to fail or to be less performant: the *Expectation Maximization (EM)* algorithm and *Spectral Clustering*, of which the basic functionalities will be outlined. This goes alongside presenting three pedagogical applications of cluster analysis. The paper concludes with a critical summary and outlook.

2 Clustering in Learning Analytics

2.1 Clustering in a nutshell

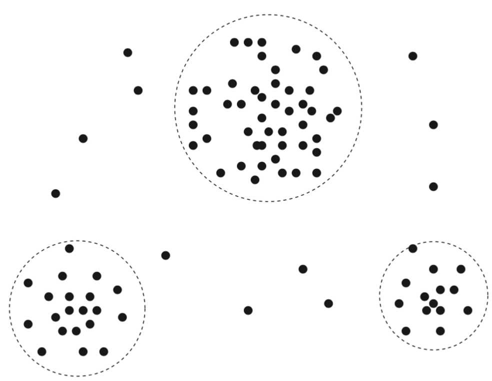

Figure 1: Three clusters, [[HKP11]: 20]

The outcome of well-working clustering algorithms performed on a dataset is typically that objects within a cluster are very similar to another and objects from different clusters are very dissimilar. In other words, clustering aims at maximizing intracluster similarity and intercluster dissimilarity at the same time, see [[HKP11]: 2]. Similarity is defined via specific attributes like e.g. belonging to the same performance group and usually computed via distance measures, as e.g. the Euclidean distance. Objects that do not fit into any cluster are so-called *outliers*. Figure 1 illustrates this: The datapoints surrounded by the dotted lines forming circles, are clusters. The datapoints lie close to eachother within a cluster and far apart when they are from different clusters. The points not captured in any circle are the outliers. Clustering is sometimes also referred to as "automatic classification" [[HKP11]: 445]; being an unsupervised learning method it runs on datasets without any given prior labels and the output is a classification of objects within groups. A cluster can hence be viewed as "implicit class" ibid. There are different kinds of clustering algorithms, such as e.g. *partitioning* algorithms which aim at forming clearly seperated clusters where each object is assigned to only one single cluster and there are algorithms involving e.g. *fuzzy* clustering where objects may belong to more than one cluster.

2.2 The K-Means algorithm

2.2.1 Defining the K-Means algorithm

The K-Means algorithm is one of the most well-known and widely used of all clustering algorithms. It is a partitioning algorithm. The following definition is based on Han et al. 2011 [[HKP11]: 451-454]. The algorithm is called K-*Means* because the mean value of the data points within a cluster represents the centre in each cluster. The cluster centres are also called *centroids*. Figure 2 serves as an illustration of K-Means' overall functioning and is based on [Tut18]. The algorithm takes as input a set of datapoints and a number k defining the number of clusters expected in the output. In the beginning k data points are randomly chosen as centroids. In 2a) k is set to three and the three centroids are represented by colored stars among the still unassigned data points in black. The x-axis and y-axis represent undefined attributes. Using the distance measure *euclidean distance*, the distance between each data point and the centroids is calculated. The datapoints are then assigned to those centroids to which they are nearest, see figure 2b): for illustration purposes the former black data points now take the same color as the centroids they are assigned to.

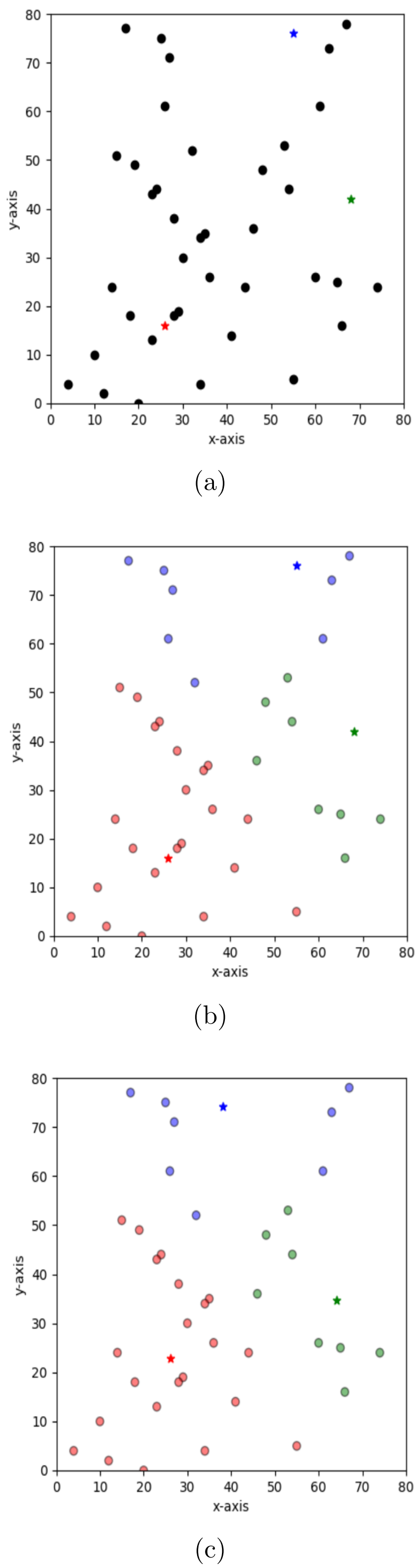

Figure 2: (a) Random data points
(b) Clustering Step 1 (c) Clustering Step 2

Hence, the now blue data points are assigned to the blue centroid, the green data points to the green centroid and the red data points to the red centroid. The cluster means are then updated, i.e. recalculated from the data points that were assigned to them. The updated centroids are illustrated in 2c), notice a change in the positions of the centroids. Then the distance of the datapoints from the new centroids is calculated anew and the data points are assigned to the updated clusters. In other words step b) and c) repeat until further updating no longer results in the clusters changing. The output of the K-Means clustering algorithm are hence k clusters of data points. K-Means' time complexity is *O(nkt)*, n being the number of data points, k the amount of clusters and t the number indicating how many interations are necessary, see [[HKP11]: 453]. It is important to note that the resulting clusters are not necessarily those that represent a global optimum; it is by contrast quite likely that K-Means converges at a local optimum, see [[HKP11]: 453], [Jai10], which is one of its main weaknesses. In order to avoid this, K-Means has to be run several times with different initial centroids. Optimality of a cluster, i.e. minimal intercluster variation, is calculated by the *sum of squared errors* between the objects within a cluster and the cluster center, see [[HKP11]: 451]. The clusters with lowest intercluster variation are the most ideal ones. However, already for the minimal number of two clusters the problem of finding a global optimum is NP-hard in the Euclidean space, see [Jai10], [[HKP11]: 452]. This also determines the notion of stability of the K-Means algorithm, i.e. whether the clusters are the same if run more than once on the same data set for a fixed k, see [VL10]. Only if there were a method that ensures to always find a unique global optimum, K-Means could be seen as a stable algorithm, a subject which is still open to research, see [RC07], [KV06]. A further and maybe the most obvious weakness of K-Means is that the number of k clusters has to be determined in advance. There are several approaches to how arrive at the value k. Among these is the "rule of thumb" [KM13], [RC07] referring to determining clusters based on similarity. This goes hand in hand with finding optimal clusters. Ultimately, the number k tends to be determined by the domain expert who knows which k is most suitable for the domain context, see [Jai10]. Hence, what a cluster actually is tends to be subjective, see [ST10]. K-Means furthermore has a strong sensitivity to outliers

[HKP11] and tends to perform poorly on high dimensional data, see [TPSH10].

2.2.2 Applying the K-Means algorithm to data from an Exploratory Learning Environment

Amershi et al. 2006 [AC06] apply K-Means in order to enable automatic recognition of learner groups given logged data of interface usage of a so-called *Exploratory Learning Environment* (ELE in the following). The main idea of an ELE is that learning via exploration can facilitate and intensify understanding in contrast to learning e.g. by a textbook, see ibid. An ELE can take the form of so-called *Algorithm Visualisation Systems* (AVS in the following), i.e. software that enables understanding of algorithms via discovering them. Amershi et al. made use of the *Constraint Satisfaction Problem* (CSP). The CSP consists of variables, variable domains and constraints and its goal is to find an arrangement of the variables that fulfills all constraints. Within the ELE the user has little to no limits of how to act. User characteristics such as metacognitive skills like self-control strategies [[NDH+08]: 73] determine how much users can profit from using ELEs. This brings up the question of how effective ELEs are in practice. The lack of structure provided by the ELE can be an obstacle to succesful learning and even experts are hardly able to derive characteristic behaviour traits of users in ELEs as Amershi et al. argue. In order to maximize the advantage for learning when using an ELE, provision of *adaptive support* for the users is desirable. Adaptive support is defined by giving students the help they need individually, matching their skillset, see [DWRK10]. K-Means enables to provide the technology needed as basis for implementation of adaptive support. In a first step, Amershi et al. implemented an offline K-Means enabling identification of learner groups on the basis of interaction patterns within the ELE. In a second step, they provided an online version of K-Means which enabled to assign users of the ELE to the generated clusters of the offline K-Means. To begin with the offline K-Means: the data used as input was the study participants' logged interaction data, generated while using the ELE interface. All in all, 24 feature vectors served as input data, each corresponding to one study participant's navigation style. The observed dimensions were e.g. frequency of and pause duration between interactions in the ELE. Among the interactions were e.g. so-called *fine-stepping*, i.e. taking steps within the ELE like clicking on a button or submitting a result. Another interaction that proved to be relevant for the study results was *resetting* of the application. Taking the external feature of post-test results into consideration, results showed e.g. a correlation between a high frequency of usage of *fine-stepping* and being low-performant in tests as well as a correlation between a long pause duration between usage of different methods and gaining better test results. Based on these findings, the clusters HL for *high average learning outcome* and LL *low average learning outcome* were obtained when setting k to 2, see figure 3.

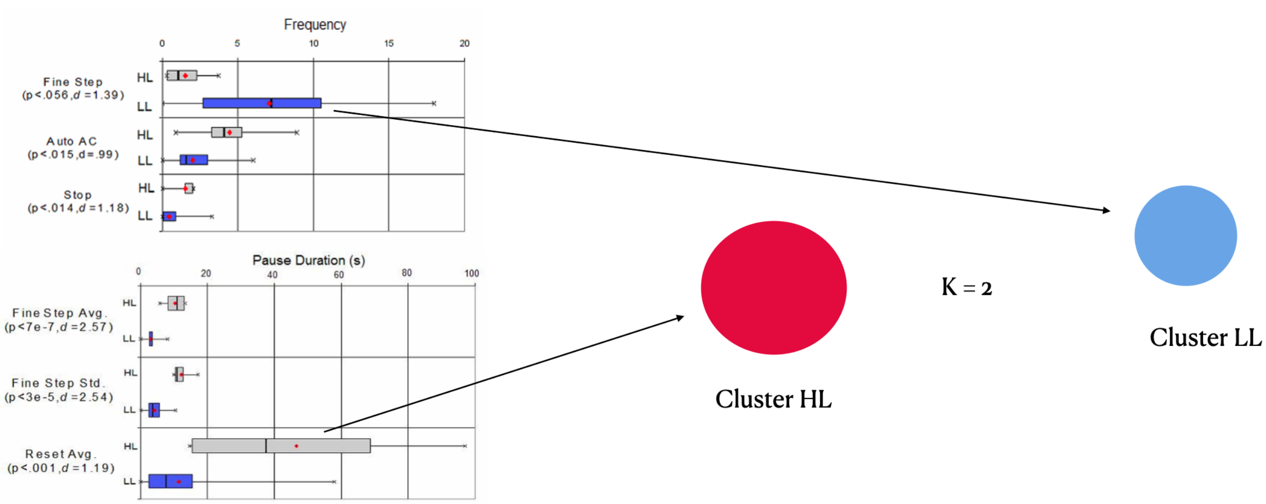

Figure 3: Histogram [AC06] giving frequency of fine steps and derived cluster LL and pause duration between resetting and derived cluster HL for k set to 2

It was argued that a high frequency of fine steps could be seen as indicator for a user not putting much thought into his or her approach to the task which was seen as detrimental for the learning outcome. Longer pauses between resetting the application, by contrast, could be seen as an indicator for a user being the opposite. Amershi et al. state, however, that without the external measure of post-test results it would be difficult to draw these conclusions. The implicit classes of performance groups created by the offline K-Means were now basis for an online K-Means: the algorithm was now applied at the same time a user interacted with the ELE. Again every user of the ELE was represented by his or her feature vector containing the information on how he or she dealt with the ELE's methods of interactive execution. These feature vectors were the input for the online K-Means. In the beginning of the interaction in the ELE it was unclear to which performance group a user fit best, see figure 4.

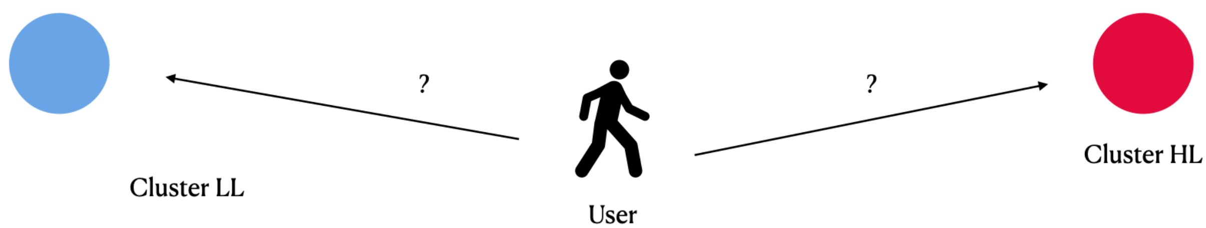

Figure 4: Unassigned user (feature vector) and clusters

Throughout interaction with the ELE the user-generated data was then saved and updated in the user's feature vector. Via this continuous updating, the user's feature vector was then assigned to the cluster with least distance to it, i.e. the user was assigned to the best-matching performance group. This is illustrated in figure 5 where the user is ultimately assigned to the LL cluster based on the interaction with the ELE and classification with K-Means.

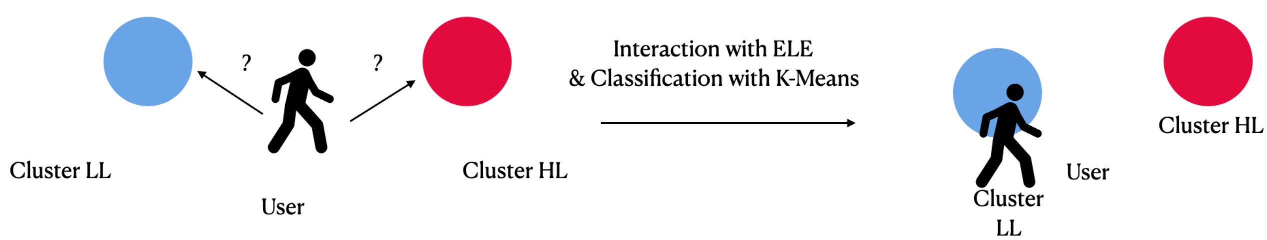

Figure 5: Process of assigning user to clusters during interaction with the ELE

The authors experimented with k set to 2 and 3 ran 20 trials of the K-Means algorithm in order to avoid convergence at local optima. This ultimately enabled to offer the information needed to offer real-time adaptive support. Although the papers leaves it open how this was then being put into practice this proves to be an example how K-Means can be used to improve the learning process. In other words this was an example of the application of clustering within LA meeting the LA objective of adaptation. The results could furthermore be used for monitoring, analyzing and ultimately assessing a student.

2.3 The EM-Algorithm

2.3.1 Defining the EM-Algorithm

It is not always possible to assign data points to exactly one single cluster as done with K-Means. The so-called *Expectation Maximization* algorithm (EM algorithm in short) offers a generalization of K-Means. It is a combination of fuzzy clustering and probabilistic clustering that enables assigning a data point to more than one cluster probabilistically, see [[HKP11]: 501, 505]. The main idea of the EM-algorithm is hence that clusters are defined as probabilistic models and every object belongs to each cluster with a specific probability [[Bis06]: 436-439]. The parameters of the distribution are adapted iteratively until a certain threshold is reached, see ibid. As with K-Means the EM-algorithm needs a fixed input value k, see [TG04] and is likely to converge at local optima, see [[HKP11]: 508].

2.3.2 Applying the EM-Algorithm to data from an LMS

Talavera et al. 2004 [TG04] use EM-clustering as a tool for creating behaviour profiles of users based on interaction patterns within a so-called *Learning Management System* (LMS in the following) in order to offer information for assessment purposes, as shall now be outlined: A LMS is a web-based software that organizes content, actors, interactions and feedback functions, see [RVG08], enabling computer-aided collaborative learning (CSCL), see [TG04]. In collaborative learning at least two people have a common learning goal and try to reach it together, see [Dil99]. A well-known example for an LMS is *Moodle*. Similar to an ELE, a LMS does not offer lot of structure for users, which is especially the case regarding interaction when using collaboration tools within an LMS like chat and forum, see [TG04]. LMS often exhibit database implementations that do not only save learning content but also interactions that are performed in the workspace. The information saved by an LMS is usually merely of quantative nature, tracking e.g. the amount of messages a person sent in a chat. In their paper, Talavera et al. used exactly this kind of data as input for the application of the EM-algorithm.

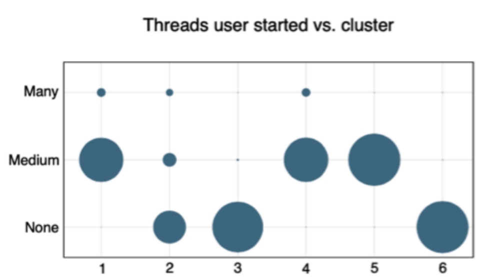

Figure 6: Illustration of results of clustering with the EM-Algorithm, [TG04]

After going through the classic steps of data mining described in the introduction, the logged interaction data was subdivided into the ranges *high*, *medium* and *low*. As in Amershi et al. 2006, the external, domain-specific feature of test results was added to determine the number of clusters as well as their semantics. Ultimately, the study resulted in six clusters, each representing a typical student behaviour profile within an LMS in correlation with performance. Figure 6 illustrates an example of cluster profiles of cooperative behaviour within the LMS: the amount of threads started by a user, presented by the ranges *many*, *medium* and *none* on the y-axis, given per cluster on the x-axis. The area of each circle corresponds to the probability of a datapoint belonging to a cluster. As focus on a single feature such as the amount of threads started did not offer enough insights, a combination of features was necessary in order to obtain more meaningful student profiles. These further features included e.g. amount of replies to user threads and messages in the LMS chat and forum. After combining the results of clustering for the individual properties, the six resulting final clusters represented student behaviour profiles: students from cluster one seemed to be especially cooperative and eager to help, which was e.g. derived from high participation in forums. They passed the exam with a rate of 77 per cent. The derived student profile for cluster six by contrast was that students from this group show average interactions and often use the chat function. They only passed the exam with a rate of 0.89 per cent. Clusters two to five corresponded to behaviour profiles between these two extremes. Although there is no causal link betweens these findings, Talavera et al. argue that the results can be used e.g. to form effective groups, as members of group one would e.g. make ideal moderators. This proves to be an example of application of clustering within LA that enables assessment of interaction.

2.4 Spectral Clustering

2.4.1 Defining Spectral Clustering

As K-Means usually has the Euclidean distance as distance measure, it is limited to finding clusters of spherical shape, see [Jai10]. Another set of algorithms known as *Spectral Clustering* algorithms enables to detect clusters of more complex shape such as concentric circles, ibid. Whereas K-Means operates directly on the data, Spectral Clustering only needs a representation of the data, i.e. a similarity graph, a weighted graph. Data points correspond to the nodes in the graph and the weights describe the similarity between two datapoints. Goal of Spectral Clustering is to find partitionings in the graph so that weights between nodes in the same group are high and between different groups low, see [TPSH10] for this definition of Spectral Clustering.

2.4.2 Applying Spectral Clustering to data from an Intelligent Tutoring System

Especially in countries like the USA, where education is based on high tuition-fees, it is quite common to try to detect students who might fail future classes, see e.g. [AP12]. This is a large field of application within *Acadamic Analytics*, a neighbouring discipline of LA that focuses on the needs of institutions rather than those of the individual learner. Trivedi et al. 2010 [TPSH10] implemented Spectral Clustering in order to prove that an *Intelligent Tutoring Systems* (ITS) implementing *dynamic assessment* can improve prediction of exam results compared to ITS that does not. An ITS is software that is comparably efficient to a human tutor in offering adaptive support to its users, see [ABR85]. Dynamic assessment refers to the amount of immediate support a student needs from an ITS in order to obtain a correct answer. If a student has a high level of knowledge the amount of help needed is likely to be low and vice versa, as the authors argue. Measuring the quantity of immediate feedback provided by the ITS can hence be used for prediction of exam results but also as measurement for formative as well as summative assessment purposes. The latter makes the study relevant for LA. The same study, using the same data, had already been conducted Feng et al. 2011, see [FHK11]. Feng et al. had however used K-Means and not Spectral Clustering. Input data was e.g. average time spent on and number of attempts needed for each item. The results proved to be statistically more significant using Spectral Clustering compared to K-Means. Trivedi et al. argue this to be reasoned in the similarity of data being independent of its geometry when using Spectral Clustering. Via conducting their study, Trivedi at al. 2010 furthermore used the success in prediction of exam results to evaluate the efficiency of the ITS. When students perform well on tests taken after practicing with an ITS, this can be interpreted as an indicator that the ITS was beneficial for the learning process. Hence, in this study Spectral Clustering helped to meet the LA objectives of monitoring and analysis, prediction and tutoring.

3 Summary and critical outlook

In this paper an overview of applications of clustering techniques within LA with special focus on K-Means was provided. K-Means proved to be able to identify learner groups, which can be used as basis for offering real-time adaptive support within an ELE. An extended version of K-Means, the EM algorithm, was successful in creating learner profiles, which can be used as basis for assessment of interaction, enabling e.g. to improve offline classroom methodology like group work. Finally, Spectral Clustering, a technique applicable to data in which K-Means fails to detect patterns, was introduced. Spectral Clustering achieved better results than K-Means, while running on data generated from usage of an ITS, proving that an ITS offering dynamic support enabled better prediction of test results. The weaknesses of K-Means such as the amount of clusters k that needs to be determined in advance, the risk of stopping at local instead of global optima and failure to detect patterns within high-dimensional data were discussed. There are many critical voices regarding ethical and privacy issues when applying LA methods like clustering in pedagogical settings. Perrotta et al. 2018 [PW18] even go so far as to argue that instead of being neutral methods for computer-aided learning, LA tools "creat[e] the educational realities they claim to measure" and define the "learner as a data construct" [PW18]. The notion of "data construct" and clustering being "structure-imposing" [MBR84] gets special weight when considering the mentioned fact that the semantics of clusters are ultimately defined subjectively, by a domain expert. If it is sufficient to implement a user's navigation style within an e-learning application, as illustrated in detail in 2.2.2, as criterion for being placed into a performance group, remains questionable - even if an external feature of performance within a test was added to draw respective conclusions. The risk of reducing a learner to (predefined) data is hence given. However, today's unprecedented need for schooling from distance, enabled by technology, brings to light many problems: simply transforming the regular face-to-face classroom methodologies to a *Zoom* conference is not proving to be convincingly successful. There is a great need for tools like ELEs and ITSs, offering automatic adaptive support and dynamic assessment to the students. This enables active learning without the physical presence of a teacher being a necessity. Especially when the user's data is not forwarded in any way and is only used as input for a learning application, benefits for learning seem to outweigh potential risks. Possible ethical problems seem to especially arise as soon as the data is used for summative assessment purposes in the sense of making judgements and giving grades. Yet, especially grading via distance schooling provides great difficulty. In particular, when it comes to grades that rely on monitoring a student for e.g. making grades for in-class participation. Hence, indeed using e.g. navigation style within an e-learning environment in order to also formatively assess a student might be a promising first attempt to solve pressing problems of our times in the field of education. In sum, the applications of clustering algorithms in this paper enabled to meet especially the LA objectives of monitoring and analysis, prediction, tutoring and assessment. The focus of LA is to improve the learning experience for students and teachers and the illustrated examples show that this is possible.

References

[ABR85] Anderson, John R. ; Boyle, C F. ; Reiser, Brian J.: Intelligent tutoring systems. In: *Science* 228 (1985), Nr. 4698, S. 456–462

[AC06] Amershi, Saleema ; Conati, Cristina: Automatic recognition of learner groups in exploratory learning environments. In: *International Conference on Intelligent Tutoring Systems* Springer, 2006, S. 463–472

[AP12] Arnold, Kimberly E. ; Pistilli, Matthew D.: Course signals at Purdue: Using learning analytics to increase student success. In: *Proceedings of the 2nd international conference on learning analytics and knowledge*, 2012, S. 267–270

[Bis06] Bishop, Christopher M.: *Pattern recognition and machine learning.* springer, 2006

[CDST12] Chatti, Mohamed A. ; Dyckhoff, Anna L. ; Schroeder, Ulrik ; Thüs, Hendrik: A reference model for learning analytics. In: *International Journal of Technology Enhanced Learning* 4 (2012), Nr. 5-6, S. 318–331

[Clo12] Clow, Doug: The learning analytics cycle: closing the loop effectively. In: *Proceedings of the 2nd international conference on learning analytics and knowledge*, 2012, S. 134–138

[Dil99] Dillenbourg, Pierre: *What do you mean by collaborative learning?* 1999

[DWRK10] Diziol, Dejana ; Walker, Erin ; Rummel, Nikol ; Koedinger, Kenneth R.: Using intelligent tutor technology to implement adaptive support for student collaboration. In: *Educational Psychology Review* 22 (2010), Nr. 1, S. 89–102

[FHK11] Feng, Mingyu ; Heffernan, Neil ; Koedinger, Kenneth: Student modeling in an intelligent tutoring system. In: *Intelligent tutoring systems in e-learning environments: Design, implementation and evaluation* (2011), S. 208–236

[GD12] Greller, Wolfgang ; Drachsler, Hendrik: Translating learning into numbers: A generic framework for learning analytics. In: *Journal of Educational Technology & Society* 15 (2012), Nr. 3, S. 42–57

[HKP11] Han, Jiawei ; Kamber, Micheline ; Pei, Jian: Data mining concepts and techniques third edition. In: *The Morgan Kaufmann Series in Data Management Systems* (2011), S. 83–124

[ID20] In: Ifenthaler, Dirk ; Drachsler, Hendrik: *Learning Analytics.* Berlin, Heidelberg : Springer Berlin Heidelberg, 2020. – ISBN 978–3–662–54368–9, 515–534

[Jai10] Jain, Anil K.: Data clustering: 50 years beyond K-means. In: *Pattern recognition letters* 31 (2010), Nr. 8, S. 651–666

[KM13] Kodinariya, Trupti M. ; Makwana, Prashant R.: Review on determining number of Cluster in K-Means Clustering. In: *International Journal* 1 (2013), Nr. 6, S. 90–95

[KV06] Kuncheva, Ludmila I. ; Vetrov, Dmitry P.: Evaluation of stability of k-means cluster ensembles with respect to random initialization. In: *IEEE transactions on pattern analysis and machine intelligence* 28 (2006), Nr. 11, S. 1798–1808

[MBR84] Mark, Aldenderfer ; Blashfield Roger, K: *Cluster Analysis, Beverly Hills.* 1984

[NDH+08] Niegemann, Helmut M. ; Domagk, Steffi ; Hessel, Silvia ; Hein, Alexandra ; Hupfer, Matthias ; Zobel, Annett: *Kompendium multimediales Lernen*. Springer Science & Business Media, 2008

[PW18] Perrotta, Carlo ; Williamson, Ben: The social life of Learning Analytics: cluster analysis and the 'performance'of algorithmic education. In: *Learning, Media and Technology* 43 (2018), Nr. 1, S. 3–16

[RC07] Rakhlin, Alexander ; Caponnetto, Andrea: Stability of k-means clustering. In: *Advances in neural information processing systems* 19 (2007), S. 1121

[RVG08] Romero, Cristóbal ; Ventura, Sebastián ; García, Enrique: Data mining in course management systems: Moodle case study and tutorial. In: *Computers & Education* 51 (2008), Nr. 1, S. 368–384

[ST10] Shamir, Ohad ; Tishby, Naftali: Stability and model selection in k-means clustering. In: *Machine learning* 80 (2010), Nr. 2-3, S. 213–243

[TG04] Talavera, Luis ; Gaudioso, Elena: Mining student data to characterize similar behavior groups in unstructured collaboration spaces. In: *Workshop on artificial intelligence in CSCL. 16th European conference on artificial intelligence* Citeseer, 2004, S. 17–23

[TPSH10] Trivedi, Shubhendu ; Pardos, Zachary ; Sárközy, Gábor ; Heffernan, Neil: Spectral clustering in educational data mining. In: *Educational Data Mining 2011*, 2010

[Tut18] `https://www.youtube.com/watch?v=1XqG0kaJVHY`

[VL10] Von Luxburg, Ulrike: Clustering stability: an overview. (2010)