Chapter 9

Cognitive Computing Applications in Education and Learning

M.T. Irfan* **and V.N. Gudivada**[†,1]
**Bowdoin College, Brunswick, ME, United States*
†East Carolina University, Greenville, NC, United States
[1]Corresponding author: e-mail: gudivadav15@ecu.edu

ABSTRACT

Education and learning stand out among many application areas of cognitive computing due to their practical appeal as well as their research challenge. We can gauge the level of research challenge once we recognize the broad spectrum of human learning, the complex and not fully understood human learning process, and various influencing factors of learning, such as pedagogy, technology, and social elements. We survey some of the key research problems and recent efforts to solve them. In particular, we examine the important roles played by the Educational Data Mining and Learning Analytics communities in these research efforts. We find evidence of significant progress over the last few years. We also find evidence of concerted efforts by academia and industry in bringing some of these research results to the public in the form of learning and educational software.

Keywords: Cognitive learning, Learning Analytics, Educational Data Mining, Cognitive systems, Cognitive computing, Personalized learning, Data science

1 INTRODUCTION

Much of the excitement about cognitive computing is spurred by its enormous potential in learning, only a small fraction of which has so far been realized. The overarching goal here is to devise computational frameworks to help us learn better by exploiting data about our learning processes and activities. There are two important aspects of it—the mechanisms or insights about *how* we actually learn and the external manifestations of our learning activities. The state of the art at this moment reveals that a lot of energy and focus have gone into the latter front, while a lot remains to be done on the former front. An example will help distinguish these two.

Handbook of Statistics, Vol. 35. http://dx.doi.org/10.1016/bs.host.2016.07.008

It is well known that humans learn from a very small number of examples. For instance, it takes us only one example of a stop sign and one example of a lobster to differentiate one from the other, even though both are red. It is an extremely complex question how our brains process these two different objects in terms of their representations as well as their generalizations. The representation of a stop sign may encode some visual clues about a stop sign, such as its polygonal shape and its color, whereas its generalization would enable us to recognize stop signs from various distances as well as angles. Devising a computational framework to learn from a very small number of examples is currently at a nascent stage and obviously has an enormous potential for future work (Lake et al., 2015).

In contrast, we can devise a computational framework for the same task *if* we are allowed to use a large number of examples. More specifically, we can "train" a machine learning algorithm with a large number of labeled examples of both stop signs and lobsters (Murphy, 2012). Typically, these examples are represented by various features. In some applications these features are handcrafted while in others they are automatically discovered by the algorithm itself. The important requirement here is a large amount of data. In education and learning, we often generate a huge amount of data through our learning activities, which presents us with both an enormous potential and a great challenge. The reader may readily relate to this as *Big Data*.

Big Data are typically characterized by five V's—volume (data are large-scale), variety (data come in many different forms), velocity (data are generated over time, often times in a "streaming" fashion), veracity (data have some elements of uncertainty), and value (extracting actionable intelligence). In the application settings of education and learning, the first four V's are prominently visible. For example, a Learning Management System (LMS) like Moodle (2016) or any Massive Open Online Course (MOOC) generates a huge volume of data (see Section 2). Moreover, the data come in a variety of forms, such as a student's answers to quantitative questions as well as essay-type questions, which are very different in many respects. It is also obvious that the data are generated continually with great velocity and that there will be uncertainty in the data if we collect data about students' level of engagement or their understanding of materials. The cognitive computing community has embraced the emergence of Big Data, thereby shaping up several subfields relevant to education and learning. The ongoing efforts have led to many success stories.

In the next section, we discuss Educational Data Mining (EDM) and Learning Analytics (LA)—two major areas of cognitive computing dealing with education and learning. Next, we present a survey of some recent results in EDM and LA. We conclude this chapter by touching upon some success stories and giving pointers to several challenging research directions. The following acronyms are used in this chapter.

ACT-R	Adaptive Control of Thought-Rational
AFM	Additive Factor Model
EDM	Educational Data Mining
ISPeL	Interactive System for Personalized eLearning
ITS	Intelligent Tutoring Systems
LA	Learning Analytics
LAK	Learning Analytics and Knowledge (Conference)
LMS	Learning Management System
MOOC	Massive Open Online Course
PFA	Performance Factors Analysis

2 EDM AND LA

EDM and LA are two emerging areas that embody the major applications of cognitive computing to education and learning. These two areas are not separated by any hard boundary and to the contrary, they often share common objectives and stakeholders. However, they fundamentally differ in several dimensions, including research methodology. These differences have been beautifully explained by Siemens and Baker (2012). In brief, humans play a central role in LA, whereas the bulk of the work in EDM is performed through automation. Humans do play a role in EDM, but that is peripheral in comparison to LA where they are the key components of the apparatus. Furthermore, LA models and analyzes the learning system as a whole, in contrast to EDM's divide-and-conquer approach. Next, we address these two areas separately. We also give an illustrative example for each.

2.1 Educational Data Mining

EDM has been inspired by advances in data mining and machine learning. The overarching goal is to help teachers, students, and other stakeholders achieve their respective objectives by utilizing Big Data in education. This critically hinges on understanding and modeling how students learn. As a result, The International Educational Data Mining Society (2016) has defined the field as "concerned with developing methods for exploring the unique and increasingly large-scale data that come from educational settings, and using those methods to better understand students, and the settings which they learn in." We should note here that this society organizes the annual International Conference on Educational Data Mining, which debuted in 2008.

The recurrent research themes in EDM are: mining educational data from various sources that include MOOCs and Intelligent Tutoring Systems (ITS); mining human–computer interaction (HCI) data in the educational setting; assessing students' affective state using multimodal sensory data; modeling students and predicting their performance; modeling, curating, and evaluating learning pathways; and connecting EDM with other related disciplines, including emerging areas like LA (see Section 2.2) as well as more mature areas like learning theory and learning sciences.

Although EDM started as a subfield of data mining, it now boasts an interesting and varied research landscape. As we will see in Section 3, it has welcomed tools and techniques from HCI and computer vision. In fact, many of the recent works in EDM can be thought of as data-centric applications of cognitive computing in education and learning.

For the most part, the methodologies in EDM research are rooted in several major areas of computer science, such as data mining and machine learning. Although EDM research typically keeps humans in the loop, the major focus is on automation through computational modeling, tools, and techniques. Next, we present a simplistic example to illustrate this point.

2.1.1 An EDM Example

For this fictitious example, imagine that a tutor has access to multidimensional data about her students' learning behavior on an online learning system. In particular, the system has recorded each student's number of sign-ins and the percentage of lesson completion as a measure of the student's progress. The input data points are shown as black dots in Fig. 1A. This figure is generated using an online visualization tool (Mohan, 2016). The tutor's goal is to *classify* her students according to their behavior to better assist them. However, the labeling for this classification is not clear a priori. This is a typical unsupervised learning problem in machine learning, also known as clustering (Murphy, 2012).

Among many different approaches to clustering, we pick the well-known k-means clustering algorithm for this example. The parameter k represents the number of clusters desired. An illustration is shown in Fig. 1 for $k = 4$. The first step is to choose k points for cluster centers (aka centroids), often times randomly. The initial centroids are shown as triangles in Fig. 1A.

Assuming that the points lie in a 2D Cartesian coordinate plane, the x coordinate of the centroid is the average of the x coordinates of all the points that belong to the cluster. The y coordinate of the cluster is calculated similarly. The same principle applies for points in an n-dimensional space. However, a suitable distance metric needs to be defined to measure distance between two points. The centroids need not be actual data points. Next, each data point is classified to its closest centroid based on a distance metric. This is shown in Fig. 1B, where different classes of points are shown in different colors. This classification is not an ideal grouping of points since cluster boundaries are not crisp. However, this completes one iteration of the algorithm.

In the next iteration, for each class, its new centroid is recalculated. The recalculated centroids are shown in Fig. 1C. Since the centroids have changed,

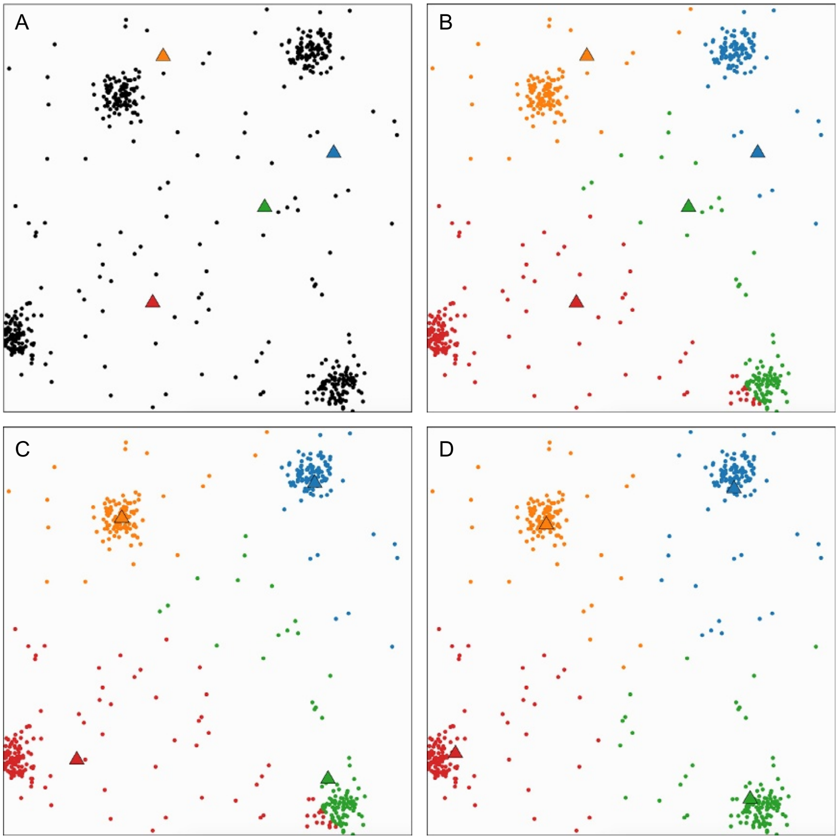

FIG. 1 (A) Input data points and four initial centroids (*triangles* filled with *different colors* (different *gray shades* in the print version)) are shown. The *x*-axis represents the number of times a student has signed in and the *y*-axis the percentage of lesson completion. Each data point corresponds to one student. (B) The data points are classified according to their proximity to the centroids. (C) The centroid of each of the four classes is recalculated by taking the average of the *x* and *y* coordinates of the data points in that class. (D) Successive iterations of centroid (re-)calculation and data point classification ultimately converge to this clustering. An interpretation of these clusters requires human experts with domain knowledge.

some of the points will now move into a different class. The recalculation of the centroids and the classification of the points are repeated, ultimately converging to the scenario shown in Fig. 1D.

The interpretation of clustering is most often left to human experts with domain knowledge—in this case, the tutor. The tutor may interpret the bottom left cluster shown in Fig. 1D as the group of students who made very little progress and also lacked in their level of efforts as evident from the small number of sign-ins. She may interpret the bottom right cluster as the group of students who signed in many times but still could not make much progress, perhaps giving an indication of short attention spans. Based on her interpretation of the four clusters, the tutor may design separate learning paths for different types of students.

This simple example illustrates many of Siemens and Baker's points on EDM (Siemens and Baker, 2012). First, it shows that the main focus of EDM is automated discovery. Human expertise is a means to that end. Second, EDM's approach of breaking up a problem into smaller parts is obvious here, since we are dealing with one small part of what we may call ITS. More interestingly, we can extend this example to model students with the goal of predicting learning outcomes. Lastly, Fig. 1 illustrates some of the common tools used in EDM, such as clustering and visualization.

2.2 Learning Analytics

LA is a relatively new field with the overarching goal of synergistically combining human judgment with technological advances for a wide range of learning activities, spanning from classroom learning to informal learning. The first International Conference on Learning Analytics and Knowledge (LAK) was held in 2011. At that time, the LA community identified LA field as: "Learning analytics is the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimising learning and the environments in which it occurs" (Long and Siemens, 2011). Since then the field has evolved significantly. For example, the sixth LAK conference held in 2016 Gašević and Lynch (2016) has identified five major branches of LA: (1) analytic approaches, methods, and tools; (2) theories and theoretical concepts; (3) measures of learning, change, and success; (4) learning activities, applications, and interventions; and (5) data sources.

Compared to EDM, LA embodies a broader umbrella of learning settings, tools, techniques, human and social factors, and applications areas. EDM is focused on automation and often times concerns with unearthing hidden patterns from data, whereas LA is more holistic and human-centric (Siemens and Baker, 2012). In a similar vein, as we will see in Section 3, predicting student performance is a central problem in EDM. In contrast, LA is more concerned with predicting the outcome of a learning approach or system. Also, LA is inherently an area of analytics, and therefore various analytical tools, such as discourse analysis and social network analysis, are more prominent in LA than in EDM. However, the border between these two fields is sometimes blurred due to their shared objectives and stakeholders. Furthermore, since both fields deal with predictive models (although applied to different domains), many of the computational techniques pertaining to machine learning and data mining are common between them. Next, we present an example of a cognitive model for LA.

2.2.1 An LAs Example

Given LA's holistic approach to an overall system, it is hard to come up with simple examples like the one we presented for EDM in Section 2.1.1. Here, we present a recent application architecture proposed by Gudivada (2016) for personalized learning. He notes that learning applications typically generate huge volumes of structured and unstructured data. Structured data can be easily obtained from a LMS's logs, whereas unstructured data come from a variety of sources such as blogs, emails and other forms of course messages, and discussions. Due to the ease of working with structured data, a vast majority of research on LA deals with only that type of data. However, if we consider building an effective system for personalized learning, dealing with unstructured data becomes essential. Gudivada proposes a comprehensive framework for personalized learning that can work with both structured and unstructured data. The framework is named Interactive System for Personalized eLearning (ISPeL).

Gudivada's architecture is hierarchical, roughly consisting of the following layers: network and hardware layer, databases and data analytics layer, cognitive analytics layer, and personalization layer. An illustration of the application level architecture is shown in Fig. 2. Some preliminaries are due before we describe this architecture.

First, Darwin Information Typing Architecture (DITA) is an XML data model for single source authoring and multichannel publishing (Bellamy et al., 2011; White, 2013). A DITA content (e.g., computer algorithms) is composed of DITA topics (e.g., time and space complexity, greedy algorithms, dynamic programming). It is desirable that each topic is minimal but self-contained. Second, a Resource Description Framework Schema is a general-purpose ontology language (Broekstra et al., 2001), typically used for storing domain information and topic hierarchies. Third, LMSs are software applications to facilitate education. Well-known examples of LMSs are Blackboard, Moodle, and D2L.

We are now ready to describe the ISPeL architecture in a bottom-up fashion. The bottom layer of ISPeL (Fig. 2) consists of nine databases. These databases dynamically keep track of information from various sources in order to provide the students a personalized learning experience. For example, the *DITA Topics* database keeps information about the topics being taught. The *Assessment Data* consists of each student's scores, time to complete tests, performance level on tests, and other related information. The *User Models* database keeps information about individual students, their preferences and progress, and learning styles.

The Question Generation Models, Scaffolding & Feedback Models, and Personalization Models are the main components for generating personalized questions, feedback, and lesson plans. The *Question Generation Models* module has capability to automatically generate questions for assessing learners.

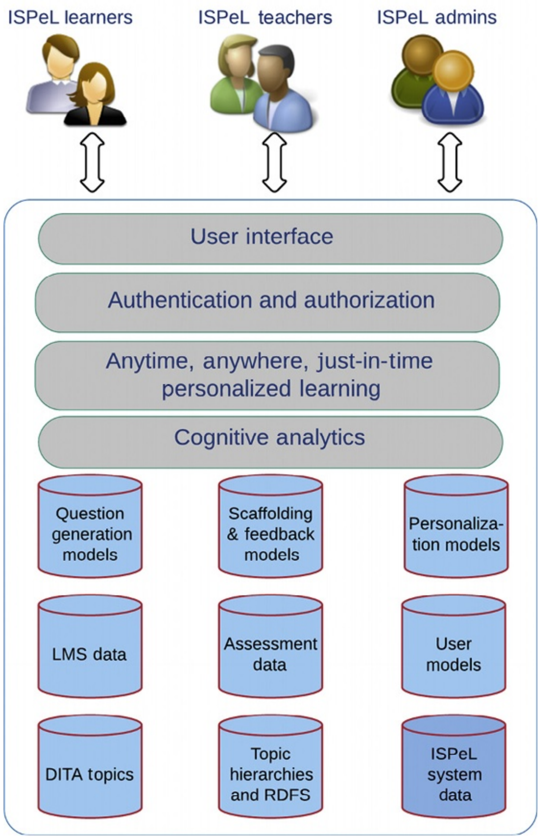

FIG. 2 ISPeL architecture for personalized learning (Gudivada, 2016).

This module also supports practice exams before the actual test. The *Scaffolding & Feedback Models* module features capability to provide guided learning and context-appropriate feedback. The *Personalization Models* component uses prerequisite dependencies among DITA topics and user models to achieve personalization. Each learner progresses at her own pace, explores learning materials in any order they choose subject to prerequisite constraints.

The cognitive analytics layer uses the data mentioned earlier together with various algorithmic and analytical techniques (e.g., data mining, machine learning, statistical analysis, natural language processing, information visualization) to provide personalized learning to students. The layer above it deals with multiple modes of personalized learning, such as anytime, anywhere, and just-in-time. The next two layers provide authentication and user interface, through which users with different credentials interact with the software application.

The ISPeL architecture shown in Fig. 2 is in tune with Siemens and Baker's call for connecting LA with EDM (Siemens and Baker, 2012). The architecture itself is holistic, but some of the individual components of the whole system are significantly challenging in their own merit. For example, for ISPeL to provide personalized learning, it is necessary to generate appropriate questions and feedback for each individual learner *automatically* and in real time. An EDM expert can help with this automation. However, this automated component must function robustly in concert with the other components of the cognitive analytics layer, which illustrates both the beauty and challenge of LA.

3 RECENT RESEARCH

In this section, we review some recent research on cognitive computing and its applications to education and learning. Many of these research results have appeared in the EDM and LAK conferences, which are the premier venues for publishing research on EDM and LA, respectively. Since EDM and LA are both very much expansive areas, it will not be possible to touch upon many of their subareas in this chapter. As a result, we will only present some recent results in a few selected problem domains. The intention is to give the reader a sampling of the progress made during the last few years. Let us first illustrate the computational challenges using the following example.

Many of our cognitive tasks involve semistructured as well as unstructured input data. Among these are tasks that we deem very much natural to us. One example is geometric word problems. Fig. 3 shows a sample problem. Problems of this type are often specified by a text description in conjunction with a diagram. In many cases, the text description alone is not sufficient to fully understand the problem, since some critical information is only given in the diagram. For example, the text description in Fig. 3 omits the important fact that AD intersects with BC at point D. Inferring such facts by combining a text description with the corresponding diagram is natural to us. An extremely challenging problem is to automate it. Very recently, Seo et al. (2015) have presented an automated solver for SAT-level word geometric problems. Their method consists of the following main components: a text parser, a diagram

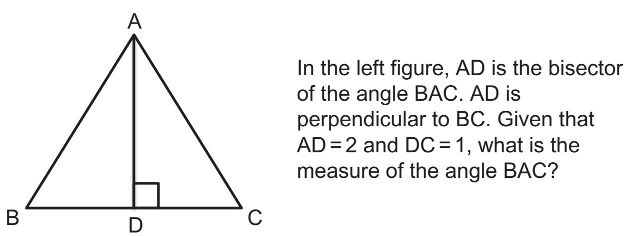

FIG. 3 A sample geometric word problem, similar to the examples given by Seo et al. (2015).

parser, and an optimizer. These components work together to produce the best possible formal representation of the problem that can be fed to a standard geometric problem solver. On unseen, official SAT geometry problems, their automated solver achieves a remarkable 49% score.

As evident from the earlier example, the full spectrum of human cognition and learning poses a grand, multiprong research challenge. This challenge is only magnified when we also consider pedagogy, technology, human judgment, social factors, and various contextual elements. We next present a survey of some recent research results in several selected categories.

3.1 Intelligent Tutoring Systems

Since the 1970s (Carbonell, 1970), ITS have been at the forefront of artificial intelligence research with a diverse array of application areas ranging from physics (Vanlehn et al., 2005) and mathematics (Matsuda and VanLehn, 2005; Melis and Siekmann, 2004) to adult education (Cheung et al., 2003) and nurse education (Hospers et al., 2003). Here, we will narrow down on some recent ITS research on learning programming. Since the publication of Anderson and Reiser's seminal work on an LISP tutoring system (Anderson and Reiser, 1985), this field has gained a lot of attention over the years. However, it is fair to say that we are still far from realizing its full potential.

Any discussion on modern-day ITS must begin with the cognitive architectures behind ITS. One of the most influential cognitive architectures is Anderson's Adaptive Character of Thought-Rational (ACT-R) theory. The central tenet of the ACT-R theory is that human cognition is the result of interactions among numerous small, indivisible units of knowledge in certain ways. First, these basic units of knowledge are of two types: declarative and procedural. For any cognitive task, the declarative units of knowledge represent the states of cognition, or in Anderson's words, "objects in the environment" (Anderson, 1996). In contrast, procedural units of knowledge, also known as *productions*, transform human cognition from one state to another.

The ACT-R theory gives the details of how these knowledge units interact, which is out of scope for this chapter. However, we should mention that ACT-R is not an abstract theory of human cognition. It is rather a concrete framework similar to a programming language. Using a programming language, we can write programs and run them. Similarly, using ACT-R, we can create a cognitive model for a specific cognitive task (e.g., computing the factorial of a number) and simulate it. The simulation details the steps in human cognitive process for that particular task under various modeling assumptions. These assumptions are usually grounded in extensive psychological experiments. We should also be noted that the ACT-R theory—in its original form—comes with a LISP interpreter where simulations can be run. Today, there are also Python and Java implementations of the ACT-R theory (Harrison, 2002; Stewart and West, 2005).

Founded on several decades of research on human cognition and intelligence, ITS is now a fast growing area in academia and industry. We now turn our attention to some cutting-edge research on ITS in a specific learning domain: programming. Several recent papers deal with the problem of helping students learn programming, in particular by giving them useful hints in real-time while they are coding. This is a vastly challenging problem, mainly because even for very simple programming tasks there are a multitude of different solution approaches, both syntactically and semantically. Even if we restrict the semantic aspect (i.e., the underlying algorithm) to a single one, the syntactic variations of implementing the algorithm present a daunting task for hint generation.

Not surprisingly, earlier work on hint generation focused on more specific problem-solving tasks, such as logic (Barnes et al., 2008; Stamper et al., 2013) and linked lists (Fossati et al., 2009). New efforts to tackle broader programming problems are at a nascent stage and use previous students' solutions to a programming task to generate hints for a new student who is working on the same task (Price and Barnes, 2015; Price et al., 2016). The basic technique in this new line of work is to first represent the previous student–tutor interactions in the form of an *interaction network* (Eagle et al., 2012). When a new student asks for a hint, that student's interaction pattern is matched with some part of the interaction network and the student is directed to an appropriate next step that ultimately leads to a solution. It is not hard to imagine the potential impact of such work on any ITS that teaches programming.

In a related work, Eagle et al. (2015) present a social network analysis of the interaction network for multiple problem domains. They found that the interaction networks exhibit power-law degree distribution. They also showed the importance of several network properties like *degree assortativity* for comparing interaction networks across different domains.

Closely connected to improving an ITS is evaluating its adaptive tutoring feature, which is very much applicable to ITS for programming. Traditionally, the literature suggests using evaluation schemes based on machine learning, such as the Performance Factors Analysis (PFA) cognitive model. This poses a challenge to course designers without a background in machine learning. A recent work addresses this issue and presents two new evaluation metrics better suited for adaptive tutoring than PFA (González-Brenes and Huang, 2015).

3.2 Clustering and Student Modeling

Clustering is a common technique in EDM for aggregating student data in order to examine student behavior. Klingler et al. (2016) have recently presented an evolutionary clustering technique for sequential student data. In comparison with previous works, their technique improves the stability of clustering for noisy data. In addition, they show that their work can be incorporated into any ITS as a black box.

Another central problem in EDM is *student modeling*. A lot of interesting research has emerged on this topic recently. One such work is on modeling learning curves. Using data from Duolingo (2016), Streeter (2015) has used probabilistic mixture models to capture the learning curves of language learners. Here, an individual's learning curve represents her error percentage over time. This is an interesting research direction, because prior to this, *knowledge tracing* has been the dominant criterion for modeling student learning (Corbett and Anderson, 1994). Streeter's work generalizes knowledge tracing and offers an elegant probabilistic model for modeling learning curves. The parameters of the model have been learned using the well-known expectation–maximization algorithm. Based on the large-scale Duolingo dataset, the mixture model outperforms many of the previous approaches, including popular cognitive models like Additive Factor Model (AFM) (Cen et al., 2006) and PFA (Pavlik et al., 2009).

Others have also worked to improve the AFM and PFA models. A recent cognitive model by MacLellan et al. (2015) has extended these in order to account for the commonly observed phenomenon that students often make mistakes on previously learned skills, which the authors have termed *slipping*. Their slipping models outperform the baseline AFM and PFA models on five datasets.

Although a lot of work has been done on modeling student learning, modeling prior knowledge has received relatively less attention. In a recent work, Nižnan et al. (2015) have presented several models to estimate prior knowledge. They started with the basic model of *Elo rating* and extended it in several ways. They evaluated the models on prior knowledge of geography, which is a subject of widely varying prior knowledge. Perhaps surprisingly, they showed that the more complex extensions of the basic Elo rating model do not outperform Elo rating by much. Although not mentioned explicitly in their paper, this could be attributed to the basic Elo rating hitting the sweet spot between overfitting and underfitting (Murphy, 2012) for that particular geography dataset.

3.3 Predicting Student Performance

Predicting student performance based on various factors has been a popular line of research in EDM. Traditionally, student performance has been measured in specific disciplines or topics, such as algebra (Stapel et al., 2016), programming in Java (Tomkins et al., 2016), or even as specific as learning fractions in mathematics (Olsen et al., 2015). Although many of the research questions on student performance are domain-specific, there are some unifying themes: How would one predict student performance? What factors are most important for a student's success? How can an ITS make use of this prediction?

Tomkins et al. (2016) answer some of these questions using their case study of a high school computer science MOOC. The MOOC is a separate course with its own evaluation, but the students taking the MOOC ultimately take the computer science Advanced Placement (AP) exam. As a result, there are two performance measures: one coming from the MOOC and the other from the AP exam. It has been empirically observed that a student's score in the AP exam is a better predictor of the student's future success than the student's performance in the MOOC. Based on their machine learning framework, Tomkins et al. identify the most important factors in MOOC for students' high and low scores in the AP exam (not MOOC). One such factor is coaching. Many of the students received coaching while taking the MOOC, while others studied independently. The ones that were coached, showed a greater level of activity in the MOOC's forum through questions, answers, and other contributions. The coached students performed better than the independent students in the MOOC. However, on the actual AP exam, the independent students scored higher. The authors also connect other features, such as textual content of the students' forum activities, to their AP exam performance.

Another new study in the context of MOOCs predicts student drop-out and proposes an intervention study (Whitehill et al., 2015). Student drop-out is an extremely common phenomenon in MOOCs (Onah et al., 2014; Rivard, 2013). Whitehill et al. note that the reasons for drop-out are not necessarily related with the quality of the course as there could be exogenous factors. More importantly, a post-course survey is ineffective in detecting the reasons for drop-outs, since the response rates for such surveys are usually very low. The authors present a classification technique for detecting student drop-outs and propose an intervention strategy in the form of early surveys in order to retain students. They used the HarvardX MOOC platform to evaluate the effectiveness of their strategy.

3.4 Affect Detection and Student Engagement

Affective states are closely related to learning and cognition. We react differently to different experiences during our learning process. Assessing affective states of students is now gaining traction within both the EDM and LA communities. As we have briefly mentioned in Section 2.1, these problems often transcend the boundary of EDM beyond traditional data mining and bring computer vision and multimodal sensory analysis into the fold of EDM research.

Based on recent research trajectories, we must start with the eyes. Episodes of mind wandering is common in various learning activities like reading (Mills and D'Mello, 2015) as well as interacting with an ITS (Mills et al., 2015). Multiple research works have shown that our eyes hold a clue to detecting whether we are paying attention or not (Bixler et al., 2016; Hutt et al., 2016). The interesting observation here is that eye-tracking has been an active research topic in HCI since the inception of HCI and is now contributing to cognitive computing in learning.

As we have mentioned earlier, students' affective states are closely tied to their learning and therefore to their performance. Based on over one thousand middle-school students' interactions with a math tutoring system, San Pedro et al. (2015) find that the fine-grained variations of affect over time ultimately impact a student's test score. For affect detection, the authors used the data of student interactions with the math tutoring system mentioned earlier. There exists a long line of prior work on such interaction-based affect detection (Pardos et al., 2013; Wang et al., 2015) and that many of the affect detectors today are video based.

Kai et al. (2015) have done a comparative study of these two major affect detection methods—video based and interaction based—for various affective states, such as boredom, confusion, delight, frustration while the students play a game called physics playground. They obtained the ground truth about the students' affective states through field observations. They showed that overall the video-based affect detector slightly outperforms the interaction-based one. This can be largely attributed to the affective state of delight for which the video-based detector is very much the superior between the two.

4 CONCLUSION

The frequent interactions between academia and industry are a remarkable feature of cognitive computing, especially when it comes to its applications in education and learning. For example, companies like IBM are embracing cognitive computing to harness the power of Big Data in multiple application areas, including education (Davis, 2016). In another success story, the ACT-R theory that we have discussed in Section 3.1 is the key driver of Carnegie Learning's *MATHia* as well as *Cognitive Tutor* software (Carnegie Learning, Inc., 2016). The Cognitive Tutor software has been adopted by the Miami-Dade county public schools in Florida since 2003, following the finding by an independent research group that the software is indeed more effective than the traditional curriculum alone. Furthermore, the new generation of LMS software like Desire2Learn (D2L) boasts predictive analytics features to identify and help at-risk students (D2L, 2016). Many online platforms for class discussion like BlikBook, which can often work in conjunction with an LMS, have built-in student behavior analytics (BlikBook, 2016). There are many other success stories stemming from the exchanges between scientific research and real-world applications.

Two premier conferences in this field—EDM and LA—are great contributors to a lot of these success stories. These two communities are driven by a grand vision and some clearly identified long-term goals. For example, we would like to build an ITS that can offer personalized learning for complex learning tasks like computer programming and adapt itself with each individual student's learning curve as well as prior knowledge. We would like to detect when a student loses engagement, both in a classroom setting and in an ITS session. Not only that, we would like to identify the causal structure behind it and design appropriate intervention strategies. We would like to design MOOCs with high student retention rates. We would also like to accurately predict student performance from behavioral data that comes in different varieties: structured, semistructured, and unstructured.

In summary, we would like to build cognitive computing applications that will have a greater understanding of how humans learn and can vastly improve the quality of learning and education. Are we there yet? No. Have we made any significant progress? Big yes. In fact, it is fair to say that the progress made so far is only the tip of the iceberg. What lie ahead are some extremely hard research problems and many untrodden research paths to solve those.

REFERENCES

Anderson, J.R., 1996. ACT: a simple theory of complex cognition. Am. Psychol. 51 (4), 355.

Anderson, J.R., Reiser, B.J., 1985. The LISP tutor. Byte 10 (4), 159–175.

Barnes, T., Stamper, J.C., Lehmann, L., Croy, M.J., 2008. A pilot study on logic proof tutoring using hints generated from historical student data. In: EDM, pp. 197–201.

Bellamy, L., Carey, M., Schlotfeldt, J., 2011. DITA Best Practices: A Roadmap for Writing, Editing, and Architecting in DITA, first ed. IBM Press, Indianapolis, IN.

Bixler, R., Mills, C., Wang, X., D'Mello, S., 2016. Automatic gaze-based detection of mind wandering during film viewing. In: Proceedings of EDM.

BlikBook, 2016. A new approach to classroom Q&A. https://www.blikbook.com.

Broekstra, J., Klein, M., Decker, S., Fensel, D., Van Harmelen, F., Horrocks, I., 2001. Enabling knowledge representation on the web by extending RDF schema. In: Proceedings of the 10th International Conference on World Wide Web, Hong Kong, WWW '01ACM, New York, NY, pp. 467–478.

Carbonell, J.R., 1970. AI in CAI: an artificial-intelligence approach to computer-assisted instruction. IEEE Trans. Man Mach. Syst. 11 (4), 190–202.

Carnegie Learning, Inc., 2016. MATHia and Cognitive Tutor personalized learning software. https://www.carnegielearning.com.

Cen, H., Koedinger, K., Junker, B., 2006. Learning factors analysis—a general method for cognitive model evaluation and improvement. In: Intelligent Tutoring Systems. Springer, New York, NY, pp. 164–175.

Cheung, B., Hui, L., Zhang, J., Yiu, S.M., 2003. SmartTutor: an intelligent tutoring system in web-based adult education. J. Syst. Softw. 68 (1), 11–25.

Corbett, A.T., Anderson, J.R., 1994. Knowledge tracing: modeling the acquisition of procedural knowledge. User Model. User Adap. Inter. 4 (4), 253–278.

D2L, 2016. Desire2Learn learning management system. www.d2l.com.

Davis, K., 2016. IBM launches industry's first consulting practice dedicated to cognitive business. http://www-03.ibm.com/press/us/en/pressrelease/47785.wss.

Duolingo, 2016. A game-based approach to learning natural languages. https://www.duolingo.com.

Eagle, M., Johnson, M., Barnes, T., 2012. Interaction networks: generating high level hints based on network community clustering. In: Proceedings of the 5th International Conference on Educational Data Mining. ERIC.

Eagle, M., Hicks, D., Peddycord III, B., Barnes, T., 2015. Exploring networks of problem-solving interactions. In: Proceedings of the Fifth International Conference on Learning Analytics and Knowledge. ACM, New York, NY, pp. 21–30.

Fossati, D., Di Eugenio, B., Ohlsson, S., Brown, C.W., Chen, L., Cosejo, D.G., 2009. I learn from you, you learn from me: how to make iList learn from students. Frontiers in Artificial Intelligence and Applications, vol. 200, pp. 491–498.

Gašević, D., Lynch, G. (Eds.), 2016. LAK '16: Proceedings of the Sixth International Conference on Learning Analytics and Knowledge, Edinburgh, United Kingdom. ACM, New York, NY. ISBN 978-1-4503-4190-5.

González-Brenes, J.P., Huang, Y., 2015. Your model is predictive—but is it useful? Theoretical and empirical considerations of a new paradigm for adaptive tutoring evaluation. In: Proceedings of EDM.

Gudivada, V., 2016. Cognitive analytics driven personalized learning. In: Educational Technology Magazine Special Issue—Big Data and Data Analytics in E-Learning, in press.

Harrison, A., 2002. JACT-R: Java ACT-R. In: Proceedings of the 8th Annual ACT-R Workshop.

Hospers, M., Kroezen, E., Nijholt, A., op den Akker, R., Heylen, D., 2003. An agent-based intelligent tutoring system for nurse education. In: Moreno, A., Nealon, J. (Eds.), Applications of Software Agent Technology in the Health Care Domain. Springer, New York, NY, pp. 143–159.

Hutt, S., Mills, C., White, S., Donnelly, P.J., D'Mello, S.K., 2016. The eyes have it: gaze-based detection of mind wandering during learning with an intelligent tutoring system. In: Proceedings of the 9th International Conference on Educational Data Mining, International Educational Data Mining Society, pp. 86–93.

Kai, S., Paquette, L., Baker, R., et al., 2015. Comparison of video-based and interaction-based affect detectors in physics playground. In: Proceedings of EDM, pp. 77–84.

Klingler, S., Käser, T., Solenthaler, B., Gross, M., 2016. Temporally coherent clustering of student data. In: Proceedings of the 9th International Conference on Educational Data Mining, International Educational Data Mining Society, pp. 102–109.

Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B., 2015. Human-level concept learning through probabilistic program induction. Science 350 (6266), 1332–1338. ISSN 0036-8075. http://dx.doi.org/10.1126/science.aab3050. http://science.sciencemag.org/content/350/6266/1332.

Long, P., Siemens, G. (Eds.), 2011. LAK '11: Proceedings of the 1st International Conference on Learning Analytics and Knowledge, Banff, Alberta, Canada. ACM, New York, NY. ISBN 978-1-4503-0944-8.

MacLellan, C.J., Liu, R., Koedinger, K.R., 2015. Accounting for slipping and other false negatives in logistic models of student learning. In: Proc. EDM.

Matsuda, N., VanLehn, K., 2005. Advanced geometry tutor: an intelligent tutor that teaches proof-writing with construction. In: AIED, vol. 125. pp. 443–450.

Melis, E., Siekmann, J., 2004. Activemath: an intelligent tutoring system for mathematics. In: International Conference on Artificial Intelligence and Soft Computing. Springer, pp. 91–101.

Mills, C., D'Mello, S., 2015. Toward a real-time (day) dreamcatcher: sensor-free detection of mind wandering during online reading. In: Proceedings of the 8th International Conference on Educational Data Mining. International Educational Data Mining Society, pp. 69–76.

Mills, C., D'Mello, S., Bosch, N., Olney, A.M., 2015. Mind wandering during learning with an intelligent tutoring system. In: Artificial Intelligence in Education. Springer, New York, NY, 267–276.

Mohan, K., 2016. Visualizing K-means clustering, Stanford, CA. http://stanford.edu/class/ee103/visualizations/kmeans/kmeans.html.

Moodle, 2016. A learning management system. https://moodle.org/.
Murphy, K.P., 2012. Machine Learning: A Probabilistic Perspective. MIT Press, Cambridge, MA.
Nižnan, J., Pelánek, R., Řihák, J., 2015. Student models for prior knowledge estimation. In: Educational Data Mining.
Olsen, J.K., Aleven, V., Rummel, N., 2015. Predicting student performance in a collaborative learning environment. In: Proceedings of EDM.
Onah, D.F., Sinclair, J., Boyatt, R., 2014. Dropout rates of massive open online courses: behavioural patterns. In: Proceedings of the 8th International Conference on Educational Data Mining, International Educational Data Mining Society, pp. 211–217.
Pardos, Z.A., Baker, R.S., San Pedro, M.O., Gowda, S.M., Gowda, S.M., 2013. Affective states and state tests: investigating how affect throughout the school year predicts end of year learning outcomes. In: Proceedings of the Third International Conference on Learning Analytics and Knowledge. ACM, New York, NY, 117–124.
Pavlik, P.I., Cen, H., Koedinger, K.R., 2009. 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 Modeling. IOS Press, Amsterdam, The Netherlands, pp. 531–538. http://dl.acm.org/citation.cfm?id=1659450.1659529, ISBN 978-1-60750-028-5.
Price, T.W., Barnes, T., 2015. An exploration of data-driven hint generation in an open-ended programming problem. In: Workshop on Graph-Based Data Mining held at Educational Data Mining (EDM).
Price, T., Dong, Y., Barnes, T., 2016. Generating data-driven hints for open-ended programming. In: Proceedings of the 9th International Conference on Educational Data Mining, International Educational Data Mining Society, pp. 191–198.
Rivard, R., 2013. Measuring the MOOC Dropout Rate. Inside Higher Ed, 8, https://www.insidehighered.com/news/2013/03/08/researchers-explore-who-taking-moocs-and-why-so-many-drop-out.
San Pedro, M.O.Z., Snow, E.L., Baker, R.S., McNamara, D.S., Heffernan, N.T., 2015. Exploring dynamical assessments of affect, behavior, and cognition and math state test achievement. In: Proceedings of EDM.
Seo, M., Hajishirzi, H., Farhadi, A., Etzioni, O., Malcolm, C., 2015. Solving geometry problems: combining text and diagram interpretation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP, pp. 17–21.
Siemens, G., Baker, R., 2012. Learning analytics and educational data mining: towards communication and collaboration. In: Proceedings of the 2nd International Conference on Learning Analytics and Knowledge. LAK '12. Vancouver, British Columbia, Canada. ACM, New York, NY, pp. 252–254.
Stamper, J., Eagle, M., Barnes, T., Croy, M., 2013. Experimental evaluation of automatic hint generation for a logic tutor. Int. J. Artif. Intell. Educ. 22 (1–2), 3–17.
Stapel, M., Zheng, Z., Pinkwart, N., 2016. An ensemble method to predict student performance in an online math learning environment. In: Proceedings of the 9th International Conference on Educational Data Mining, International Educational Data Mining Society, pp. 231–238.
Stewart, T.C., West, R.L., 2005. Python ACT-R: a new implementation and a new syntax. In: 12th Annual ACT-R Workshop.
Streeter, M., 2015. Mixture modeling of individual learning curves. In: International Conference on Educational Data Mining.
The International Educational Data Mining Society, 2016. Organizes the EDM conference series and publishes the journal of educational data mining. http://www.educationaldatamining.org/.

Tomkins, S., Ramesh, A., Getoor, L., 2016. Predicting post-test performance from student behavior: a high school MOOC case study. In: Proceedings of the 9th International Conference on Educational Data Mining, International Educational Data Mining Society, pp. 239–246.

Vanlehn, K., Lynch, C., Schulze, K., Shapiro, J.A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., Wintersgill, M., 2005. The Andes physics tutoring system: lessons learned. Int. J. Artif. Intell. Educ. 15 (3), 147–204.

Wang, Y., Heffernan, N.T., Heffernan, C., 2015. Towards better affect detectors: effect of missing skills, class features and common wrong answers. In: Proceedings of the 5th International Conference on Learning Analytics and Knowledge. ACM, New York, NY, 31–35.

White, L.W., 2013. DITA for Print: A DITA Open Toolkit Workbook. XML Press, Fort Collins, CO.

Whitehill, J., Williams, J.J., Lopez, G., Coleman, C.A., Reich, J., 2015. Beyond prediction: first steps toward automatic intervention in MOOC student stopout. Available at SSRN 2611750.