Article ID: 1000-5641(2014)06-0126-15

Ontology construction and mathematical modeling for the LAS engines

XING Wan-li[1], WU Yong-he[2], MA Xiao-ling[3]

(1. *School of Information Science & Learning Technology, University of Missouri-Columba, MO 65201, USA*; 2. *School of Open Education and Learning & Shanghai Engineering Research Center of Digital Education Equipment, East China Normal University, Shanghai 200062, China*; 3. *Department of Information Science, East China Normal University, Shanghai 200062, China*)

Abstract: Learning analytics system (LAS) has the potential to pull together diverse resources and services to leverage the best practices for education. As the central component of this system, current LAS engines have been limited in function and vaguely defined as well as poor scalability and extensibility to other contexts and institutions. This paper first proposed engine ontology, role, source, time and control, to describe and distinct four engine functions: Prediction, Reflection, Recommendation, and Adaptation, in order to establish a common language and practice for LA engines, and in turn improve interoperability between different LA applications. Based on those ontological engines, this study further designed a mechanism of LAS engines and applied mathematical modeling to explain its decomposition and recombination techniques. This LAS engines is expected to power an open and integrated LAS that is capable of scaling up and extensible to any context.

Key words: learning analytics; learning analytics system; learning analytics system engines; mathematical modeling

CLC number: G436　**Document code:** A

DOI: 10.3969/j.issn.1000-5641.2014.06.017

学习分析系统引擎的本体建构与数学模型

邢万里[1]，吴永和[2]，马晓玲[3]

(1. 密苏里大学 信息科学与学习技术系，密苏里洲 65201，美国；
2. 华东师范大学 开放教育教育学院暨上海数字化教育装备工程技术研究中心，上海 200062；
3. 华东师范大学 信息学系，上海 200062)

基金项目：国家社科基金重点项目(11AXW001)；上海市浦江人才计划项目(14PJC034)；上海市科委“科技创新行动计划”高新技术领域项目(145111093000)

第一作者：邢万里，男，博士生，研究方向为教育技术. E-mail: wxdg5@mssouri.edu.

第二作者：吴永和，男，博士，高级工程师，硕士生导师，研究方向为教育信息化系统及标准、学习技术系统架构、教育新媒体、学习分析等. E-mail：yhwu@dec.ecnu.edu.cn.

通信作者：马晓玲，女，讲师，研究方向为信息系统及分析．E-mail: xiaolingm@gmail.com.

摘要: 学习分析系统 (LAS) 具有把不同的资源和服务整合来为教育提供最好的实践. 学习分析引擎作为这一系统的核心组成部分, 目前在功能上的限制, 定义上的模糊, 以及可扩展性差使其不能扩展到其他内容和机构. 本文首先提出了引擎本体 (角色、来源、时间和控制) 来描述和独特的四种引擎功能: 预测、反思、建议和适应, 以建立一个共同的语言和实践学习分析引擎, 从而改善不同的LA之间的互操作性应用. 基于这些本体论的引擎, 本研究进一步设计LAS引擎的机制和应用数学模型来解释其分解和重组技术. 本文所阐述的学习分析系统引擎, 预计可用于开发一个开放和集成的, 能够扩大规模和扩展到任何环境中的LAS.

关键词: 学习分析;　学习分析系统;　学习分析系统引擎;　数学建模

0 Introduction

The amount of educational data is skyrocketing due to increased learning occurring over the Internet. Recently, educational institutions are embarking on explorationing the science of these large data sets, with the aim of improving educational experiences. As a result, a new promising field of research, leavning analytics (LA), has emerged. LA was officially defined during the 1st International Conference on Learning Analytics and Knowledge as “the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs.”[1] The NMC Horizon Report 2013 listed LA in the two to three years’ time-to-adoption window in anticipation of the wide spread adoption of this technology[2].

LA affects every tier of education, including Wed-based System (WBS)[3-5], Learning Management System (LMS)[6-8], Social Learning System (SLS) or Wed 2.0[9-11], and Live Teaching System (LTS)[12-15], Now it trends to build an integrated and open LAS other than solely data source[16-18], limited engine functionalities[19-22], and vague functionalities[22-26].

In order to apply LA at a broader and deeper scale, an open and integrated LAS with completed engine functions is needed to incorporate various data sources (LMS, Web 2.0, physical world-data etc.) to leverage the best practices for education. Verbert proposed a concept to turn the abundance of learning resources into an asset and then mined it for recommendations of resources, activities, or people[27]. However, it stayed at the conceptual level and no specific architecture to construct such a LAS was attempted. Bramucci and Gaston demonstrated that the Sherpa system, which can mine data from various educational sources to deliver multimodal (email, voice, text-to-speech, Facebook announcements), personalized services such as recommending course for students when their preferred courses were full, targeting at-risk student for academic interventions, and tailoring information about campus events to individual interests[28]. Even though it was close to being a LAS due to various engine functionalities, this design was compromised because, similar to many other applications, it did not show the extensibility to other contexts or institutions. Until now, the most extensive system was proposed by Siemens et al.[16], which was an integrated and modularized LAS open for process, algorithms and technologies to apply into different contexts. The advantage of this open platform was the capability to offer an integrated, scalable, and expandable technology that considered whole sources of educational data to inform educational decisions. In this work, Siemens et al. identified that the analytics engine was the central component in the LAS. Since it was a conceptual paper, they did not get into a detailed mechanism for this analytics engine.

This paper focusses on LAS engines' ontology and mathematical modeling. The remainder of this paper was organized as follows. The next section illustrated the basic procedure for the LAS engines making up of four distinctive engines and their functionalities from role, source, time and control perspectives. To some extent, these four views [role, source, time and control] could be considered as ontology to describe any LA engine function. This common language aims to increase interoperability and communication between applications and services. Next, mathematical methods were used to model the functional mechanism for the LAS engines. Finally, the conclusion and future work of this research is discussed. It is hoped this LAS engines could contribute to the development of an open and integrated LAS, and in turn to the entire educational paradigm change and reform to a data driven culture.

1 LAS architecture and engines ontology

1.1 Data pre-processing

Different educational systems vary significantly in their data structures. It would be rather difficult, if not impossible, to merge all the data from different educational systems. Therefore, this study explored an alternative way to integrate resources distributed in diverse educational environments rather than from the ground data merging perspective. However, since LAS system is after all a data processing system, it would be convenient if all the data were stored in the same format. For the purpose of this paper, we simply provided data ontology as [user, temporal, place, operation and content] to provide a sense of the data format. Users could be a single user, or a group of people etc; temporal could be a point of time or duration; place could be in a single educational system or across systems; operation could be reply, remove, update, read, email etc; content could be a web page, a discussion board, or a paper questionnaire in physical world. Based on our preliminary analysis, Contextualized Attention Metadata (CAM) which centers on an event, User Interaction Context Model (UICO) which focuses more on the tasks that users carry out while interacting with the resources, Learning Object Metadata (LOM) and NSDL Para data are object-centric, every data format could be mapped back to this data structure. To illustrate the scalability and functionality of this ontology, for example, several entities could be involved depending on the operation, i.e. when replying a post, there is a "sender" in the user, at least one person with the user "receiver" and content with the "post". It also stores the temporal factor when the operation happens and the place could be Facebook, Wiki or Blackboard etc. Thus, no matter which educational system a piece of data comes from, it can be processed through Code Base (Fig. 1) to this data ontology and stores in the Data Repository.

1.2 Data analysis and engine ontology construction

The unified data stream in each education system can be automatically decomposed to three semantic data domains: User Domain, Pedagogical Domain and Resource Domain (Fig. 1). A key rational for this decomposition method is that any type of education mode (online, live or hybrid) is made of these three elements, user, pedagogy and resource. Each domain is a composite of a number of relevant data pieces. These data pieces can be measured on different scales such as frequency of the operation or the time spent in one system. These domains work as the basis for further functional processing.

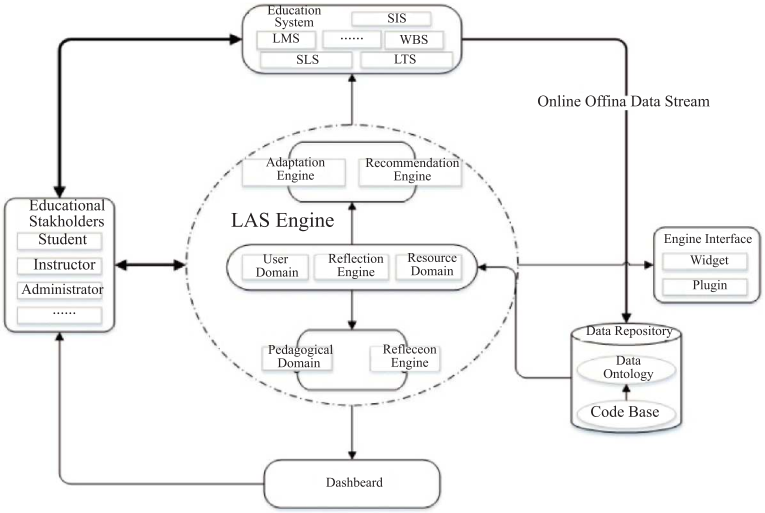

Fig. 1 LAS engines architecture

Domain data then is individually input into LAS engines making up of four functions: Prediction, Reflection, Adaptation and Recommendation (Fig. 1). Due to the vague description and definition of the engine functions in previous studies, this work describes these engines from four perspectives: role, source, time and control, to identify their difference and generate ontology.

Prediction focuses only on the role - not on the source dimension. In this context, role means students (extensible to instructors or other educational stakeholders in the future). This engine functions as making predictions about students' future academic outcomes. For example, the system can detect whether a student is at-risk or potential risk or their success etc. The prediction expresses itself in time In terms of control, this engine could function automatically to make predictions, or instructors could manipulate the domains or systems to personalize predictions for specific context. For instance, if the learning mainly happens in a face-to-face classroom context, then when considering the various education systems as a whole, teachers could adjust the Web 2.0 environments to share in the predictive function. Or if learning is heavily relies on resources usage, then the pedagogical domain could be tuned down for its effect in prediction.

Reflection considers source dimensions solely. Source here has two aspects: first, basic statistics of educational systems source usage data with no further processing. The system gathers the data and presents to the users such data as times of documents downloaded, or the percentage of correct answers for a test; second, advanced algorithms are applied to process the basic usage data to make implications such as students' mood, or learning disposition. From the time perspective, reflection has real time characteristics. Information would be directly delivered to the users in Dashboard or the corresponding educational system (i.e. Facebook message). From the control point of view, different users are able to access different information based on their access rights. For example, students might be granted access to basic source usage data, their social network, dispositions, or the average metrics for the class etc.; instructors, by contrast, can obtain information on each student's performance, a group's performance or the class performance as a whole; administrators might be able to look at detailed information across different courses or departments etc.

Recommendation works both for the role and source dimensions. In this situation, role could mean students and instructors because the recommendation engine is able to recommend resources for both instructors and students. From the source perspective, they could be instructional actions and learning materials and paths, or a student to pair up. When it comes to time dimension, all the resources recommended might not be incorporated or accepted by the users immediately. From the control point of view, it is similar to prediction because recommendation can be made automatically by users. To illustrate, based on the algorithms, the recommendation engine can suggest a book or learning path for a student. Meanwhile, an instructor can also advise a book or a learning path to a particular student based on the student' learning style; and a student can recommend resources to another student. However, a student might take those recommendations or end up not using them at all.

Adaptation acts on role and source dimensions in that it modifies the system (interface or behavior), and in turn personalizes users' experiences. Role in this engine context means students (Adaptation may have the potential for instructors or other users in the future, too). For the resources view, the adaptation engine is able to changie a learning path in the learning analytics system for a student; instructors can modify an education system materials or sources etc. to provide intervention for the student. In terms of the time perspective, adaptation function is real time. When it comes to control, the system can automatically modify the system itself to best meet students' needs; the instructor can also change the system to access a particular student or provide interventions directly to the student.

Adaptation shares a huge commonness with recommendation engines, which is perhaps the major reason why most of the LAS or applications do not differentiate these two functions. These functions have two major differences: First, adaptation is focusing on what the next step is either for the user or for a pedagogical method or for resources while recommendation is what is useful for the user or pedagogical method or for resources etc. Second, adaptation is usually in time while recommendation involves more freedom because a student may ignore the LAS recommendations or lag behind for action of the recommendation. Several papers mentioned the intervention and personalization engines, but failed to explain the functionality of these engines clearly. Based on the above discussion, we argued that the capability of four distinctive engines (prediction, reflection, recommendation and adaption)is able to power an integrated LAS with no further need to incorporate intervention or personalization engines. In addition, these descriptions [role, source, time, control] for the four engines could work as an ontology for LA functions. Researchers and developers could place any new application under this ontology so that they can clearly understand where they are. Moreover, this ontology could also provide a common language and practice for the LA applications and development, which has the benefitof increasing communication and interoperability between the new tools and services.

On the other hand, since learning analytics aims to have a broader and deeper scale of comparing and leveraging tools utilizing datasets from diverse sources, it is imperative to develop a flexible infrastructure that can support the development of tools and services. An infrastructure like this requires basic technical agreement on common standards and protocols. As long as the applications complying with these rules, these tools and services can be embedded as small application components such as widgets or plugins into this LAS engines (Fig. 1). It is hoped this engine functional ontology could initiate and contribute to the common language, practice and rule construction in LA field.

1.3 Data post-processing

LAS Engines has one input from the Data Repository but has four outlets: Engine Interface, Educational System, Educational Stakeholders and Dashboard (Fig. 1). To illustrate, Engine Interface is mainly designed for third party widgets and plugins; the interaction between the LAS engines and the Educational System has two types: One is automatic change based on the engine functions. For example, adaptation is able to adjust the education system to fit for a student's personal needs. The second is based on the information obtained from the engines to change the educational system. For example, instructors relying on the recommendation from the engine could change the instructional design for a course; LAS engines has a two way communication with Educational Stakeholders. From the engine ontology control perspective, users can participate in engine's data processing e.g. instructors adjust the Web 2.0 Educational System share in predictive function if learning mainly took place in traditional classroom. On the other hand, LAS engines can also provide information for users e.g. offer suggestions by recommendation function.

The fourth outlet is Dashboard to provide visual feedback to stakeholders. On some level, Dashboard can also be considered as an engine because it depends on visual data analytics techniques. According to Johnson et al. visual data analytics merges advanced computational methods with sophisticated graphic engines to expand the ability of users to see patterns and structures in complex visual presentations[28]. Furthermore, Bienkowski et al. stated that visual data analytics was able to contribute to expose patterns, trends, exceptions and more in dealing with large heterogeneous and dynamic datasets collected form complex systems[25]. LAS as a complex system can benefit its users using visual data analytics to display various results on Dashboard. For example, a common practice for prediction function is to employ different colors to represent students risk levels e.g. red indicates at risk while green means success. Books or other resource suggestions from the recommendation can be displayed at its home Educational System and can also be accessed from the dashboard. Basic usage or advanced learning disposition information gaining from reflection engine can also be shown on the Dashboard.

2 Mathematical modeling

The central component in the LAS is the set of LAS Engines, which is an integrated processor for identifying and processing data based on various analysis modules. In the previous section, we discussed the distinctive modules that made up the LAS Engines as well as their functions respectively. Now the biggest challenge turns into what mechanism can coordinate those various LAS Engines to function as a whole so that it can incorporate data from different Educational Systems (e.g., LMS, Web 2.0, physical world data etc.) and leverage best educational practices out of them. At the same time, this mechanism of LAS Engines also has extensibility and scalability to other contexts and institutions.

In response to this challenge, decomposition-and-recombination method was designed to deal with these heterogeneous data sets as well as overcome the limitation to various contexts and settings for the LAS Engines. Mapping back to the previous Data Analysis section, decomposition procedure means to deconstruct the educational process into three semantic domains which lays the ground work for recombination. The latter attempts to blend multiple domains and Educational Systems to effectively express and manage complex and diverse patterns of the educational process. Stacking is one of the blending methods, in which a collection of base models is given to a second-level modeling algorithm, also named a meta-model. The second level algorithm is trained to combine the input meta-models into an optimal final set of outputs. In our system, we first build meta-models for each domain in the decomposition process. Then using meta-model as an input, a generalization strategy as a recombinationmethod is used to combine those meta-models to a best-fit output. Typically, optimization can be realized by using algorithms such as Expectation Maximization (EM) algorithm.

This methodology is able to extend and scale the LAS Engines to any context and institution because it enables the selection of variables from an entire collection of data space and blend those variables appropriately. To illustrate, every educational system has a sub-space in LAS Engines and every domain in that system has its own space, too. LAS Engines in all represents the collected data space. Space is the place where the data states are formed. Interactions and relationships between different variables are built and implemented by algorithms to realize functions. Then for each function, the LAS Engines can blend the index between the three domains in an educational system sub-space and then combine the indicators from different educational systems based on context or situation appropriately. In addition, since the meta-model is built in each domain by the system, this methodology avoids combining the different data structures from various educational systems by simply mapping data tothe proposed ontology.

2.1 Data preprocessing and input

According to the Fig. 1, the LAS first registers all the data in one representation (data ontology) instead of combining them. Use $\boldsymbol{o}$ to notate a single data point, then each data point should have five dimensions–User, temporal, place, operation, and content. Therefore data source n (with aforementioned five dimensions) of each system in one domain can be expressed as $\boldsymbol{o}_n^{ij} = o_{1n}^{ij}, o_{2n}^{ij}, o_{3n}^{ij}, o_{4n}^{ij}, o_{5n}^{ij} (n = 1, 2, \cdots, N)$, where i denotes different systems, j denotes different domains and N is the total observations of data.

2.2 Decomposition-Meta-Modeling

In order to build meta-models for each domain and to account for different functions, data fusion model is applied.

In terms of the user domain: information such user knowledge, user behavior, user motivation, user experience, and user satisfaction is tracked. When it comes tospecific data points, different education systems have different data sets to reflect those user's attributes. Using Web-based intelligent system as an example, it may need to collect data sets such as correctness of responses, time spent before making an attempt, repetitions of wrong answers, and errors made. However, for a Web 2.0 such as a Wiki or Google Doc, the tracking events might be how many words one puts in, how much time one spends on the system, or how many edits one makes. Therefore, each educational system has its own separate space, rather than assembling them into one ensemble. Similarly, in terms of event tracking for the pedagogical domain, tracking and collected data consist of the learning component, instructional principle, and curriculum. For the resource domain, information about responses and actions on learning objects over time or how the learning objects are used is put together.

Each domain has a different focus on the aspects of learning and teaching. User Domain focuses on users or subjects' attributes; Pedagogical Domain focuses on methods and process, and Resources Domain centers on content role in education. Specifically, for prediction engine which focuses on user's success or failure in one course or a program, User Domain can predict the student's success based on his or her behavior, experience, knowledge, motivation etc.; Pedagogical domain can predict his or her success relying on the learning or teaching procedure or method one takes; Resource domain can predict his or her success depending on the materials one uses and how much he or she uses those materials. Similarly, based on recommendation engine results, a student knows what is most beneficial for his or her learning. For instance, participating in an online discussion, reading a website or to spend less time on Facebook. Pedagogical Domain consists of recommendations such as what instructional design or method a student or teacher should use.InResources Domain, the system might suggest a book for a student to read or another student to study with. Therefore, every domain is granted its own space in an educational system. As a result, different algorithms or models are built for different domains to perform different functions.

Data overlap can exist between domain events. The following examples will illustrate this point. In order to track user behavior in user domain, the correctness of responses data point or event is tracked; in order to track the instructional practice effects in promoting learning in pedagogy domain, student correctness of answers is tracked; in order to track whether one particular website is effective in teaching and learning in resource domain, one's responses or correctness is kept. Therefore the number of the overall combination of the tracked events in different domains in certain education system is larger than the events take place in that system due to the overlap between domains. Considering the possible overlap between domain events, the statistical processes such as the Cross tabulation (or crosstabs for short), endogeneity tests and the Canonical Correlation Analysis (CCA) can be conducted before the model development. Crosstabs, also known as contingency table, provides a basic picture of the interrelation between two variables and can help find interactions between them. In a statistical model, endogeneity means the situation when there is a correlation between the parameter or variable and the error term. This implies that the regression coefficient in an ordinary least squares (OLS) regression is biased, however if the correlation is not contemporaneous, then it may still be consistent. The commonly used endogeneity tests include instrumental variable regression, Heckman selection correction and Hausman's test. CCA is a multivariate statistical model which facilitates the study of interrelationships among multiple dependent variables and multiple independent variables. It is one of the dimension reduction techniques. CCA is the second-most general application of the General Linear Model (GLM) following structural equation modeling. The underlying logic of CCA involves the derivation of a linear combination of variables from each of the two sets of variables so that correlation between the two sets is maximized.

Assuming there are N data observations in domain j of system i, the dataset, $\boldsymbol{D}^i j$, from one domain of each system can be expressed as follows.

$$\boldsymbol{D}^i j = \boldsymbol{o}_1^{ij}, \boldsymbol{o}_2^{ij}, \cdots, \boldsymbol{o}_n^{ij}, \cdots, \boldsymbol{o}_N^{ij}$$

where, $i \in [0, I]$ represents different education systems, with a total of I systems; $j \in [0, J]$ represents different domains in an education system, with a total of J domains.

Assuming that $\boldsymbol{X}^{ij} = \boldsymbol{x}_n^{ij}$ is a vector of observed explanatory variables and $\boldsymbol{Y}^{ij} = \boldsymbol{y}_n^{ij}$ is a vector of observed dependent variables or indicators ($n = 1, 2 \cdots, N$), then both the independent variables $\boldsymbol{X}^{ij}$ and dependent variables $\boldsymbol{Y}^{ij}$ are extracted from the dataset $\boldsymbol{D}^{ij}$:

$$\boldsymbol{X}^{ij} \in \boldsymbol{D}^{ij}, \quad \boldsymbol{Y}^{ij} \in \boldsymbol{D}^{ij}.$$

Moreover, different engines require different algorithms to function such as prediction, adaptation, recommendation and reflection. For illustration and abstract purpose, all these functions can then be presented as: $(\widehat{y}_{(m),i,j}, \widehat{p}_{(m)})$ where $\widehat{p}_{(m)} = p(\boldsymbol{y}^{ij} = \widehat{y}_{(m),i,j} | \boldsymbol{x}^{ij})$, $\widehat{y}_{(m),i,j}$ denotes the possibility of education system i at particular function (prediction, reflection, recommendation, adaptation) m in domain j, $\boldsymbol{x}^{ij}$ denotes direct data of domain j in education system i.

In the LAS, engine functions are also initially defined but not perhaps limited to those functions with the development in LA field. For prediction engine, the main purpose would be to indicate a student or user at risk, potential risk or success or additional categories either in a program or a particular course. In this context, $\boldsymbol{x}_n^{ij} \longrightarrow (\boldsymbol{y}_n^{ij}, p \in [0,1]^c)$, $\forall n$, where $\boldsymbol{y}_n^{ij}$ are the risk indicators or categories, p is the associated probability, and c is the number of risk indicators/categories. The indicator $\boldsymbol{y}_n^{ij} = F[\boldsymbol{x}_1^{ij}, \boldsymbol{x}_2^{ij}, L, \boldsymbol{x}_n^{ij}, L, \boldsymbol{x}_N^{ij}] \longrightarrow p$. The results of F function, $\boldsymbol{y}_n^{ij}$, are the numerical numbers of indicators. Based on predefined category, for example, if p is below 0.4, then the user or student can be reflected as at risk. If the values follows in the range of (0.4, 0.6], then the user or student is at risk. If the value is in the range of (0.6, 1.0], then the student is a success. All of these are simply based on one domain j in one education system i.

Adaption engine and recommendation, on some level are similar but they are different in the sense of results as stated above. Nevertheless, in mathematical modeling, they are similar in format. Using adaptation as an example: $\boldsymbol{x}_n^{ij} \longrightarrow (\boldsymbol{y}_n^{ij},\, p \in [0,1]^c)$, $\forall n$, where $\boldsymbol{y}_n^{ij}$ indicates the adapted steps either in user action, or pedagogical principle, or resource, p is a value indicating the associated probability where $\boldsymbol{y}_n^{ij} = G[\boldsymbol{x}_n^{ij}] = G[\boldsymbol{x}_1^{ij}, \boldsymbol{x}_2^{ij}, L, \boldsymbol{x}_n^{ij}, L, \boldsymbol{x}_N^{ij}] \longrightarrow p$. Similar to adaptation, recommendation would replace the next step with another resource. The function algorithm would be different due to the reason discussed above.

Reflection is different from others. Typically the research in reflection or reflective learning present the factual information to the education system users. Therefore, the system can simply offer the direct data from $D^{ij} = \boldsymbol{o}_1^{ij}, \boldsymbol{o}_2^{ij}, L, \boldsymbol{o}_n^{ij}, L, \boldsymbol{o}_N^{ij}$ to the users. However, social learning research proposed to provide users with information such as mood, or learning disposition calculation based on basic factual information which involves various algorithms.

2.3 Recombination

A significant aspect of this LAS Engines mechanism is the combination process: how to recompose different domains in different systems to lead to one function (prediction, reflection, recommendation, and adaptation). We propose a strategy that follows two lines: one is two dimensions of combination; the other is automatic combining process and intervened combining process.

For the two levels of combination, it means the combinations take place in two spaces for one function. For prediction function, the domain indicators are integrated in one model in the first level and then they are combined, with weights, across different systems. For others engines, the first level combination happens in individual spaces of different domains in one system; the second level is to rank it as a whole or just give the whole results (Reflection engine). To illustrate, for the system i and domain j, the indicators matrix can be

Systems	Domain Indicators						
	$\cdots$	$\cdots$					
1	y_{11}	y_{12}	y_{13}	$\cdots$	y_{1j}	$\cdots$	y_{1J}
2	y_{21}	y_{22}	y_{23}	$\cdots$	y_{2j}	$\cdots$	y_{2J}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
i	y_{i1}	y_{i2}	y_{i3}	$\cdots$	y_{ij}	$\cdots$	y_{iJ}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
I	y_{I1}	y_{I2}	y_{I3}	$\cdots$	y_{Ij}	$\cdots$	y_{IJ}

For different engines, different models can be developed for y_{ij} as follows:

For Prediction function, $y_{ij}^f = F(\boldsymbol{x}_1^{ij}, \boldsymbol{x}_2^{ij}, L, \boldsymbol{x}_n^{ij}, L, \boldsymbol{x}_N^{ij})$;

For Adaptation function, $y_{ij}^g = G(\boldsymbol{x}_1^{ij}, \boldsymbol{x}_2^{ij}, L, \boldsymbol{x}_n^{ij}, L, \boldsymbol{x}_N^{ij})$;

For Recommendation function, $y_{ij}^h = H(\boldsymbol{x}_1^{ij}, \boldsymbol{x}_2^{ij}, L, \boldsymbol{x}_n^{ij}, L, \boldsymbol{x}_N^{ij})$;

For Reflection function, $y_{ij}^{k}=K(\boldsymbol{x}_{1}^{ij},\boldsymbol{x}_{2}^{ij},L,\boldsymbol{x}_{n}^{ij},L,\boldsymbol{x}_{N}^{ij})$.

These models are developed to link the probability of an event occurrence with direct data collected by LAS Engines. Typically, to find the best-fit model, the methodological approaches, such as logit and conditional logit models and Bayesian modeling approach, can be used in the LAS Engines. For example, the conditional logit models can be developed based on case-controlled data in which the learning conditions prior to event (e.g., failure/risk) are used as cases while the matched conditions are used as controls. This proposed matched risk-non-risk analysis is expected to evaluate the impacts of learning conditions on risk likelihood while controlling for the effects of other confounding variables, such as pedagogical methods or resources' characteristics. The logit models can be used to link risk likelihood of student learning with real-time user data, pedagogical data and resource features. The difference between the abovementioned models is that the logit models consider the pedagogical data and resource features as the important variables simultaneously. In addition to the logit models and their variations, other traditional statistical models such as the multivariate probit model, Bayesian discriminate analysis and Fisher discriminate analysis can be adopted as well.Moreover, the Bayesian logit model and Bayesian conditional logit models can also be used for risk prediction. If the learning variables have varying effects on risk/failure across observations (heterogeneous effects of learning conditions), the Bayesian random-parameters logit model and Bayesian hierarchical model with cross-level interactions can be utilized to link the varying parameters of the learning variables with higher level factors.

To illustrate, the example meta-model can be developed based on the following denotations. For each domain indicator y_{ij}, $\boldsymbol{x}_{(1),n}=[x_{(1),1,n},x_{(1),2,n},\cdots,x_{(1),K(1),n}]$ is a $1\times K(1)$ vector of explanatory variables that can be denoted by $x_{(1),k,n}(k=1,2,3,\cdots,K(1))$. The vector $\boldsymbol{\beta}_{(1),n}=[\beta_{(1),1,n},\beta_{(1),2,n},\cdots,\beta_{(1),K(1),n}]^{T}$ is the parameter vector for the explanatory variable vector $x_{(1),n}$. These parameters are assumed to be varied across observations. The $x_{(2),n}=[x_{(2),1,n},x_{(2),2,n},\cdots,x_{(2),K(2),n}]$ is a $1\times K(2)$ vector of another explanatory variables that can be denoted as $x_{(2),k,n}(k=1,2,3,\cdots,K(2))$. The vector $\boldsymbol{\beta}_{(2)}=[\beta_{(2),1},\beta_{(2),2},\cdots,\beta_{(2),K(2)}]^{T}$ is the parameter vector for the explanatory variable vector $x_{(2),n}$. Each parameter in the $\boldsymbol{\beta}_{(2)}$ is assumed to be fixed across observations.The constant β 0,n can also be assumed to be varied or fixed across observations. More variables and parameter vectors can be defined in the similar way, either varied or fixed across observations. With these specifications, the examplemeta-model can be written as:

$$y_{ij}=\beta_{0,n}+\boldsymbol{\beta}_{(1),n}\boldsymbol{x}_{(1),n}+\boldsymbol{\beta}_{(2)}\boldsymbol{x}_{(2),n}+\cdots+\varepsilon_{m}$$

where the term ε_{m} represents the unobserved components. The parameter $\boldsymbol{\beta}_{(1),n}$ is assumed to be normal distributions:

$$\boldsymbol{\beta}_{(1),n}\sim N\Big(\mu_{(1)},\sum_{(1)}\Big)\text{ And }\boldsymbol{\beta}_{(2)}\sim \text{Normal}\Big(\overline{\mu}_{(2)},\overline{\sum}_{(2)}\Big)\text{ with}$$

$$\mu_{(2)} = \begin{bmatrix} \mu_{(1),1} \\ \mu_{(1),2} \\ M \\ \mu_{(1),K(1)} \end{bmatrix}, \quad \sum_{(1)} = \begin{bmatrix} \sum_{(1),1} & 0 & L & 0 & 0 \\ 0 & \sum_{(1),2} & L & 0 & 0 \\ M & M & O & M & M \\ 0 & 0 & L & \sum_{(1),K(1)-1} & 0 \\ 0 & 0 & L & 0 & \sum_{(1),K(1)} \end{bmatrix}$$

The likelihood function of the meta-model can be given as follows:

$$f(Y|\Theta) = \prod_{n=1}^{N} f(y_n|\beta_{0,n}, \boldsymbol{\beta}_{(1),n}, \beta_{(2)}, L)$$

where the vector of all parameters Θ includes the random parameter vector $\boldsymbol{\beta}_{(1)}$, the fixed parameter vector $\boldsymbol{\beta}_{(2)}$, the random constant $\boldsymbol{\beta}_0$, the random parameters mean vector $\mu_{(1)}$, the random parameters variance vector $\sum_{(1)}$, the random constant mean μ_0, and the random constant variance $\sum_0$ and so on. Thus,

$$\Theta = \left[\ \boldsymbol{\beta}_0, \boldsymbol{\beta}_{(1)}, \boldsymbol{\beta}_{(2)}, \mu_0, \textstyle\sum_0, \boldsymbol{\mu}_{(1)}, \sum_{(1)}, L \ \right].$$

Since that is the only one system expression, the engine cannot function well without considering other systems. Therefore, the process of recombination is conducted for different engines.

2.3.1 For Prediction Engine

2.3.1.1 Recombination at domain level (Level I)

For system i prediction result can be combined to z_j,

$$\begin{aligned} z_i &= w_{i1}F(\boldsymbol{x}_{11}) + w_{i2}F(\boldsymbol{x}_{i2}) + w_{i3}F(\boldsymbol{x}_i3) + \cdots + w_{ij}F(\boldsymbol{x}_{ij}) \cdots + w_{iJ}F(\boldsymbol{x}_{iJ}) \\ &= w_{i1}y_{i1} + w_{i2}y_{i2} + w_{i3}y_{i3} +, + w_{ij}y_{ij} \cdots + w_{iJ}y_{ij} = \sum_{j=1}^{J} w_{ij}y_{ij}. \end{aligned}$$

It represents different domain has its own prediction to whether a user or student is going to succeed in this course or program (based on the data using) or at risk.

2.3.1.2 Recombination at system level (Level II)

The other combination space takes place at system levels: Since there are i systems, to aggregate the best result for prediction is

$$Z = w_1'z_1 + w_2'z_2 + w_3'z_3 + \ldots + w_i'z_i \ldots + w_I'z_I = \sum_{i=1}^{I} w_i'z_1$$

where Z denotes whether the student is at risk/success or other similar meanings.

2.3.2 For Adaptation and Recommendation engine

The recombination method for adaptation and recommendation is similar. Using adaptation as an example: every engine algorithm processes the data in every domain would produce an adaptation result. Thus, for education system i, it produces a set of adaptation step as

$$\boldsymbol{y}_{ij}^{g} = G(\boldsymbol{x}_1^{ij}, \boldsymbol{x}_2^{ij}, \cdots, \boldsymbol{x}_n^{ij}, \cdots \boldsymbol{x}_N^{ij}).$$

In education system i, $\boldsymbol{y}_{i1}^{g}$ represents what user should do next, or what system adapts for the user, $\boldsymbol{y}_{i2}^{g}$ represents what pedagogical method should use next or being adapted, $\boldsymbol{y}_{i3}^{g}$ reflects what content should be adapted, and $\boldsymbol{y}_{ij}^{g}$ states what that particular domain is adapted.

Next, the system will run another algorithm to on the set of all the adaptation advices

$$y_{11}^{g}, y_{12}^{g}, y_{13}^{g}, \cdots, y_{1J}^{g}, y_{21}^{g}, y_{22}^{g}, y_{23}^{g}, \cdots, y_{2J}^{g}, \cdots, y_{i1}^{g}, y_{i2}^{g}, y_{i2}^{g}, \cdots, y_{ij}^{g}, \cdots, y_{I1}^{g}, y_{I2}^{g}, y_{I3}^{g}, \cdots, y_{IJ}^{g}$$

As a result, the system will understand what the most important step for adaptation is.

Unlike y_{ij}^{f} (a category to predict student being at risk or potential risk or success), $\boldsymbol{y}_{ij}^{g}$ is a sorted list by significance. Therefore, the adaption for the user action is the most emergent, and then adaptation for a pedagogical method is the immediately second most important one etc. Similarly for recommendation:

$$y_{ij}^{f} = H(\boldsymbol{x}_1^{ij}, \boldsymbol{x}_2^{ij}, \cdots, \boldsymbol{x}_n^{ij}, \cdots, \boldsymbol{x}_N^{ij}),$$

it will also generate a sorted list of recommended sources by importance.

2.3.3 For reflection engine

In terms of the reflection engine: the reflection is the fact or other inference knowledge based on the analysis purpose. Assuming that $\boldsymbol{y}_{ij}^{k}$ is a vector of binary observed indicators. If the reflection is the fact or direct presentation of usage information, $k = 1$, the result is $\boldsymbol{x}_{ij}$. For example, it could be how many times the student replies a post in LMS or how many connections he or she has in learning network; otherwise, $k = 0$. Then the results would be the function on the basic usage data $K[\boldsymbol{D}^{ij}]$. It means using advanced algorithms to process the basic usage data to reflect other attributes of learning. For instance, it could be the indicators of students' learning dispositions, mood etc.

$$y_{ij}^{k} = K(\boldsymbol{o}_1^{ij}) \begin{cases} \boldsymbol{x}^{ij}, \boldsymbol{x}^{ij} \in \boldsymbol{D}^{ij}, & k = 1 \\ K[\boldsymbol{D}^{ij}] = K[\boldsymbol{o}_1^{ij}, \boldsymbol{o}_2^{ij}, \cdots, \boldsymbol{o}_n^{ij}, \cdots \boldsymbol{o}_N^{ij}], & k = 0 \end{cases}$$

When the system does the recombination function at either domain level or system level, two ways can be applied. One is automatic weight blending. By selecting an automatic blending method, the blending weights associated with individual meta-model are optimized to provide a best-fit model. If selecting the intervened method, the user is offered the option to adjust the weight of different based functions. For example, when a course is mainly using LMS and Web 2.0 tools, users or students are able to dampen the effect of the other system model effects on the overall functions either prediction, adaptation, recommendation or reflection.

3 Conclusion and future work

Learning is increasingly distributed in various environments or educational systems. LAS has the potential to pull together diverse resources and services to empower educational stakeholders. However, as a central component of LAS, current LA enginesare limited in function are vaguely defined and are hindered bypoor interoperability between different tools. This paper first proposed engine ontology (time, role, source, control) to define and incorporate distinct engine functions, with the hope to establish a common language and practice for LA engine functions or applications, and in turn improve interoperability. Then we applied mathematical modeling to describe the decomposition and recombination mechanism for the LAS Engines to support the integrated LAS. Decomposition works on deconstructing the data from every Educational System into three domains: user domain, pedagogical domain and resource domain. Next, two levels of blending: domain and system recombination strategy with two methods of combination were described. This LAS Engines are able to extend and scale up to any context and institutions.

The architecture we proposed is centered on the LAS Engines working mechanics rather than the system architecture. Future work requires developing a complete LAS infrastructure and investigating its mechanics between different modules. In addition, it also needs to develop a prototype for LAS engines to test if the proposed mechanism works on real world data sets. In addition, it is critical to develop standards and protocols for the learning analytics applications in order to embed various LA tools in this LAS.

Learning analytics is a sphere about deciphering trends and patterns from educational big data. With the development of LAS, it could further advance a personalized and supportive educational experience.

[References]

[1] International Conference on Learning Analytics and Knowledge [EB/OL]. 2011-03-27[2014-10-12]. https://tekri.athabascau.ca/analytics.

[2] New Media Consortium. 2013 Horizon report[R/OL]. [2014-10-12]. http://www.nmc.org.

[3] KOULOCHERI E, M(N)XENOS. Considering formal assessment in learning analytics within a PLE: the HOWZLEARN case [C]//Proceedings of the Third International Conference on Learning Analytics and Knowledge. [S.L.]: ACM, 2013: 28-32.

[4] PARDOS Z A, BAKER R SJD, SAN PEDRO M OCZ, et al. Affective states and state tests: Investigating how affect throughout the school year predicts end of year learning outcomes [C]//Proceedings of the Third International Conference on Learning Analytics and Knowledge. [S.L.]: ACM, 2013: 117-124.

[5] KIZILCEC R F, PIECH C, SCHNEIDER E. Deconstructing disengagement: analyzing learner subpopulations in massive open online courses [C]//Proceedings of the Third International Conference on Learning Analytics and Knowledge ACM, 2013: 170-179.

[6] LONN S, KRUMM A E, WADDINGTON R J, et al. Bridging the gap from knowledge to action: Putting analytics in the hands of academic advisors [C]//Proceedings of the 2nd International Conference on Learning Analytics and Knowledge. ACM, 2012: 184-187.

[7] BOWMAN M, DEBRAY S K, Reasoning about naming systems [J]. ACM Trans Program Lang Syst, 1993, 15(5): 795-825.

[8] GARCÍA-SOLÓRZANO D, MORÁN J A, COBO G, et al. Educational monitoring tool based on faceted browsing and data portraits. In Proceedings of the 2nd International Conference on Learning Analytics and Knowledge ACM, 2012: 170-178.

[9] GUNNARSSON B L, ALTERMAN R. Understanding promotions in a case study of student blogging [C]//Proceedings of the Third International Conference on Learning Analytics and Knowledge. ACM, 2013: 57-65.

[10] SOUTHAVILAY V, YACEF K, REIMANN P, et al. Analysis of collaborative writing processes using revision maps and probabilistic topic models [C]//Proceedings of the Third International Conference on Learning Analytics and Knowledge. ACM, 2013: 38-47.

[11] SHUM S B, FERGUSON R. Social learning analytics [J]. Educational Technology and Society, 2012, 15(3): 3-26.

[12] BECKER K, GHEDINI C, TERRA E. Using kdd to analyze the impact of curriculum revisions in a Brazilian university [C]//Eleventh International Conference on Data Engineering. Proceedings of the SPIE 14th Annual International Conference on Aerospace/Defense, Sensing, Simulation and Controls. Orlando, 2010: 412-419.

[13] LUAN J. Data mining, knowledge management in higher education, potential applications [C]//Workshop Associate of Institutional Research International Conference. Toronto, 2002: 1-18.

[14] BLIKSTEIN P. Multimodal learning analytics [C]//Proceedings of the Third International Conference on Learning Analytics and Knowledge. ACM, 2013: 102-106.

[15] BROOKS C, EPP C D, LOGAN G, et al. The who, what, when, and why of lecture capture [C]//Proceedings of the 1st International Conference on Learning Analytics and Knowledge. ACM, 2011: 86-92.

[16] SIEMENS G, GASEVIC D, HAYTHORNTHWAITE C, et al. Open Learning Analytics: an integrated and modularized platform [R/OL]. 2011[2014-10-12]. http://solaresearch.org.

[17] NIEMANN K, WOLPERS M, et al. Aggregating social and usage datasets for learning analytics: data-oriented challenges [C]//Proceedings of the Third International Conference on Learning Analytics and Knowledge. ACM, 2013: 245-249.

[18] D'AQUIN M, JAY N. Interpreting data mining results with linked data for learning analytics: motivation, case study and directions [C]//Proceedings of the Third International Conference on Learning Analytics and Knowledge. ACM, 2013: 155-164.

[19] ARNOLD K E, PISTLLI M D. Course signals at Purdue: Using learning analytics to increase student success. In Proceedings of the 2nd International Conference on Learning Analytics and Knowledge. ACM, 2012: 267-270.

[20] BARBER R, SHARKEY M. Course correction: using analytics to predict course success [C]//Proceedings of the 2nd International Conference on Learning Analytics and Knowledge. ACM, 2012: 259-262.

[21] ESSA A, AYAD H. Student success system: risk analytics and data visualization using ensembles of predictive models [C]//Proceedings of the 2nd International Conference on Learning Analytics and Knowledge. ACM, 2012: 158-161.

[22] SANTOS J L, GOVEARTS S, VERBERT K, et al. Goal-oriented visualizations of activity tracking: a case study with engineering students [C]//Proceedings of the 2nd International Conference on Learning Analytics and Knowledge. ACM, 2012: 143-152.

[23] U S Department of Education. Transforming American Education: Learning Powered by Technology [R/OL]. 2010[2014-10-12]. http://www.ed.gov/technology/netp-2010.

[24] SIEMENS G, LONG P. Penetrating the fog: Analytics in learning and education [J]. Educause Review, 2011, 46(5), 30-32.

[25] BIENKOWSKI M, FENG M, MEANS B. Enhancing teaching and learning through educational data mining and learning analytics: An issue brief [R]. Washington, DC: SRI International, 2012.

[26] VERBERT K, MANOUSELIS N, DRACHSLER H, et al. Dataset-driven research to support learning and knowledge analytics [J]. Educational Technology and Society, 2012, 15(3): 133-148.

[27] BRAMUCCI R, JIM G. Sherpa: increasing student success with a recommendation engine [C]//Proceedings of the 2nd International Conference on Learning Analytics and Knowledge. ACM, 2012: 82-83.

[28] JOHNSON L, LEVINE A, SMITH R, et al. The 2010 Horizon Report [R]. Austin, TX: [s.n.], 2010.

(责任编辑 王善平)