Designing a Social Machine for Mediated Learning Environments

SRINATH SRINIVASA, Gooru Labs, IIIT Bangalore
PRASAD RAM, Gooru Inc
ROBERT SCHULER, Gooru Inc

The web is seen as a promising platform for designing scalable educational practices across large populations. Much of the efforts in this space use the web primarily as an amplifier over existing models of learning – that are based on the classroom. In this work, we argue for modeling large-scale online learning environments as *social machines*, rather than as large classrooms. We propose a pedagogy model called "mediated learning" where the web acts as a platform to nurture a learning community, by continuously mediating between knowledge need and expertise. Mediated learning has the potential to "invert the learning pyramid" by interfacing the learner with several experts as part of a single learning experience. Supporting mediated learning needs an approach that is data-intensive, as well as be driven by social semantics, making the paradigm of social machines, a natural choice. This document outlines the Gooru model for pedagogy, called mediated learning. The model comprises of two primary components: A user-end "navigator" component that provides a rich interface for users to independently navigate through a learning space; and a back-end "community" component, that performs meaningful mediations between participants in the logical learning space.

Additional Key Words and Phrases: Mediation, Semantics, Learning, Pedagogy, Social Machines

ACM Reference format:
Srinath Srinivasa, Prasad Ram, and Robert Schuler. 2017. Designing a Social Machine for Mediated Learning Environments. 1, 1, Article 1 (June 2017), 15 pages.
DOI: none

1 INTRODUCTION

Education is a fundamental human right. The challenge of making formal education accessible to everyone in the world, has been pursued for ages. With increasing penetration of the web in different corners of the world, it has become a natural platform of choice for designing scalable education models.

However, much of the efforts in this direction have viewed the web mainly as an amplifier to increase the impact of conventional models of learning, based on the *classroom*.

A popular example of this is the MOOC (Massive Online Open Course) [6, 23] model of learning. Even though MOOC environments typically support interactive activities and community interactions, the core element of the model is still that of a classroom scaled over a massive student population. While there are famous commercial success stories of MOOCs, they have also elicited criticism from various quarters. Several teachers saw the massive scale of the student population in MOOCs, as a threat not only to their jobs, but to pedagogic credibility in general [2]. Among the several concerns expressed by faculty members include the predominant model of a MOOC as a "one-size-fits-all vendor-designed blended courses" [15] – in other words, the classroom model on steroids.

The idea of a classroom has its roots in the Prussian industrial concept of the *factory* [16, 35]. The factory model of the classroom can be described as a "pyramid" representing a one-to-many relationship between the teacher and the students, called the "teacher-student ratio" (TSR). The

2017. XXXX-XXXX/2017/6-ART1 $
DOI: none

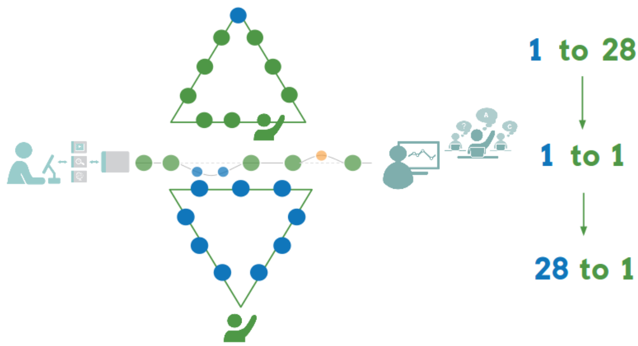

Fig. 1. Inversion of the learning pyramid: Classroom, ITS and Gooru models

classroom is designed to cater to the "average" student, and the pyramid model is geared for efficiency in instruction to the average, and provides time-bound guarantees for covering the syllabus.

However, the classroom cannot guarantee effectiveness of pedagogy itself. As noted by Rose [26], the idea of an "average student" is a myth. With increasing dimensions of concern, the probability of finding an average individual across all dimensions, rapidly decreases. Pedagogy is a multi-dimensional concern that need to address several kinds of competency. A classroom experience designed to cater to the average along all dimensions is likely to cater to none of the individual learners. For this reason, *personalization* is an important design element to increase the quality of outcomes. A characteristic property of personalization is "elasticity" – the system, even though designed for the mythical average student, needs to adapt to the actual individual student to various degrees.

Other models of online learning, like Intelligent Tutoring Systems (ITS) or Adaptive Learning Environments (ALE), address pedagogy from this perspective [1, 14, 24, 30]. These approaches draw from a variety of fields like cognitive psychology, learning theory and artificial intelligence, to create an automated tutor agent that maintains an ongoing *relationship* with the student [5, 7]. The tutor continuously adapts the learning experience based on how the student is performing. The automated nature of the tutor is critical in this approach, as it is impractical for a human teacher to continuously adapt their pedagogy for every learner.

The main challenge in ITS is the complexity of automated tutoring. Pedagogy is replete with several latent, unobservable factors that are critical to both instruction and comprehension. Automated tutoring raises debatable questions as to whether such latent factors can be reliably captured and codified in a computational model. Also, ITS aims to *replace* the teacher with a computational agent, leading to other deeper questions about the importance of the human element in pedagogy. In many classroom environments, the teacher is typically also a source of inspiration, a role model and a source of moral support to students, in addition to being the subject matter expert. The human connection between the teacher and the student is an important element of effective learning.

Given this backdrop, the "Gooru" approach to pedagogy as presented in this paper, aims to go beyond the classroom and adaptive learning models. Rather than placing the classroom or an adaptive agent at the core, the Gooru approach models learning practices in the context of a *social machine*. The core element of this model is a *learning community*, comprising of several human participants, whose interactions are facilitated by algorithms. Scale and personalization is achieved not just by automating the process of pedagogy, but also by *mediating* between learning needs and expertise in the community.

The primary aspiration in this approach is called "inverting the learning pyramid" – that is, change the teacher-student ratio (TSR) from the conventional one-to-many model to a many-to-one model (Figure 1). In other words, help students to interact seamlessly and meaningfully with several experts as part of a single learning experience. This includes not just formal, subject-matter experts and teachers, but also interaction with other peers and community members who may have the required expertise in some dimension required for learning the subject.

The Gooru system comprises of two components. Users interact with a learning agent called the Gooru navigator. The navigator aids the learner by representing the learning experience as a navigation in a logical space, somewhat akin to GPS based navigation agents like the Google navigator. Users can not only navigate on routes suggested by the navigator to achieve mastery over specific competencies, they can also explore the logical space and obtain an overview of the subject area based on how learning resources are organized in this space.

The second component of the Gooru system is the back-end "community" engine. The logical space in which the navigator operates is a shared space used by several participants playing different roles. The community component suggests and facilitates several forms of meaningful interactions between the participants in this space.

A combination of the community and independent learning components makes the Gooru model a *social machine*. The term "social machine" was coined by Berners-lee and Fischetti [3] to denote an ensemble of human and algorithmic components, whose emergent properties depends on both human and algorithmic decisions.

2 RELATED LITERATURE

The importance of learning communities has been recognized and addressed in different forms over several years [11, 20, 28, 31, 32]. Learning communities are driven by various motivations. These range from the need for harnessing the tacit element of learning [19], catalyzing independent learning [33] and as a means of continuous professional development [10].

A significant amount of literature on learning communities are set in professional and community environments, rather than formal educational institutions. However, educational psychologists like Slavin [28, 29] have focused on collective learning strategies in classroom settings with encouraging outcomes.

The somewhat related idea of "learning organizations" that was first mooted by Senge [27], had elicited a lot of interest from senior management and academics in the 1990s. Learning Organizations take a holistic systems thinking approach to organizational learning, focusing on collective outcomes. However, the approach was primarily oriented towards improving management practices implemented almost entirely by humans with little or no central role for algorithms and formal models. Some of the criticisms towards learning organizations cite this element as its limiting factor. Coopey [8] notes that learning organizations require its key players to be completely apolitical in their approach to organizational learning, which is often an unrealistic assumption.

The concept of social machines was not in vogue in the 1990s when learning organizations were a popular idea. In hindsight today, it is clear that any holistic or systemic approach requires seamless orchestration between algorithmic and human factors for achieving predictable outcomes.

Another relevant area of literature is Computer Supported Collaborative Learning (CSCL) [17, 21, 22]. CSCL has its genesis around 1989 and has seen significant amount of interest from researchers in pedagogy as well as from information system designers. A variety of perspectives have been addressed in the CSCL literature, making it an inter-disciplinary field of inquiry.

Fig. 2. Learning maps and navigation

While CSCL addresses *operational* issues of collaboration, the idea of mediated-learning operates at a *structural* level. The objective of mediation is to "facilitate" collaboration, rather than to "drive" collaboration. This removes the need for encoding detailed computational models of pedagogy. The pedagogy is still driven by humans. However, there is still a need for significant algorithmic sophistication in understanding underlying semantics, in order to make meaningful mediation.

A class of learning models that operate at a structural level are the *blended learning* models [12, 13]. The main idea behind blended learning is a structural transformation of the learning environment to combine elements of online learning and offline face-to-face environments. An example of blended learning is the "flipped classroom" model [4, 34]. In this model, conventional lectures are recorded as videos which the students watch online in an asynchronous fashion. The synchronous classroom in turn, hosts collective problem-solving activities rather than lectures. What was conventionally "homework" would now be performed in class, while the conventional classroom lectures would be taken up by students at home.

Mediation-based learning as presented in this paper, is a form of blended learning, focusing on implementing sustaining innovations. The primary motivation is "inversion of the learning pyramid" through web-enabled learning mediations – effectively achieving personalization at scale, without changing the fundamental model of learning and its measurement.

3 THE LEARNING MAP

The Gooru learning experience for any given subject is situated in a logical space called the "learning map" for a given subject. Users interact with the system either through a web browser or a mobile device, with the help of the *Gooru navigator* to explore the space. Figure 2 depicts a screenshot of the learning map as it appears to the user, as well as a navigation route computed by the navigator to help the student complete a learning objective.

Gooru maintains a vast library of learning resources that are created, collected and curated by a large, in-house content management team. Resources in the Gooru library include open educational material available on the web, as well as educational material created by different providers and educators using the Gooru system.

Content in the Gooru library is organized into one or more "subjects" – each of which has its own separate learning space. A subject is presented to the user as a 2-dimensional map in which there are several learning resources. The axes of the learning map corresponds to "domains" and "pedagogic depth" and these axes are labeled as Q and P axes respectively.

A subject is shared by several participants who are learning or contributing to the resources about that subject. The x-axis of this map (called the Q-axis) represents a set of topics (called "domains" in pedagogic terminology) under that subject. The y-axis (called the P axis) represents increasing levels of pedagogic depth for every topic. For instance, a subject called "High School Math" would be organized as a learning space where students can learn mathematical concepts at a high school level. The Q axis would refer to different topics covered in high school math, like Linear Equations, Quadratic Equations, Theorems of Eucledian Geometry, etc. The P axis would be organized as graded levels showing increasing pedagogic depth like: Grade 8, Grade 9, etc. Even the topics forming the Q axis are organized in increasing order of complexity as pertinent to the subject. Hence, moving horizontally along the Q axis at a fixed value of the P axis corresponds to learning topics in a course in the sequence presented by the course.

Every point in this discrete space that has a coordinate is called a *competency*. A competency represents the basic unit of pedagogy. Learning objectives are modeled in terms of competencies that need to be achieved by the student.

Formally, the Learning Map is defined as follows:

$$LM = (C, A, Q, P, \gamma, \delta, Badges) \tag{1}$$

Here, C is a set of *competencies* (also called *concept nodes* or *competency nodes*), that represent the basic unit of learning. Every competency $c \in C$ is uniquely identifiable by a pair of dimensions $(q, p) \in Q \times P$ that represents a "domain" and "pedagogic depth" respectively. The term $\gamma : C \rightarrow Q \times P$ represents the mapping function that assigns a coordinate for every competency in the system. The set Q represents the set of all domains or topics of interest in the learning map, while the set P represents the set of all values for pedagogic depth.

The term A represents a set of "learning activities." These are abstract containers that can host different kinds of activities – be it an offline classroom interaction or an online interactive lesson or simply a learning resource like a video or a document. Learning activities represent the basic unit in which the dynamics of the system are characterized.

The term $\delta : A \rightarrow C$ represents a mapping function that maps a learning activity to a competency node in the learning map. A given competency may have several learning activities mapped to it. Different learning activities mapped to the same competency represent different available alternatives to help achieve the said competency.

Any competency $c \in C$ is in turn, made up of the following elements:

$$c = (Entry, Capacity, Assessment) \tag{2}$$

Every competency is associated with it, a set of assessments. The term *Assessment* refers to the collection of all assessments in the given competency node. Assessments may be of the form "Pre-tests" and "Post-tests" that are administered before and after any learning activity mapped to this node. A special form of assessment called the "benchmark assessment" or the "signature assessment" is associated with every competency. This is used for certifying whether the learner has obtained the said competency.

Successful completion of a signature assessment by a learner gets them a "badge" or a "micro-credential" drawn from a global pool of badge types called *Badges* (defined in Eqn 1). The set

Badges forms a poset defined by a function $\sqsubseteq$ such that for any $b_i, b_j \in Badges$, $b_i \sqsubseteq b_j$ or $b_j \sqsubseteq b_i$ or both. When both conditions hold, the badges are said to be equivalent $b_i \equiv b_j$. A special badge $\perp$ (called "newbie") is defined such that $\forall b \in Badges, \perp \sqsubseteq b$.

Badges earned by learners influences the way they can stake a claim to gain entry into other competency nodes. The term *Entry* in Equation 2 refers to a set of "entry criteria" or "pre-requisites" that determine whether a learner is eligible to take up the signature assessment in a given competency node. The entry criteria set is of the form:

$$Entry(c) = \{(c', b') \mid c' \in C, b' \in Badges\} \tag{3}$$

Here c' is another competency which forms the pre-requisite for the given node, and b' is the minimum badge level that is expected from the learner from c'. Learners can enter competency node c if for all its entry criteria of the form (c', b') they have obtained a badge b from c', such that $b' \sqsubseteq b$.

The term *Capacity* in Eqn 2, defines a set of "roles" that a user can adopt in the competency node. Each $role \in Capacity$ defines a set of criteria similar in structure to Eqn 3, specifying a set of competencies and minimum expected badge levels. A user can enter a competency node several times in different roles. When a learning activity mapped to a given competency is presented to the user, the set of all possible roles that they are eligible for, is also presented. Users may choose to activate a given role from the set of all roles that they are eligible for.

4 COMPETENCY MAP

As learners perform learning activities and successfully complete signature assessments, they are said to gain "mastery" over competencies in the subject. At any time, the user may be working on several competencies and may have achieved mastery over any number of competencies. The performance of the user at any point in time for a given learning map, is represented by a *competency map* for the user.

Figure 3 shows an example competency map for a user. Cells colored green represent "mastery" achieved by the user, cells colored yellow depict competencies that the user is currently pursuing. Grey colored cells represent competencies that the user has not attempted.

A user is said to have mastered a competency after successfully completing the signature assessments associated with the competency node. Alternatively, users may also just "assert" that they already have a said competency, or a given competency may be "inferred" by the system based on how the user is performing in other competencies and dependencies among competencies. In the latter two cases, the mastery is said to be tentative, and a green colored cell may be reverted to yellow or gray if the assertion or inference is invalidated by conflicting evidence.

The user's "location" in the learning map is represented not as a single point, but as a set of competencies which can potentially form starting points for achieving the next learning objective. This set of points is represented by the competency "Skyline" denoted by the maximal points along the pedagogy axis for each domain. The Skyline is shown by the black overlay line in Figure 3.

Whenever the user achieves mastery over a competency (or asserts mastery, or mastery is inferred), the skyline is updated. Algorithm 1 introduces the algorithm for updating a user's competency skyline.

Users can navigate through the space using pathways created and curated by humans in the form of lessons and courses. Alternatively, users can also choose target competencies in the space and ask the navigator to compute a route to achieve the target competency. Figure 2 shows an example of a route computed by the navigator in response to the user's wish to achieve a given competency.

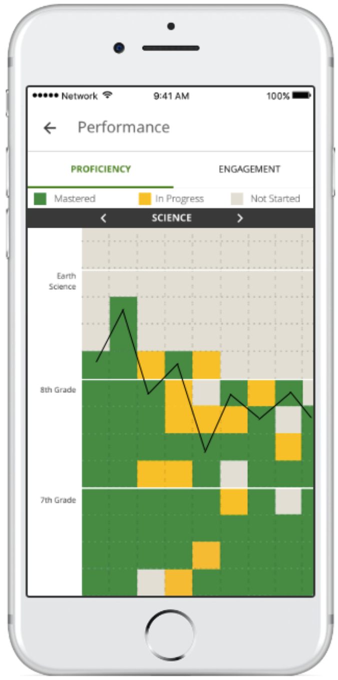

Fig. 3. Competency map for a navigator user

Algorithm 1 Algorithm for updating user's competency skyline

procedure UPDATESKYLINE($User\ u, Competency\ c$) ▷ Called when user u masters competency c.
    $skyline \leftarrow skyline(u)$ ▷ $skyline(u)$ returns the skyline list for user u, which is initialized to the null list [] to begin with
    $(q, p) \leftarrow dimensions(c)$ ▷ Get the coordinates of c
    if skyline contains a competency of the form (q, p') **then**
        if $p > p'$ **then**
            Replace (q, p') with (q, p) in $skyline$
        else
            ignore (q, p) ▷ Not a skyline competency. Do nothing.
    else
        Add (q, p) to $skyline$
    return $skyline$ ▷ Return the updated skyline

Finding a route to a user selected destination competency involves starting from the destination and unraveling its set of pre-requisites, until we reach any point on the user's current skyline. Algorithm 2 depicts the basic theory behind the route-finder algorithm.

In practice, since the route has to be computed in real-time in response to a user's query, the pre-requisite tree is not fully unfurled. If the specified target competency is too far with respect to the user's current skyline, a new virtual target competency is computed by the navigator by

Algorithm 2 Algorithm for computing a route to a target competency specified by the user

procedure ROUTE($User\ u, Competency\ c$) ▷ Called when user u wishes to navigate to competency c.
    $skyline \leftarrow skyline(u)$
    $route \leftarrow []$ ▷ Initialize route to the empty list
    $route << c$ ▷ Push c onto the route
    for $c \in route$ **do**
        if $c \notin skyline$ **then**
            $prereq \leftarrow prerequisites(c)$
            $route << prereq$ ▷ Get pre-requisites of c and push them on the route
        else
            break ▷ Keep unfolding the pre-requisites tree until we reach the skyline
    return $reverse(route)$

coming down the P-axis until there exists some competency mastered by the user at that pedagogic depth. This is used as a new virtual target and the user is directed to first reach mastery in this competency, before attempting to master the chosen competency.

5 LESSONS

Human curated learning experiences over the learning map, are in the form of "Lessons" where each lesson is a set of "Collections." Each collection in turn, is a "playlist" of learning activities. A lesson is also seen as a "navigation" on the learning map.

Figure 4 shows an example of the user interaction with a lesson. While the user is performing learning activities corresponding to different elements in the lesson, the navigator component (at the top of the user screen) provides the context of the learning activity, by representing the lesson as a navigational route and locating the user on the route. Similarly, the community management component called the "Suggest subsystem" connects the user with the rest of the participant community for this subject. Both these elements are visible in Figure 4.

Each learning activity in a lesson is mapped to an underlying competency. As noted in the previous section, each competency imposes certain constraints and provides some form of micro-certification to the learner.

Lessons are manually curated playlists formed by creating a sequence of learning activities by teachers and subject matter experts. In cases where the user wishes to achieve a target competency and invokes the navigator to compute a route, a lesson needs to be created automatically.

Algorithm 2 returns a list of competencies that need to be mastered by the user to achieve the learning objective. Each competency node in the list may have several learning activities associated with it. The navigator invokes the services of the Semantic subsystem to associate a suitable learning activity with each competency in the route.

Creating a sequence of learning activities from a given sequence of competencies is called the "Narrative Arc" problem – which is currently an open research problem.

Learners who are not eligible to enter a given competency node, cannot participate in any learning activity mapped to that node that is presented to them as part of a lesson. Similarly, a given learner may not find the presented learning activity to be best suited to his/her learning style and may prefer to participate in another activity mapped to the same competency. Such deviations from a given collection are handled by the semantic processing subsystem called the "Suggest" subsystem that is introduced in the next section.

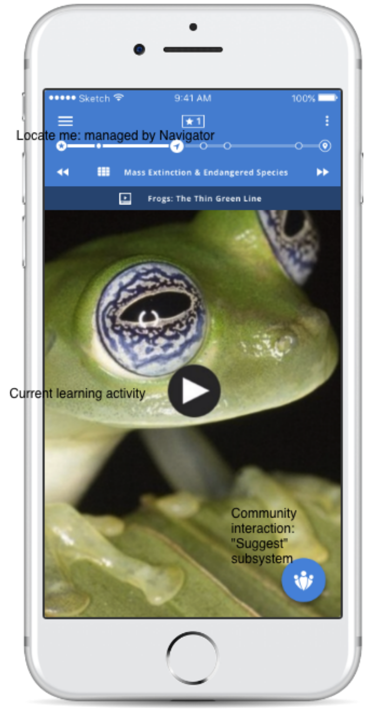

Fig. 4. Navigating through a lesson

6 THE "SUGGEST" SUBSYSTEM

The community component of the Gooru architecture is called the "Suggest" subsystem. This subsystem suggests different kinds of mediated interactions between participants in order to enrich the overall learning experience. Some of the fundamental guiding principles for building learning communities, are based on the following characteristics:

Asymmetry: Any community of learners interested in a given subject represent an uneven distribution of knowledge, skills and insight pertaining to that subject. While some asymmetry (such as between teachers and students) may be stark or explicit, asymmetry is a characteristic of the community as a whole. This means that there is a need, as well as an opportunity, to continuously detect "demand" and "supply" mismatch and mediate between the two.

Construct responses: When learning is facilitated by mediation, rather than by automated tutoring, it opens the possibility of "construct responses." An automated tutor can only choose resources from a given repository in response to a learner's needs. However, with human tutors, it is possible to *create* new resources and/or learning experiences dynamically, in response to the needs of a learner.

Serendipity: Construct responses also increase the possibility of *serendipity* in the learning experience, by creating valuable connections between knowledge demand and supply that was not part of the originally planned learning experience.

Figure 5 shows a schematic description of the Suggest subsystem. Broadly, the subsystem is divided into four levels: *Data*, *Analytics*, *Semantics* and *Mediation*.

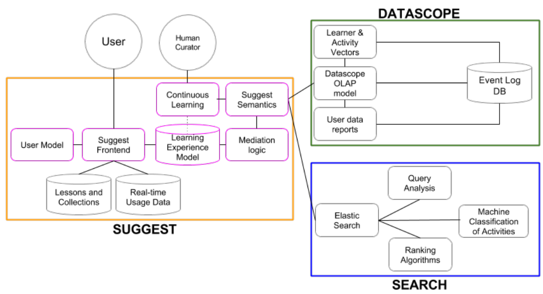

Fig. 5. Suggest Subsystem: schematic representation

The *Data* layer is concerned with management of two forms of data: data relating to learning resources, and data related to activities. The first kind of data is managed in the "Search" subsystem. This subsystem is centered around a keyword-based search index that indexes all resources including learning materials and learner profiles. Activity data are managed by an application called the "Datascope". This implements a data warehouse built from all the Event logs. The warehouse is organized in the form of a snowflake schema, comprising several fact tables, each of which, are described by a set of dimension tables. Essential facts managed by the datascope include the following: *Competency node*, *Learning Activity*, *Learner*, *Collection* and *Lesson*.

The *Analytics* layer comprises of a set of algorithms that computes several forms of aggregate information based on data stored in the Search and Datascope subsystems. Some examples are described below:

Activity Vector. The "activity vector" is a vector associated with every learning activity, scoring it along the following dimensions: *Relevance*, *Engagement* and *Efficacy*. They are described as follows:

Relevance: The relevance score of a learning activity a, that is mapped to competency c, is a measure of its usefulness, based on how often a was used whenever competency c was required:

$$relevance(a) = \frac{collections(a)}{collections(c(a)) + \epsilon} \tag{4}$$

Here $collections(a)$ is the set of all Collections that use activity a, and $collections(c(a))$ is the set of all collections that use any activity from c.

Engagement: This is a measure of how engaging is a given activity, calibrated based on its usage and reactions by users.

Let $reactions(a)$ represent the set of all reactions (likes, recommendations, etc.) from users to activity a, and let $use(a)$ represent represent the set of all users who have successfully completed activity a. The engagement score of a is calculated as:

$$engagement(a) = relevance(a) * \tag{5}$$

$$\left[\alpha \frac{|reactions(a)|}{\max\limits_{t \in c(a)} |reactions(t)|} + (1-\alpha)\frac{|use(a)|}{\max\limits_{t \in c(a)} |use(t)|}\right] \tag{6}$$

Here $0 \leq \alpha \leq 1$ is a tuning parameter that can be used to vary the mix between reaction-related scores and use-related scores, and $c(a)$ is the competency node for a.

Efficacy: The efficacy score for a learning activity measures how effective it was in making an *observable difference* in outcomes displayed by the learner after performing this activity. This is computed using the model of Bayesian Knowledge Tracing (BKT) [9], which is quite popularly used in automated tutoring. BKT is based on computing the probability of acquiring a competency, based on four priors:

- P_{init}: the probability that a learner already has the said competency
- $P_{transit}$: the probability that a learner will transit from not having the competency to having the competency after participating in the learning activities of this node
- P_{slip}: the probability of a learner having the said competency failing to show it in the signature assessment
- P_{guess}: the probability that a learner not possessing the said competency, clears the signature assessment by luck.

Given the above, the probability of a given learner u obtaining the said competency based on positive and negative outcomes, is given as follows:

$$P_u(c|outcome = 1) = \frac{P_{init}(1 - P_{slip})}{P_{init}(1 - P_{slip}) + (1 - P_{init})P_{guess}} \tag{7}$$

$$P_u(c|outcome = 0) = \frac{P_{init}P_{slip}}{P_{init}P_{slip} + (1 - P_{init})(1 - P_{guess})} \tag{8}$$

The overall probability of a learner obtaining the competency is given by:

$$P_u(c) = P_u(c|outcome) + P_{transit}(1 - P_u(c|outcome)) \tag{9}$$

The efficacy score for a competency node c is computed by summing up all the competency scores for all users who have completed the signature assessment:

$$efficacy(c) = \sum_u P_u(c) \tag{10}$$

The efficacy score for every learning activity a that is mapped to c is computed by multiplying the cosine score of its relevance and engagement scores with the efficacy score of the competency:

$$efficacy(a) = \frac{relevance(a)engagement(a)}{\sqrt{relevance^2(a) + engagement^2(a))}} efficacy(c(a)) \tag{11}$$

Learner Vector. Along similar lines to activity vectors, each learner is also characterized by a vector of aggregate properties. A brief description of these properties are as follows:

Proficiency: The proficiency score of a learner is the expected value of the badge that the learner is likely to obtain, if the learner takes an arbitrary signature assessment at an arbitrary competency node (after completing its corresponding learning activity). The expected value is calculated by computing the average percentile score across all badges earned by the learner, compared to others who have completed the corresponding signature assessment. Then, a global distribution of badges is computed showing the distribution of badges earned. The computed percentile score is then projected onto the global badge distribution to determine the expected badge by the learner in an arbitrary assessment.

Authority and Citizenship: Authority and citizenship are two scores associated with learners that represent the social value of their participation. They are computed based on a number of social feedback mechanisms that are implemented by learning activities. These include mechanisms for recommendations, endorsements, likes, etc. that learners can offer to other learners or to the activity or content created by them. Authority and citizenship are considered duals of one another and are defined as follows: *A good authority is one who is recognized by a good citizen, and a good citizen is one who recognizes good authorities.*

The recursive formulation of these two scores makes it amenable for implementation using the well known HITS algorithm [18].

In addition to the activity and learner vectors, the analytics layer also computes a number of other aggregate information that are relevant for semantics and mediation.

The third layer of the Suggest subsystem is the *Semantics* layer. This layer computes semantic associations between entities in the learning map in order to aid in mediation.

The semantic layer is aided by a global data structure called the *co-occurrence graph*, of the form $G = (T, E, w)$. This is a weighted, undirected graph, where edges from the set E depict pair-wise co-occurrences of terms from T anywhere in the system – in descriptions of resources, learning activities, forum posts, user profiles, etc. The weight w associated with any edge refers to the frequency of co-occurrence.

This graph is used to compute semantic associations, based on the 3-layer model introduced in [25]. Some example semantic associations are described below:

Characteristic anchors: Given a learning activity a or a learner profile l along with their activities data, this algorithm computes a set of topical terms (from the set of all topics T from Eq. 1 that are managed by the learning map), that may be considered to characterize a or l. Each entity in the system is associated with it, a set of keywords that are extracted using a keyphrase extraction algorithm.

A set of highly relevant keyphrases from the entity are provided as input to the Topical Anchor algorithm introduced in [25]. This algorithm tries to provide terms that are likely to topically describe the input set of terms. The set of highly rated terms from this algorithm that match terms in the controlled vocabulary T are returned as characteristic anchors.

Characteristic anchors are computed in a context specific fashion. For instance, given a learner profile l, the characteristic anchors can be computed based on the learner's activity in a competency c, by considering only those keywords that associate the activities of l in c. Similarly, characteristic anchors can be computed for contexts like a topic or a pedagogic depth.

Characteristic anchors are represented as a probability density function, comprising of a set of anchor terms and a probability value associated with it.

Activity and learner siblings: Given an entity (learning activity or learner profile), this algorithm suggests "sibling" entities based on the similarity in their characteristic anchors. Siblings are computed based on a notion of "replacability". Given a context c and entities (learning activities or learner profiles) A and B, with their characteristic anchors defined as $\vec{A_c}$ and $\vec{B_c}$ respectively, the sibling value of B over A is defined as the information gain in the characteristic anchors when A is replaced by B.

This is computed using the K-L divergence score from A to B as follows:

$$D(A||B) = \sum_i A_c(i) \frac{A_c(i)}{B_c(i)} \tag{12}$$

where $A_c(i)$ and $B_c(i)$ refers to the probability value assigned to the i^{th} characteristic anchor under this context c for A and B respectively.

In addition to the above, several other semantic associations are computed, which collectively form the semantic layer of the Suggest subsystem.

7 MEDIATION AND EMERGENCE

The topmost layer of the community component of Gooru is called the *Mediation* layer. The objective of this layer is to foster the emergence of desired collective outcomes from learning communities by strategically mediating between knowledge needs and expertise.

Mediation can be seen both as an *operational* or as a *structural* construct. Operational mediation involves identifying opportunities for mediation in the course of the learning activities of two or more users and suggest mediated interactions between them. Some examples are as follows:

OC1: *A student tries to open a learning activity in a collection and is denied entry as s/he does not satisfy the entry criteria.*

In such cases, the Suggest subsystem may either recommend pre-requisite learning activities to the student, or may mediate the student with a teacher or a tutor who may curate a customized lesson for the student.

OC2: *A course creator wishes to jointly offer a course with a suitable collaborator.*

In such cases, the Suggest subsystem computes "sibling users" using algorithms from the semantics layer to suggest as potential collaborators.

One of the underlying guiding principles for operational mediation is the *Pareto improvement* guideline – or a non-negative value addition to all the parties involved in the suggested activity.

Structural Mediations. In contrast to operational interventions, mediation can also be seen as a structural construct. Here, different structural and policy elements of the system are introduced in order to create a strategic disposition for the system towards producing intended collective outcomes.

Structural mediation is an active area of research involving creation of semantic ontologies and modeling system dynamics using different paradigms like agent-based modeling, complex networks modeling, etc.

8 IMPLEMENTATION

The Gooru system presented in this paper is a live system whose initial versions have been deployed at different schools across the United States.

The current implementation comprises the Learning Map, Collections, Search subsystem and the Datascope. A basic version of the Suggest subsystem is also available, that provides two forms of operational mediation.

Other forms of mediation and a richer version of the semantic layer are under implementation and would be dependent on sufficient operational data collected from existing installations.

While, formal calibration models and experiments are yet to be conducted on Gooru installations, performance of students from the Leadership Public Schools (LPS) in Oklahoma, California, who have adopted the Gooru model of learning, have shown an average of 2.8 years of growth per year, as measured by the independent NWEA[1] Measurements of Academic Progress (MAP) assessments[2].

9 CONCLUSIONS

In this paper, we presented a case for modeling web-based educational platforms as a social machine, rather than as an extension of the conventional classroom. The core idea is based on the tenet that a learning experience is largely a sequence of mediated human interactions. Almost all forms of educational experiences – be they sports training, technical education or professional education, the human interaction sequence is central. Learning environments are best designed to support such human interactions, rather than make them obsolete in the quest for scalable models.

The Gooru system presented in this paper is a live implementation. This paper presented the overall architecture and design rationale of the system. We plan to introduce each of the research elements that make up the overall system, individually in separate research communications over time.

REFERENCES

[1] John R Anderson, C Franklin Boyle, and Brian J Reiser. 1985. Intelligent tutoring systems. *Science (Washington)* 228, 4698 (1985), 456–462.

[2] Kaustuv Basu. 2012. "Faculty groups consider how to respond to MOOCs. Inside HigherEd. Online Publication. https://goo.gl/tamk8k. (May 2012). Last accessed: 1 Jan 2017.

[3] T. Berners-Lee and M. Fischetti. 1999. *Weaving the Web: The Past, Present and Future of the World Wide Web by Its Inventor.* Orion Business. https://books.google.com/books?id=4UeymwEACAAJ

[4] Jacob Lowell Bishop and Matthew A Verleger. 2013. The flipped classroom: A survey of the research. In *ASEE National Conference Proceedings, Atlanta, GA*, Vol. 30.

[5] Charles P Bloom and R Bowen Loftin. 1998. *Facilitating the development and use of interactive learning environments.* CRC Press.

[6] Lori Breslow, David E Pritchard, Jennifer DeBoer, Glenda S Stump, Andrew D Ho, and Daniel T Seaton. 2013. Studying learning in the worldwide classroom: Research into edX's first MOOC. *Research & Practice in Assessment* 8 (2013).

[7] Konstantina Chrysafiadi and Maria Virvou. 2013. Student modeling approaches: A literature review for the last decade. *Expert Systems with Applications* 40, 11 (2013), 4715 – 4729. DOI: http://dx.doi.org/10.1016/j.eswa.2013.02.007

[8] John Coopey. 1995. The Learning Organization, Power, Politics and Ideology Introduction. *Management Learning* 26, 2 (1995), 193–213. DOI: http://dx.doi.org/10.1177/135050769502600204

[9] Ryan SJ d Baker, Albert T Corbett, and Vincent Aleven. 2008. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *International Conference on Intelligent Tutoring Systems.* Springer, 406–415.

[10] Richard DuFour and Rebecca DuFour. 2010. The role of professional learning communities in advancing 21st century skills. *21st century skills: Rethinking how students learn* (2010), 77–95.

[11] Ian Falk and Lesley Harrison. 1998. Community learning and social capital:"Just having a little chat". *Journal of Vocational Education and training* 50, 4 (1998), 609–627.

[12] D.R. Garrison and N.D. Vaughan. 2008. *Blended Learning in Higher Education: Framework, Principles, and Guidelines.* Wiley. https://books.google.co.in/books?id=2iaR5FOsoMcC

[13] D Randy Garrison and Heather Kanuka. 2004. Blended learning: Uncovering its transformative potential in higher education. *The internet and higher education* 7, 2 (2004), 95–105.

[14] Arthur C Graesser, Mark W Conley, and Andrew Olney. 2012. Intelligent tutoring systems.. In *APA educational psychology handbook, Vol 3: Application to learning and teaching.* American Psychological Association.

[15] Kevin Hartnett. 2013. San Jose State to Michael Sandel: Keep your MOOC off our campus. Boston Globe, Online Article. https://goo.gl/lLnePp. (May 2013). Last accessed: 1 Jan 2017.

[16] Michael B. Horn and Meg Evans. 2013. A factory model for schools no longer works. Milwaukee-Wisconsin Journal Sentinel Online. https://goo.gl/qngLwN. (June 2013). Last accessed: 1 Jan 2017.

[1] http://www.newa.org/

[2] A report on this is accessible here: http://www.christenseninstitute.org/wp-content/uploads/2016/07/Connecting-ed-and-tech.pdf

[17] Heisawn Jeong and Cindy E. Hmelo-Silver. 2010. An Overview of CSCL Methodologies. In *Proceedings of the 9th International Conference of the Learning Sciences - Volume 1 (ICLS '10)*. International Society of the Learning Sciences, 921–928.

[18] Jon M. Kleinberg. 1999. Authoritative Sources in a Hyperlinked Environment. *J. ACM* 46, 5 (Sept. 1999), 604–632. DOI: http://dx.doi.org/10.1145/324133.324140

[19] Clive Lawson and Edward Lorenz. 1999. Collective learning, tacit knowledge and regional innovative capacity. *Regional studies* 33, 4 (1999), 305–317.

[20] Kariene Mittendorff, Femke Geijsel, Aimee Hoeve, Maarten de Laat, and Loek Nieuwenhuis. 2006. Communities of practice as stimulating forces for collective learning. *Journal of Workplace Learning* 18, 5 (2006), 298–312.

[21] Naomi Miyake. 2007. Computer supported collaborative learning. *The SAGE handbook of e-learning research* (2007), 248–265.

[22] C. O'Malley. 2012. *Computer Supported Collaborative Learning.* Springer Berlin Heidelberg.

[23] Laura Pappano. 2012. The Year of the MOOC. *The New York Times* 2, 12 (2012), 2012.

[24] Alexandros Paramythis and Susanne Loidl-Reisinger. 2003. Adaptive learning environments and e-learning standards. In *Second european conference on e-learning*, Vol. 1. 369–379.

[25] Aditya Ramana Rachakonda, Srinath Srinivasa, Sumant Kulkarni, and M. S. Srinivasan. 2014. A Generic Framework and Methodology for Extracting Semantics from Co-occurrences. *Data Knowl. Eng.* 92 (July 2014), 39–59. DOI: http://dx.doi.org/10.1016/j.datak.2014.06.002

[26] T. Rose. 2016. *The End of Average: How to Succeed in a World That Values Sameness.* Penguin Books Limited. https://books.google.co.in/books?id=w7-zCgAAQBAJ

[27] P.M. Senge. 1990. *The Fifth Discipline: The Art and Practice of the Learning Organization.* Doubleday/Currency. https://books.google.com/books?id=bVZqAAAAMAAJ

[28] Robert E. Slavin. 1980. Cooperative Learning. *Review of Educational Research* 50, 2 (1980), 315–342. DOI: http://dx.doi.org/10.3102/00346543050002315

[29] Robert E Slavin. 1983. When does cooperative learning increase student achievement? *Psychological bulletin* 94, 3 (1983), 429.

[30] D. Sleeman and J.S. Brown. 1982. *Intelligent tutoring systems.* Academic Press. https://books.google.co.in/books?id=gfBEAAAAYAAJ

[31] Louise Stoll, Ray Bolam, Agnes McMahon, Mike Wallace, and Sally Thomas. 2006. Professional Learning Communities: A Review of the Literature. *Journal of Educational Change* 7, 4 (2006), 221–258. DOI: http://dx.doi.org/10.1007/s10833-006-0001-8

[32] Louise Stoll and Karen Seashore Louis. 2007. Professional learning communities: Elaborating new approaches. *Professional learning communities: Divergence, depth and dilemmas* (2007), 1–13.

[33] Mary Thorpe, MR Lea, and K Nicoll. 2002. From independent learning to collaborative learning. *Distributed learning: Social and cultural approaches to practice* (2002).

[34] Bill Tucker. 2012. The flipped classroom. *Education next* 12, 1 (2012).

[35] Audrey Watters. 2015. The Invented History of 'The Factory Model of Education'. Hack Education Online Article http://hackeducation.com/2015/04/25/factory-model. (April 2015). Last accessed: 1 Jan 2017.