Knewton Adaptive Learning

* **Introduction**

The Knewton platform is a flexible, scalable system for delivering adaptive learning experiences and predictive analytics across arbitrary collections of content in different learning environments. Knewton provides a set of tools and services that education companies can use to make their learning applications adaptive. These tools, which include individualized tutoring, predictive analytics, and student progress reports, are built into a platform that is focused at a fundamental level on scalability.

* **What Knewton means by adaptive learning?**

Knewton analyses learning materials based on thousands of data points — including concepts, structure, difficulty level, and media format — and uses sophisticated algorithms to piece together the perfect bundle of content for each student, constantly. The system refines recommendations that harness the power of all the data collected for all students to optimize learning for each individual student.

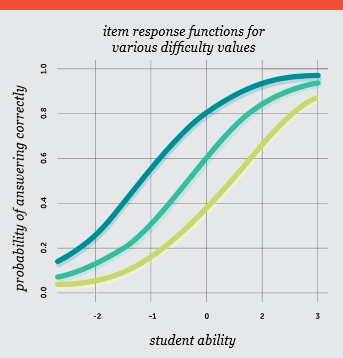
* **Overview of Knewton’s Approach**

Knewton pairs information about content (via the Knewton knowledge graph (top)) with student response data (left) to make real-time psychometric inferences about student abilities. These inferences are in turn used to power predictive analytics on student outcomes (bottom) and to generate personalized recommendations for what to study next (right). Knewton’s accuracy improves as more data are collected, since Knewton can use student response information to revise and upgrade the models and parameters used in the analytics and recommendation systems (bidirectional arrows).



* **Theories & approaches behind Knewton recommendations**
* **Item Response Theory**

IRT models student ability using question level performance instead of aggregate test level performance. Instead of assuming all questions contribute equivalently to our understanding of a student’s abilities, IRT provides a clearer view on the information each question provides about a student. It is founded on the premise that the probability of a correct response to a test question is a mathematical function of parameters such as a person’s latent traits or abilities and item characteristics (such as difficulty, “guessability,” and specificity to topic).

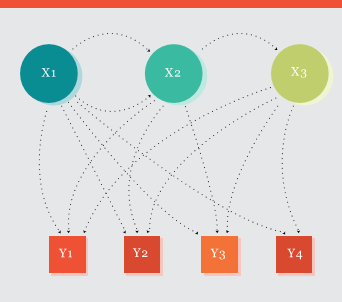


Item Response Theory

IRT models help us to better understand how a student’s performance on testing relates to his ability.

* **Probabilistic Graphical Models (PGMs)**

This framework includes statistical methods like Bayesian Network and Markov Random Fields. Knewton applies PGMs to determine which other topics student may be ready to master by using his known proficiencies. For instance, such a model might help the platform discover to what degree a mastery of fractions helps students master decimals and to what degree a mastery of decimals helps students master exponentiation. Thus, it helps to determine the relationship between mastery of fractions and mastery of exponentiation.

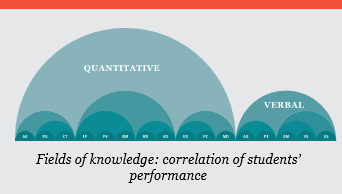


Probabilistic Graphical Model

Discovery of such relationships through the Probabilistic Graphical Model helps to refine recommendations.

* **Hierarchical Agglomerative Clustering**

It is a method of analysis which aims to construct a hierarchy or structure of clusters. At Knewton, the technique is used to detect latent structures within large groups and build algorithms that determine how students should be grouped and what features they should be grouped by.



Hierarchical Agglomerative Clustering

It can be used by the teachers to group students who are working on the same material by level of concept mastery.

* **Why Knewton is so effective?**
* **Knowledge Graphs**

Knowledge Graph takes into account academic concepts defined by sets of content and the relationships between those concepts. When visualized, the Knowledge Graph can provide a sense of a student’s potential flow through the course material. Knewton recommendations steer students on personalized and even cross disciplinary paths on the Knowledge Graph towards ultimate learning objectives based on both what they know and how they learn.

A single-point adaptive learning system evaluates a student’s performance at one point in time, and from there determines the type of instruction she receives. Knewton’s continuously adaptive learning system, on the other hand, constantly mines student performance data, responding in real time to a student’s activity on the system. Upon completion of a given activity, the system directs the student to the next activity.

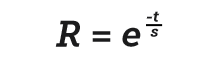
* **Spaced reinforcement**

It is a learning method in which new concepts or skills are absorbed while previously taught concepts and skills are reinforced. Because new material is introduced incrementally and woven into familiar material, spaced reinforcement typically occurs over an extended period of time. Spaced reinforcement allows Knewton recommendations to help students build their skills in a cumulative way and retain understanding once it is gained.

* **Retention & learning curves**

The Knewton recommendation engine needs to be able to take the degradation of skill (or forgetting) into account. That is, it needs to be able to detect such occurrences and provide actionable recommendations as a result. So, for this purpose, Knewton uses Hermann Ebbinghaus’s work on memory retention and learning curves.

These curves are governed by the following premise: each time students are exposed to content associated with a given topic, they receive a “bump” in their virtual ability level for a topic; likewise, if they are not exposed to some other topic, they likely “forget” that topic over time. The forgetting curve itself that governs rate of retention is roughly described by the following formula:

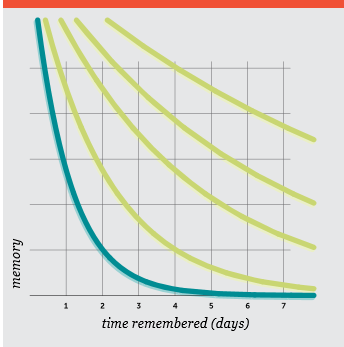


Where,

R is memory retention,

S is the relative strength of memory,

t is time.



The Forgetting Curve

* **Student learning profile**

If a student who has already taken a Knewton-powered course enrols in another, the course starts “warm” with that student’s data (as opposed to starting “cold” with no data). The course takes into account the student’s recently mastered concepts and skills and unique trajectory through the material, and uses this knowledge to maximize student learning continuously from that point forward. Once enough data is collected, the platform will uncover patterns in the student’s learning, likely blind spots; modality and medium preferences; and granular strengths and weaknesses.

* **The Knewton Inferential Engine**

Knewton provides adaptive recommendations and predictive analytics via its inferential engines. There are two types of engines:-

* **The Recommendation Engine**

It consists of models that power the adaptive system behind Knewton’s recommendation service.

Example: **Modelling Engagement**

Knewton incorporates a model of student engagement to make recommendations as effective as possible.

When a student’s engagement drops, her productivity also tends to drop, sometimes to the point of ending the session entirely. The data hint that perhaps some content is more likely to cause a student to quit working than others, and indeed, that is frequently the case. One might think that content with a high quitting rate should never be recommended to students. But in certain cases, the content might provide reasonable instruction, just on a very difficult concept. In these cases, Knewton’s proficiency models can reveal whether students tend to perform better after having worked through this particular piece of content, even though it is challenging, and can use that information to make more informed recommendations.

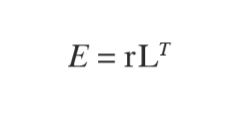
* **The Analytics Engine**

In parallel with the recommendation engine, Knewton’s analytics engine provides real-time inferences for predictive analytics and student reports.

**Example:** **Active Time**

The Active Time metric tracks how much productive time students spend working with educational materials. For this purpose, an inferential model is used.

Each interaction from student **s** on item **i** has a duration **d(s, i)** that is assumed to be a product of a student-specific rate **r(s)** and an item length **l(i**), plus a component **e(s, i)** which represents deviations from the expected duration due to loss of engagement with the content. In typical learning data, many students interact with the same item, and a single student can interact with many different items. The model uses these relationships by collecting the interaction durations into a sparse matrix **D**, where entry **s**, **i** in the matrix represents the duration of student s’s interaction with item **i**. This sparse matrix is then approximated via a rank-1 estimate E defined as



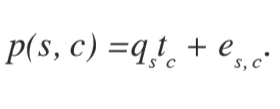
where r = (r(1), r(2), … r(n)) and L = (l(1), l(2), … l(m)), so that the deviation between **E** and **D** (at the existing entries of **D**) is due to non-zero engagement components es, **i**.

Active Time uses a simple median calculation to find the factorized matrix E due to the advantage of efficiency. Once the parameter vectors r and L are computed, each new interaction duration that is received can be compared to the predicted duration given the student and item components. This comparison is used to determine the active time awarded for the interaction.

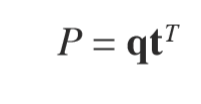
**Example:** **Work Remaining**

The Work Remaining metric predicts how much activity Knewton expects that a student will have to complete before she is proficient in a content area.

For each student **s** and content area **c**, there is assumed to be a quickness parameter **q(s)** and a toughness parameter **t(c)** that together with an error term **e(s, c)** form a prediction for how much practice it takes a student to gain proficiency in a content area:

.

The Work Remaining metric assumes a rank-1 approximation to a matrix P with the form



where q = (q(1), q(2), … q(n) ) and t = (t(1), t(2), … t(n) ). The vectors **q** and **t** are computed using a large-scale matrix completion algorithm.

While these estimates (together with estimates of proficiency) are useful for understanding a student’s progress on the content they are currently working on, by combining these estimates with the learning paths implied by the Knewton knowledge graph, they can also be used to predict how much instruction and practice a student will require for a larger group of content, or for educational goals several days or weeks in the future.

* **Big data & adaptive infrastructure**
* **Big Data and Education**

Big data unleashes a range of productive possibilities in the education domain in particular, since data that reflects cognition is structurally unique from the data generated by user activity around web pages, social profiles, and online purchasing habits. Because there is a very high degree of correlation between educational data (mastery of fractions and mastery of exponentiation, for example), there is tremendous potential to optimize user experiences over time and provide tangible value for students.

* **Adaptive Infrastructure**

Knewton has designed its own framework called AltNode which works by dividing work between machines and then sending continuous updates between the minimal necessary number of machines. All significant updates are stored in a distributed Cassandra database. If one machine fails, another one nearby automatically takes its place, recovering recent values from the database and resuming work. One unique feature of AltNode is that it allows models to recover from any state and respond to new data as it arrives.

* **Engagement of Students**

To improve student’s engagement with the platform, Knewton uses the following:

* **Instant Feedback**

Students are less likely to lose focus if feedback is immediate and they can quickly self-correct. A continuously adaptive learning system is able to deliver personalized feedback to both multiple choice and free response questions quickly.

* **Community & Collaboration**

An adaptive system can improve student engagement by weaving a social component into coursework. Knewton Math Readiness, for instance, provides a dashboard that allows teachers to group students who are working on the same material together.

* **Gamification**