

Recommendation based Learning Apps

Version 1.0.0.3

Vikramaditya Jakkula Real Simple Edu

The functional specification of the recommendation based learning approach for RealSimpleEdu solution offering

Table of Contents

[1 Introduction 1](#_Toc305434124)

[2 Proposed Solution 1](#_Toc305434125)

[3 Detailed Design 2](#_Toc305434126)

[3.1 User Interface 2](#_Toc305434127)

[3.2 Algorithm Details 3](#_Toc305434128)

[3.2.1 Work flow 3](#_Toc305434129)

[3.2.2 Core Algorithm Implementation 4](#_Toc305434130)

[4 Evaluation Methodology 4](#_Toc305434131)

[5 Summary 4](#_Toc305434132)

[6 vNext 5](#_Toc305434133)

[7 Version 5](#_Toc305434134)

# Introduction

RealSimpleEdu currently offers on the go learning experience via mobile solutions and immersive learning via tablet based solution offering. The need for true user learning experience has been identified. The need for recommendation based learning approach would help personalize the learning experience, hence enhancing the overall user experience as well retaining and improving the current solution offering.

# Proposed Solution

Given the current solutions are being offered via mobile and tablet offering, we currently have identified computation limitation both in terms of processing time and memory consumption. We propose two ways to approach this problem; however we plan to focus on the first for now and later invest into the second approach. Our primary solution is a simple collaborative filtering approach which can reside on the device and the second solution is a client and server based approach with collaborative filtering fusing with clustering which is more computationally intense and offers improved experience.

# Detailed Design

## User Interface

|  |
| --- |
|  |
| Details:  This screen gives us a simple interface for the user.  The user has three options:   * Pre-Test = Gives a simple 20 question test and finally displays the scores. * Learn = Gives a simple reference to the key topics from existing MathRef Application from RealSimpleEdu existing offering. * Test = Full feature test based on the mathematical area. |

|  |
| --- |
|  |
| Details: The test screen   * The Interface is simple with question and options displayed with an option to look at the solution once an answer is selected. When the user selects next, it navigates to the next screen. * When the question is loaded a time counter is started. * When the user clicks next in the back ground the CF – Slope One algorithm is run in the background. And then the next question is loaded and time counter is reset. |

|  |
| --- |
|  |
| Details:   * The final score is displayed along with areas of focus in the test and a percentile correct. * This is displayed at the very end after the completion. |

## Algorithm Details

We divide the questions into levels of 1-5 and recommend the level based on domain and level. Domain here is an ID for the type of problem, for instance, the domain of ID 1 could be mathematics addition operation and level 5 would be adding 5 digit decimal numbers;

### Work flow

Step 1: Start with a low rating level question based on the selected domain.

Step 2: If answered pass it to the algorithm to recommend a question.

Step 3: The algorithm returns the domain and level of question.

### Core Algorithm Implementation

The algorithm used for this purpose if [slope one algorithm](http://en.wikipedia.org/wiki/Slope_One#Slope%5Fone%5Fcollaborative%5Ffiltering%5Ffor%5Frated%5Fresources) from the collaborative filtering space. Rating-based collaborative filtering is the process of predicting how a user would rate a given item from other user ratings. We propose three related slope one schemes with predictors of the form f(x) = x + b, which pre-compute the average difference between the ratings of one item and another for users who rated both. Slope one algorithms are easy to implement, efficient to query, reasonably accurate, and they support both online queries and dynamic updates, which makes them good candidates for real-world systems.

Input: Previous history or set of <Domain ID>, <Level>

Output (Predicted): New <Domain ID>, <Rating/Level>

# Evaluation Methodology

* User Feedback: Have a user feedback screen which pop-up once a month locally on the app and later this data can be sent to the server for analysis. We can also have a feedback form to monitor any feedback from users who are using this app.
* App purchase/Download Trend: Observer and monitor the App purchase and download trend and look for trends which would help analyses.
* Increased no of tests or user interaction:

1. Have a monthly collection of App usage data sent to server to analyses how the app is being used.
2. Have a record of number of tests taken by the user and have this data sent to server once a month.

* The above identified data can be used to effectively measure or evaluate the user experience with the application empowered with recommendation based learning approach.
* For generic testing before release, we can have a controlled group of users use the application and provide feedback.

# Summary

The recommendation approach looks very promising and with server based solution offering would enhance the user learning experience significantly.

# vNext

|  |  |
| --- | --- |
| Version | Feature |
| 2.0.0.1 | Server based solution (Increased dimensional approaches as well) |
| 2.0.0.1 | Improving the UI |

# Version

|  |  |  |
| --- | --- | --- |
| Version | Changes | Date |
| 1.0.0.1 | Document Created | 10/1/2011 |
| 1.0.0.2 | Major Changes | 10/3/2011 |
| 1.0.0.3 | Minor errors and other small changes | 10/3/2011 |