**CodeTrix**

2014 Semester 1

**Attack Simulator on Recommender Systems**

**INB302 - Capstone Project**

**John Wilkinson, Philippe Gref-Viau, Corey Thompson, Mikhail Sutormin**

**Supervisor: Associate Professor Yue Xu**

Contents

[Section 1 Project Summary 4](#_Toc387925800)

[Requirements 4](#_Toc387925801)

[Scope 4](#_Toc387925802)

[Quality Objectives 4](#_Toc387925803)

[Section 2 Comparison of Planned versus Actual 4](#_Toc387925804)

[Original scope and history of approved changes 4](#_Toc387925805)

[Original scheduled Deadline vs. Actual Completion Date 4](#_Toc387925806)

[Test Plans and Test Results 5](#_Toc387925807)

[Section 3 Outstanding Issues 5](#_Toc387925808)

[Itemised List and expected completion 5](#_Toc387925809)

[Any Ongoing Support and Duration 5](#_Toc387925810)

[Appendix Project Documentation List 5](#_Toc387925811)

[Application Manual 5](#_Toc387925812)

[Technical Report 5](#_Toc387925813)

[Database Information 5](#_Toc387925814)

[Algorithms 5](#_Toc387925815)

[Classes 6](#_Toc387925816)

[Methods 6](#_Toc387925817)

[Test Cases 6](#_Toc387925818)

[Team Process 6](#_Toc387925819)

[Sprint 1 (Week 2) 6](#_Toc387925820)

[Sprint 2 (Week 3) 7](#_Toc387925821)

[Sprint 3 (Week 4) 7](#_Toc387925822)

[Sprint 4 (Week 5) 7](#_Toc387925823)

[Sprint 5 (Week 6) 7](#_Toc387925824)

[Sprint 6 (Week 7) 8](#_Toc387925825)

[Sprint 7 (Week 8) 8](#_Toc387925826)

[Sprint 8 (Week 9) 8](#_Toc387925827)

[Sprint 9 (Week 10) 8](#_Toc387925828)

[Sprint 10 (Week 11) 8](#_Toc387925829)

# Section 1 Project Summary

Recommender Systems are used widely by companies such as Amazon, EBay, Spotify and many other companies to give recommendations to their users. However they present a security risk, as attackers who cannot be readily distinguished from real users can inject fake profiles into the system to distort the results and in turn, produce recommendations which can unfairly produce better results for the attacker. Attackers can also use public accessible data to enhance the precision, invisibility and effectiveness of their attacks.

In the Attack Simulator on Recommender Systems project, we build an application to simulate these attacks on recommender systems. The resulting data after the simulated attack can then be compared with the original data to illustrate the effectiveness of different types of attacks on different types of recommender systems.

## Requirements

The Attack Simulator on Recommender Systems Project requires an application that can simulate injection attacks on various Recommender Systems. In order to complete the project, the following requirements must be met.

* Importing of ratings data
* Implementation of User-Based Collaborative Filtering System, based on pre-existing literature (provided algorithms) on generic lists of users/products/ratings.
* A well designed GUI to allow easy comparison of pre-attack and post-attack results.
* Simulation of Profile Injection Attacks through a well designed GUI which includes
  + Control over number of random item ratings for a fake user
  + Control over target item ratings for fake user
  + Control over number of fake profiles to inject

## Scope

There are many various types of attacks that can be simulated. For this project, the attacks are limited to Profile Injection Attacks.

Various forms of filtering exist, however this project is limited to the use of User-Based Collaborative Filtering.

Data used for filtering is limited to the simple format of User, Rating, and Item.

## Quality Objectives

The application will provide accurate and usable results which will be ensured through thorough testing during application development.

The application can be used for demonstrations so will GUI will be consistent, user-friendly and provide information in an organised and efficient way.

# Section 2 Comparison of Planned versus Actual

## Original scope and history of approved changes

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Original | Approved Changes | Comments |
| Application language | Java | C# | Difficulty implementing GUI with Java, so switched to C# due to prior experience |

## Original scheduled Deadline vs. Actual Completion Date

|  |  |  |
| --- | --- | --- |
| Item | Schedule | Completion |
| Team Contract | 11/03/2014 | 11/03/2014 |
| Project Plan | 25/03/2014 | 24/03/2014 |
| Team Conflict Presentation | 02/04/2014 | 02/04/2014 |
| Finished Application and Documentation | 21/05/2014 |  |
| Project Presentation | 28/05/2014 |  |
| Project Showcase | 04/06/2014 |  |

## Test Plans and Test Results

# Section 3 Outstanding Issues

## Itemised List and expected completion

## Any Ongoing Support and Duration

# Appendix Project Documentation List

## Application Manual

## Technical Report

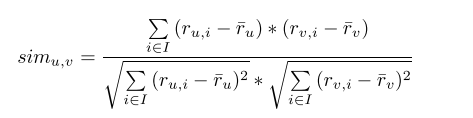
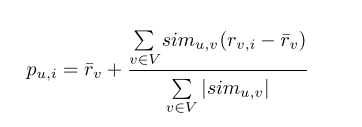
### Database Information

The project can import information into a new database from a plain text document that fits the schema of userID, itemID, rating. Alternatively, an existing database can be used that has the same schema.

### Algorithms

#### User-Based Collaborative Filtering

The user-based collaborative filtering method (*k*NN) is as follows;

1. Calculate similarity of all users to target user using correlation formula below  
     
   u = target user  
   v = neighbouring user  
   I = set of all items that can be rated  
   *ru,i* = rating of item *i*for target user *u  
   rv,i* = rating of item *i* for neighbouring user *v u* = average of ratings *u* over those items in I that *u* and *v* have in common  
   *v* = average of ratings *v* over those items in I that *u* and *v* have in common   
   
2. Select *k* most similar users to the target user
3. Filter out any users with similarity rating of less than 0.1
4. Calculate prediction of item *i* for target user *u* using formula below  
     
   V = set of *k* similar users  
   *rv,i* = rating of item *i* for neighbouring user *v  
   v* = average rating for target user over all rated items  
   *simu,v* = mean-adjusted Pearson correlation from Step 1  
   **

#### Rating Predictions

### Classes

### Methods

### Test Cases

## Team Process

CodeTrix implemented an Agile methodology for the project, holding weekly scrum meetings. The CodeTrix team was finalised in Week 2 of Semester 1, 2014 and so Sprint 1 corresponds to Week 2, Sprint 2 to Week 3 and so forth.

### Sprint 1 (Week 2)

* Finish and submit Team Contract
* Research Project Options
* Setup communications (Skype, Email, exchange phone numbers, Facebook Group)

RESULTS

* The Team Contract was complete on schedule.
* We researched potential projects and attempted to arrange a meeting for Thursday in regards to a short list of projects, however this was not suitable for the project supervisor and was postponed.
* Communications were setup successfully

### Sprint 2 (Week 3)

* Go to meeting with Project Supervisor and finalise project
* Draw up Project Plan

RESULTS

* The meeting was arranged, however the projects were not suitable for the team and so we continued our research into project options.
* More research done on potential projects and various emails sent out to supervisors

### Sprint 3 (Week 4)

* Finalise project
* Finalise project plan and submit

RESULTS

* Project was finalised with Associate Professor Yue Xu to do the Recommender System Attack Simulator Application
* Project Plan was completed and submitted

### Sprint 4 (Week 5)

* Prepare for Team Conflict presentation
* Prepare documents for Team Conflict scenario
* Complete Team presentation

RESULTS

* Preparation for presentation complete
* Presentation completed successfully

### Sprint 5 (Week 6)

* Create GUI for Database Selection
* Implement class to execute GUI user instructions

RESULTS

* GUI was completed
* Class for GUI execution completed but not actually linked with GUI

### Sprint 6 (Week 7)

* Begin work on Project Documentation

RESULTS

* Documentation template created

### Sprint 7 (Week 8)

* Create GUI

RESULTS

* Basic GUI created but not yet working with code properly

### Sprint 8 (Week 9)

* Prepare for practice presentation

RESULTS

* Good peer review marks from presentation

### Sprint 9 (Week 10)

* Add commenting to code
* Finish GUI
* Test Cases for Formula Calculations
* Continue Documentation

RESULTS

* Started application again using C# due to hassles with Java
* GUI 90% finished
* Test Case spreadsheet created for manual calculations
* Project Summary of documentation completed.

### Sprint 10 (Week 11)

* Finish GUI
* Test formula calculations
* Start creating slides for Presentation and Showcase