Algorithm of Recommendation System For Clicker System

Anoop Dobhal

May 11, 2014
# Contents

1 Algorithm for Recommendation System  
1.1 Algorithm for modifying the Parameters .................................. 1  
1.2 Scenarios for User Ui ................................................................. 2  
1.3 Scenarios for User Qj ................................................................. 2  
1.4 Tree Representation of Four Cases Discussed for Qj and Ui .......... 3  
1.5 Examples User Attempting Questions ........................................ 3
Abstract

This report describe an algorithm written for recommendation system used in AakashClicker.
Chapter 1

Algorithm for Recommendation System

Qi is difficulty score for question Qk where

\[ 1 \leq k \leq n \]  \hspace{1cm} (1.1)

n is total number of questions

Ui Every user in the system assigned with a number Ui, this parameter is user intelligence, more the number, he/she is more capable of solving problems. Value of this parameter varies between 0 and 1.

Initial Values
A new user or new question is assigned with unbiased 0.5 values. Based on the questions taken the parameters are expected to go UP or DOWN.

1.1 Algorithm for modifying the Parameters

Pi = Probability that user will answer the question correctly, as a function of (Ui and Qj),[1],[2]

if( Ui taken Qj )

\[ P_i = e^{U_i-Q_j} / (1 + e^{U_i-Q_j}) \]  \hspace{1cm} (1.2)

if( correct )

\[ U_i = U_i + (1 - U_i) * (1 - P_i) \]  \hspace{1cm} (1.3)

\[ Q_j = Q_j - Q_j * P_i \]  \hspace{1cm} (1.4)

else

\[ U_i = U_i - U_i * P_i \]  \hspace{1cm} (1.5)
\[ Q_j = Q_j + (1 - Q_j) \times (1 - P_i) \]  \hspace{1cm} (1.6)

1.2 Scenarioes for User Ui

1. if the users is highly expected to answer correctly (pi is larger, so 1-pi is smaller), then he/she answers correctly. no surprise, add little intelligence to his bucket

\[ U_i + (1 - U_i) \times (1 - P_i) \]

2. if the users is not expected to answer correctly (pi is smaller, so 1-pi is larger), then he answers correctly, it is a surprise, add intelligence parameter is tuned much

\[ U_i + (1 - U_i) \times (1 - P_i) \]

3. if the users is highly expected to answer correctly (pi is larger, so 1-pi is smaller), then he answers wrongly, it is surprise, penalize intelligence parameter

\[ U_i - (U_i) \times (P_i) \]

4. if the users is not expected to answer correctly (pi is smaller, so 1-pi is larger), then he answers wrongly, no surprise, penalize little

\[ U_i - (U_i) \times (P_i) \]

1.3 Scenarioes for User Qj

1. if the users is highly expected to answer correctly (pi is larger, so 1-pi is smaller), then he/she answers correctly. no surprise, add little intelligence to his bucket

\[ Q_j = Q_j - Q_j \times P_i \]

2. if the users is not expected to answer correctly (pi is smaller, so 1-pi is larger), then he answers correctly, it is a surprise, add intelligence parameter is tuned much

\[ Q_j = Q_j - Q_j \times P_i \]

3. if the users is highly expected to answer correctly (pi is larger, so 1-pi is smaller), then he answers wrongly, it is surprise, penalize intelligence parameter
\[
Q_j = Q_j + (1-Q_j)*(1-P_i)
\]

4. if the user is not expected to answer correctly (pi is smaller, so 1-pi is larger), then he answers wrongly, no surprise, penalize little

\[
Q_j = Q_j + (1-Q_j)*(1-P_i)
\]

1.4 Tree Representation of Four Cases Discussed for Qj and Ui

According to responses of user Ui(user intelligence) score will change for that particular user. Same in case of Qj(question difficulty score) will change as responses from user collected. Score of Qj and Ui normalised in equations so that value will not fall outside. Figure 1.1 shows the tree representation of scenarios discussed for QJ and Ui.

\[0 < value \leq 1\] (1.7)

![Tree Representation of Four Cases Discussed for Qj and Ui](image)

Figure 1.1: sequence of correctness

Figure 1.1 gives insight into scenarios explained in section 1.2 and 1.32, where all the four cases represented as tree structure.

1.5 Examples User Attempting Questions

Case1: let \(U_i=0.5\), \(Q_j=0.7\) and \(P_i=0.75\) then
if (correct)

\[ U_i = 0.5 + (1 - 0.5)(1 - 0.75) = 0.625 \]

\[ Q_j = 0.7 - 0.7 \times 0.75 = 0.175 \] \hspace{1cm} (1.8)

else

\[ U_i = 0.5 - 0.5 \times 0.75 = 0.125 \] \hspace{1cm} (1.9)

\[ Q_j = 0.7 + (1 - 0.7) \times (1 - 0.75) = 0.775 \] \hspace{1cm} (1.10)

Case 2: let \( U_i = 0.5, Q_j = 0.7 \) and \( P_i = 0.25 \)
then
if (correct)

\[ U_i = 0.5 + (1 - 0.5)(1 - 0.25) = 0.875 \] \hspace{1cm} (1.12)

\[ Q_j = 0.7 - 0.7 \times 0.25 = 0.575 \] \hspace{1cm} (1.13)

else

\[ U_i = 0.5 - 0.5 \times 0.25 = 0.375 \] \hspace{1cm} (1.14)

\[ Q_j = 0.7 + (1 - 0.7) \times (1 - 0.25) = 0.925 \] \hspace{1cm} (1.15)

As we see, all the values are normalised, and scenarios for \( Q_j \) and \( U_i \) are giving exact defined results.
References
