Algoritme

Dit algoritme beschrijft een manier om vragen uit een ELO te suggereren aan gebruikers afhankelijk van de moeilijkheid van de vragen en het niveau van de gebruiker. Het programma moet hiermee gebruikers binden aan het systeem. Dit door vragen aan te bieden die aansluiten bij het niveau van de gebruiker alsmede een ranglijst bij te houden van gebruikers onderling.

Om de moeilijkheid van vragen te bepalen meten we hoelang het gemiddeld duurt om een correct antwoord te geven op de vraag. Om gokken niet te laten lonen krijgt men ook een aanzienlijke tijdstraf bij ieder verkeerd gegeven antwoord. De score (het niveau) van een gebruiker is de gemiddelde tijd dat het hem (inclusief straftijd) gekost heeft om de vragen die hij heeft gedaan correct te beantwoorden. Het systeem houdt ook een totaal gemiddelde bij. Dit is de gemiddelde tijd dat het álle gebruikers heeft gekost om de vragen die ze hebben gedaan correct te beantwoorden (inclusief straftijd). Als een gebruiker hoger dan gemiddeld scoort (een kortere tijd dan gemiddeld als score heeft), krijgt hij vragen aangeboden waar men evenredig langer over heeft gedaan dan gemiddeld. Andersom krijgt een gebruiker die gemiddeld langer dan gemiddeld over zijn vragen heeft gedaan, vragen aangeboden waar men evenredig korter over heeft gedaan. De tijdscore van aangeboden vragen noemen we de ‘tegengestelde tijd’ en dit wordt als volgt berekend:

Tegengestelde tijd = 2x het totaal gemiddelde – het gemiddelde van de gebruiker

De gebruiker krijgt de vraag aangeboden die het dichtst bij de uitkomst van deze formule ligt. Vragen die hij al eerder heeft gemaakt worden hierbij overgeslagen. Hij mag dus niet zelf kiezen welke volgende vraag hij gaat doen. Als het totaal gemiddelde verandert krijgt hij mogelijk wel een andere vraag aangeboden door het systeem. Vragen waarbij geen antwoord gegeven wordt of waarbij de gebruiker een pauze neemt worden niet meegerekend. Je kunt iedere vraag ook maar één keer starten.

De straftijd bij ieder verkeerd gegeven antwoord wordt berekend aan de hand van de volgende formule:

Straftijd = gemiddelde tijd dat men over de vraag doet / gemiddeld aantal foute antwoorden bij die vraag

Dit betekent dat als je een gemiddeld aantal foute antwoorden geeft, dat je straftijd dan de gemiddelde tijd is dat men over de vraag doet. Je moet dus in ieder geval minder foute antwoorden geven dan gemiddeld om op een bovengemiddelde tijd uit te komen. Als iemand met een onder gemiddeld aantal fouten een bovengemiddelde tijd scoort loopt de straftijd op (want gemiddeld aantal fouten loopt terug en gemiddelde tijd loopt op). De straftijd loopt terug als iemand een onder gemiddeld aantal fouten maakt en als zijn tijd minder invloed heeft op de gemiddelde tijd voor de vraag dan zijn aantal fouten op de gemiddelde foutenlast van de vraag.

Als een vraag voor het eerst gemaakt wordt bestaat er nog geen gemiddelde tijd en geen gemiddeld aantal foute antwoorden. De gemiddelde tijd wordt dan het totaal gemiddelde maar als deze ook nog niet bestaat voert de docent een schatting in hoelang de vraag moet duren. Het gemiddelde aantal ‘foute antwoorden’ wordt bij een nieuwe vraag door de volgende formule berekend:

Gemiddelde aantal foute antwoorden = (Aantal antwoorden – 1) / 2

Bij een vraag met 4 mogelijke antwoorden is de gemiddelde foutenlast (als men gokt) volgens bovenstaande formule 1.5. Dit klopt want men geeft (met een gelijke kans) 0, 1, 2 of 3 verkeerde antwoorden alvorens een goed antwoord te geven. Het gemiddelde hiervan is 1.5. De kans om bij ieder aantal verkeerde antwoorden uit te komen is gelijk (1/4e bij 4 mogelijke antwoorden, 1/3e bij 3 mogelijke antwoorden, 1/5e bij 5 mogelijke antwoorden enz). Dit kun je narekenen. Vandaar bovenstaande formule.

Nadat een vraag voor het eerst is gemaakt wordt de hierbij gescoorde tijd en het gescoorde aantal fouten bij een volgend antwoord op de vraag gebruikt om de straftijd te berekenen. De gemiddelde begintijd of de tijd die door de docent is ingevuld én de gemiddelde foutenlast worden dus maar één keer gebruikt (alleen bij de eerste beantwoording van de vraag).

Het systeem suggereert altijd eerst vragen die nog door niemand zijn gemaakt, alvorens de tegengestelde tijd te gaan gebruiken.

**Oude opzet:**

De tijdstraf is een factor waarmee je verlopen tijd vermenigvuldigd wordt op het moment dat je een verkeerd antwoord geeft. Deze factor wordt berekend door de volgende formule: 1 + verwachte tijd/verlopen tijd. De verwachte tijd is de gemiddelde tijd die andere gebruikers hebben gescoord voor de vraag. Bij vragen die nog niet eerder zijn beantwoord is dit een constante of een waarde opgegeven/ingeschat door de docent. Na het toepassen van de tijdstraf wordt de nieuwe verlopen tijd (incl. tijdstraffen) in deze formule gebruikt bij elk volgend verkeerd antwoord. Als je op de verwachte tijd een fout antwoord geeft is je factor 2 (dit is de gemiddelde factor van deze grafiek). Vooral aan het begin is deze factor relatief erg hoog, hierdoor heeft gokken dus geen zin. Na de verwachte tijd komt deze factor echter steeds dichter bij één, dus dan wordt je straf relatief steeds draaglijker bij het geven van een laat fout antwoord. Geen antwoord geven resulteert in een erg lang gescoorde tijd. Verder kun je iedere vraag maar één keer maken/starten.

De gemiddelde tijd die je voor gemaakte vragen hebt gescoord wordt je score ten opzichte van anderen. Ook wordt bij iedere vraag uitgerekend/bijgehouden hoe lang er gemiddeld op deze vraag gescoord is. Verder houdt het systeem een totaal gemiddelde bij, dit is de gemiddelde score voor álle correct beantwoorde vragen. Aan de hand van deze gegevens wordt je volgende vraag voor je uitgekozen door het systeem. Je mag dus niet zelf een vraag uitkiezen. Het systeem berekent het verschil tussen jouw gemiddelde en het totaal gemiddelde. Als je gemiddeld 2 minuten lager staat dan totaal gemiddeld, dan krijg je de vraag waarbij het gemiddelde het dichtst bij 2 minuten hoger dan het algehele gemiddelde ligt. Andersom werkt dit op dezelfde manier. Vragen die je al hebt gemaakt worden niet gesuggereerd. Hierdoor word je altijd uitgedaagd om je snelle tijd waar te (blijven) maken (of verbeteren/verstevigen) in moeilijkere vragen of je slechte tijd te verbeteren via makkelijkere vragen. Als het gemiddelde van vragen verandert of het totaal gemiddelde verandert, dan verandert mogelijk ook de vraag die je aanbevolen krijgt door het systeem.

Handleiding

Voor dit programma hebben we MySQL gebruikt in plaats van SQLite (deze laatste hebben we eerst gebruikt, de code staat nog in een file). Dit omdat deze database op het web veel gebruikt wordt en we dit dus wilden leren gebruiken. U moet dus MySQL eerst installeren op uw computer alvorens u het programma kunt gebruiken. Download MySQL hier: <https://dev.mysql.com/downloads/> . Het root wachtwoord wat u moet instellen tijdens de installatie staat bovenaan in de code (in Main.py). Ook bovenaan in de code staat de ingeschatte tijdsduur in seconden voor de eerste vraag die gemaakt wordt. Dit mag u wijzigen maar hij staat standaard ingesteld op 5 minuten (300 seconden). Daarbij staat ook een vraagnummer ingesteld voor de vraag van gemiddelde moeilijkheid die hoort bij de geschatte tijdsduur. Deze vraag is de eerste vraag die door een gebruiker gemaakt gaat worden.

De database tables kunt u bekijken en aanpassen met HeidiSQL. U kunt HeidiSQL downloaden en installeren via <https://www.heidisql.com/download.php>. Hiermee kunt u bijvoorbeeld vragen toevoegen of wijzigen. U kunt hier zowel de input als de output van het programma nagaan. Maak in HeidiSQL eerst een nieuw profiel aan voor MySQL en voer het root wachtwoord in wat bovenaan in de code staat (andere instellingen hoeven niet veranderd te worden).

De sample databases zijn bijgevoegd als bijlage in de repository. U moet deze databases eerst met HeidiSQL importeren via File > Run SQL file (daarna eventueel refresh). De SQL code in deze sample files overschrijft bestaande databases met dezelfde naam. Vul vervolgens bovenin de code de naam in van de database in waarmee u wilt verbinden. Dit is de inhoud van de sample databases:

* sample1: Lege database met 4 vragen erin (het antwoord is de vraag zelf; de vraag geeft dus het juiste antwoord aan).

Testrapport

Het volgende is door middel van systeemtesten getest:

* Correct database schema wordt aangemaakt als de database met tables nog niet bestaan
* Nieuwe gebruiker wordt toegevoegd bij opgeven nieuwe username
* Eerste gebruiker krijgt default values aangeboden bij de eerste vraag die gemaakt wordt
* Correcte penalty en avgnumofpenalties bij eerste vraag vermeld
* Programma vraagt om het antwoord en de verlopen tijd alvorens antwoorden te verwerken
* Bij een verkeerd antwoord krijgt gebruiker de kans om nog een antwoord te geven, straftijd en verlopen tijd bij een verkeerd antwoord worden onthouden
* Bij het geven van het correcte antwoord stopt het programma en wordt de correcte totale tijd (incl. penalties) toegevoegd aan de database (zowel bij de gebruiker als bij de vraag).
* Niet eerder gemaakte vragen worden altijd eerder aangeboden dan eerder gemaakte vragen
* Bij het verdergaan onder een bestaande gebruikersnaam wordt zijn score correct teruggevonden
* Als er vragen gemaakt zijn wordt de totalaveragetime hiervan ook gebruikt bij vragen die nog niet eerder gemaakt zijn.
* Bij niet eerder gemaakte vragen wordt avgnumofpenalties 1.5 steeds toegepast
* Als een gebruiker meerdere vragen maakt wordt zijn gemiddelde correct herberekend aan de hand van zijn oude gemiddelde en zijn nieuwe score voor de nieuw gemaakte vraag.
* Als een vraag meermaals gemaakt wordt, dan wordt de correcte gemiddelde tijd voor deze vraag steeds correct uitgerekend (aan de hand van oude gemiddelde en nieuwe score) en ingevoerd in de database.
* Als alle vragen minstens één keer zijn gemaakt suggereert het systeem de vraag het dichtst bij de oppositetime van de gebruiker
* Als een gebruiker alle vragen heeft gemaakt zegt het systeem dat en sluit het programma
* Zolang er geen fouten zijn gemaakt bij een vraag (en er dus geen avgnumofpenalties berekend kan worden) is avgnumofpenalties 1.5
* Als er fouten zijn gemaakt bij een vraag wordt bij deze vraag de correcte avgnumofpenalties uitgerekend en gebruikt om de straftijd te berekenen