

Project template

Matjaž Pogačnik

faculty of computer and information science
večna pot 113
1000 ljubljana

Abstract. Kratak opis ideje

Key words: eQuiz, rating

1 INTRODUCTION

Ena izmed funkcionalnosti aplikacije eQuiz je rating študentov in nalog preko Elo rating sistema. Študent se sooči z nalogo in na ta način se iztočasno izračuna rating študenta in rating naloge, kot bi igrala študent in naloga šah. Študent lahko na nalogo odgovori prav ali napačno, kar je ekvivalentno temu, da študent naloge premaga ali proti njej izgubi. Pojavijo pa se pomankljivosti takega rating sistema.

Če uporabljamo eQuiz za spremljanje ratinga študentov in nalog v sklopu določenega predmeta, bo proti številu študentov, ki bodo v tem letu aktivni, število nalog navadno mnogo večje. To pa pomeni, da bodo, dokler študentje ne rešujejo samih nalog, ti prejeli posodobitev ratinga veliko večkrat kot kot posamezna naloga. Še več posodobitev pa študentje, ki so pri reševanju nalog bolj aktivni. Torej ratingini, pripisani določeni nalogi ali študentu lahko izvirajo iz več ali manj reševanj, kar naredi rating bolj ali manj verodostojen. Zaupanje v določen rating lahko tako iz posamezne številke raje razširimo na interval zaupanja.

2 GLICKO RATING

Za tak pristop je primeren Glicko rating, ki poleg ratinga za posameznega igralca (študent ali naloga) vpeljuje še deviacijo RD , ki nam omogoča predstavitev ratinga posameznega igralca kot 95% interval zaupanja:

$$(r - 1.96RD, r + 1.96RD) \quad (1)$$

pri čemer se RD manjša ob vsakem updatu ratinga, kjer se igralec sooči z drugim igralcem (ali več njih)

$$RD' = \sqrt{\left(\frac{1}{RD^2} + \frac{1}{d^2}\right)^{-1}} \quad (2)$$

kjer RD predstavlja deviacijo pred posodobitvijo, d^2 pa je definiran kot

$$d^2 = \left(q^2 \sum_{j=1}^m (g(RD_j))^2 E(s|r, r_j, RD_j) (1 - E(s|r, r_j, RD_j)) \right)^{-1} \quad (3)$$

$$q = \frac{\ln 10}{400} = 0.0057565 \quad (4)$$

$$E(s|r, r_j, RD_j) = \frac{1}{1 + 10^{-g(RD_j)(r-r_j)/400}} \quad (5)$$

$$g(RD) = \frac{1}{\sqrt{1 + 3q^2(RD^2)/\pi^2}} \quad (6)$$

Nov rating r' se izračuna po formuli

$$r' = r + \frac{q}{1/RD^2 + 1/d^2} \sum_{j=1}^m g(RD_j) (s_j - E(s|r, r_j, RD_j)) \quad (7)$$

Zgornje formule so posplošene za posodobitev ratinga r in deviacije RD igralca proti skupini m nasprotnikov z deviacijami RD_1, RD_2, \dots, RD_m in ratingi r_1, r_2, \dots, r_m . s_1, s_2, \dots, s_m predstavljajo izide, ki so lahko 0 ali 1, za izgubo ali zmago.

Igralčev RD se ne posodablja samo, ko se ta sooča, temveč tudi ob preteku določene časovne periode, kar predstavlja zniževanje verodostojnosti trenutnega ratinga, če igralec določen čas ne igra. Tako se njegov RD posodobi kot

$$RD = \min \left(\sqrt{RD_{old}^2 + c^2}, 350 \right) \quad (8)$$

konstanta c je, poleg maksimalne deviacije, ki je v tem primeru 350, edini parameter, ki ga nastavlja administrator sistema. Več o tem v kontekstu equiza v poglavju X

Matematične izpeljave formul so na voljo na tukaj

Iz formul vidimo, da je r' odvisen od RD igralca in ni uravnotežen tako, kot je pri Elo sistemu, kjer se zmagovalcu zviša rating toliko, kot se poražencu zniža. Ker naj bi velikost RD odražala koliko informacije imamo o igralcu, se v iteraciji ocenjevanja, kjer ima en igralec velik RD , drug pa majhen, prvemu zviša veliko več, kot se drugemu zniža, saj vemo, da je rating prvega nezanesljiv, rating drugega pa je. Enako se v tem primeru ne more upravičeno znižati rating drugega igralca toliko, kot bi se, če bi bil rating njegovega soigralca zanesljiv.

3 IZRACUNI IZ EQUIZ PODATKOV

Če študent reševal skozi semester izgleda rating po Elo tako:

(Če pa nimamo rednih podatkov): zgenerirana tabela reševanja po kolokvijih, sprotnih kvizih in random reševanjih. Na random ratingih nalog zajetih iz equiz ali pa ce je kaj informacije o tem.

iz zgornjih tabel se da ugotavljati kasnejše ugotovitve napisane spodaj

(morda) imamo informacijo samo o izpitu. (primerjava ratinga Elo vs Glicko na izpitu, tudi za naloge)

Če

Trenutni podaki (lahko) baza za nove,

Po rating updatih lahko določimo cas (in verjetno kaj je ucenec reseval) – match lahko uporabimo kar prvi rating in matchamo casovno naslednje preizkuse

Ali imamo podatke naključnih reševanj? (Nejc Subic diplomska: Da?) Če ne lahko tudi umetno prikazemo ucinek za sprotno delo – naključno izberem nekaj nalog(lahko tudi randomly generiram podobne podatke kot trenutne naloge v equizu).

- Prikaz ratingov po izpitu (tudi za naloge). kot poseben primer za prikaz da je se vedno accurate – izstopanje nalog
- Prikaz ratingov po rednih preverjanjih, kot prikaz kako se vzdržuje rating in devianca(graf deviance za studenta) – c samo pokaze koliko globoko bo dipnilo med preverjanji in s tem koliko pomembno je sprotno delo?

Zeljeni podatki:

- izpit (sasa)
- cas posodobitev ratingov + kaj je ucenec takrat reseval (za sestavljene naloge (izpite) + naključno reševanje)

S tem pridobimo v primeru študentov informacijo o sprotne delu, saj bo RD manjši za učence, ki so pogostejše ocenjeni (poleg opravljenih sprotne preverjanj še naključna reševanja kvizov/nalog), kar kaže na več/manj sprotne dela. (Glede na spreminjanje RD lahko spremljamo delo učenca kot nekakšen koeficient napredka/ak.vnos.) Za nasprotnika učencu – nalogo, pa deviacija predstavlja kdaj je bila naloga nazadnje ocenjena, torej koliko je njena ocena zanesljiva. V primeru učenca, ki izbira naloge iz zbirke na equizu, pa lahko naloge z veliko deviacijo idenfificiramo kot nepriljubljene (malo učencev se lo. naloge). Reciklirane naloge bodo torej tudi večkrat nastopile na preverjanjih in bo natančnost njenih rangov večja (dobro za parametrizirane naloge, ker se zaradi njihove narave lahko večkrat pojavijo v preverjanjih) Ra.ng nalog, ki se pojavijo na izpitu, pa bo še vedno zanesljiv, saj bo v is. iteraciji nalogo reševalo veliko učencev. Ra.ng učenca/naloge, ki je neak.ven čez določen časovni interval se ustrezno zmanjša, kar je poleg maksimalne deviacije edini parameter, ki ga je potrebno nastavi. kot administrator. Kako sistema.čno določamo ta parameter je opisano v dodatnem članku o glicko, vendar ga lahko verjetno večemo na redna preverjanja kot npr. kolokvije. Ker se da predmet opravi. samo s kolokviji celotno snov sta.s.ke razdelijo na 5 delov. Torej bo študent, ki opravi samo eno preverjanje od pe.h ustrezno imel manjši ra.ng predvsem pa večjo deviacijo (če je opravljal samo prvega se bo njegov RD š.rikrat zviševal do konca semestra), saj imamo informacijo o njegovem znanju čedalje bolj nezanesljivo, ker vemo čedalje manj koliko študent dejansko ve o sta.s.ki odkar je reševal .s. kolokvij (pričakuje se, da študent tekom leta ve več in več o sta.s.ki).

Lemma 3.1. $\ell = \varepsilon + t(p-1)(q-1)$.

Proof.

$$\begin{aligned}\ell &= \varepsilon + r(q-1) \\ &= \varepsilon + t(p-1)(q-1)\end{aligned}$$

■

Theorem 3.2. $\ell \equiv \varepsilon \pmod{\varphi(n)}$.

Proof. By Lemma 3.1 and (??) we have $\varepsilon \bmod \varphi(n) = \ell$.

■

Corollary 3.3. $\exists k \in \mathbb{Z} : \ell - \varepsilon = k\varphi(n)$.

Proof. Follows by Theorem 3.2.

■

4 CONCLUSION

conclusion