

PROJEKTNA DOKUMENTACIJA

# **Umjetna inteligencija u adaptivnom učenju**

*Jelena Nemčić, Marin Ovčariček, Zvonimir Sučić, Ivana Žeger*

Voditelj:

Zagreb, kolovoz 2020.

# 1. Uvod

Adaptivno učenje u e-learning softverima tipično funkcionira tako da mjeri razinu znanja korisnika kroz početni skup pitanja, virtualnu simulaciju i/ili dodijeljene zadatke. Na temelju podataka prikupljenih iz odgovora korisnika, takav softver u stvarnom vremenu procjenjuje razliku između korisničkog znanja i znanja potrebnih za određenu kompetenciju, te odabire lekcije i zadatke za korisnika tako da minimizira količinu edukacijskog sadržaja koji tom korisniku prikazuje.

Konstrukcija algoritma za određivanje puta za adaptivno učenje tipično se radi na dva načina: 1) kreiranjem formalnog modela znanja za određenu domenu, a koji kreiraju eksperti iz te domene ili 2) koristeći algoritamski pristup baziran na teorijama Bayesian Knowledge Tracing i Item Response Theory koji na temelju odgovora polaznika edukacije procjenjuje vjerojatnost da je polaznik usvojio određenu vještinu/koncept (što ponovno zahtijeva unaprijed definirane vještine/koncepte).

U posljednje vrijeme, a s obzirom na dostupnost sve većih količina podataka (big data) ponovno su oživjele i dodatno se razvijaju tehnologije kreiranja grafova znanja (npr. pomoću dubokog učenja), a koji s obzirom na to da su kreirani statistički mogu biti mnogo kompleksniji i uže segmentirani (precizniji) u odnosu na one koje kreiraju eksperti nekog područja. Dodatno, prikupljanje velike količine informacija u različitim domenama omogućuje da kreiranje grafova znanja ne bude ograničeno samo na one tvrtke koje imaju golem broj korisnika kao što su Google ili Facebook). Na temelju inicijalnog istraživanja vjerujemo da se ova metoda može primijeniti i na kreiranje grafova znanja za adaptivno učenje, te time omogućiti s jedne strane znatno veću adaptivnost, a s druge strane veću jednostavnost kreiranja takvih grafova.

Ovo je posebno važno za područja edukacije izvan formalnog obrazovanja, gdje nisu strogo definirane ishodi učenja i testovi kojima se mjeri je li neki ishod učenja dostignut kod pojedinog polaznika. Dva primjera za to su instrukcije, gdje učeniku često nedostaju i predznanja iz drugih područja koje je ranije u školi trebao usvojiti, te korporativne edukacije, koje često obuhvaćaju ljude različitih struka i različitim znanjima iz domene za koju nastoje dobiti certifikat.

## 2. Cilj i postupak

Stoga je cilj ovog projekta provjeriti sljedeće:

- Provjera mogućnosti kreiranja grafa znanja isključivo na temelju točnosti odgovora na zadacima iz jedne domene znanja koje korisnici daju i informacije o redoslijedu zadavanja zadataka pojedinom korisniku
- Ako je navedeno moguće, potrebno je provjeriti može li se kreirati graf znanja na temelju rješavanja zadataka za istu domenu na temelju parcijalnog broja zadataka (što realnije reprezentira dostupne zadatke za stvarne domene – rijetko su dostupna baš sva znanja iz neke domene da bi se mogla kreirati zadaci koji pokrivaju baš svaku informaciju u toj domeni)
- Ako je navedeno moguće, potrebno je provjeriti može li se isto napraviti i za neku domenu realnog znanja

Ukratko, ovim se projektom provjerava može li se graf znanja potreban za adaptivno učenje kreirati metodama dubokog učenja na temelju ponašanja korisnika na zadacima (probabilistički), a bez potrebe za time da eksperti unaprijed određuju koncepte/vještine u koje se grupiraju zadaci ili čak sam graf znanja.

Za potrebe ove provjere kreirat će se:

- umjetni, zatvoreni graf znanja
- zadaci koji pokrivaju sve informacije prisutne u tom zatvorenom grafu znanja

Potom će se zadaci dati testerima na rješavanje, tako da se:

- varira redoslijed zadataka koje pojedini tester dobiva kako bi pokrio sve kombinacije
- bilježi točnost odgovora testera na zadatak
- u slučaju netočnog odgovora testeru se prikazuje točna informacija.

## 3. Bayesian knowledge tracing (BKT)

### 3.1. Općenito o BKT

Bayesian Knowledge Tracing koristi Hidden Markov Model i ima 4 osnovna parametra:

- $p(L_0)$  - vjerojatnost da je korisnik a priori savladao gradivo
- $p(G)$  - vjerojatnost da je korisnik pogodio točan odgovor bez da ima potrebno znanje
- $p(S)$  - vjerojatnost da je korisnik krivo odgovorio iako ima potrebno znanje
- $p(T)$  - vjerojatnost da je znanje prešlo iz NE ZNA u ZNA nakon prilike da se primjeni znanje

Kao izlaz dobivaju se vrijednosti:

- $p(L)$  - vjerojatnost ovladavanja vještinom (eng. probability of skill mastery)
- $p(C)$  - vjerojatnost da će korisnik ispravno primijeniti vještinu u budućnosti (eng. probability of the student correctly applying the skill on a future practice)

$$p(L_t \mid obs = correct) = \frac{p(L_t) * (1 - p(S))}{p(L_t) * (1 - p(S)) + (1 - p(L_t)) * p(G)} \quad (3.1)$$

$$p(L_t \mid obs = wrong) = \frac{p(L_t) * p(S)}{p(L_t) * p(S) + (1 - p(L_t)) * (1 - p(G))} \quad (3.2)$$

$$p(L_{t+1}) = p(L_t \mid obs = correct) + (1 - p(L_t \mid obs = correct)) * p(T) \quad (3.3)$$

$$p(C_{t+1}) = p(L_{t+1}) * (1 - p(S)) + p(L_{t+1}) * p(G) \quad (3.4)$$

## 3.2. Ideje

Prvobitna ideja je bila da se  $p(L0)$  računa iz inicijalnih pitanja, vrijednosti  $p(G)$  i  $p(S)$  bi se prema preporuci iz rada (trebalo bi pronaći kojeg i baciti referencu) stavile na interval  $[0,0.3]$ ,  $[0,0.1]$  te bi se  $p(T)$  postavio prema preporuci eksperta što ne želimo jer je cilj ovog projekta da smanjimo zadatke eksperata na minimum.

To je ukazalo na potrebu pronalaska algoritama koji bi uz pomoć nekog skupa podataka aproksimirali parametre za BKT.

## 3.3. Problemi i zadaci

- proučiti parametar  $p(T)$
- proučiti kodove sa githuba kako bi se dobila ideja kako algoritam funkcionira
- napraviti malu implementaciju s malo pitanja i provjeriti radi li
- proučiti parameter fitting uz pomoć EM algoritma, stochastic gradient descenta ili neke druge metode
- kako napraviti input dataset, prikupiti podatke

## 3.4. Dobivanje BKT parametara

### 3.4.1. EM (expectation-maximization) algoritam

- iterativni algoritam za pronalaženje (aproksimiranje) najveće izglednosti (eng. maximum likelihood) ili maksimalne a posteriori (MAP) procjene parametara u statističkim modelima
- model ovisi o nepoznatim latentnim varijablama
- EM iteracija sadrži 2 koraka:
  - korak očekivanja (E), koji stvara funkciju za očekivanje log-izglednosti koja se procjenjuje pomoću trenutne procjene parametara, procjenjuju se vrijednosti latentnih varijabli
  - korak maksimizacije (M), koji izračunava parametre distribucije koji maksimiziraju očekivanu log-izglednost pronađenu u E koraku, ti se parametri zatim koriste za procjenu latentnih varijabli u sljedećem E koraku

- primjenjuje se kada želimo odrediti parametre distribucije (normalna, eksponencijalna, ...)
- problem: za korištenje potrebo znati distribuciju podataka ili točne vrijednosti (eng. true values) traženih parametara

Kroz ovo istraživanje nije pronađena niti jedna implementacija EM algoritma za aproksimaciju BKT parametara niti je napravljena vlastiti implementacija zbog prevelikog praga znanja matematike.

### 3.4.2. Grid search i Simulated Annealing

Pronađen je kod napisan u Javi koji računa BKT parametre tehnikom simuliranog kaljenja <https://github.com/wlmiller/BKTSimulatedAnnealing>. U README na githubu se također spominjao kod koji je bio baza za to, on je koristio običan grid search kako bi izračunao parametre. Oba koda su prevedena u python i prilagođena našim skupovima podataka. Na kraju se ispostavilo da je "simulirano kaljenje" povoljnije te se grid search odbacio.

## 3.5. Rezultati

- napravljen google forms kviz sa 20 pitanja iz biologije, ispitanici moraju odgovoriti na svih 20 pitanja kako bi podaci ušli u dataset
- napravljena python skripta koja pretvara podatke dobivene iz google formsa u oblik prikladan za treniranje BKT-a i pronalaženje parametara
- pronađen je kod u Javi koji tehnikom simuliranog kaljenja aproksimira parametre za BKT uz pomoć danog dataseta, kod je preveden u python skript
- napravljena python skripta za BKT koja određuje vjerojatnost da je ispitanik naučio/ savladao gradivo
- uz pomoć skripte za aproksimaciju BKT parametara, nađene su njihove vrijednosti za svaku vještinu iz ASSISTMENTS dataseta i pohranjenje u google sheets tablicu
- dobiveni parametri algoritmom simuliranog kaljenja uspoređeni su s onima dobivenima pomoću grid search metode -> vrijednosti parametara su skoro iste, vrlo male razlike
- napravljen google forms kviz sa po 6 pitanja iz 5 koncepata, izračunati su parametri za taj dataset

- BKT kod i kod za aproksimaciju BKT parametara su se dalje koristili u bilježnicama za izgradnju grafa probabilističkim metodama

## 3.6. Poveznice

### 3.6.1. BKT

[https://en.wikipedia.org/wiki/Bayesian\\_Knowledge\\_Tracing](https://en.wikipedia.org/wiki/Bayesian_Knowledge_Tracing)  
<http://www.cs.cmu.edu/~./ggordon/yudelson-koedinger-gordon-individualized-bayesian-knowledge-tracing.pdf>  
<https://github.com/CAHLR/pyBKT/blob/master/README.md>  
<https://www.learnlab.org/uploads/mypslc/publications/bca2008v.pdf>  
[https://www.upenn.edu/learninganalytics/ryanbaker/paper\\_143.pdf](https://www.upenn.edu/learninganalytics/ryanbaker/paper_143.pdf)  
<https://github.com/yemao616/Bayesian-Knowledge-Tracing>  
<http://www.cs.cmu.edu/~./ggordon/yudelson-koedinger-gordon-individualized-bayesian-knowledge-tracing.pdf>  
<https://www.fi.muni.cz/~xpelanek/publications/umuai-overview.pdf>  
<https://medium.com/@joyboseroy/modelling-a-students-learning-34375b0131dd>  
[https://www.math.vu.nl/~sbhulai/publications/data\\_analytics2018c.pdf](https://www.math.vu.nl/~sbhulai/publications/data_analytics2018c.pdf)

### 3.6.2. Pronalaženje parametara

<https://www.fmrib.ox.ac.uk/datasets/techrep/tr00yz1/tr00yz1/node9.html>  
[https://github.com/wlmiller/BKTSimulatedAnnealing/blob/master/computeKTparams\\_SA.java](https://github.com/wlmiller/BKTSimulatedAnnealing/blob/master/computeKTparams_SA.java)  
[https://www.upenn.edu/learninganalytics/ryanbaker/paper\\_143.pdf](https://www.upenn.edu/learninganalytics/ryanbaker/paper_143.pdf)  
[https://educationaldatamining.org/files/conferences/EDM2018/papers/EDM2018\\_paper\\_14.pdf](https://educationaldatamining.org/files/conferences/EDM2018/papers/EDM2018_paper_14.pdf)

<http://yudelson.info/hmm-scalable/>  
<https://www.educationaldatamining.org/EDM2015/proceedings/short364-367.pdf>  
<https://concord.org/wp-content/uploads/2016/12/pdf/tracking-student-progress-in-a-game-like-learning-environment.pdf>  
<https://tinyheero.github.io/2016/01/03/gmm-em.html>  
<https://machinelearningmastery.com/expectation-maximization-em-algorithm/>  
[http://rstudio-pubs-static.s3.amazonaws.com/1001\\_3177e85f5e4840be840c84452780db52.html](http://rstudio-pubs-static.s3.amazonaws.com/1001_3177e85f5e4840be840c84452780db52.html)  
[https://www.colorado.edu/amath/sites/default/files/attached-files/em\\_algorithm.pdf](https://www.colorado.edu/amath/sites/default/files/attached-files/em_algorithm.pdf)



## 4. Bayes graf

### 4.1. Uvod

Ideja je napraviti vlastiti model koji iz skupa podataka računa vjerojatnost savladavanja koncepata te se prema njima gradi graf znanja.

Prvi pristup je račun uvjetnih vjerojatnosti  $p(Y_j | X_i)$ , odnosno vjerojatnost da korisnik zna (savladao je) koncept Y ako zna X. Koristi se klasična formula Bayesove uvjetne vjerojatnosti čime se gradi matrica odnosa koncepata.

$$p(Y | X) = \frac{p(X \wedge Y)}{p(X)} \quad (4.1)$$

Vjerojatnost Y uz X nam govori koliko je poznavanje X bitno za poznavanje Y. Pretpostavka je da će korisnik s većom vjerojatnošću znati jednostavniji koncept ako zna kompliciraniji (nadogradnju jednostavnog), ako su X uz Y i Y uz X podjednaki (potreban je neki prag) može se pretpostaviti da su koncepti nezavisni ili bi se mogli spojiti kao jedan koncept.

Drugi pristup je bio pokušaj da se gleda kad su koncepti X i Y točni s obzirom koji od njih je položen prvi, pretpostavka je bila da bi se tako mogla odrediti relacija nadogradnje, odnosno prethodnika. Pristup nije uspio jer ili nije moguć ili nije bio dobro definiran / implementiran.

Najveći problem u oba pristupa je to što pokušavamo dobiti ovisnosti među konceptima, a ne pitanjima. Za razliku od pitanja za koncepte se ne može reći da ih se zna/ne zna jer se oni sastoje od više pitanja te se može samo gledati postotak riješenosti za svakog studenta / prosječan postotak riješenosti. Prosječan postotak rješenosti se može gledati kao pripadnost neizrazitom skupu ZNA odnosno 1- postotak riješenosti kao pripadnost skupu NE ZNA, to komplicira stvari kod Bayesovog zaključka. Potrebno je pronaći/proučiti postoji li ekvivalent Bayesovog zaključka kada se koriste neizraziti skupovi odnosno kontinuirane vrijednosti  $[0,1]$ .

## 4.2. Problem kontinuiranih vrijednosti

Običan Bayesov zaključak radi sa diskretnih vrijednostima (true/false, 1/0). Dok je to istina kod izračuna vjerojatnosti i točnosti pitanja, isto se ne može reći za koncepte. Može se gledati da je koncept položen s nekom vjerojatnošću ili da određen pokušaj ispita pripada skupo "položen" s nekom pripadnošću, to je obično postotak točnih bodova i to je racionalan broj. Iako postoji Bayesov zaključak za kontinuirane vrijednosti nije jednostavan za izračunati jer se teško mogu dobiti latentne razdiobe bodova za neku populaciju. Pošto je Bayes kontinuiranoj domeni prekomplikiran za računati potražena je alternativa te se došlo do zaključka da se treba nekako vratiti u diskretnu domenu, odnosno napraviti model koji može odlučiti kad neki koncept je ili nije položen.

Najjednostavnije rješenje je ručno za svaki koncept zadati bodovni prag te bi svi korisnici koji zadovolje prag imali oznaku da su savladali koncept. To i nije najbolji pristup jer se ne uzimaju u obzir razne druge varijable kao što su npr. vjerojatnost da korisnik pogađa ili slučajno pogriješi.

Kako bi se riješio taj problem, pronađena je tehnika Bayesian Knowledge Tracing 3 i kombinirana sa vlastitom tehnikom raspodjele po Gaussu kako bi se bolje aproksimiralo trenutno znanje korisnika.

## 4.3. Skaliranje korisnika prema Gaussu

Kao alternativa BKT-u implementirano je skaliranje studenata prema Gaussu:

- za svakog studenta i za svaku vještinu izračunat je percentil kojemu taj student pripada u ovisnosti o odgovorima svih studenata
- postavi se threshold koji odgovara percentilu iznad kojeg se nalaze studenti koji su položili vještinu

Paralelno se za svakog studenta računa prolaznost (0 ili 1) prema BKT-u i prema Gaussu:

- ako se one poklapaju, to se uzima kao vrijednost za tog studenta
- ako je zbroj te dvije jačine veći od 0, uzima se da je student položio vještinu
- ako su različite gleda se jačina s kojom model tvrdi da je student pao/položio, ona se računa na sljedeći način:

## 4.4. Izvještaji

### 4.4.1. 22.07.2020.

#### Planovi, problemi:

- pronaći dobru vrijednost za cutoff kod micanja poveznica kod grafa
- iteriranje po retcima dataframeova - nije dobro, treba pronaći efikasnije načine obrade tablica
- isprobati program na malom datasetu
- isprobati program na biologija dataset, ali gledajući da je svako pitanje jedan koncept (napraviti i program koji bi pretvorio pitanja u koncepte)
- proučiti librarye za crtanje usmjerenih grafova (boje, labela, debljine poveznica)
- dopuniti program da crta usmjerene grafove
- pokušati naći implementacije EM i grid search algoritma za računanje BKT parametara
- pokušati pronaći dobro objašnjenog bayesa u kontinuiranoj domeni
- pronaći naprednije implementacije BKT-a koje uzimaju više parametara
- osmisлити strukture podataka koje bi predstavljale čvorove i povezivale pitanja, koncepte i ostale čvorove u jednu cjelinu, omogućavale obilazak grafa te nalaženje najkraćeg puta (po nekom kriteriju) od početnog do ciljnog čvora
- napraviti novi test koji bi imao mali broj pitanja iz različitih područja koja su međusobno povezana (npr. stanica, što je to -> građa stanice -> građa različitih organela) i na tome isprobati program

#### Rezultati

- napravljena skripta za izgradnju grafa pomoću uvjetnih vjerojatnosti i Bayesove formule
  - ulaz: datoteka koja sadrži BKT parametre za sve vještine i datoteka koja sadrži tablicu studenata i njihovih odgovora
  - izlaz: matrica koja predstavlja matricu susjedstva grafa, gdje svaki element predstavlja “jačinu”/”vjerojatnost” s kojom je koncept  $i$  povezan s konceptom  $j$

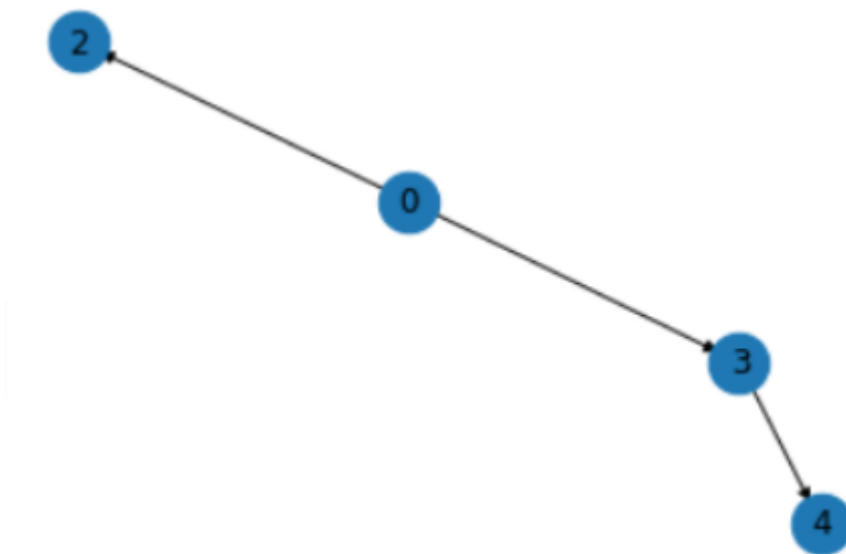
- skripta sadrži razred BKT i funkcije za računanje vjerojatnosti  $p(X)$ , računanje zajedničke vjerojatnosti  $p(X \text{ and } Y)$  i računanje matrice ovisnosti
- skripta je isprobana na dosad prikupljenim podacima testa za biologiju (74 odgovora) i dobiveni su realni rezultati, ali nismo imali sa čime usporediti budući da je to samo jedan koncept
- izrađeni su grafovi na malom umjetnom datasetu i datasetu biologija gdje su se pitanja gledali kao zasebni koncepti
- za potrebu prilagodbe pitanja u koncept napravljena je skripta koja uzima pitanja kao koncepte, računa vjerojatnosti te crta graf
- za crtanje grafova koristio se `networkx` library
- vrijednost cutoffa za izgradnju grafova bi prema procjeni trebala u intervalu  $[0.8, 0.95]$ . Nismo našli nikakve službene izvore/preporuke tako da će se za sada cutoff morati ručno procijeniti.
- odrađena optimizacija koda gdje se boljim iteriranjem po dataframeu dobiva ljepši i brži kod, daljnja optimizacija je vjerojatno moguća, ali trenutno nije prioritet trošiti vrijeme na istraživanje kako to napraviti

#### **4.4.2. 28.07.2020.**

- napravljen je novi kviz sa pet koncepata iz biologije
- svaki koncept ima 6 pitanja
- problemi: kviz ima više pitanja nego prošli, u prosjeku je mentalno zahtjevniji, znanje koncepta vjerojatno nije dobro pokriveno sa tih 6 pitanja, neki koncepti su su značajno zahtjevniji nego drugi te raspodjela pitanja po težini nije ista u svakom konceptu i vjerojatno nije optimalna budući da se trenutno svi odgovori vrednuju jednako
- u trenutku pisanja izvještaja imamo samo 25 odgovora na kviz, pokrenuli smo model da vidimo kako reagira na manjak podataka
- rezultati su očekivano lošiji nego prije, u početku su od 5 koncepata na grafu bila prikazana samo 3 te smo dobili NaN vrijednosti matrici povezanosti, što se dogodilo jer prema pragu BKT-a niti jedan student nije položio predmet te se pojavilo dijeljenje sa nulom. Nakon što smo spustili prag prolaska sa 0.95 na 0.9, pojavio se dodatan čvor na grafu i nismo imali NaN, ovo ukazuje na

problem osjetljivosti BKT-a na manjak podataka i na redoslijed točno/netočno odgovora studenata

- također same povezanosti grafa nisu imali smisla (npr. da je živčana stanica preduvjet za diobu stanice)
- ovi rezultati nisu iznenađujući jer je samo svojstvo BKT-a da je osjetljiv na redoslijed i uzastopno ponavljanje točno/netočno u nizu odgovora te su modeli bazirani na vjerojatnostima jako osjetljivi na manjak ulaznih podataka
- IDEJA: isprobati jednostavno skaliranje bodova studenata na temelju Gaussove raspodjele te odrediti prag prema kojem bi se dijelili na prolazak/pad. Ovako bi se maknula loša svojstva BKT-a, ali bi model postao još jednostavniji te bi vjerojatno imao slabija svojstva predviđanja.
- treba proučiti kako napraviti da linije povezanosti imaju različite debljine s obzirom na jačinu povezanosti, onda bi se mogao malo spustiti cutoff da se dobije bolji pregled ovisnosti između koncepata



**Slika 4.1**

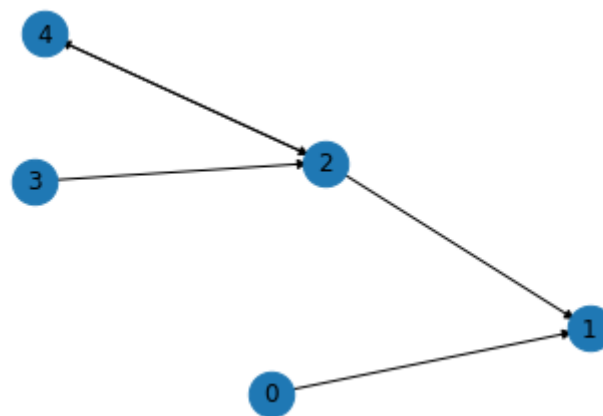
Legenda:

0: Stanica 1: DNA 2: Stanični metabolizam 3: Dioba stanice 4: Živčana stanica

#### **4.4.3. 29.07.2020.**

- napravljen program koji generira proizvoljan umjetan dataset

- obavezni ulazni argumenti su broj studenata i trojka (broj pitanja po konceptu, prosječni udio točnih odgovora u tom konceptu, standardna devijacija)
- skripta pretpostavlja da svaki student odgovara na sve koncepte i na sva pitanja
- prema gaussovoj raspodjeli svakog koncepta se svakom studentu pridjeljuje postotak točno riješenih zadataka te mu se prema tome generiranju rješenja
- dataset se sprema u klasičnom obliku kojeg naše skripte za crtanje mogu koristiti
- u trenutku pisanja ima 75 odgovora na novu anketu
- pokrenuta je obrada podataka i izračunavanje bkt parametara za trenutne odgovore
- cutoff threshold je 0.5



**Slika 4.2**

Legenda:

0 : Stanica 1 : DNA 2 : Živčana stanica 3 : Stanični metabolizam 4 : Dioba stanice

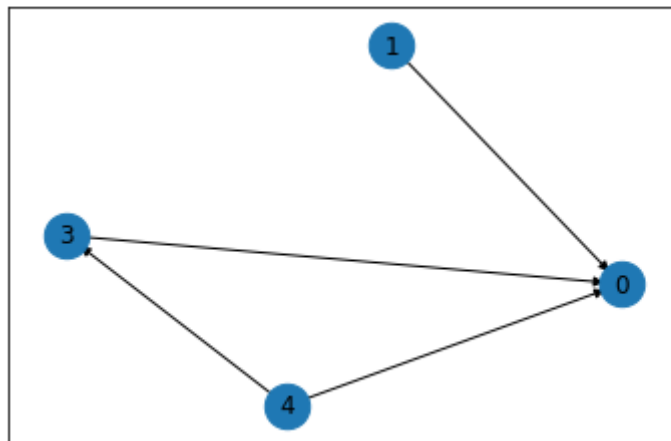
**Zaključci:** ljudi koji su znali diobu stanicu i stanični metabolizam generalno su znali i živčanu stanicu, ljudi koji su znali diobu stanice obično su znali i živčanu stanicu te ljudi koji su znali DNA su obično znali i stanicu/živčanu stanicu. Ovo ne prikazuje realan slijed učenja biologije, ali nije ni očekivano da prikazuje jer biologija nema tako strogi uvjetni slijed koncepata kao npr. matematika

### **Rezultati umjetnih datasetova**

BKT-annealing daje predzadnjem konceptu jako male vrijednosti parametara zbog

čega nitko ne prolazi taj predmet, dobivaju se NaN vrijednosti te se ne crta taj čvor. Ovo ukazuje na veliku osjetljivost aproksimacije parametara na broj i redoslijed točnih

```
[1.          0.41176471 0.          0.41176471 0.29411765]
[0.77777778 1.          0.          0.22222222 0.11111111]
[          nan          nan          nan          nan          nan]
[0.77777778 0.22222222 0.          1.          0.33333333]
[1.          0.2          0.          0.6          1.          ]]
[0, 1, 2, 3, 4]
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.
if sys.path[0] == '':
```



Slika 4.3

odgovora studenata, ovo se može poboljšati pažljivim odabirom mean i stddev vrijednosti kod generiranja umjetnog dataseta (ne odabrati premali mean te pogotovo ne uzimati veliki stddev kod malog meana i uz mali broj studenata jer se može dogoditi da većina njih dobije male postotke riješenih zadataka što daje loše bkt parametre). Drugi način je staviti nekakvo zaglađivanje kako bi se onemogućilo dobivanje NaN vrijednosti s nekom malom konstantom (ovo samo sprječava errore, ali ne pomaže u točnosti modela). Zadnje što ostaje je pronaći i razviti alternativu BKT-u koja nije toliko osjetljiva.

#### 4.4.4. 30.07.2020.

- ideja: trenutno je za model potrebno da svi studenti rješavaju sve koncepte, to nije pojava u realnim skupovima podataka pa se predlaže sljedeće rješenje - kod računa `joint_probability` uzeti presjek studenata koji su rješavali koncept A i koncept B po njihovom `student_id`-u, sam izračun `pX` se ostavi kao što je i bio. Ova pretpostavka bi mogla vrijediti jer ako je realan skup podataka dovoljno

velik te ako je presjek također dovoljno velik oni će imati dovoljno informacije da se da realni prikaz ovisnosti između koncepata

- ideja2: iako je bitno da studenti odgovaraju istim redoslijedom na pitanja taj uvjet se može izbjeći naknadnim sortiranjem po `quesiton_id`-u. Ovo nije prioritet za implementirati jer gotovo svi standardizirani ispiti daju pitanja istim redoslijedom te daju svakom studentu ista pitanja (treći uvjet za funkcioniranje modela)
- ideja3: napraviti skriptu iz koja bi bila skup svi do sad proizvedenih skripti. Ta skripta bi zapravo bila skup “main” funkcija dobivenih pozivanjem određenih funkcija iz ostalih bilježnica (ovdje se radi importanje, a ne kopiranje postojećeg koda)
- Mogućnosti sjedinjene skripte:
  - Izgradnja grafa
  - Generiranje umjetnog dataseta
  - BKT simulated annealing
  - Obrada podataka s google formsa
  - Gledanje pitanja kao koncepata

#### **4.4.5. 31.07.2020.**

- napravljene su potrebne izmjene skripti kako bi se broj studenata na pojedinim vještinama mogao razlikovati, što više odgovara realnim podacima (neće svaki student odgovarati na pitanja iz svake vještine)
- kod računanja vrijednosti `joint_probability` uzima se presjek studenata koji su rješavali koncept A i koncept B (gleda se njihov `student_id`) i računa njihova uspješnost
- kod generiranja umjetnog dataseta dodan je parametar `broj_studenata` za svaki koncept
- napravljena je skripta koja sjedinjuje sve važne dosad napravljene skripte u niz funkcija koje se lako pozivaju kako bi se moglo raditi importanje `.ipynb` fileova potrebno je u istom kazalu gdje su fileovi dodati praznu `__init__.py` datoteku
- dodana je mogućnost odabira pragova `cutoffa`, `gaussa`, `bkt-a` kod pozivanja funkcije `build_graph()`.



- trenutna greška u skripti pitanja->koncepti, izračun  $pX$ -a se pokreće dva puta, prvi put je sve u redu dok se drugi put neki keyevi ne nalazi u dictionaryu, uzrok tome je drugi poziv funkcije calculate, kod splitanja se maknu neki indeksi, greška je u pisanju koda, svugdje se gleda range(noStudents) umjesto list studenata, kod kfolda se određeni studenti miču i ostaju samo neki id-ovi i noStudents se smanjuje i može se desiti npr. da se makne student s id-om 1 što znači da ga više nema u dictionaryju ali for petlja će ga i dalje pokušati dohvatiti, treba napisati program da radi prema student id-u, ako se ne nađe bolje rješenje maknut će se k-fold izračun jer trenutna implementacija nije dobra - POPRAVLJENO
- treba u oba crtanja grafa riješiti problem dijeljenja sa nulom - DJELOMIČNO POPRAVLJENO - radi, ali rješenje nije najbolje (ako  $pX$  bude nula, ne dijeli se nego se joint probability postavlja na 0)
- taj kod bi se možda mogao koristiti kao provjera povezanosti pitanja
- zadnje što preostaje nakon popravljavanja i uljepšavanja koda koji pretvara pitanja u koncepte jest smisliti obradu google formsa tako da je omogućeno da bude različit broj pitanja po konceptu, s tim da su pitanja grupirana odnosno nisu pomiješana po ispitu

#### 4.4.6. 14.08.2020.

Prebačene su skripte iz colaba u .py fileove na git, obavljeno refaktoriranje kako bi se malo više standardizirale konvencije imenovanja i ostala uljepšavanja koda. Napravljena main skripta koju se lagano može programirati da radi razne operacije uz pomoć funkcija koje pozivaju importane module.

### 4.5. Poveznice

[https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/readings/MIT18\\_05S14\\_Reading13a.pdf](https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/readings/MIT18_05S14_Reading13a.pdf)

[https://www.probabilitycourse.com/chapter9/9\\_1\\_2\\_MAP\\_estimation.php](https://www.probabilitycourse.com/chapter9/9_1_2_MAP_estimation.php)

<https://www.sciencedirect.com/science/article/abs/pii/S0951832017300674>

<https://arxiv.org/pdf/1610.09156.pdf>

[http://www.dia.fi.upm.es/~mgremesal/MIR/slides/Lesson%206%20\(Inference%20from%20Conditional%20Fuzzy%20Propositions\).pdf](http://www.dia.fi.upm.es/~mgremesal/MIR/slides/Lesson%206%20(Inference%20from%20Conditional%20Fuzzy%20Propositions).pdf)

<https://www.intechopen.com/books/fuzzy-logic/some-methods-of-fuzzy-conditional-inference-for-application-to-fuzzy-control-systems>

## **5. Zaključak**

Zaključak.

## **6. Literatura**

## **Umjetna inteligencija u adaptivnom učenju**

### **Sažetak**

Sažetak na hrvatskom jeziku.

**Ključne riječi:** Ključne riječi, odvojene zarezima.

### **Title**

### **Abstract**

Abstract.

**Keywords:** Keywords.