

# Bayesian knowledge tracing (BKT)

August 19, 2020

# Uvod

## 1.1 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)

$$p(L_t \mid obs = correct) = \frac{p(L_t) * (1 - p(S))}{p(L_t) * (1 - p(S)) + (1 - p(L_t)) * p(G)} \quad (1.1)$$

$$p(L_t \mid obs = wrong) = \frac{p(L_t) * p(S)}{p(L_t) * p(S) + (1 - p(L_t)) * (1 - p(G))} \quad (1.2)$$

$$p(L_{t+1}) = p(L_t \mid obs = correct) + (1 - p(L_t \mid obs = correct)) * p(T) \quad (1.3)$$

$$p(C_{t+1}) = p(L_{t+1}) * (1 - p(S)) + p(L_{t+1}) * p(G) \quad (1.4)$$

## 1.2 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.

## 1.3 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

## Dobivanje BKT parametara

### 2.1 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 pronaenu 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 pronaena niti jedna implementacija EM algoritma za aproksimaciju BKT parametara niti je napravljena vlastiti implementacija zbog prevelikog praga znanja matematike.

### 2.2 Grid search i Simulated Annealing

Pronaen 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 prilagoena našim skupovima podataka. Na kraju se ispostavilo da je "simulirano kaljenje" povoljnije te se grid search odbacio.

## 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
- pronaen 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 odreuje vjerojatnost da je ispitanik naučio/ savladao gradivo
- uz pomoć skripte za aproksimaciju BKT parametara, naene su njihove vrijednosti za svaku vještinu iz ASSISTMENTS dataseta i pohranjenje u google sheets tablicu
- dobiveni parametri algoritmom simuliranog kaljenja usporeeni su s onima dobivenima pomoću grid search metode -i, 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

## Poveznice

### 4.1 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)

### 4.2 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-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)

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)